

## VS1005g - Audio Processing Platform IC

### Analog Hardware Features

- Three channels of 24-bit audio ADC
- Two 24-bit audio DACs
- Stereo earphone driver for 30  $\Omega$  load
- Internal microphone amplifiers
- Stereo FM radio receiver with RDS
- 10-bit ADC, 3-5 external inputs
- Operation from single power supply, four programmable internal regulators

### Digital Hardware Features

- 100 MIPS VS\_DSP<sup>4</sup> processor core
- 128 KiB program RAM (32 KiWord)
- 128 KiB data RAM (2 $\times$ 32 KiWord)
- Protected 8 Mi-bit FLASH (Optional)
- USB 2.0 Hi-Speed (480 Mbit/s) Device / Host
- I2S and SPDIF digital audio interfaces
- NAND FLASH interface with EEC
- SD Card interface
- 2 SPI bus interfaces
- 10BaseT Ethernet controller
- UART interface
- 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 with battery backed memory
- Reed-Solomon error correction
- HW debug support with VSIDE via JTAG

### Firmware and VSOS Features

- Decoders: MP3, WMA, Ogg Vorbis, AAC, HE-AAC, FLAC, WAV PCM
  - Encoders: MP3, Ogg Vorbis, WAV PCM
  - File I/O for SD cards and NAND flash
  - FM tuner and RDS decoder
  - USB host and slave libraries
  - Graphical display with resistive touch panel
  - Extensive audio DSP library
  - IP stack of Ethernet
  - Flexible boot options
  - Pre-emptive multitasking
  - Easy-to-write software interface with VSIDE
- Note: All VSOS features not available yet.

### Applications

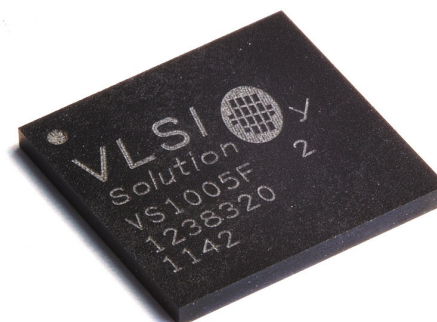
- Portable recorders
- Digital docking stations
- MP3 players
- Internet radio
- Wireless headphones
- Audio co-processor

### Overview

VS1005g is a flexible audio platform device. It is built around VS\_DSP<sup>4</sup>, 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 inputs and outputs, and the control ADC can be used for example for interfacing a resistive touch panel.

VS1005g has an embedded FLASH memory of 8 Mi-bits (1 MiByte) for customization by VLSI, customers or third parties. The firmware and hardware are designed to prevent access to the embedded FLASH in protected mode. After FLASH memory programming VS1005g can be booted from it as a fully customized stand-alone audio processor.

VS1005g is offered in six different variants (see Chapter 3 for details).



## Contents

<b>VS1005g Front Page</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>6</b>
<b>1 Disclaimer</b>	<b>7</b>
<b>2 Definitions</b>	<b>7</b>
<b>3 Product Variants</b>	<b>7</b>
<b>4 Characteristics &amp; Specifications</b>	<b>8</b>
4.1 Absolute Maximum Ratings . . . . .	8
4.2 Recommended Operating Conditions . . . . .	9
4.3 Analog Characteristics of Audio Outputs . . . . .	10
4.4 Analog Characteristics of Audio Inputs . . . . .	11
4.5 SAR Characteristics . . . . .	11
4.6 FM Characteristics . . . . .	11
4.7 Analog Characteristics of Regulators . . . . .	12
4.8 Analog Characteristics of VHIGH voltage monitor . . . . .	13
4.9 Analog Characteristics of CVDD voltage monitor . . . . .	13
4.10 Power Button Characteristics . . . . .	13
4.11 Digital Characteristics . . . . .	13
<b>5 Package and Pin Descriptions</b>	<b>14</b>
5.1 LFGA-88 Package . . . . .	14
5.2 Pin Assignments . . . . .	16
5.3 VS1005g Pin Descriptions . . . . .	17
5.3.1 PCB Layout Recommendations . . . . .	19
<b>6 Example Schematic</b>	<b>20</b>

<b>7 VS1005g General Description</b>	<b>21</b>
7.1 VS1005g Internal Architecture . . . . .	22
7.1.1 Regulator Section . . . . .	22
7.1.2 I/O Section . . . . .	23
7.1.3 Digital Section . . . . .	23
7.2 Analog Section . . . . .	25
<b>8 Oscillator and Reset Configuration</b>	<b>26</b>
<b>9 Firmware Operation</b>	<b>27</b>
9.1 SPI Boot . . . . .	27
9.2 NAND FLASH Probe . . . . .	28
9.3 UART Boot/Monitor . . . . .	29
9.4 Default Firmware Features . . . . .	30
9.4.1 SD Card Test . . . . .	30
9.4.2 USB Mass Storage and Audio Device . . . . .	30
9.5 Supported Audio Decoders . . . . .	31
9.5.1 Supported MP3 (MPEG layer III) Decoder Formats . . . . .	31
9.6 Supported Audio Encoders . . . . .	32
9.6.1 Supported MP3 (MPEG layer III) Encoder Formats . . . . .	32
<b>10 VS1005g Peripherals and Registers</b>	<b>33</b>
10.1 The Processor Core . . . . .	33
10.2 VS1005g Memory Map . . . . .	33
10.3 VS1005g Peripherals . . . . .	34
10.4 Interrupt Controller . . . . .	35
10.4.1 Interrupt Controller Registers . . . . .	36
10.4.2 Enable INT_ENABLE[L/H][0/1] . . . . .	36
10.4.3 Origin INT_ORIGIN[0/1] . . . . .	36
10.4.4 Vector INT_VECTOR . . . . .	37
10.4.5 Enable Counter INT_ENCOUNTER . . . . .	37

10.4.6	Global Disable INT_GLOB_DIS . . . . .	37
10.4.7	Global Enable INT_GLOB_ENA . . . . .	37
10.5	DSP Clock Domain Registers . . . . .	38
10.5.1	General Purpose Software Registers . . . . .	38
10.5.2	Peripheral I/O Control . . . . .	38
10.5.3	PLL Clock Control . . . . .	38
10.6	XTAL Clock Domain Registers . . . . .	40
10.6.1	Analog Control Registers . . . . .	40
10.6.2	Regulator and Peripheral Clock Control Registers . . . . .	42
10.7	Audio Playback Interfaces . . . . .	44
10.7.1	Primary Audio Path 24-bit Sample Rate Upconverter with Filters . . . . .	45
10.7.2	Primary Audio Path Volume Control . . . . .	46
10.7.3	Secondary Audio Path: DAC Offset Registers . . . . .	47
10.7.4	Filterless Sample Rate Converter (SRC) Registers . . . . .	48
10.8	SPI Peripherals . . . . .	49
10.9	Common Data Interfaces . . . . .	53
10.9.1	Ethernet Controller . . . . .	56
10.9.2	Reed-Solomon Codec . . . . .	59
10.9.3	Nand Flash Interface . . . . .	63
10.9.4	SD Card Interface . . . . .	65
10.10	USB Peripheral . . . . .	67
10.10.1	USB Peripheral Registers . . . . .	68
10.10.2	USB Clocking Modes . . . . .	71
10.10.3	USB Host . . . . .	71
10.11	Interruptable General Purpose IO Ports 0-2 . . . . .	72
10.12	S/PDIF Peripheral . . . . .	74
10.12.1	S/PDIF Receiver . . . . .	74
10.12.2	S/PDIF Receiver Registers . . . . .	75
10.12.3	S/PDIF Transmitter . . . . .	77
10.12.4	S/PDIF Transmitter Registers . . . . .	78

10.13 UART Peripheral . . . . .	81
10.13.1 UART Peripheral Registers . . . . .	81
10.14 Watchdog Peripheral . . . . .	83
10.14.1 Watchdog Registers . . . . .	83
10.15 24-bit Analog to Digital Converters (ADC) . . . . .	84
10.15.1 Configuring Signal Paths for ADC1 and ADC2 . . . . .	85
10.15.2 Digital Filter Operation Modes . . . . .	86
10.16 FM Receiver . . . . .	88
10.16.1 Configuring RF and Analog Modules for FM Receiver Mode . . . . .	88
10.16.2 Configuring FM Demodulator . . . . .	89
10.16.3 Radio Data System (RDS) . . . . .	91
10.17 I2S Peripheral . . . . .	93
10.17.1 I2S Peripheral Registers . . . . .	93
10.18 Timer Peripheral . . . . .	95
10.18.1 Timer Peripheral Registers . . . . .	95
10.19 Real Time Clock (RTC) . . . . .	97
10.19.1 RTC Peripheral Registers . . . . .	97
10.20 10-Bit Successive Approximation Register Analog-to-Digital Converter (SAR) . . . . .	99
10.21 Pulse Width Modulation Unit . . . . .	101
10.22 Special Features . . . . .	102
10.22.1 Software Protection . . . . .	102
<b>11 VS1005g Debugger</b>	<b>103</b>
<b>12 Document Version Changes</b>	<b>106</b>
<b>13 Contact Information</b>	<b>107</b>

## List of Figures

1	VS1005g top view, LFGA-88. . . . .	14
2	VS1005g bottom view, LFGA-88. . . . .	15
3	VS1005g side view, LFGA-88 . . . . .	15
4	VS1005g 88-pin LFGA pin assignment. . . . .	16
5	VS1005g default pin usage. . . . .	20
6	VS1005g external interfaces . . . . .	21
7	VS1005g block diagram . . . . .	22
8	User's memory map. . . . .	33
9	VS1005g playback (DA) audio paths. . . . .	44
10	Block diagram of data interfaces. . . . .	53
11	VS1005g USB block diagram. . . . .	67
12	VS1005g UTM functional block diagram. . . . .	68
13	S/PDIF frame format. . . . .	74
14	S/PDIF sub-frame format. . . . .	74
15	RS232 serial interface protocol . . . . .	81
16	VS1005g recording (AD and FM) signal paths . . . . .	84
17	Block diagram of FM receiver, RF and analog section . . . . .	88
18	RDS data structure . . . . .	91
19	I2S frame format. . . . .	93
20	JTAG state machine. . . . .	103

## 1 Disclaimer

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

## 2 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.

## 3 Product Variants

Device ID (Order Code)	MP3 Encoder	MP3 Decoder	Embedded 8 Mi-bit Flash
VS1005G-Q		X	
VS1005G-F-Q		X	X
VS1205G-Q	X	X	
VS1205G-F-Q	X	X	X
VS8005G-Q			
VS8005G-F-Q			X

### 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 <sup>3</sup>		-0.3	IOVDD+0.3 <sup>1</sup>	V
Voltage at power Button	PWRBTN	-0.3	3.6	V
Voltage at RTC Pins	XTALI_RTC, XTALO_RTC	-0.3	CVDD+0.3 <sup>4</sup>	V
Total Injected Current on Pins			±200 <sup>2</sup>	mA
Operating Temperature		-40	+85	°C
Storage Temperature		-65	+150	°C

<sup>1</sup> Must not exceed 3.6 V

<sup>2</sup> Latch-up limit

<sup>3</sup> Except RTC and PWRBTN pin

<sup>4</sup> 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 <sup>1</sup>	AGND DGND		0.0		V
Regulator input voltage <sup>2</sup>	VHIGH	AVDD+0.3	4.0	5.25	V
Analog positive supply <sup>3</sup>	AVDD	2.75	2.8	3.6	V
Digital positive supply <sup>3</sup>	CVDD	1.65	1.8	1.95	V
Digital RTC supply	RTCVDD	1.2	1.5	1.95	V
I/O positive supply <sup>3</sup>	IOVDD	1.8	2.8	3.6	V

<sup>1</sup> Must be connected together as close the device as possible for latch-up immunity.

<sup>2</sup> At least 4.0 V is required for compliant USB level.

<sup>3</sup> Regulator output of the device.

Oscillator Specification					
Parameter	Symbol	Min	Typ	Max	Unit
Input clock frequency <sup>1</sup>	XTALI	11	12.288 <sup>2</sup>	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			98	MHz
RTC clock frequency <sup>3</sup>	XTALI_RTC		32768		Hz
RTC frequency tolerance			+/-100		ppm
RTC oscillator startup time			1000		ms

<sup>1</sup> 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.

<sup>2</sup> 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.

<sup>3</sup> The 32 kHz crystal is optional, but required for RTC time counter.

### 4.3 Analog Characteristics of Audio Outputs

Unless otherwise noted: AVDD=3.6V, CVDD=1.8V, IOVDD=2.8V, Vref=1.6V, TA=+25°C, XTALI=12 MHz, Internal Clock Multiplier 3.0×. 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. Microphone test amplitude 50 mVpp, f=1 kHz, Line input test amplitude 2.2 Vpp, f=1 kHz. FM test signal input level -70 dBm, deviation 75 kHz, pre-emphasis 50 us, f=1 kHz.

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 <sup>1</sup>	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 <sup>2</sup>		Ω
Analog output load capacitance	AOLC			100 <sup>3</sup>	pF
DC level, V <sub>ref</sub> =1.2V (CBUF, LEFT, RIGHT)		1.1		1.3	V
DC level, V <sub>ref</sub> =1.6V (CBUF, LEFT, RIGHT)		1.5		1.7	V
CBUF disconnect current (short-circuit protection)			130	200	mA

<sup>1</sup> Loaded from Left/Right pin to analog ground via 100 μF capacitors.

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

<sup>3</sup> 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 Analog Characteristics of Audio Inputs

ADC Characteristics					
Parameter	Symbol	Min	Typ	Max	Unit
ADC Resolution			24		bits
Microphone input amplifier gain	MICG		20		dB
Microphone input amplitude			50	140 <sup>1</sup>	mVpp AC
Microphone Total Harmonic Distortion	MTHD		0.02	0.10	%
Microphone S/N Ratio, A-weighted	MSNR	50 <sup>2</sup>	75		dB
Line input amplitude			2200	2800 <sup>1</sup>	mVpp AC
Line input Total Harmonic Distortion	LTHD		0.015	0.10	%
Line input S/N Ratio	LSNR	80 <sup>2</sup>	90	100	dB
Sample rate		24		192	kHz
Line and Microphone input impedances			100		kΩ

<sup>1</sup> Above typical amplitude the Harmonic Distortion increases.

<sup>2</sup> Limit Min due to noise level of production tester.

### 4.5 SAR Characteristics

SAR Characteristics					
Parameter	Symbol	Min	Typ	Max	Unit
SAR resolution			10		bits
Input amplitude range		0		AVDD	V
SAR sample rate <sup>1</sup>				100	kHz
Integral Nonlinearity	INL		+/-2		LSB
Differential Nonlinearity	DNL		+/-0.5		LSB

<sup>1</sup> XTAL dependent

### 4.6 FM Characteristics

FM Characteristics						
Parameter	Symbol	Min	Typ	Max	Unit	
Channel frequency range, 10 kHz steps		76		108	MHz	
FM mono S/N Ratio, deviation 22kHz, Input level -50dBm <sup>1</sup>	LSNR		72		dB	
FM stereo S/N Ratio, deviation 45kHz, Input level -50dBm <sup>1</sup>	LSNR		47		dB	
FM mono S/N Ratio, deviation 22kHz, Input level -90dBm <sup>1</sup>	LSNR		45		dB	
FM stereo S/N Ratio, deviation 45kHz, Input level -90dBm <sup>1</sup>	LSNR		27		dB	
Total harmonic distortion, deviation 75 kHz	THD		0.1	0.3	%	
Stereo separation			40		dB	

<sup>1</sup> Measured over whole FM band

### 4.7 Analog Characteristics of Regulators

Parameter	Symbol	Min	Typ	Max	Unit
<b>IOVDD</b>					
Recommended voltage setting range		1.7		3.6	V
Voltage setting step size		55	60	65	mV
Default setting, reset mode <sup>1</sup>			1.8		V
Default setting, active mode <sup>2</sup>			1.8/3.6 <sup>3</sup>		V
Load regulation			4.0		mV/mA
Line regulation from VHIGH			2.0		mV/V
Continuous current			30 <sup>4</sup>	60	mA
<b>IOVDD2</b>					
Recommended voltage setting range		1.7		3.6	V
Voltage setting step size		55	60	65	mV
Default setting, reset mode <sup>1</sup>			1.8		V
Default setting, active mode <sup>2</sup>			1.8/3.6 <sup>3</sup>		V
Load regulation			4.0		mV/mA
Line regulation from VHIGH			2.0		mV/V
Continuous current			30 <sup>4</sup>	60	mA
<b>CVDD</b>					
Recommended voltage setting range		1.65		1.95	V
Voltage setting step size		25	30	35	mV
Default setting, reset mode <sup>1</sup>			1.8		V
Default setting, active mode <sup>2</sup>			1.8		V
Continuous current			25 <sup>4</sup>	70	mA
Load regulation			2.0		mV/mA
Line regulation from VHIGH			2.0		mV/V
<b>AVDD</b>					
Recommended voltage setting range		2.6		3.6	V
Voltage setting step size		35	40	45	mV
Default setting, reset mode <sup>1</sup>			2.5		V
Default setting, active mode <sup>2</sup>			2.7		V
Continuous current			30 <sup>4</sup>	70	mA
Load regulation			1.5		mV/mA
Line regulation from VHIGH			2.0		mV/V
<b>PWRBTN</b>					
Minimum startup voltage			0.9		V
Minimum startup pulse			100		ms

<sup>1</sup> Device enters reset mode when XRESET pin is pulled low.

<sup>2</sup> Device enters active mode when XRESET pin is pulled high after reset mode. Regulator settings can be modified when booted from external memory (see Section 9).

<sup>3</sup> Depends on GPIO0\_7 pin status in boot (see Section 9).

<sup>4</sup> Device is tested with a 30 mA load.

### 4.8 Analog Characteristics of VHIGH voltage monitor

Parameter	Symbol	Min	Typ	Max	Unit
Trigger voltage	AMON		$1.07 \times AVDD$		V
Hysteresis			50		mV

### 4.9 Analog Characteristics of CVDD voltage monitor

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

### 4.10 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

### 4.11 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 <sup>1</sup>		$0.7 \times IOVDD$			V
Low-Level Output Voltage, 1.0 mA load <sup>1</sup>				$0.3 \times IOVDD$	V
XTALO high-level output voltage, -0.1 mA load		$0.7 \times IOVDD$			V
XTALO low-level output voltage, 0.1 mA load				$0.3 \times IOVDD$	V
Input leakage current		-1.0		1.0	$\mu A$
Rise time of all output pins, load = 30 pF <sup>1</sup>				50	ns

<sup>1</sup> Pins GPIO0\_[15:0], GPIO1\_[15:0], GPIO2\_[13:0].

## 5 Package and Pin Descriptions

### 5.1 LFGA-88 Package

LFGA-88 is a 10x10x0.8 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 Figures 1, 2 and 3. For more information about the LFGA-88 package and its dimensions visit <http://www.vlsi.fi/>.

Note that the pins do not extend to the sides of the IC (see Figure 2).

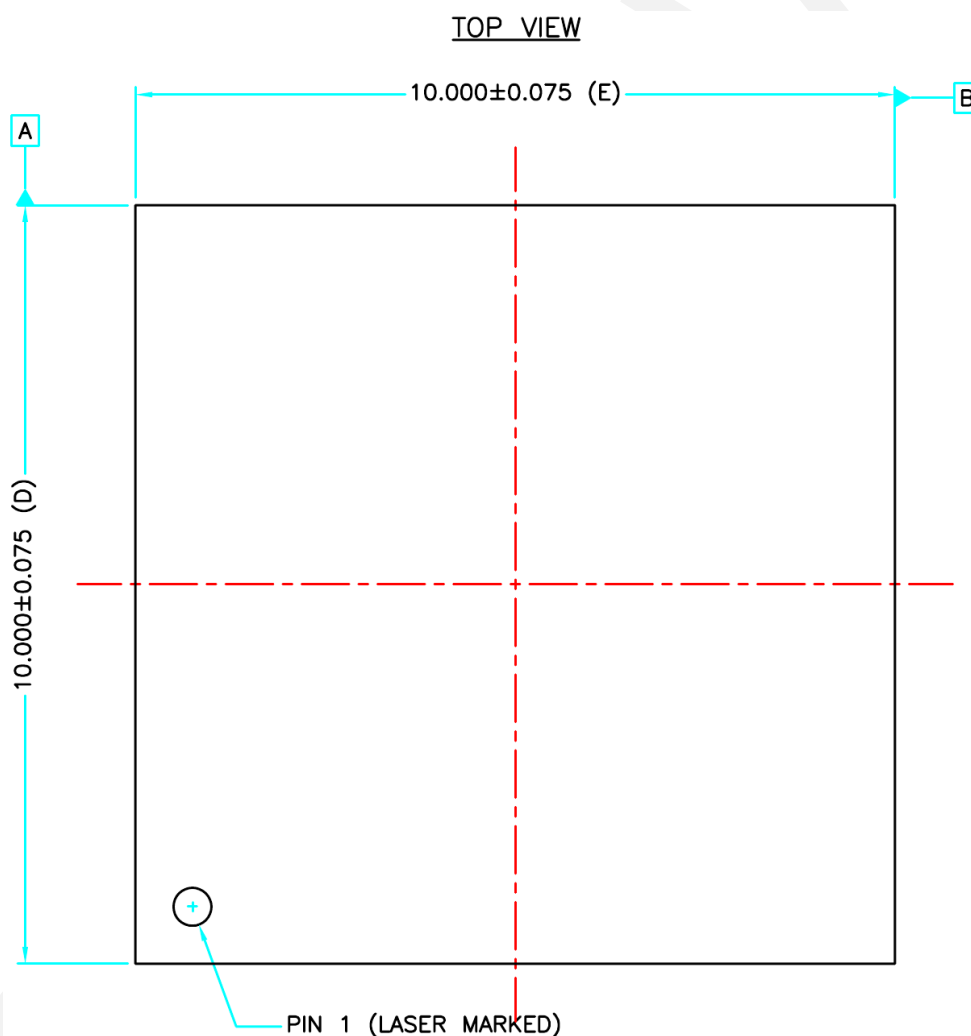


Figure 1: VS1005g top view, LFGA-88.

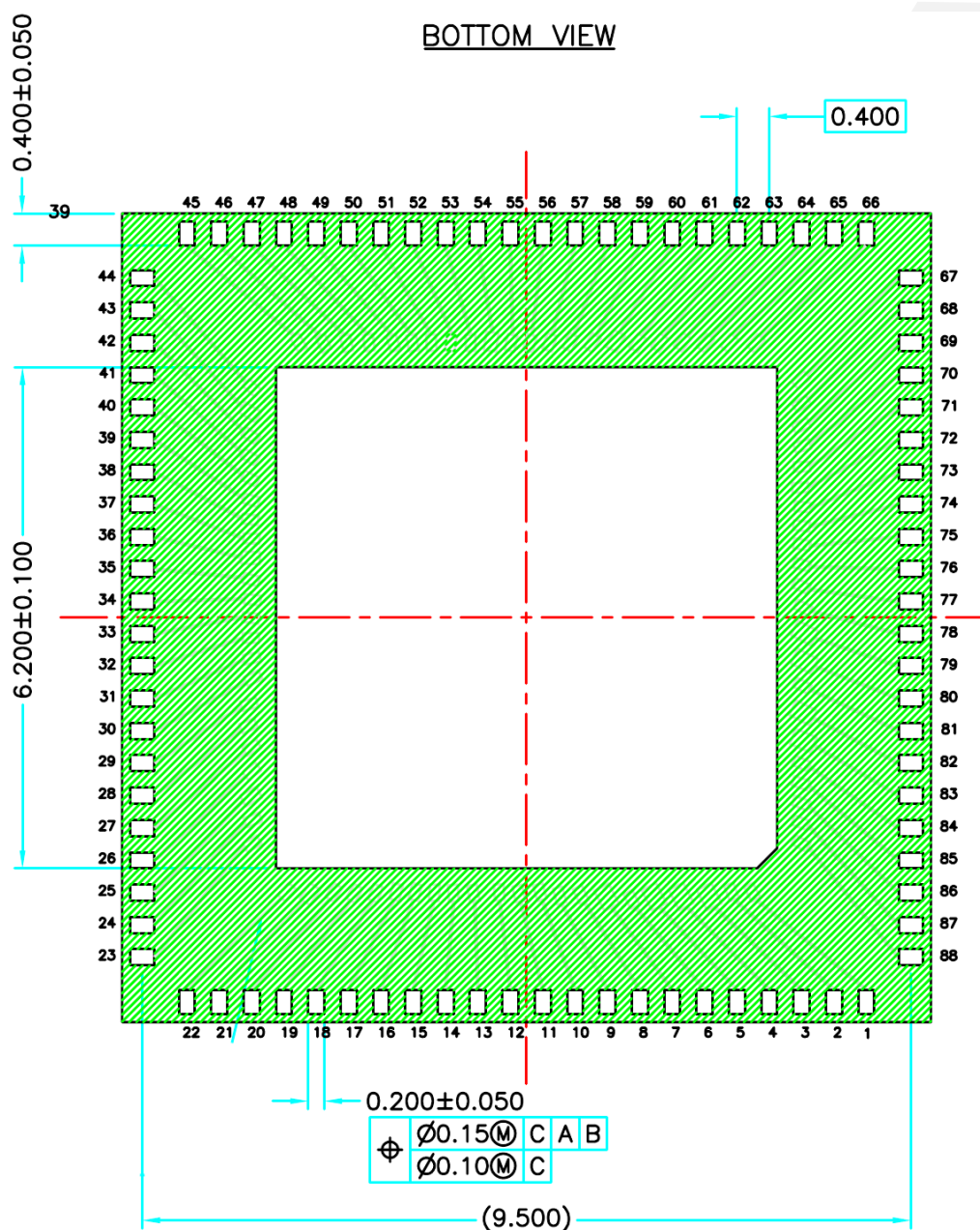


Figure 2: VS1005g bottom view, LFGA-88.

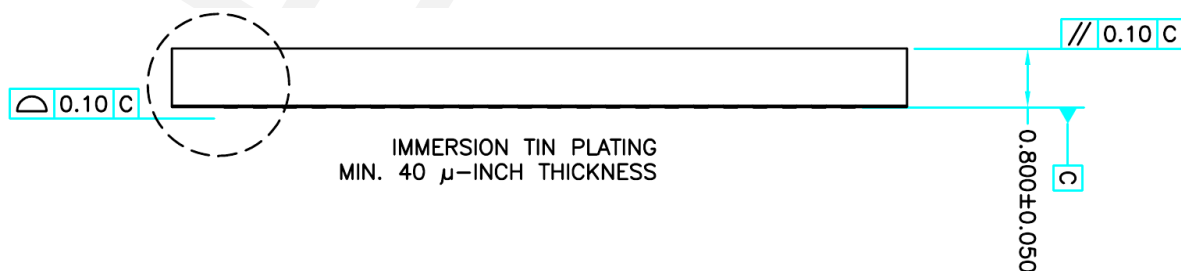


Figure 3: VS1005g side view, LFGA-88



### 5.2 Pin Assignments

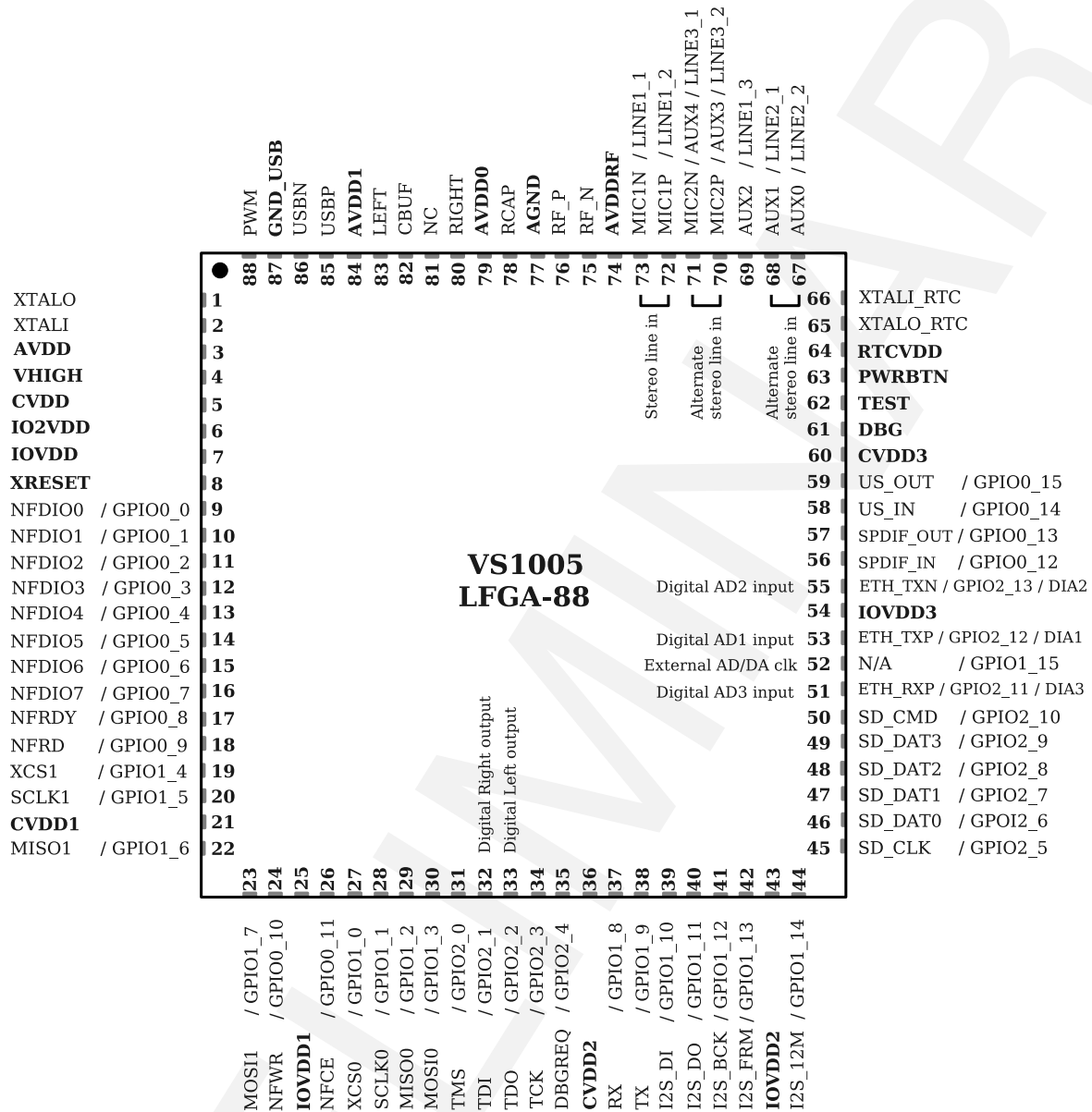


Figure 4: VS1005g 88-pin LFGA pin assignment.



### 5.3 VS1005g Pin Descriptions

Pin Name	LFGA Pin	Pin Type	Function
XTALO	1	AO	Crystal output
XTALI	2	AI	Crystal input
AVDD	3	APWR	Analog power supply, Regulator output
VHIGH	4	PWR	Power supply, Regulator input
CVDD	5	CPWR	Core power supply, Regulator output
IO2VDD	6	IO2PWR	Serial Flash power supply, Regulator output
IOVDD	7	IOPWR	I/O power supply, Regulator output
XRESET	8	DI	Active low asynchronous reset, schmitt-trigger input
NFDIO0 / GPIO0_0	9	DIO	Nand-flash IO0 / General-purpose IO Port 0, bit 0
NFDIO1 / GPIO0_1	10	DIO	Nand-flash IO1 / General-purpose IO Port 0, bit 1
NFDIO2 / GPIO0_2	11	DIO	Nand-flash IO2 / General-purpose IO Port 0, bit 2
NFDIO3 / GPIO0_3	12	DIO	Nand-flash IO3 / General-purpose IO Port 0, bit 3
NFDIO4 / GPIO0_4	13	DIO	Nand-flash IO4 / General-purpose IO Port 0, bit 4
NFDIO5 / GPIO0_5	14	DIO	Nand-flash IO5 / General-purpose IO Port 0, bit 5
NFDIO6 / GPIO0_6	15	DIO	Nand-flash IO6 / General-purpose IO Port 0, bit 6
NFDIO7 / GPIO0_7	16	DIO	Nand-flash IO7 / General-purpose IO Port 0, bit 7
NFRDY / GPIO0_8	17	DIO	Nand-flash READY / General-purpose IO Port 0, bit 8
NFRD / GPIO0_9	18	DIO	Nand-flash RD / General-purpose IO Port 0, bit 9
XCS1 / GPIO1_4	19	DIO	SPI1 XCS / General-Purpose I/O Port 1, bit 4
SCLK1 / GPIO1_5	20	DIO	SPI1 CLK / General-Purpose I/O Port 1, bit 5
CVDD1	21	CPWR	Core power supply, connect to regulator CPWR
MISO1 / GPIO1_6	22	DIO	SPI1 MISO / General-Purpose I/O Port 1, bit 6

Pin Name	LFGA Pin	Pin Type	Function
MOSI1 / GPIO1_7	23	DIO	SPI1 MOSI / General-Purpose I/O Port 1, bit 7
NFWR / GPIO0_10	24	DIO	Nand-flash WR / General-purpose IO Port 0, bit 10
IOVDD1	25	IOPWR	I/O power supply, connect to regulator IOPWR
NFCE / GPIO0_11	26	DIO	Nand-flash CE / General-purpose IO Port 0, bit 11
XCS0 / GPIO1_0	27	DIO	SPI0 XCS / General-Purpose I/O Port 1, bit 0
SCLK0 / GPIO1_1	28	DIO	SPI0 CLK / General-Purpose I/O Port 1, bit 1
MISO0 / GPIO1_2	29	DIO	SPI0 MISO / General-Purpose I/O Port 1, bit 2
MOSI0 / GPIO1_3	30	DIO	SPI0 MOSI / General-Purpose I/O Port 1, bit 3
TMS / GPIO2_0	31	DIO	JTAG TMS / General-Purpose I/O Port 2, bit 0
TDI / GPIO2_1	32	DIO	JTAG TDI / General-Purpose I/O Port 2, bit 1
TDO / GPIO2_2	33	DIO	JTAG TDO / General-Purpose I/O Port 2, bit 2
TCK / GPIO2_3	34	DIO	JTAG TCK / General-Purpose I/O Port 2, bit 3
DBGREQ / GPIO2_4	35	DIO	Debug interrupt / General-Purpose I/O Port 2, bit 4
CVDD2	36	CPWR	Core power supply, connect to regulator CPWR
RX / GPIO1_8	37	DIO	UART RX / General-Purpose I/O Port 1, bit 8
TX / GPIO1_9	38	DIO	UART TX / General-Purpose I/O Port 1, bit 9
I2S_DI / GPIO1_10	39	DIO	I2S data in / General-Purpose I/O Port 1, bit 10
I2S_DO / GPIO1_11	40	DIO	I2S data out / General-Purpose I/O Port 1, bit 11
I2S_BCK / GPIO1_12	41	DIO	I2S bit clock / General-Purpose I/O Port 1, bit 12
I2S_FRM / GPIO1_13	42	DIO	I2S frame sync / General-Purpose I/O Port 1, bit 13
IOVDD2	43	IOPWR	I/O power supply, connect to regulator IOPWR
I2S_12M / GPIO1_14	44	DIO	I2S 12.288MHz clock output (XTALI) / General-Purpose I/O Port 1, bit 14

Pin Name	LFGA Pin	Pin Type	Function
SD_CLK / GPIO2_5	45	DIO	SD card clock / General-Purpose I/O Port 2, bit 5
SD_DAT0 / GPIO2_6	46	DIO	SD card data line 0 / General-Purpose I/O Port 2, bit 6
SD_DAT1 / GPIO2_7	47	DIO	SD card data line 1 / General-Purpose I/O Port 2, bit 7
SD_DAT2 / GPIO2_8	48	DIO	SD card data line 2 / General-Purpose I/O Port 2, bit 8
SD_DAT3 / GPIO2_9	49	DIO	SD card data line 3 / General-Purpose I/O Port 2, bit 9
SD_CMD / GPIO2_10	50	DIO	SD card cmd line / General-Purpose I/O Port 2, bit 10
ETH_RXP / GPIO2_11 / DIA3	51	DIO	Ethernet RXP / General-Purpose I/O Port 2, bit 11 / Digital ADC 1 input
GPIO1_15	52	DIO	General-Purpose I/O Port 1, bit 15
ETH_TXP / GPIO2_12 / DIA1	53	DIO	Ethernet TXP / General-Purpose I/O Port 2, bit 12 / Digital ADC 1 input
IOVDD3	54	IOPWR	I/O power supply, connect to regulator IOPWR
ETH_TXN / GPIO2_13 / DIA2	55	DIO	Ethernet TXN / General-Purpose I/O Port 2, bit 13 / Digital ADC 1 input
SPDIF_IN / GPIO0_12	56	DIO	S/PDIF data in / General-Purpose I/O Port 0, bit 12
SPDIF_OUT / GPIO0_13	57	DIO	S/PDIF data out / General-Purpose I/O Port 0, bit 13
GPIO0_14	58	DIO	General-Purpose I/O Port 0, bit 14
GPIO0_15	59	DIO	General-Purpose I/O Port 0, bit 15
VDD3	60	CPWR	Core power supply, connect to regulator CPWR
DBG	61	DI	Debug mode enable (active high), connect to DGND
TEST	62	DI	Test mode input (active high), connect to DGND
PWRBTN	63	AIO	Power button for Regulator startup (and Power Key)
RTCVD	64	RTCPWR	Real time clock power supply
XTALO_RTC	65	AO	Real time clock crystal output
XTALI_RTC	66	AI	Real time clock crystal input

Pin Name	LFGA Pin	Pin Type	Function
AUX0 / LINE2_2	67	AI	SAR AD input 0 / Line 2 input 2
AUX1 / LINE2_1	68	AI	SAR AD input 1 / Line 2 input 1
AUX2 / LINE1_3	69	AI	SAR AD input 2 / Line 1 input 3
MIC2P / AUX3 / LINE3_2	70	AI	Microphone 2 pos. differential input, self-biasing / SAR AD input 3 / Line 3 input 2
MIC2N / AUX4 / LINE3_1	71	AI	Microphone 2 neg. differential input, self-biasing / SAR AD input 4 / Line 3 input 1
MIC1P / LINE1_2	72	AI	Microphone 1 pos. differential input, self-biasing / Line 1 input 2
MIC1N / LINE1_1	73	AI	Microphone 1 neg. differential input, self-biasing / Line 1 input 1
AVDDRF	74	APWR1V8	1.8V RF power supply, connect to regulator CPWR
RF_N	75	AI	RF FM antenna negative differential input
RF_P	76	AI	RF FM antenna positive differential input
AGND <sup>1</sup>	77	APWR	Analog reference ground
RCAP	78	AIO	Filtering capacitance for reference
AVDD0	79	APWR	Analog power supply, connect to regulator APWR
RIGHT	80	AO	Right channel output
N/A	81	N/A	Not connected pin
CBUF	82	AO	Common voltage buffer for headphones
LEFT	83	AO	Left channel output
AVDD1	84	APWR	Analog power supply, connect to regulator APWR
USBP	85	AIO	USB differential + in / out, controllable 1.5k $\Omega$ pull-up
USBN	86	AIO	USB differential - in / out
GNDUSB	87	APWR	USB ground, connect to ground network in PCB
PWM	88	DO	PWM output

<sup>1</sup> Connect to RCAP capacitor without vias in PCB.

Alternate pin functions in VS1005g package			
Pin Name	LFGA Pin	Pin Type	Function
Analog Line input 1	71	AI	Alternate analog input pin for Line input 1
Analog Line input 2	70	AI	Alternate analog input pin for Line input 2
Analog Line input 1	68	AI	Alternate analog input pin for Line input 1
Analog Line input 2	67	AI	Alternate analog input pin for Line input 2
Digital DA/AD Clock	52	DO	Digital DA/AD clock output, xtal/2/4
Digital DAC Right	32	DO	DAC right channel digital output, xtal/2
Digital DAC Left	33	DO	DAC left channel digital output, xtal/2
DIA1	53	DI	Digital ADC 1 input, xtal/2
DIA2	55	DI	Digital ADC 2 input, xtal/2
DIA3	51	DI	Digital ADC 3 input, xtal/2
TMS	31	DI	Jtag Test Mode Select
TDI	32	DI	Jtag Test Data In
TDO	33	DO	Jtag Test Data Out
TCK	34	DI	Jtag Test Clock
DBGREQ	35	DO	Hardware debug state pin

Pin type descriptions:

Type	Description	Type	Description
DI	Digital input, CMOS Input Pad	APWR	Analog power supply pin or ground
DO	Digital output, CMOS Input Pad	APWR1V8	Analog power supply pin, 1.8V
DIO	Digital input/output	RTCPWR	Real time clock power supply pin, 1.8V
AI	Analog input	DGND	Core or I/O ground pin
AO	Analog output	CPWR	Core power supply pin
AIO	Analog input/output	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.

### 5.3.1 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.
- RF\_P and RF\_N traces should be isolated from all other signal traces.

## 6 Example Schematic

Default pin usage is shown in Figure 5.

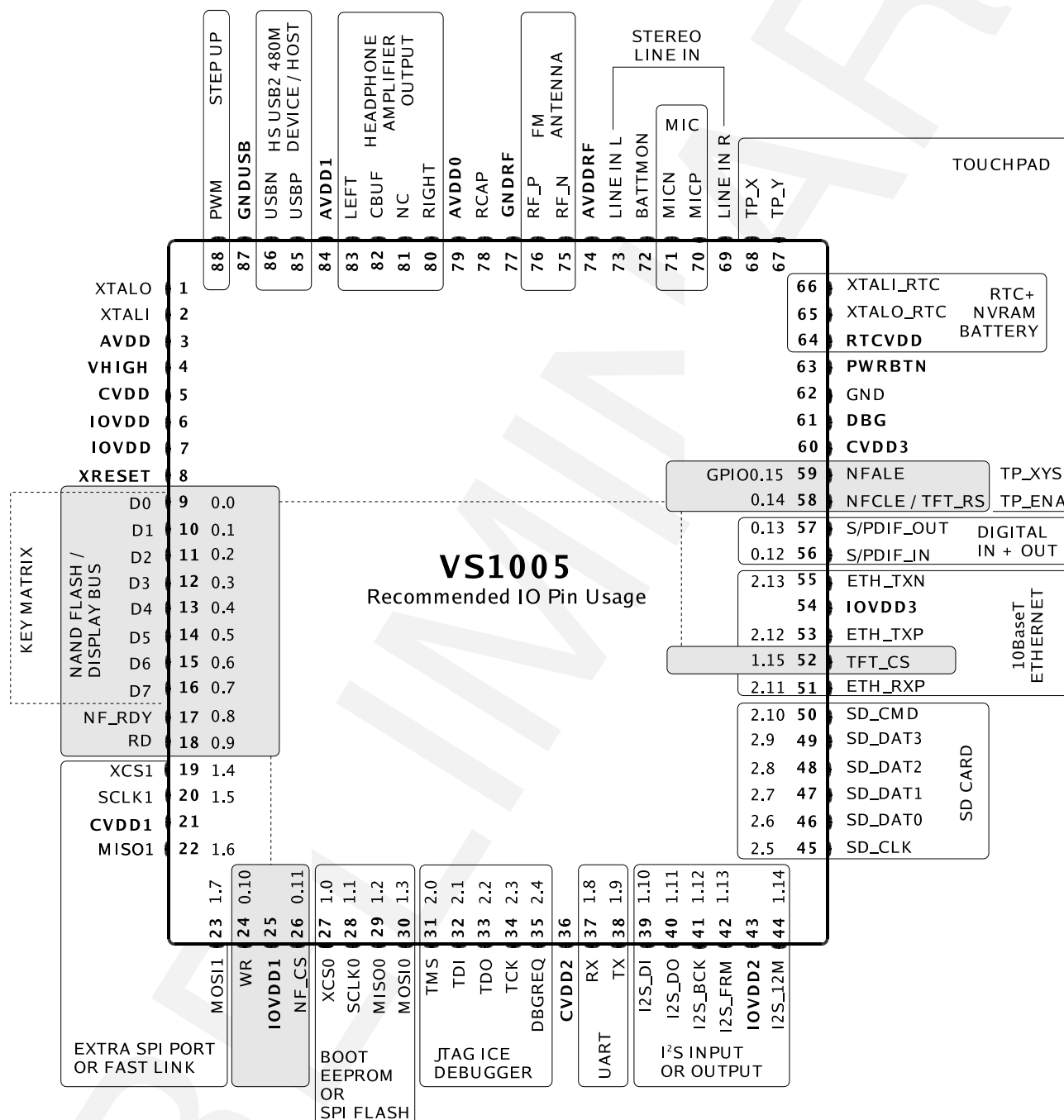


Figure 5: VS1005g default pin usage.

## 7 VS1005g General Description

VS1005g 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. VS1005g has two clock domains which are clocked by PLL. Analog interfaces are clocked with a XTAL clock but the dsp, digital interfaces and memories are clocked with a multiplied clock. VS1005g external interfaces are shown in Figure 6.

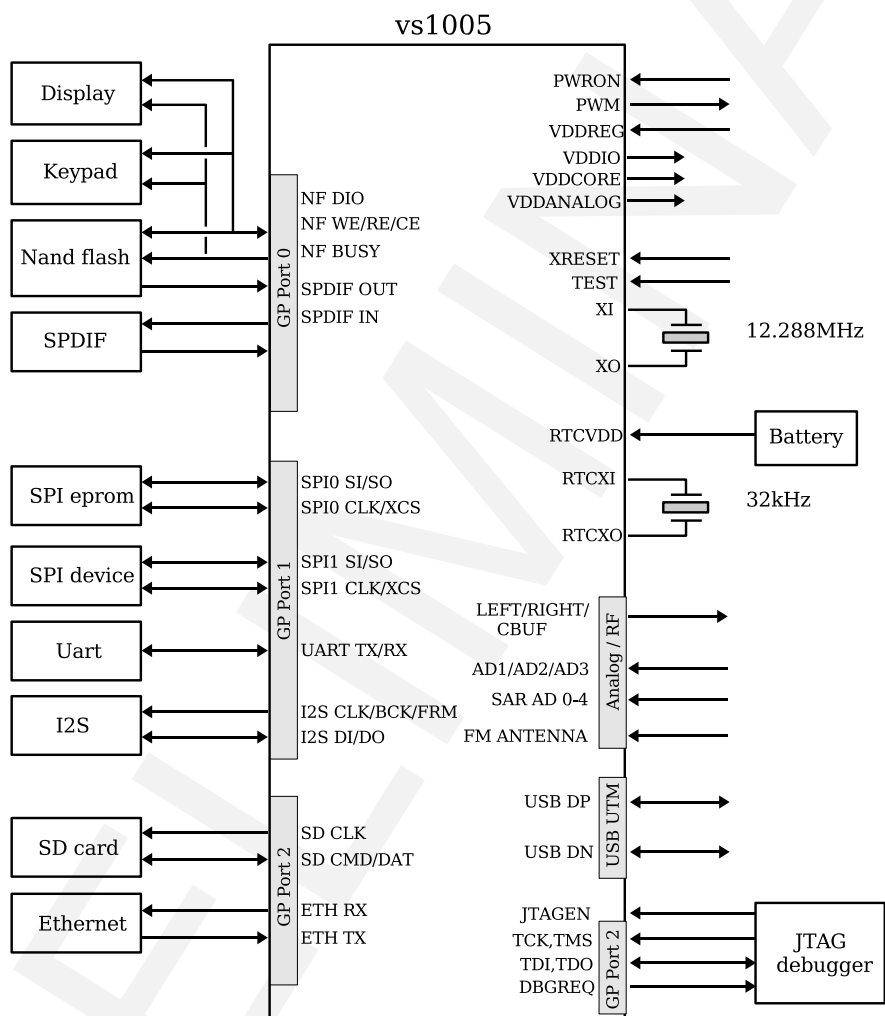


Figure 6: VS1005g external interfaces

## 7.1 VS1005g Internal Architecture

VS1005g block diagram is shown in Figure 7.

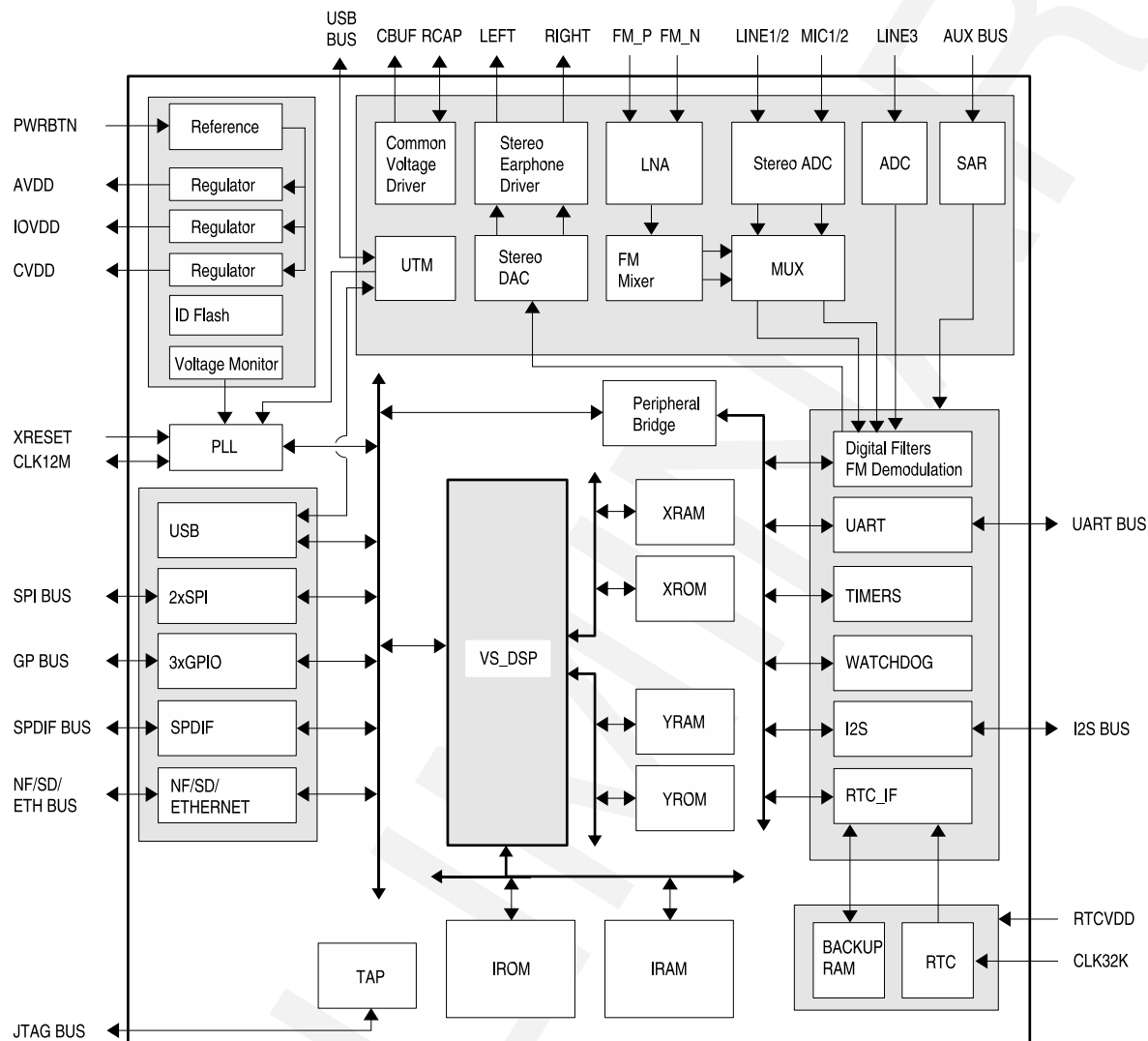


Figure 7: VS1005g block diagram

### 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, IOVDD2 and IOVDD3 input pins. Proper bypass capacitors should also be used.

The firmware uses GPIO0\_7 to select I/O voltage level. After reset the I/O voltage is 1.8 V. If GPIO0\_7 has a pull-down resistor, 1.8 V I/O voltage is used. If GPIO0\_7 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, CVDD2 and CVDD3 input pins. 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 VS1005g 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  $1.0\times$ ,  $1.5\times$ , ...,  $8.0\times$ . When USB is connected, the clock is  $5.0\times 12\text{ MHz} = 60\text{ MHz}$ . 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 switched to RTC clock and the clock oscillator is powered down.

## VSDSP<sup>4</sup>

VSDSP<sup>4</sup> 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 VS1005gx<sup>F</sup> 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.

## NAND FLASH Interface

The NAND FLASH peripheral calculates a simple error-correcting code (ECC), and automates some of the communication with a NAND FLASH chip. The firmware uses the peripheral to access both small-page (512+16 B pages) and large-page (2048+64 B pages) NAND FLASH chips. The first sector in the FLASH tells the firmware how it should be accessed.

The NAND FLASH interface pins can also be used as general-purpose I/O. The default firmware uses GPIO0\_[4:0] for keys, and GPIO0\_[7:6] for other purposes. Pull-up and pull-down resistors must be used for these connections so that the data transfer to and from the NAND FLASH isn't disturbed when keys are pressed.

## 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.



## Ethernet Controller

Ethernet Controller is an interface to 10base-t network. The interface uses digital signal levels and external components are required to connect to ethernet. The core clock must be switched to 60 MHz when ethernet peripheral is used.

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

## USB

The USB peripheral handles USB 2.0 Full Speed and Hi-Speed hardware protocols. Low speed communication is not supported, but is correctly ignored. The USBP pin has a software-controllable 1.5k $\Omega$  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.

The firmware uses the USB peripheral to implement both USB Mass Storage Device and USB Audio Device. Which device is activated depends on the state of GPIO0\_6 when the USB connection is detected. If GPIO0\_6 has a pull-up resistor, VS1005g appears as an USB Audio Device. If GPIO0\_6 has a pull-down resistor, VS1005g appears as an USB Mass Storage Device.

## 7.2 Analog Section

The third regulator provides power for the analog section.

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

The regulator AVDD output pin must be connected to AVDD1 and AVDD2 pins with proper bypass capacitors, because they are not connected internally. The AVDDRF pin is connected to regulator CVDD pin with proper bypass capacitors or with external regulator from VHIGH.

The USB pins use the internal AVDD voltage, and the firmware configures AVDD to 3.6 V when USB is attached.

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
- *dbg\_rst* : Debugger 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 Firmware Operation

The firmware uses the following pins (see the example schematics in Section 6):

Pin	Description
PWRBTN	High level starts regulator, is also read as the Power button Key.
GPIO0_0	external 1 M $\Omega$ pull-down resistor, Key 1 connects a 100 k $\Omega$ pull-up resistor <sup>1</sup>
GPIO0_1	external 1 M $\Omega$ pull-down resistor, Key 2 connects a 100 k $\Omega$ pull-up resistor
GPIO0_2	external 1 M $\Omega$ pull-down resistor, Key 3 connects a 100 k $\Omega$ pull-up resistor
GPIO0_3	external 1 M $\Omega$ pull-down resistor, Key 4 connects a 100 k $\Omega$ pull-up resistor
GPIO0_4	external 1 M $\Omega$ pull-down resistor, Key 5 connects a 100 k $\Omega$ pull-up resistor
GPIO0_6	external pull-down resistor for USB Mass Storage Device, pull-up for USB Audio Device
GPIO0_7	external pull-down resistor for 1.8 V I/O voltage, pull-up resistor for 3.3 V I/O voltage
NFCE	external pull-up resistor for normal operation
XCS	external pull-up to enable SPI EEPROM boot
SI	Power LED control during firmware operation
SO	Feature LED control during firmware operation
USBN	external 1 M $\Omega$ pull-up
USBP	external 1 M $\Omega$ pull-up

<sup>1</sup> Smaller pull-down resistors may be needed for keys if the capacitance on the GPIO pins is high.

Boot order:

Stage	Description
Power on	Power button (PWRBTN) pressed when VHIGH has enough voltage
Reset	Power-on reset, XRESET, or watchdog reset causes software restart
UART Boot	Almost immediately after power-on UART can be used to enter emulator mode. UART boot remains possible after the following steps, too.
SPI EEPROM boot	If XCS is high, SPI Boot is tried.
NAND FLASH probed	If NFCE is high, NAND FLASH is checked.
SD boot	If VS1005G.SYS exists, it is used as a boot file.
Default firmware	The firmware in ROM takes control.

### 9.1 SPI Boot

The first boot method is SPI EEPROM. If GPIO1\_0 is low after reset, SPI boot is skipped. If GPIO1\_0 is high, it is assumed to have a pull-up resistor and SPI boot is tried.

First the first four bytes of the SPI EEPROM are read using 16-bit address. If the bytes are "VLS5" (for protected host) or "WLS5" (for unprotected host), a 16-bit EEPROM is assumed and the boot continues. If the last 3 bytes are read as "VLS", a 24-bit EEPROM is assumed

and boot continues in 24-bit mode. Both 16-bit and 24-bit EEPROM should have the “VLS5” or “WLS5” string starting at address 0, and the rest of the boot data starting at address 4. If no identifier is found, SPI EEPROM boot is skipped.

Boot records are read from EEPROM until an execute record is reached. Unknown records are skipped using the data length field.

Byte	Description
0	type 0=l-mem 1=X-mem 2=Y-mem 3=execute
1, 2	data len lo, hi – data length in bytes
3, 4	address lo, hi – record address
5..	data*

## 9.2 NAND FLASH Probe

If NAND FLASH chip select (NFCE) is high, a NAND FLASH is assumed to be present and the first sector is read. The access methods (nandTypes 0..5) are tried in order to find the “VLN5” identification. If the first bytes are “VLN5”, a valid boot sector is assumed. This sector gives the necessary information about the NAND FLASH so that it can be accessed in the right way.

NandFlash Header		
Byte	Value	Description
0, 1, 2, 3	0x56 0x4c 0x4E 0x35	'V' 'L' 'N' '5' – Identification
4, 5	0x00 0x03	NandType (0x0003 = large-page with 3-byte block address), See table
6	0x08	BlockSizeBits ( $2^8 * 512 = 128$ KiB per block)
7	0x13	FlashSizeBits ( $2^{19} * 512 = 256$ MiB flash)
8, 9	0x00 0x46	NandWaitNs – NAND FLASH access time in ns (e.g. 0x46)
10, 11	0x00 0x01	Number of 512-byte blocks for boot (e.g. 0x0001)
12, 13	0x00 0x00	EWRD = How many words to skip before 'B' 'o' 'O' 't' string
14 ... 14+EWRD-1 (14, 15, 16, 17) + EWRD	0x42 0x6f 0x4f 0x74	Extra words (this field exists only if EWRD != 0)
18+EWRD ... 511		'B' 'o' 'O' 't' – Optional boot ident code

NandFlash Type Configuration	
Low byte (byte 0x4)	Description
0	512+16 B small-page flash with 2-byte block address ( $\leq 32$ MiB)
1	2048+64 B large-page flash with 2-byte block address ( $\leq 128$ MiB)
2	512+16 B small-page flash with 3-byte block addr. ( $> 32$ MiB, $\leq 8$ GiB)
3	2048+64 B large-page flash with 3-byte bl.addr. ( $> 128$ MiB, $\leq 32$ GiB)
4	512+16 B small-page flash with 4-byte block address ( $> 8$ GiB)
5	2048+64 B large-page flash with 4-byte block address ( $> 32$ GiB)

If bytes (14-16)+EWRD contain “BoOt”, the value in bytes 10 and 11 determines how many sectors are read from NAND-flash. Note that 0 is interpreted as 1. After the data is read into memory, the boot records in this data are processed, transferring code and data sections into the right places in memory and possibly executed. If an unknown boot record is encountered, booting is stopped and control returns to the firmware code.

NandFlash Record Configuration	
Code byte	Description
17, 16	type 0x8000=I-mem 0x8001=X-mem 0x8002=Y-mem 0x8003=execute
19, 18	data length in (words -1) : 0 = 1 word, 1 = 2 words, etc.
21, 20	address – record address
22..	data

### 9.3 UART Boot/Monitor

When byte 0xef is sent to RX at 115200 bps, the firmware enters monitor mode and communicates with **vs3emu**. Memory contents can be displayed, executables can be loaded and run, or the firmware code can be restarted or continued.

The UART is also a convenient way to program the NAND FLASH boot sector(s) or the SPI EEPROM.

## 9.4 Default Firmware Features

### 9.4.1 SD Card Test

If an SD card is connected to VS1005g, the firmware searches for a boot file called VS1005G.SYS. If found, VS1005g executes the code in that file.

If VS1005G.SYS is not found, the firmware tries to open the file TEST.MP3. If successful, that file is played back.

Note: Playback is provided only as a functional test. Playback quality is not indicative of VS1005g default playback quality.

Note: MP3 playback is not supported by VS8005g.

### 9.4.2 USB Mass Storage and Audio Device

If no SD card boot or test file was found, the firmware goes into USB detection mode.

When USB cable insertion is detected by the firmware, USB handling code is started. The internal clock is configured to 48 MHz (assuming XTALI = 12.288 MHz), the analog power is configured to 3.6 V, the USB peripheral is initialized, and the USB pull-up resistor is enabled.

If GPIO0\_6 has a pull-up resistor, VS1005g appears as an USB Audio Device. The audio device contains a HID interface which allows for code to be loaded to VS1005g with a custom program. If GPIO0\_6 has a pull-down resistor, VS1005g appears as an USB Mass Storage Device.

If during power-on the NAND FLASH contained a valid boot sector, the NAND FLASH disk will be used with the mass storage device. The NAND FLASH disk requires a filesystem-level formatting before it can be used.

## 9.5 Supported Audio Decoders

VS1005g ROM firmware supports decoding MP3 files (only VS1005 and VS1205). Support for other file formats is implemented as link libraries.

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

### 9.5.1 Supported MP3 (MPEG layer III) Decoder Formats

The decoder supports all MP3 samplersates and bitrates.

MPEG 1.0<sup>1</sup>:

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

MPEG 2.0<sup>1</sup>:

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

MPEG 2.5<sup>1</sup>:

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

<sup>1</sup> All variable bitrate (VBR) formats are also supported.

## 9.6 Supported Audio Encoders

VS1005g ROM firmware supports encoding MP3 files (only VS1205). Support for other file formats is implemented as link libraries.

### 9.6.1 Supported MP3 (MPEG layer III) Encoder Formats

VS1005g supports all MP3 samplerates and bitrates, in stereo and mono, both with constant bitrate (CBR) or variable bitrate (VBR). The following tables apply to constant bitrate.

Conventions	
Symbol	Description
++	Format is supported and recommended for this channel configuration and bitrate.
+	Format is supported.
x	Format is supported but use is strongly discouraged for quality reasons.
v	Format is supported but for best quality lower samplerate with same bitrate is recommended.
<	Format is supported but lower bitrate will give same quality.
-	Format exists but isn't supported.
	Format doesn't exist.

MPEG 1.0 layer III (MP3 full rates), stereo:

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

MPEG 2.0 & 2.5 layer III (MP3 low rates), stereo:

Samplerate / Hz	Bitrate / kbit/s, stereo													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24000	x	v	v	v	v	v	v	+	++	++	+	+	+	<
22050	x	v	v	v	v	v	+	+	+	+	+	+	<	<
16000	x	v	v	v	+	+	+	++	+	+	+	+	<	<
12000	v	v	v	+	++	++	++	+	+	+	+	+	<	<
11025	v	v	v	+	+	+	+	+	+	+	+	+	<	<
8000	++	++	++	++	+	+	+	+	+	+	+	<	<	<

MPEG 1.0 layer III (MP3 full rates), mono:

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

MPEG 2.0 & 2.5 layer III (MP3 low rates), mono:

Samplerate / Hz	Bitrate / kbit/s, mono													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24000	v	v	+	+	+	++	+	+	+	+	+	<	<	<
22050	v	v	+	+	+	+	+	+	+	+	+	<	<	<
16000	v	v	+	++	++	+	+	+	+	<	<	<	<	<
12000	v	v	++	+	+	+	+	+	<	<	<	<	<	<
11025	v	v	+	+	+	+	+	+	<	<	<	<	<	<
8000	++	++	+	+	+	+	<	<	<	<	<	<	<	<



## 10 VS1005g Peripherals and Registers

### 10.1 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.

### 10.2 VS1005g Memory Map

VS1005g's Memory Map is shown in Figure 8.

	Instruction (32-bit)	X (16-bit)	Y (16-bit)
0x0000	System vectors		
0x003F			
	User Instruction RAM	X Data RAM	Y Data RAM
0x1800		Stack	Stack
0x1880			
		X Data RAM	Y Data RAM
0x8000	Instruction ROM	X Data ROM	Y Data ROM
0xF400			
			USB RX Buffer
0xF800			
			USB TX Buffer
0xFC00			
			Hardware Registers
0xFFFF			

Figure 8: User's memory map.

## 10.3 VS1005g Peripherals

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

Peripheral address spaces are summarized in the following table:

VS1005g Peripheral Addresses	
Address	Device
<b>PLL clocked peripherals</b>	
0xFC00 - 0xFC1F	Interrupt controller
0xFC20 - 0xFC3F	DSP interface registers
0xFC40 - 0xFC4F	SPI 0
0xFC50 - 0xFC5F	SPI 1
0xFC60 - 0xFC65	10base-t ethernet controller
0xFC66 - 0xFC6C	DSP interface for peripheral data buffer
0xFC70 - 0xFC76	Reed-Solomon codes
0xFC77 - 0xFC7A	Nand flash interface
0xFC7B - 0xFC7F	SD card interface
0xFC80 - 0xFC9F	Hi-Speed USB
0xFCA0 - 0xFCBF	16-bit GPIO port 0
0xFCC0 - 0xFCDF	16-bit GPIO port 1
0xFCE0 - 0xFCFF	14-bit GPIO port 2
0xFD00 - 0xFD1F	S/PDIF
<b>XTAL clocked peripherals</b>	
0xFE00 - 0xFE1F	UART
0xFE20 - 0xFE3F	Watchdog
0xFE40 - 0xFE5F	FM and A/D interface
0xFE60 - 0xFE7F	I2S
0xFE80 - 0xFE9F	Timers
0xFEA0 - 0xFEBF	RTC interface
0xFEC0 - 0xFEDF	Control and configuration registers for 12 MHz clock

## 10.4 Interrupt Controller

VS1005g has 28 maskable interrupt vectors and 33 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.

VS1005g interrupt vectors			
Interrupt source	Interrupt vector	IRQ	Interrupt device
INT_SAR	27	32	10-bit ADC (SAR)
INT_PWM	26	31	Pulse width modulator
INT_REGU	25	30	Power button
INT_STX	23	28	S/PDIF transmitter
INT_SRX	22	27	S/PDIF receiver
INT_RDS	21	26	FM RDS
INT_RTC	20	25	RTC time alarm
INT_DAOSET	19	24	DAC offset
INT_SRC	18	23	DAC sample rate converter
INT_FM	17	22	FM interrupt (192 kHz)
INT_TIMER2	16	21	Timer 2

VS1005g interrupt vectors (continued)			
Interrupt source	Interrupt vector	IRQ	Interrupt device
INT_TIMER1	15	20	Timer 1
INT_TIMER0	14	19	Timer 0
INT_UART_RX	13	18	UART receive
INT_UART_TX	12	17	UART transmit
INT_I2S	11	16	I2S transmitter/receiver
INT_MAC2	10	15	A/D 3 (mono AD)
INT_GPIO2	9	14	Gpio port 2
INT_GPIO1	8	13	Gpio port 1
INT_GPIO0	7	12	Gpio port 0
INT_MAC0	6	11	A/D 1/2 (stereo AD)
INT_MAC1	5	10	FM decimation filter
INT_SPI1	4	9	SPI 1
INT_SPI0	3	8	SPI 0
INT_XPERIP	2	7	Reed-Solomon Decoder
INT_XPERIP	2	6	Reed-Solomon Encoder
INT_XPERIP	2	5	SD Card Controller
INT_XPERIP	2	4	Nand Flash Controller
INT_XPERIP	2	3	SPI Slave Controller
INT_XPERIP	2	2	Ethernet Controller
INT_USB	1	1	Full/Hi-Speed USB
INT_DAC	0	0	DAC

## 10.4.1 Interrupt Controller Registers

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
0xFC00	r/w	0	INT_ENABLEL0	Interrupt Enable Low 0
0xFC01	r/w	0	INT_ENABLEL1	Interrupt Enable Low 1
0xFC02	r/w	0	INT_ENABLEH0	Interrupt Enable High 0
0xFC03	r/w	0	INT_ENABLEH1	Interrupt Enable High 1
0xFC04	r/w	0	INT_ORIGIN0	Interrupt Origin 0
0xFC05	r/w	0	INT_ORIGIN1	Interrupt Origin 1
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

## 10.4.2 Enable INT\_ENABLE[L/H][0/1]

Interrupt enable registers selectively masks interrupt sources. Enable registers 0 contain sources 0..15 and enable registers 1 contain sources 16..31. Each source has two enable bits: one in the enable low and one in the enable high register. If both bits are zero, the corresponding interrupt source is not enabled, otherwise the bits select the interrupt priority.

High	Low	Priority
0	0	Source disabled
0	1	Priority 1
1	0	Priority 2
1	1	Priority 3

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 part of code where the interrupts have been disabled.

## 10.4.3 Origin INT\_ORIGIN[0/1]

If an interrupt source requests an interrupt, the corresponding bit in the interrupt origin register (ORIGIN0 or ORIGIN1) will be set to '1'. If an interrupt source is enabled (using ENABLE

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 bits in the interrupt origin register. All '1'-bits in the written value cause the corresponding bits in the interrupt origin register to be cleared. All zero-bits cause the corresponding bits in the interrupt origin register to keep their state. For example writing a value 0x00ff will clear the lowest eight bits in the interrupt origin register, while leaving the upper bits as-is.

#### 10.4.4 Vector INT\_VECTOR

The last generated vector value can be read from the vector register.

#### 10.4.5 Enable Counter INT\_ENCOUNT

The global interrupt enable/disable is used to control whether an interrupt request is sent to the processor or not. Whenever this 3-bit counter value 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, thus disabling further interrupts.

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

Don't write directly to INT\_ENCOUNT. Use INT\_GLOB\_DIS and INT\_GLOB\_ENA to manipulate the value of this register.

#### 10.4.6 Global Disable INT\_GLOB\_DIS

A write (of any value) to global disable register increases the global interrupt enable/disable counter by one. If the counter is zero, interrupt signal generation is enabled. When the interrupt arbitrator generates an interrupt request for VS\_DSP core, it automatically increases the counter. The user must write to the global enable register (once) to enable interrupts.

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

#### 10.4.7 Global Enable INT\_GLOB\_ENA

A write (of any value) to global enable register decreases the global interrupt enable/disable counter by one. If the counter is zero, interrupt generation is enabled.

The user must write to this register once in the end of the interrupt handler to enable further interrupts. This should be done in assembly language.

## 10.5 DSP Clock Domain Registers

### 10.5.1 General Purpose Software Registers

SW\_REG0, SW\_REG1, SW\_REG2 and SW\_REG3 are software registers for user purposes. They are zeroed 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

### 10.5.2 Peripheral I/O Control

VS1005g has three general purpose I/O ports. Ports 0 and 1 are 16-bits and port 2 is 14 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
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 VS1005g pins are at GPIO mode and all GPIOx\_MODE register are reset. If a peripheral mode is required the pin's GPIOx\_MODE bit must be set ('1').

### 10.5.3 PLL Clock Control

VS1005g 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 Bits			
Name	Bits	type	Description
CLK_CF_EXTOFF	15	r/w	S/PDIF peripheral clock gate control
CLK_CF_NFOFF	14	r/w	NF, SD and R-S peripherals 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_DIV1	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\_DIV1 controls the input divider of PLL's vco. If CLK\_CF\_DIV1 is set the vco input clock is divided by two. If CLK\_CF\_DIV1 is reset the vco input clock is the xtal 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 xtal 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. The pad must be in peripheral mode in order to output clock. The output driver has glitch removal.

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\_NFOFF and CLK\_CF\_USBOFF control peripheral clock gates. CLK\_CF\_NFOFF controls Nand flash, SD card, ethernet, Reed-Solomon codecs and peripeheral data buffer clocks. CLK\_CF\_EXTOFF controls S/PDIF peripheral clock. CLK\_CF\_USBOFF controls USB peripheral clock.



## 10.6 XTAL Clock Domain Registers

Peripheral control registers control the logic that is clocked with xtal clock (12 MHz).

### 10.6.1 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

ANA_CF0 Bits		
Name	Bits	Description
	15:11	Reserved, use '00000'
ANA_CF0_M1LIN	10	Line input mode select for ADC 1
ANA_CF0_M2LIN	9	Line input mode select for ADC 2
	8:6	Reserved, use '000'
ANA_CF0_M2MIC	5	Mic input mode select for ADC 2
ANA_CF0_LCKST	4	2 GHz vco lock status
ANA_CF0_LCKCHK	3	2 GHz vco lock check init
ANA_CF0_M1MIC	2	Mic input mode select for ADC 1
ANA_CF0_M2FM	1	FM input mode select for ADC 2
ANA_CF0_M1FM	0	FM input mode select for ADC 1

ANA\_CF0\_LCKCHK and ANA\_CF0\_LCKST are used to poll 2 GHz vco lock status. When ANA\_CF0\_LCKCHK is first set and reset the lock status can be read from ANA\_CF0\_LCKST. If ANA\_CF0\_LCKST remains set the 2 GHz VCO is locked.

For details on how to program register bits ANA\_CF0\_M[1/2]LIN, ANA\_CF0\_M[1/2]MIC, and ANA\_CF0\_M[1/2]FM, see Chapter 10.15.1 on Page 85.



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 reset disable
	11	Reserved, use '1'
ANA_CF1_DBG	10	Debug mode pin state
ANA_CF1_XTDIV	9	Input clock divider for 24 MHz xtal oscillator
ANA_CF1_SAR_ENA	8	SAR power down
	7	Reserved, Use '0'
ANA_CF1_DA_ENA	6	DAC power enable
ANA_CF1_DAGAIN	5:4	Reserved, use '00'
ANA_CF1_DRV_ENA	3	DAC driver power enable
	2	Reserved, use '0'
	1:0	Reserved, use "00"

ANA\_CF1\_XTDIV is the input clock prescaler control register. When register is set the input clock is divided by 2. ANA\_CF1\_SAR\_ENA, ANA\_CF1\_DA\_ENA and ANA\_CF1\_DRV\_ENA are analog module's enable signals. When register is set the module is enabled.

ANA_CF2 Bits		
Name	Bits	Description
	15:14	Reserved
ANA_CF2_TSTE	13	Hardware debug test enable, read only
ANA_CF2_VCMST	12	Ground buffer short circuit monitor
ANA_CF2_VCMDIS	11	Ground buffer driver short circuit protection disable
ANA_CF2_UTM_ENA	10	Hi-Speed USB UTM enable
ANA_CF2_LNA_ENA	9	Low Noise Amplifier enable
ANA_CF2_2G_ENA	8	2 GHz VCO enable
ANA_CF2_AMP1_ENA	7	Microphone amplifier 1 enable
ANA_CF2_AMP2_ENA	6	Microphone amplifier 2 enable
	5	Reserved, use '0'
ANA_CF2_HIGH_REF	4	Analog reference voltage, 1.2V (0) or 1.6V (1)
ANA_CF2_REF_ENA	3	Analog reference power enable
ANA_CF2_M3_ENA	2	ADC 3 power enable
ANA_CF2_M2_ENA	1	ADC 2 power enable
ANA_CF2_M1_ENA	0	ADC 1 power enable

ANA\_CF2 register controls several analog module power enables. Each module is enabled when the power enable register bit is set.

ANA_CF3 Bits		
Name	Bits	Description
ANA_CF3_480_ENA	15	480 MHz clock enable
ANA_CF3_UTMBIAS	14	USB pad bias enable
ANA_CF3_FMDIV[1:0]	13:12	FM divider selection 16, 20 or 24
ANA_CF3_DIV[1:0]	11:10	VCO divider select register
ANA_CF3_GAIN2[2:0]	9:7	ADC 2 gain register
ANA_CF3_GAIN1[2:0]	6:4	ADC 1 gain register
ANA_CF3_2GCNTR[3:0]	3:0	VCO center frequency register

ANA\_CF3\_FMDIV is the VCO divider selection register for FM receiver. When the register is set the VCO clock is divided by 20 (FM mode). When the register is reset the divider value is 16 (HS USB mode). ANA\_CF3\_FMDIV2 register selects the divider 24. In this divider mode the ANA\_CF3\_FMDIV should be set. The VCO frequency is therefore FM tuning frequency multiplied by 16, 20 or 24.

FM Divider Bits			
FMDIV[1]	FMDIV[0]	Divider	Description
1	1	24	FM frequency is VCO frequency divided by 24
0	1	24	Don't Use (reserved)
1	0	20	FM frequency is VCO frequency divided by 20
0	0	16	FM frequency is VCO frequency divided by 16

ANA\_CF3\_2GCNTR register is used to match VCO's center frequency to programmed value (CCF). ANA\_CF3\_DIV[1:0] controls the VCO's dividers. These dividers are used to set VCO's output frequency range.

ANA\_CF3\_480ENA is the 480 MHz clock driver enable for UTM. When set the clock driver is enabled.

ANA\_CF3\_GAIN1 and ANA\_CF3\_GAIN2 set the ADC 1 and 2 gains. ADC gain can be adjusted to four values.

ADC Gain	
Register value	Gain
001	20 dB (max)
010	17 dB
100	14 dB
000	11 dB (min), default after reset

## 10.6.2 Regulator and Peripheral Clock Control Registers

VS1005g has four internal regulators, one regulator for each power domain. The voltage can be adjusted in about 50mV step size. To save power some of the peripheral clocks can be switched off.

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_SNFVOLT <sup>1</sup>	11:7	Serial Flash voltage configuration
REGU_CF_SNFOFF	6	Serial Flash voltage regulator shutdown
REGU_CF_ADOFF	5	AD filter clock gate control
REGU_CF_FMOFF	4	FM demodulator clock gate control
REGU_CF_REGCK	3	Regulator latch enable
REGU_CF_AOFF	2	Analog voltage regulator shutdown
REGU_CF_IOOFF	1	IO voltage regulator shutdown
REGU_CF_COFF	0	Core voltage regulator shut down

<sup>1</sup> User should not modify this register if embedded serial flash is used.

REGU\_CF\_ADOFF and REGU\_CF\_FMOFF control the AD and FM peripheral clocks. When these registers are set the clocks are cut off.

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

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

## 10.7 Audio Playback Interfaces

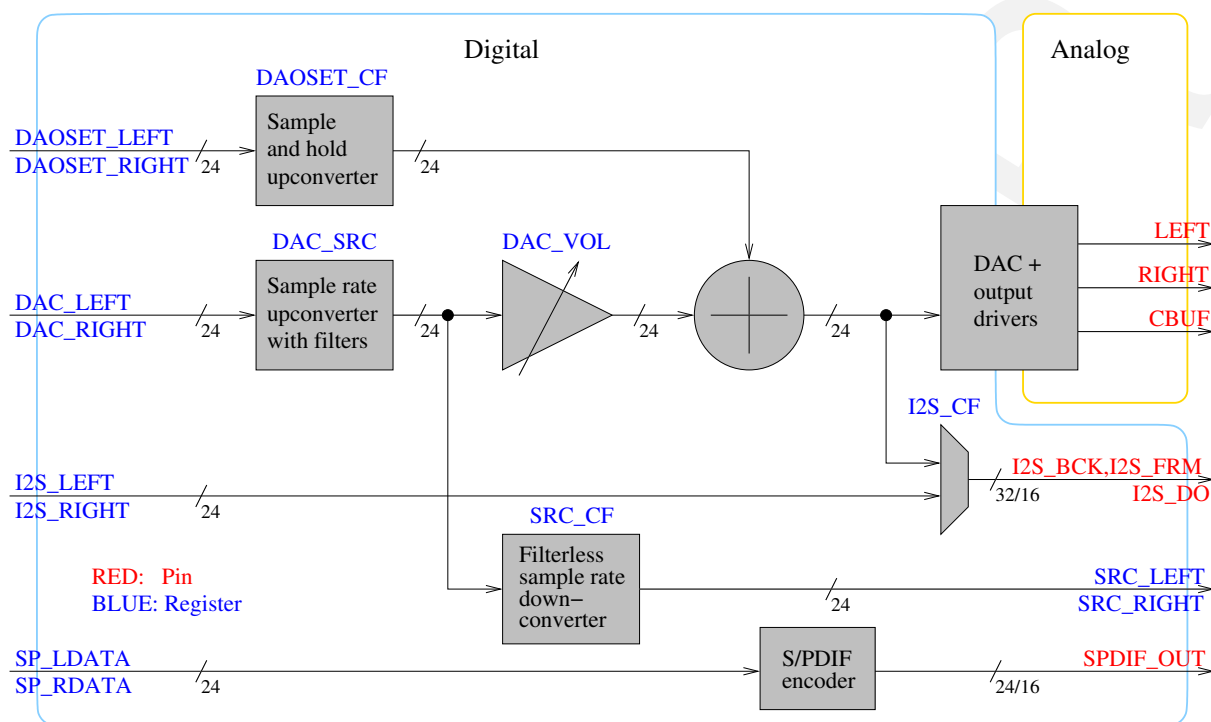


Figure 9: VS1005g playback (DA) audio paths.

The VS1005g playback audio paths are shown in Figure 9.

The nominal high-quality audio path begins 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.

## 10.7.1 Primary Audio Path 24-bit Sample Rate Upconverter with Filters

VS1005g 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

The DAC interpolator frequency is defined with registers DAC\_SRCH and DAC\_SRCL. 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 can be calculated from the equation

$$fs = (f_{clk}/2^{27}) * ifreq$$

where ifreq can have values from 1 to 1048575 (0xFFFFF) and  $f_{clk}$  is the xtal clock frequency. 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 the xtal frequency is 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.

## Configuring Analog DAC Modules

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

Analog Control Register example for DAC Operation			
Address	Register	Value	Description
0xFEBC	ANA_CF1	0x0048	DAC and output driver power down
0xFED2	ANA_CF2	0x0018	Reference voltage select and reference power down

## 10.7.2 Primary Audio Path Volume Control

In VS1005g 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.

## 10.7.3 Secondary Audio Path: DAC Offset Registers

In VS1005g 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:

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

dacoffset\_cf\_fs can have values from 0 to 4095 (0xFFF) and  $F_{clk}$  is the xtal clock frequency. E.g. value 0xFFF gives sample rate of  $12.288 \text{ MHz} / (0xFFF + 1) = 3.0 \text{ kHz}$ .

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.



## 10.7.4 Filterless Sample Rate Converter (SRC) Registers

VS1005g 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
Xtal 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 <sup>1</sup>	0 Hz - 96 kHz	Input sample rate
Output sample rate <sup>1</sup>	$0.97 \times FS_{in}$ - 192 kHz	Output sample rate
Filter delay <sup>2</sup>	19 input samples	
Gain	0.78	

<sup>1</sup> Assuming 12.288 MHz XTAL clock.

<sup>2</sup> 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]
0xFED8	r/w	0	SRC_LEFT	SRC left sample bits [23:8]
0xFED9	r/w	0	SRC_RIGHT_LSB[15:12]	SRC right sample bits [7:0]
0xFEDA	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.



## 10.8 SPI Peripherals

VS1005g has two SPI (serial-to-parallel) peripherals which can be configured as a master or a slave. Before SPIs can be used the VS1005g 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
- *Embedded Serial Flash disabled* : SYSTMPD\_SFENA bit reset when using SPI0 (also boot device)
- *Buffered SPI slave disabled* : ETH\_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.

SPI pins and their GPIOx_MODE register					
SPI id	VS1005g pin	Type	SPI pin	GPIO_MODE register	Description
SPI0	XCS0/GPIO1[0]	i/o	xcs	GPIO1_MODE[0]	Master/slave chip select
SPI0	SCLK0/GPIO1[1]	i/o	sclk	GPIO1_MODE[1]	Master/slave clock
SPI0	MISO0/GPIO1[2]	i/o	miso	GPIO1_MODE[2]	Master input / slave output
SPI0	MOSI0/GPIO1[3]	i/o	mosi	GPIO1_MODE[3]	Master output / slave input
SPI1	XCS1/GPIO1[4]	i/o	xcs	GPIO1_MODE[4]	Master/slave chip select
SPI1	SCLK1/GPIO1[5]	i/o	sclk	GPIO1_MODE[5]	Master/slave clock
SPI1	MISO1/GPIO1[6]	i/o	miso	GPIO1_MODE[6]	Master input / slave output
SPI1	MOSI1/GPIO1[7]	i/o	mosi	GPIO1_MODE[7]	Master output / slave input

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[10:0]	Configuration
0xFC41	0xFC51	r/w	0	CLKCF	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_SRESET	11	SPI software reset
SPI_CF_RXFIFOMODE	10	'0' = interrupt always when a word is received, '1' = Interrupt only when FIFO register full or CS deasserted with receive register full
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\_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 at 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.

## 10.9 Common Data Interfaces

VS1005g has a 3kB data buffer which is a dedicated peripheral memory. The memory can be configured to be used with:

- Ethernet interface
- Nand Flash Interface
- SD Card Interface
- Reed-Solomon Codecs

Block diagram of the data interfaces is shown in Figure 10.

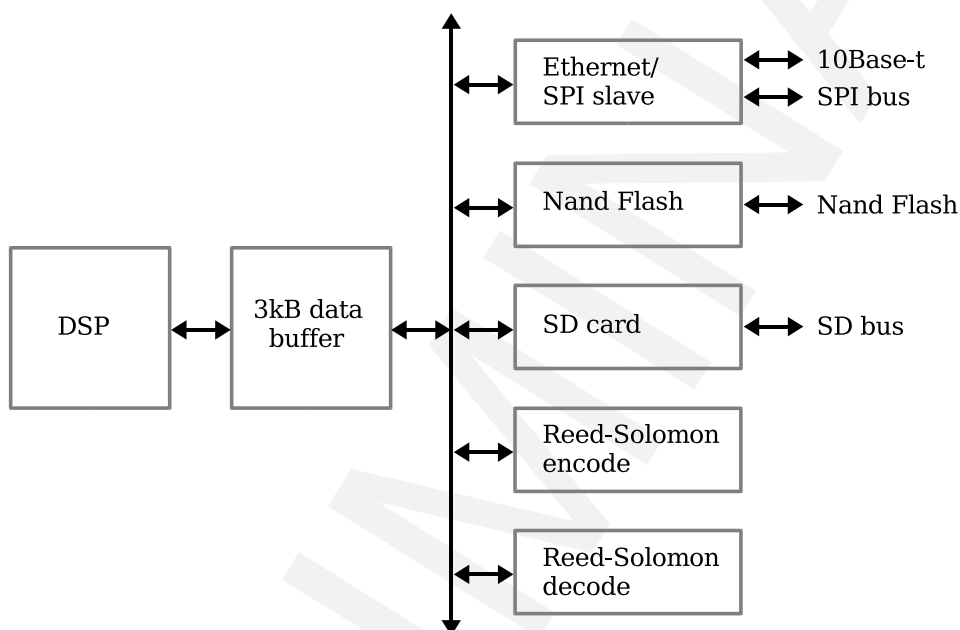


Figure 10: Block diagram of data interfaces.

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 and some idle cycles are required if several devices are enabled simultaneously. A guideline is that every 16th read cycle is idle. Dsp interface has Error Correctin 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_ODAT	15:12	RS_ODATA mux control
XP_CF_ECCRST	9	ECC reset
XP_CF_ECCENA	8	ECC enable
XP_CF_WRBUF	1	Data buffer write enable
XP_CF_RDBUF	0	Data buffer read enable

XP\_CF\_ODAT is a control register for RS\_ODATA register.

XP\_CF\_ECCRST and XP\_CF\_ECCENA control the ECC unit. XP\_CF\_ECCRST reset the unit when set. The register is reset automatically after one clock cycle.

XP\_CF\_ECCENA register enables the ECC calculation. column parity (CP) and line parity (LP) registers are modified when data is read from XP\_ODATA or written to XP\_IDATA register and XP\_CF\_ECCENA is set.

XP\_CF\_WRBUF and XP\_CF\_RDBUF 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\_ODATA) or written (XP\_IDATA) to memory.

Data interfaces can generate only one interrupt request for the DSP. The interrupt source is stored in interrupt status register.

XP\_STATUS register is used to track the interrupt source of the peripherals using data buffer memory. The interrupt requests are reset when the register bit is set by software. Interrupt sources are listed in the table.

XP_STATUS Bits		
Name	Bits	Description
XP_ST_INT_ENA	14	Interrupt enable for data buffer peripherals
XP_ST_ETXRB_FULL	13	Ethernet transmit ring buffer full
XP_ST_ETXRB_HALF	12	Ethernet transmit ring buffer half full
XP_ST_ERXRB_FULL	11	Ethernet receive ring buffer empty
XP_ST_ERXRB_HALF	10	Ethernet receive ring buffer half empty
XP_ST_SPIERR	9	SPI slave error, transfer was interrupted middle of byte
XP_ST_RSEC_RDY	8	Reed-Solomon decode error correction data ready
XP_ST_RSDEC_RDY	7	Reed-Solomon decode ready
XP_ST_RSENC_RDY	6	Reed-Solomon encode ready
XP_ST_SD_INT	5	SD card interface ready interrupt
XP_ST_NF_INT	4	Nand flash interface ready interrupt
XP_ST_SPI_STOP_INT	3	SPI slave stop interrupt, chip select to inactive state
XP_ST_SPI_START_INT	2	SPI slave start interrupt, chip select to active state
XP_ST_ETHRX_INT	1	Ethernet receiver new packet interrupt
XP_ST_ETHER_TX_INT	0	Ethernet transmitter ready interrupt

XP\_ST\_INT\_ENA is the peripheral interrupt enable. When set the interrupt requests are forwarded to interrupt controller. Interrupt requests in XP\_STATUS register are modified regardless the value of XP\_ST\_INT\_ENA.

The SPI slave error register (XP\_ST\_SPIERR) 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.



## 10.9.1 Ethernet Controller

VS1005g has a controller for interfacing 10base-t ethernet bus. Additionally this peripheral can be configured to SPI slave mode to be used with VLSI Solution's RF link. In this mode the SPI1 pins are used and they must be configured to peripheral mode with GPIO1\_MODE[7:4] registers.

Ethernet Controller Registers				
Reg	Type	Reset	Abbrev	Description
0xFC60	r/w	0	ETH_TXLEN	Ethernet transmitter packet length
0xFC61	r/w	0	ETH_TXPTR	Ethernet transmitter memory address pointer
0xFC62	r/w	0	ETH_RXLEN	Ethernet receiver packet length
0xFC63	r/w	0	ETH_RXPTR	Ethernet receiver memory address pointer
0xFC64	r/w	0	ETH_RBUF	Ethernet transmitter/receiver ring buffer configuration
0xFC65	r	0	ETH_RXADDR	Ethernet receiver memory address, 11 bits

ETH_TXLEN Bits		
Name	Bits	Description
ETH_TXLEN_META	15	SPI slave synhronization configuration
ETH_TXLEN_RX_BE	14	Set big endian SPI slave receiver bit order
ETH_TXLEN_TX_BE	13	Set big endian SPI slave transmitter bit order
ETH_TXLEN_LEN[11:0]	11:0	Ethernet transmitter packet size in bytes

ETH\_TXLEN\_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 in reset. In ethernet mode this register is don't care.

ETH\_TXLEN\_RX\_BE and ETH\_TXLEN\_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). In ethernet mode these registers are don't care.

ETH\_TXLEN[11:0] register is loaded with packet length (in bytes) before the transmitter is enabled. When transmitter is enabled this register is decremented after a byte has been sent. When the length register reached zero the transmitter returns to idle state. In SPI slave mode this register is zero.

ETH_TXPTR Bits		
Name	Bits	Description
ETH_TXPTR_SPI_TX_ENA	15	SPI slave transmitter enable
ETH_TXPTR_SPI_RX_ENA	14	SPI slave receiver enable
ETH_TXPTR_BUSY	13	Ethernet transmitter busy
ETH_TXPTR_START	12	Ethernet transmitter start-to-send packet
ETH_TXPTR_PTR[10:0]	10:0	Ethernet transmitter memory address pointer

ETH\_TXPTR\_SPI\_TX\_ENA and ETH\_TXPTR\_SPI\_RX\_ENA are the SPI slave mode enables for transmitter and receiver. SPI start and stop interrupts are generated even though these reg-



isters would be reset. It should be noted that when ETH\_TXPTR\_SPIRE is set the transmitter address pointer must be initialized to data start address. In ethernet mode these registers are don't care.

ETH\_TXPTR\_BUSY is the ethernet transmitter busy flag. In SPI slave mode this flag is set if transmitter is enabled and chip select line is in its active state (low).

ETH\_TXPTR\_START enables the ethernet transmitter. When this register is set the transmitter changes from idle to busy state and sends ETH\_TXLEN[11:0] number of bytes. Before this register is set the packet data must be stored in peripheral memory and tx address pointer and tx packet length registers must be configured. In SPI slave mode this register is zero.

ETH\_TXPTR[10:0] is the ethernet/SPI transmitter memory address pointer. This pointer is loaded with packet start address before transmitter is enabled.

ETH_RXLEN Bits		
Name	Bits	Description
ETH_RXLEN_SPIMODE	15	Peripheral mode select: Ethernet (0) / SPI slave mode (1)
ETH_RXLEN_SPIINVCLK	14	SPI slave transmitter clock configuration
ETH_RXLEN_LEN[11:0]	11:0	Ethernet receiver packet size in bytes

ETH\_RXLEN\_SPIMODE register configures the peripheral to ethernet mode or to SPI slave mode. When register is reset (default state) the peripheral is in ethernet mode.

ETH\_RXLEN\_SPIINTCLK 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. In ethernet mode this register is don't care.

ETH\_RXLEN\_LEN[11:0] register is loaded with ethernet/SPI receiver packet length counter when receiver returns from busy state to idle (packet end). Packet length is given in bytes.

ETH_RXPTR Bits		
Name	Bits	Description
ETH_RXPTR_CRCOK	15	Ethernet receiver crc status flag
ETH_RXPTR_NEWPKT	14	Ethernet receiver packet received flag
ETH_RXPTR_BUSY	13	Ethernet receiver busy
ETH_RXPTR_ENA	12	Ethernet receiver enable
ETH_RXPTR_PTR[10:0]	10:0	Ethernet receiver memory address pointer

ETH\_RXPTR\_CRCOK is the received packet crc status flag. Receiver sets the flag if the received packet crc was correct. Flag must be reset by user (write '1'). In SPI slave mode the crc flag is set if last four bytes were 0xFF.

ETH\_RXPTR\_NEWPKT is the flag for incoming packet. The receiver sets the flag when it changes its state from busy to idle. Flag must be reset by user (write '1'). In spi mode this register is zero.

ETH\_RXPTR\_BUSY is a busy flag for ethernet/SPI slave receiver. This receiver sets the flag when changes its state from idle to busy state.

ETH\_RXPTR\_ENA register places the ethernet receiver on hold for incoming packet when set. When packet start is detected the receiver switches from idle to busy state. Receiver address pointer must be configured before this register is set. In SPI slave mode this register controls the SPI receiver enable. When register is set the SPI transmit end automatically enables the SPI receiver.

ETH\_RXPTR[10:0] is the ethernet/SPI receiver memory pointer. This pointer is loaded with packet start address before receiver is enabled. When receiver changes its state from idle to busy this register is loaded to memory write address pointer register.

ETH_RBUF Bits		
Name	Bits	Description
ETH_RBUF_CLKCF	9:8	Reserved, use "00"
ETH_RBUF_TXENA	7	Ethernet receiver ring buffer enable
ETH_RBUF_TXCF	6:4	Ethernet receiver ring buffer configuration
ETH_RBUF_RXENA	3	Ethernet transmitter ring buffer enable
ETH_RBUF_RXCF	2:0	Ethernet transmitter ring buffer configuration

ETH\_RBUF\_TXENA and ETH\_RBUF\_RXENA are ring buffer enable registers for transmitters and receiver respectively. Ring buffer size is defined with ETH\_RBUF\_TXCF and ETH\_RBUF\_RXCF registers as explained in next table.

Ring buffer configuration bits			
CF register	Ring buffer size	Locked bits	Incremented bits
111-100	1024 words	[10]	[9:0]
011	512 words	[10:9]	[8:0]
010	256 words	[10:8]	[7:0]
001	128 words	[10:7]	[6:0]
000	64 words	[10:6]	[5:0]

ETH\_RXADDR register is the current memory address where receiver stores data. This register is loaded with ETH\_RXPTR[10:0] when new packet start is detected in bus.

Ethernet controller generates an interrupt each time a new packet is received or transmitter has finished sending a packet. 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.

## 10.9.2 Reed-Solomon Codec

VS1005g has a Reed-Solomon encoder and decoder for error correction e.g. from nand flash data. Reed Solomon is a forward error correction code which adds redundancy at the end of the message. The code word length  $n$  is defined as  $k + 2 \cdot t$  where  $k$  is the maximum number of data symbols (pay load) and  $2 \cdot t$  is the number of parity check symbols. The algorithm can fix up to  $t$  symbols from code word and detect  $2 \cdot t$  errors.

The Reed Solomon codec in VS1005g supports two different code lengths

- NF:  $n = 1023$  symbols and  $t = 4$ . This makes  $2 \cdot t = 8$  10-bit parity check symbols. The data symbols are interpreted as 8-bit symbols where the two msb bits are always zero.
- RF:  $n = 255$  symbols and  $t = 16$ . This makes  $2 \cdot t = 32$  8-bit parity check symbols. Also the data symbols are 8 bits wide.

VS1005g Reed Solomon Codecs				
Codec	Data Symbols (k)	Symbol Width	Check Symbols ( $2 \cdot t$ )	Typical Code Width (n)
NF	$\leq 1015$	8 / 10 bits	8 (10 bits)	518 + 8
RF	$\leq 223$	8 bits	32 (8 bits)	223 + 32

The NF codec is used for multi level cell (MLC) flash error detection and correction. Therefore the symbol width is limited to 8 bits for data symbols. The R-S encoder generates eight 10-bit parity check symbols (80 bits) which are stored with 518-byte user data. These check symbols are organized as 10 8-bit symbols which are stored to memory chip. The code word would therefore be a total of 528 bytes long.

NF Reed Solomon encoder check symbols are outputted as five 16-bit words (80 bits total). These check symbol words can be read from RS\_DATA port when the RS encoder has finished calculation. The check symbol organization is as listed in the table.

NF Reed Solomon Encoder Check Symbol Organization		
Check Symbol	RS Encoder output word bits	
0	word[0](1:0)	& word[0](15:8)
1	word[1](11:8)	& word[0](7:2)
2	word[1](5:0)	& word[1](15:12)
3	word[2](15:8)	& word[1](7:6)
4	word[3](9:8)	& word[2](7:0)
5	word[3](3:0)	& word[3](15:10)
6	word[4](13:8)	& word[3](7:4)
7	word[4](7:0)	& word[4](15:14)

RF Reed Solomon encoder outputs the 8-bit check symbols as 16 16-bit words. Here the 32 8-bit symbols are organized in big endian format. This codec provides a means to detect 32 symbol errors and to fix 16 symbols.

The NF Reed Solomon decoder makes it possible to detect 8 symbol errors and to fix a maximum of 4 symbols. The decoder first reads the data symbols and then the parity check words.

As the check symbols are 10 bits they must be organized into memory in this format before check symbol decoding is started. The encoder expects them to be in LSB bits (9:0) in consecutive memory locations. The decoder returns the number of total errors and the number of errors in data symbols. Only the data symbol errors are returned as location / magnitude pairs when Reed Solomon decoder has finished the calculation. These errors must be fixed by software to the code word by XORing magnitude to the error location data. The location / magnitude pairs are stored in memory.

Reed Solomon codecs use a shared interrupt source INT\_XPERIP. The source of interrupt is stored in register XP\_ST where the decoder has one bit for Reed Solomon encoder and two bits for Reed Solomon decoder. To enable the interrupts the XP\_ST\_INT\_ENA must be set.

Reed Solomon interrupts in XP_STATUS Register		
Name	Bits	Description
XP_ST_INT_ENA	14	Interrupt enable, active high
XP_ST_RSEC_RDY	8	Reed-Solomon decode error correction ready
XP_ST_RSDEC_RDY	7	Reed-Solomon decode ready
XP_ST_RSENC_RDY	6	Reed-Solomon encode ready

Reed-Solomon Registers				
Reg	Type	Reset	Abbrev	Description
0xFC70	r	0	RS_ST	Reed-Solomon status for encoder and decoder
0xFC71	r/w	0	RS_CF	Reed-Solomon control and configuration register
0xFC72	r/w	0	RS_EPTR	Reed-Solomon encoder memory pointer
0xFC73	r/w	0	RS_ELEN	Reed-Solomon encoder data length
0xFC74	r/w	0	RS_DPTR	Reed-Solomon decoder memory pointer
0xFC75	r/w	0	RS_DLEN	Reed-Solomon decoder data length
0xFC76	r	0	RS_DATA	Data read port

RS_ST Bits		
Name	Bits	Description
RS_ST_DERR	12:8	Number of errors in decoded code word
RS_ST_DFFAIL	6	Decoder fix algorithm found too many errors
RS_ST_DFRDY2	5	Decoder fix algorithm completed
RS_ST_DFBUSY	4	Decoder fix algorithm is calculating magnitude and location pairs
RS_ST_DFRDY1	3	Decoder fix algorithm part 1 completed
RS_ST_DFAIL	2	Code word errors can not be fixed
RS_ST_DERR	1	Code word has errors
RS_ST_DOK	0	Code word does not contain errors

RS\_ST\_DERR is the error count register. When R-S decoder fix algorithm has completed the number of location/magnitude pairs is stored to this register.

RS\_ST\_DFFAIL register is set when the decoder algorithm could not fix all the errors in code word.

RS\_ST\_DFRDY2, RS\_ST\_DFBUSY and RS\_ST\_DFRDY1 are monitoring the status of R-S

decoder. When both ready register are set the error correcting algorithm has completed and an interrupt request is generated.

RS\_ST\_DFAIL is set when a fatal error was encountered. It is not possible to restore code word. RS\_ST\_DFAIL is modified after code end was given (RS\_CF\_DEND set by user)

RS\_ST\_DERR flag is set if code word has errors. RS\_ST\_DERR is modified after code end was given (RS\_CF\_DEND was set by user). If this flag was set the error correcting algorithm is started automatically.

RS\_ST\_DOK flag is set if code word does not has errors. RS\_ST\_DOK is modified after code end was given (RS\_CF\_DEND was set by user).

RS_CF Bits		
Name	Bits	Description
RS_CF_DNF	13	R-S decoder nand flash mode select
RS_CF_D10B	12	R-S decoder 10-bit input data
RS_CF_DEND	11	R-S decoder code end
RS_CF_DSTR	10	R-S decoder code start
RS_CF_DENA	9	R-S decoder enable
RS_CF_DMODE	8	R-S decoder mode control
RS_CF_SEL[3:0]	7:4	R-S encoder parity select for RS_OPORT
RS_CF_ENF	3	R-S encoder nand flash mode select
RS_CF ESTR	2	R-S encoder code start
RS_CF_EENA	1	R-S encoder enable
RS_CF_EMODE	0	R-S encoder mode control

RS\_CF\_DNF selects between two data input modes. When set the decoder uses nand flash input data register as input. When reset the data is fetched from peripheral memory.

RS\_CF\_D10B selects between 10-bit and 8-bit input modes. Normally the symbols are 8-bit and two MSB zero bits are added. When RS\_CF\_D10B is set the symbols are fetched from peripheral memory as 10-bit and the two MSB bits are not zeroed. In 10-bit mode the data is in bits [9:0] and it is fetched from memory in word format. This bit is set when NF parity check symbols are decoded. When decoding the 10-bit check symbols the decoder does not generate RS decoder interrupt XP\_ST\_RSDEC\_RDY.

RS\_CF\_DEND is a code end register for decoder. When this register is set the decoder stops decoding current code word and the status can be read from RS\_ST register. If code word contained symbol errors the symbol error correction algorithm is started automatically. The location and magnitude pairs needed to fix corrupted symbols are placed in memory from RS\_DPTR address onwards. The RS\_DPTR value is not incremented during calculation and it holds the start address of the location / magnitude pairs in memory. The progress of the calculation is visible in RS\_ST register. When the location / magnitude pairs are calculated an XP\_ST\_RSEC\_RDY interrupt is generated. RS\_CF\_DEND register is automatically reset after one clock cycle.

RS\_CF\_DSTR initializes the R-S decoder i.e. starts a new decoding sequence. This register is reset automatically when first symbol is decoded.

RS\_CF\_DENA enables the R-S decoder. When RS\_CF\_DNF is set the decoder is decoding symbols as they are read from nand flash. If RS\_CF\_DNF is reset the decoder starts reading

symbols from peripheral memory from address RS\_DPTR onwards. The symbols are fetched from memory as 8-bit or 10-bit symbols but are always forwarded to decoder as 10-bit symbols where bits [9:8] are zero if RS\_CF\_D10B is reset. The decoder decodes RS\_DLEN number of symbols and then reset RS\_CF\_DENA. Also an XP\_ST\_RSDEC\_RDY interrupt request is generated.

RS\_CF\_DMODE register should be set when decoding nand flash data (10-bit NF). When reset the 8-bit code is used (RF).

RS\_CF\_SEL is used to select encoded parity symbols. The selected parity symbol can be read from RS\_DATA register.

RS\_CF\_ENF selects between two data input modes. When set the encoder uses nand flash output data register as input. When reset the data is fetched from peripheral memory.

RS\_CF\_ESTR initializes the R-S encoder i.e. starts a new encoding sequence. This register is reset automatically when first symbol is encoded. The encoder does not need encode-end register as the check symbols are updated on-the-fly and are readable from the RS\_DATA port after the current operation has finished.

RS\_CF\_EENA enables the R-S encoder. When RS\_CF\_ENF is set the encoder is encoding symbols as they are written to nand flash. If RS\_CF\_ENF is reset the encoder starts reading symbols from peripheral memory from address RS\_EPTR onwards. The symbols are fetched from memory as 8-bit data but are forwarded to encoder as 10-bit symbols where bits [9:8] are always zero. The big endian byte order is expected. The encoder encodes RS\_ELEN number of symbols and then reset RS\_CF\_EENA. Also an XP\_ST\_RSENC\_RDY interrupt request is generated.

RS\_CF\_EMODE register should be set when encoding nand flash data (10-bit). When reset the 8-bit code is used (RF).

RS\_EPTR and RS\_DPTR are the 11-bit address registers for Reed-Solomon encoder and decoder. The start address of data is written to these registers prior the encode or decode is enabled. The big endian byte order is expected.

RS\_ELEN and RS\_DLEN are the code length registers for the encoder and decoder. The length is given as the number of 8-bit symbols. For decoder the symbols can also be 10-bit parity check symbols.

RS\_DATA is a data read port for several data registers.

RS_OPORT Mux Control Bits	
XP_CF[15:12] register	Mux input
1111-1100	Reserved
1011	R-S decoder total errors (data + check symbols), 5 bits
1010-0100	Reserved
0011	R-S decoder CSF index, 10 bits
0010	R-S decoder BM index, 9 bits
0001	R-S decoder syndrome index, 10 bits
0000	R-S encoder parity word as selected with RS_CF[7:4], 16 bits



## 10.9.3 Nand Flash Interface

Nand Flash Controller Registers				
Reg	Type	Reset	Abbrev	Description
0xFC77	r/w	0	NF_CF	Nand flash configuration register
0xFC78	r/w	0	NF_CTRL	Nand flash control register
0xFC79	r/w	0	NF_PTR	Nand flash memory pointer
0xFC7A	r/w	0	NF_LEN	Nand flash data length register (bytes)

NF_CONF Bits		
Name	Bits	Description
NF_CF_SCKM	9	Slave mode clock active edge select
NF_CF_SLV	8	Slave mode enable
NF_CF_FLT	7	Nand flash output bus float enable
NF_CF_INT_ENA	6	Nand flash interface interrupt enable
NF_CF_WAITSTATES	5:0	Nand flash interface clock configuration

NF\_CF\_SCKM selects slave mode active clock edge. If set the data bus is read at rising edge of ready/busy line, when reset at falling edge.

NF\_CF\_SLV configures the nand flash interface to slave input mode. In slave mode the nand flash interface reads data from 8-bit bus and stores it to memory. The clock is the ready/busy input.

NF\_CF\_FLT leaves the data output bus (flash input bus) floating when set. When reset the bus is driven to low or high state.

NF\_CF\_INT\_ENA enables the nand flash interrupt request when set.

NF\_CF\_WAITSTATES configures the length of nand flash read enable and write enable pulses. The cycle time is  $2 \times (\text{NF\_CONF\_WS} + 1)$  dsp clock cycles.

NF_CF Bits		
Name	Bits	Description
NF_CTRL_RDY	4	Status of nand flash ready line
NF_CTRL_RDWRX	2	Read (1) or write (0) select
NF_CTRL_ENA	1	Start nand flash read or write
NF_CTRL_USEPERIP	0	Use peripheral memory

NF\_CTRL\_RDY register is monitoring the current state of nand flash ready/busy line. The line has pull-up and when it is in its low state the flash chip is busy.

NF\_CTRL\_RDWRX is a read or write select. When this register is set the operation is a nand flash read. When reset the nand flash interface writes to flash.

NF\_CTRL\_ENA starts nand flash read or write when set. When all bytes are transfered this register is reset and an interrupt request is generated.

NF\_CTRL\_DBUF configures nand flash interface to use peripheral memory when set. If NF\_CTRL\_USEPERIP is reset when nand flash interface is enabled, the data is read from XP\_IDATA register or written to XP\_ODATA register. This is a one byte transaction and big endian format is used.



NF_PTR Bits		
Name	Bits	Description
NF_PTR_RENA	14	Ring buffer enable for slave mode
NF_PTR_RCF	13:11	Ring buffer configuration for slave mode
NF_PTR_PTR[10:0]	10:0	Nand flash memory pointer

NF\_PTR[10:0] is the memory pointer register.

NF\_PTR\_RENA and NF\_PTR\_RCF configure a ring buffer for slave mode. In ring buffer mode only the lower address bits are modified and higher bits are locked. E.g. when 512 word buffer size is used the ring buffer uses memory addresses 0 - 511 when NF\_PTR[10:9] bits are 0b00, addresses 512-1023 when bits are 0b10 and so on.

Ring buffer configuration bits			
CF register	Ring buffer size	Locked bits	Incremented bits
111-100	1024 words	[10]	[9:0]
011	512 words	[10:9]	[8:0]
010	256 words	[10:8]	[7:0]
001	128 words	[10:7]	[6:0]
000	64 words	[10:6]	[5:0]

Ring buffer mode generates interrupt in the mid and end addresses of the buffer.

NF\_LEN defines the number of bytes that are read from or written to nand flash. The length is given in bytes. In read and write operations the data uses big endian format, i.e. the MSB part is transmitted first.

## 10.9.4 SD Card Interface

VS1005g 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_RDWRX	6	Read (1) or write (0) select
SD_CF_CMDDATX	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\_RDWRX register selects between read and write transfers.

SD\_CF\_CMDDATX register selects between command and data transfers.

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	3	data stop bit missing error
SD_ST_CRC16	2	crc16 error when reading data
SD_ST_CRC7	1	crc7 error when reading command response
SD_ST_NOSTARTB	0	timeout error when reading, no start bit

SD\_ST\_WAITSTATES configures the length of SD card clock cycle. The cycle time is  $2 \times (\text{SD\_ST\_WAITSTATES} + 1)$  dsp clock cycles.

SD\_ST\_RPM register 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 is set if stop bit was not found when reading data from SD card.

SD\_ST\_CRC16 is set if crc16 error was detected when reading data from SD card.

SD\_ST\_CRC7 is set if command response had a crc7 error.

SD\_ST\_NOSTARTB is set if start bit was not found during 256 SD clocks.

## 10.10 USB Peripheral

VS1005g 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 11

Simplified UTM module diagram is shown in Figure 12.

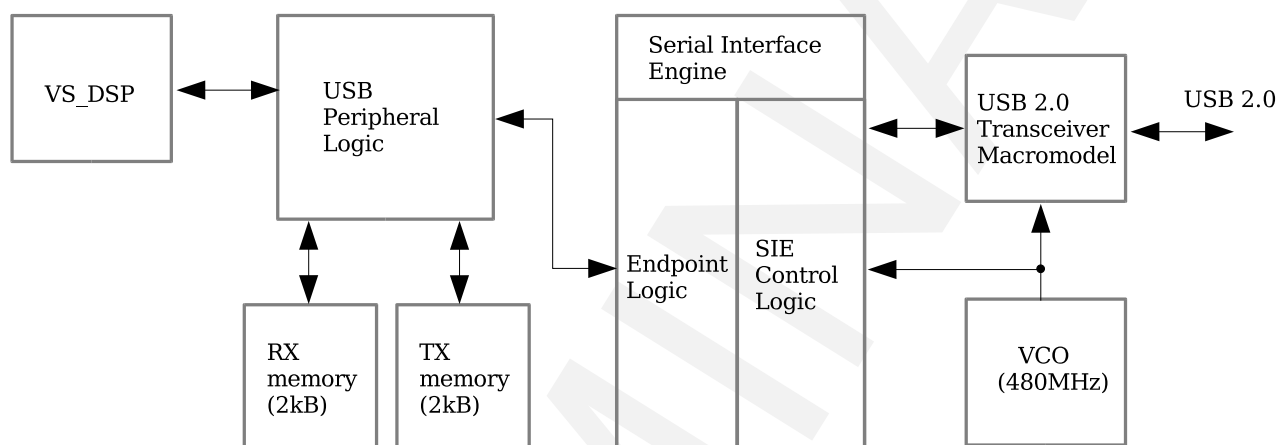


Figure 11: VS1005g USB block diagram.

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 VS1005g is based on 16-bit words).

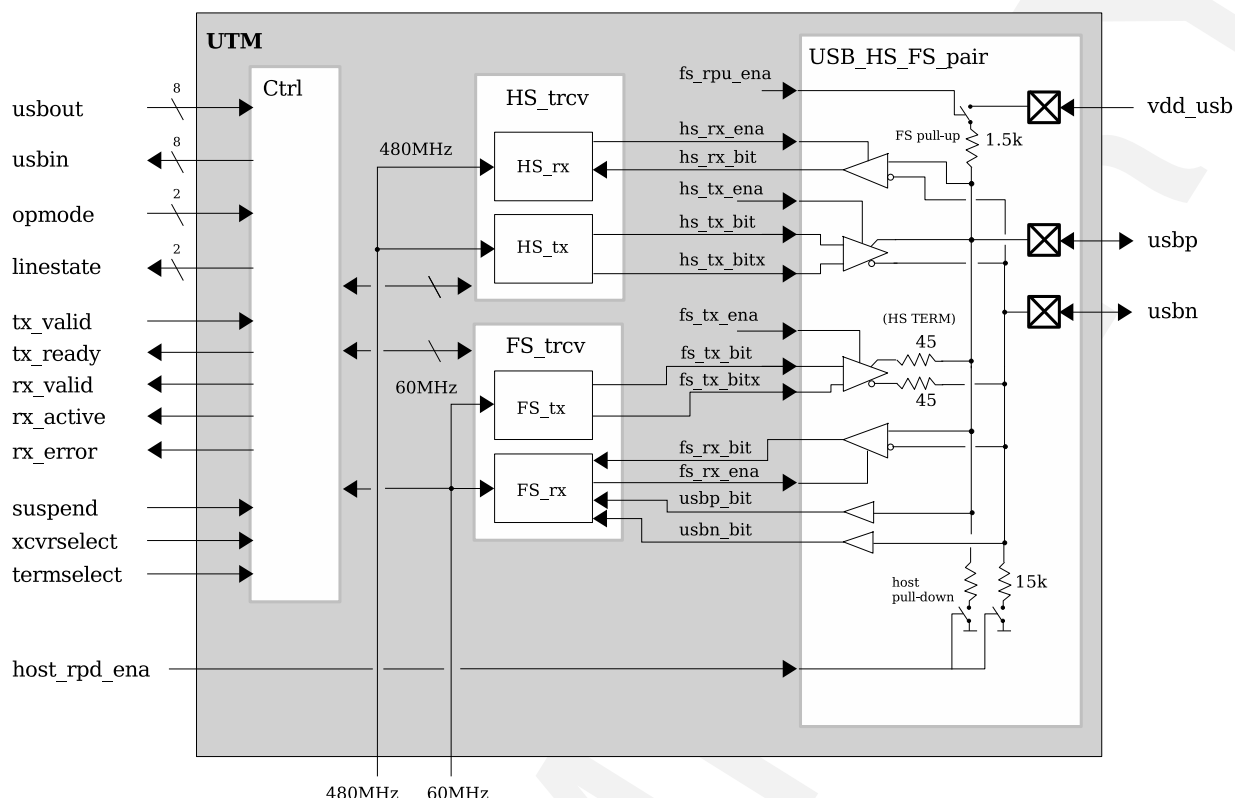


Figure 12: VS1005g UTM functional block diagram.

## 10.10.1 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)
	12:11	Reserved, use '0'
USB_CF_DTOGERR	10	Data Toggle error control (set to 0)
	9	Reserved, use '0'
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_RESUM	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
<b>EPnOUT (PC → Device) endpoint (0 .. 3) flags</b>		
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'
<b>EPnIN (Device → PC) endpoint (0 .. 3) flags</b>		
USB_EP_ST_ITYP	7:6	00=bulk 01=interrupt 11=isochronous
USB_EP_ST_INT_ENA	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'

## 10.10.2 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 by 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.

## 10.10.3 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 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

## 10.11 Interruptable General Purpose IO Ports 0-2

VS1005g 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 and 1 are 16-bits wide and GPIO port 2 is 14-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 monitors 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.

## 10.12 S/PDIF Peripheral

### 10.12.1 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, biphasic 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.

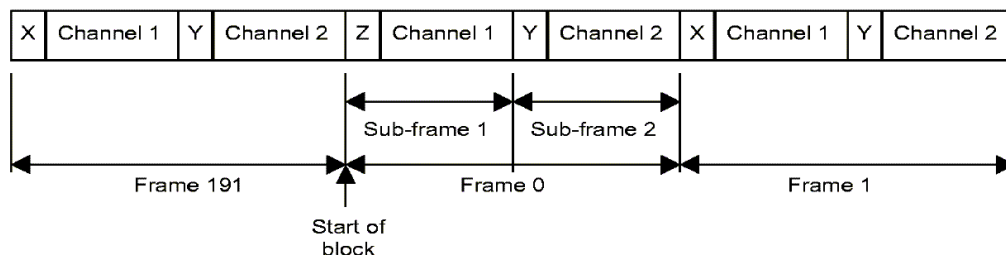


Figure 13: S/PDIF frame format.

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.

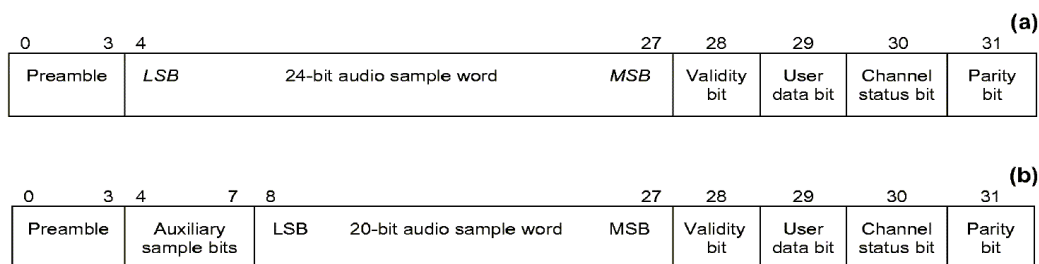


Figure 14: S/PDIF sub-frame format.

## 10.12.2 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	0x1FF0	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 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 resetted 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  $F_s$  according to master clock frequency. Default value is 8, resulting to  $F_s = 48$  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 $\neq$ 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 biphas 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_RX_BLCNT	15:8	r	Frame block count
SP_RX_FRCNT	7:0	r	Frame count

SP\_RX\_BLCNT is a fream frame block counter. This counter increment each time a new frame is received. It is zeroed after the 191th frame is received.

SP\_RX\_FRCNT is zeroed with every Z-preamble and incremented with every X-preamble.

S/PDIF Receiver uses two interrupts, a *frame received interrupt* and a *channel status chance interrupt*. Device issues an interrupt when it has received a frame. The interrupt is cleared when SP\_RX\_LDATA is read. Channel status change interrupt is set when at least one of the following conditions is satisfied:

- Channel status bit 0, selection between professional and consumer mode, is changed.
- Channel status bit 1, which indicates whether the sample word is linear PCM or not, is changed.
- Validity bit for either channel, left or right, is changed.

This interrupt is enabled by setting SP\_RX\_CF\_INT\_ENA bit.

### 10.12.3 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  $n \times 12.288$  MHz. 44.1 kHz sampling frequency is supported.

## 10.12.4 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_LD <small>ATA</small> _LSB	Left channel Audio sample bits 7-0
0xFD03	w	0	SP_LD <small>ATA</small>	Left channel Audio sample bits 23-8
0xFD04	w	0	SP_RD <small>ATA</small> _LSB	Right channel Audio bits sample 7-0
0xFD05	w	0	SP_RD <small>ATA</small>	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 1
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 Chan- nel 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\_LDADA 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\_WRDL. "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  $F_s = 48$  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.

## 10.13 UART Peripheral

RS232 UART implements a serial interface using rs232 standard.

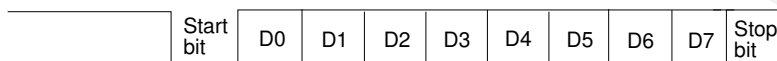


Figure 15: RS232 serial interface protocol

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.

### 10.13.1 UART Peripheral Registers

UART Registers				
Reg	Type	Reset	Abbrev	Description
0xFE00	r	0	UART_STATUS[3:0]	Status
0xFE01	r/w	0	UART_DATA[7:0]	Data
0xFE02	r/w	0	UART_DATAH[15:8]	Data High
0xFE03	r/w	0	UART_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 transmission 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 the XTAL, and  $f$  is the TX/RX speed in bps.

## 10.14 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  $65536 \times \text{WDOG\_CF}$  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 re-setting 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.

### 10.14.1 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

## 10.15 24-bit Analog to Digital Converters (ADC)

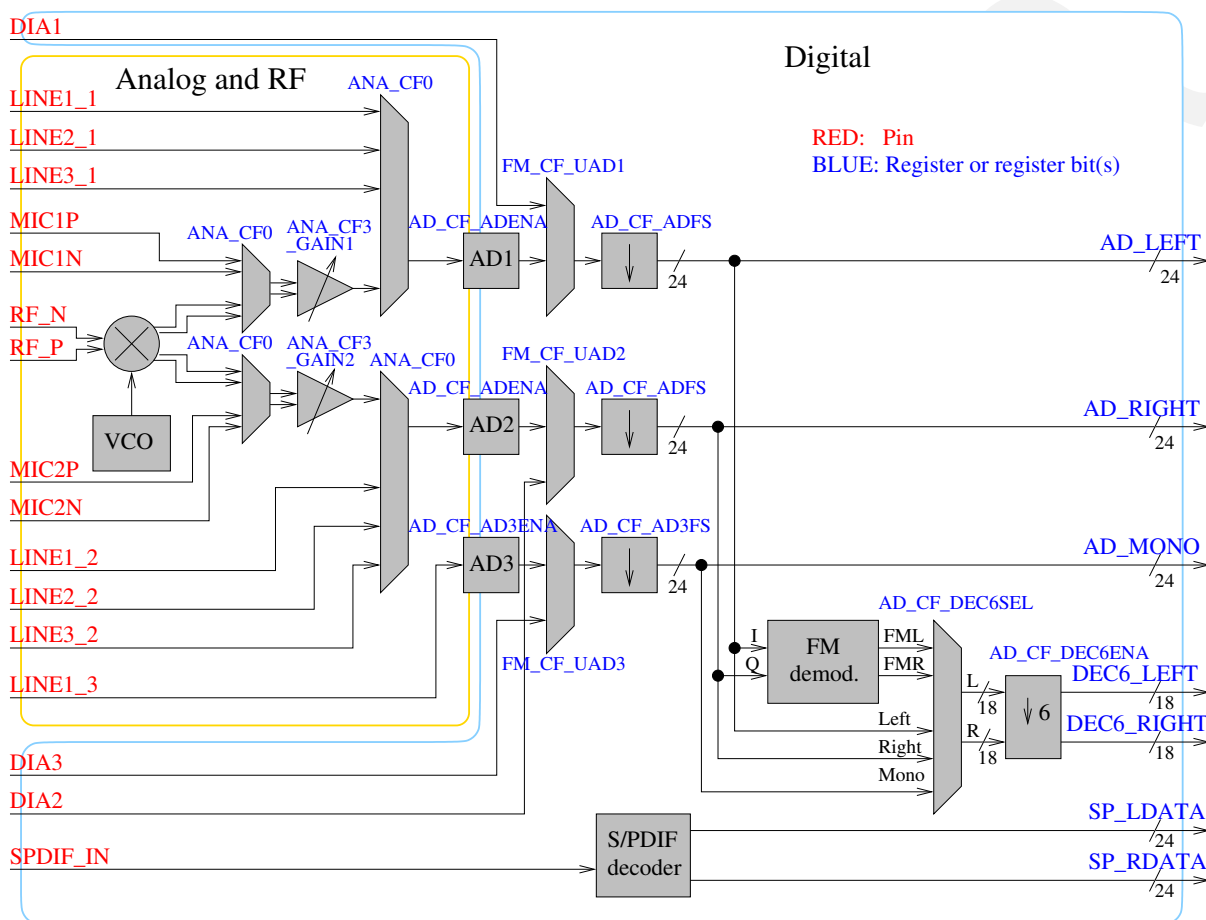


Figure 16: VS1005g recording (AD and FM) signal paths

VS1005g has three 24-bit AD input channels and an FM receiver. The signal paths for them are shown in Figure 16.

ADs 1 and 2 can be configured for mic or line input mode (stereo AD). AD 3 can be used as a line input (mono AD). When the FM receiver is used only AD 3 is available because the FM demodulator reserves the signal paths of ADs 1 and 2.

All of the logic is clocked directly with the XTALI (11-13 MHz). FM and AD digital peripheral clocks can be switched off to save power. In order to use FM or/and AD channels the master clock enable registers REGU\_CF\_ADOFF and REGU\_CF\_FMOFF must be reset. Analog and RF logic clocking is automatically switched on when blocks are enabled.

The front end of the ADs (analog section, i.e. ADC) operates always at XTALI / 2. The digital logic has a programmable sample rate. Sample rates are between 24 kHz and 192 kHz. It should be noted that the exact sample rates are XTALI-dependent and here it is assumed that XTALI = 12.288 MHz.

Note that the S/PDIF interface does not interact with any other audio paths. It is only included to show all available audio input paths, digital or analog.



AD filter's control and data registers are listed in following table.

A/D Control and Data Registers				
Reg	Type	Reset	Abbrev	Description
0xFE40	r/w	0	FM_CF	FM demodulator and AD filter configuration register
0xFE41	r/w	0	AD_CF	AD filter configuration register
0xFE46	r	0	AD_LEFT_LSB[15:8]	AD1 filter (left) channel bits [7:0]
0xFE47	r	0	AD_LEFT	AD1 filter (left) channel bits [23:8]
0xFE48	r	0	AD_RIGHT_LSB[15:8]	AD2 filter (right) channel bits [7:0]
0xFE49	r	0	AD_RIGHT	AD2 filter (right) channel bits [23:8]
0xFE4A	r	0	AD_MONO_LSB[15:8]	AD3 filter (mono) channel bits [7:0]
0xFE4B	r	0	AD_MONO	AD3 filter (mono) channel bits [23:8]
0xFE4E	r	0	DEC6_LEFT_LSB[15:14]	FM filter left channel bits [1:0]
0xFE4F	r	0	DEC6_LEFT	FM filter left channel bits [17:2]
0xFE50	r	0	DEC6_RIGHT_LSB[15:14]	FM filter right channel bits [1:0]
0xFE51	r	0	DEC6_RIGHT	FM filter right channel bits [17:2]

## 10.15.1 Configuring Signal Paths for ADC1 and ADC2

Analog signal paths for ADC1 and ADC2 are configured using register ANA\_CF0 (see Chapter 10.6.1 on Page 40).

ADC1 Signal Path Configuration					
Note: Always set ANA_CF2 bits ANA_CF2_REF_ENA and ANA_CF2_M1_ENA.					
ANA_CF0_			Input	Input	Set/Check Also
M1FM	M1LIN	M1MIC	Pin(s)	Pin Name(s)	
0	1	0	73	LINE1_1	
0	0	0	68	LINE2_1	
1	1	0	71	LINE3_1	
1	0	1	72,73	MIC1P & MIC1N	ANA_CF2_AMP1_ENA ANA_CF3_GAIN1[2:0]
1	0	0	75,76	RF_N & RF_P (I signal)	ANA_CF2_AMP1_ENA ANA_CF2_LNA_ENA ANA_CF2_2G_ENA ANA_CF3_FMDIV[1:0] ANA_CF3_DIV[1:0] ANA_CF3_GAIN1[2:0] ANA_CF3_2GCNTR[3:0] AD_CF_DEC6ENA AD_CF_DEC6SEL[1:0]

ADC2 Signal Path Configuration					
Note: Always set ANA_CF2 bits ANA_CF2_REF_ENA and ANA_CF2_M2_ENA.					
ANA_CF0_			Input	Input	Set/Check Also
M1FM	M1LIN	M1MIC	Pin(s)	Pin Name(s)	
0	1	0	72	LINE1_2	
0	0	0	67	LINE2_2	
1	1	0	70	LINE3_2	
1	0	1	70,71	MIC2P & MIC2N	ANA_CF2_AMP2_ENA ANA_CF3_GAIN2[2:0]
1	0	0	75,76	RF_N & RF_P (I signal)	ANA_CF2_AMP2_ENA ANA_CF2_LNA_ENA ANA_CF2_2G_ENA ANA_CF3_FMDIV[1:0] ANA_CF3_DIV[1:0] ANA_CF3_GAIN2[2:0] ANA_CF3_2GCNTR[3:0] AD_CF_DEC6ENA AD_CF_DEC6SEL[1:0]

Note: Decoding FM transmissions is only possible if both ADC1 and ADC2 are configured for FM reception (RF pins), and if ANA\_CF3\_GAIN1 = ANA\_CF3\_GAIN2.

## 10.15.2 Digital Filter Operation Modes

The FM\_CF register has four bits that have effect on AD functionality. FM\_CF\_ENABLE activates digital filters power down. When the register is set the digital filters are operable. This register bit can also be used to synchronize the stereo and mono AD filters when three channels are used with same sample rate (no phase error). To do this, clear FM\_CF\_ENABLE (if not already cleared), then set it again.

The input to digital filters can also be selected from external ADCs. With FM\_CF\_UAD1, FM\_CF\_UAD2 and FM\_CF\_UAD3 registers the filter's input can be taken from an external source (pins DIA1, DIA2, and DIA3, respectively). In this mode the AD input sample rate must be XTALI/2 or XTALI/4 and the input must be synchronized to VS1005g the XTALI oscillator clock. VS1005g can provide both the XTAL, XTAL/2, and XTAL/4 clocks to external circuits.

FM_CF Bits for Digital Filters (see Chapter 10.16.2 for other bits)		
Name	Bits	Description
FM_CF_UAD2	14	External input enable for stereo AD, right channel
FM_CF_UAD1	13	External input enable for stereo AD, left channel
FM_CF_UAD3	12	External input enable for mono AD
FM_CF_ENABLE	6	Software reset for AD and FM demodulator

The AD configuration register has bits to enable filters and to select sample rates. When the filter is enabled also the interrupt request is generated and forwarded to the interrupt controller. The decimation filter is included to decimate the demodulated FM signals down to 32 kHz sample rate but its input can be selected from other sources also. The filter's input bit width is 18 bits.

AD_CF Bits		
Name	Bits	Description
AD_CF_AD23_FLP	9	Flip AD2 and AD3 inputs
AD_CF_DEC6SEL[1:0]	8:7	Input selection for FM filter (decimation-by-6)
AD_CF_AD3FS[1:0]	6:5	Sample rate selection for AD filter 3 (mono AD)
AD_CF_ADFS[1:0]	4:3	Sample rate selection for AD filters 1 and 2 (stereo AD)
AD_CF_DEC6ENA	2	FM decimation filter enable
AD_CF_AD3ENA	1	AD filter 3 enable (mono AD, line input 3)
AD_CF_ADENA	0	AD filter 1 and 2 enable (stereo AD, line input 1 and 2)

AD\_CF\_AD23\_FLP register flips the input of filters 2 and 3.

FM decimation filter is used when FM is enabled and it decimates the FM signals to 32 kHz. The filter can be used in cascade with other filters. Register AD\_CF\_DEC6SEL is used to select filter input.

Decimation filter input selection	
AD_CF:DEC6SEL	Filter input
11 or 10	mono AD (left = right)
01	stereo ADs
00	FM demodulator

AD\_CF\_AD3FS and AD\_CF\_ADFS registers select the AD filters's decimation factor (sample rate). Exact sample rate is xtal dependent.

Decimation factor		
Register	Decimation factor	Sample rate @ XTALI = 12.288 MHz
11	256	24 kHz
10	128	48 kHz
01	64	96 kHz
00	32	192 kHz

AD\_LEFT, AD\_LEFT\_LSB, AD\_RIGHT, AD\_RIGHT\_LSB, AD\_MONO and AD\_MONO\_LSB are the three AD filters's output data registers. As a new data sample is calculated also an interrupt request is generated.

## 10.16 FM Receiver

The FM receiver in VS1005g is capable of receiving frequency modulated (FM) signals from 76 MHz to 108 MHz. The operation of FM receiver requires several modules:

- *RF modules* : VCO, LNA and Mixer
- *Analog modules* : Muxes, amplifiers and ADCs
- *Digital modules* : Digital filters and FM demodulator

As was shown in Figure 16. the FM receiver uses partially the same signal paths as the ADCs. When FM demodulator is used the stereo AD filter must be configured to 192 kHz sample rate and the decimation filter enabled with input selection from FM demodulator. Other modules must be powered up and FM path is selected to analog output.

### 10.16.1 Configuring RF and Analog Modules for FM Receiver Mode

The front end configuration of the FM receiver is shown in Figure 17. The VCO is digitally controlled and set to an FM band as is explained in section “Configuring FM Demodulator”.

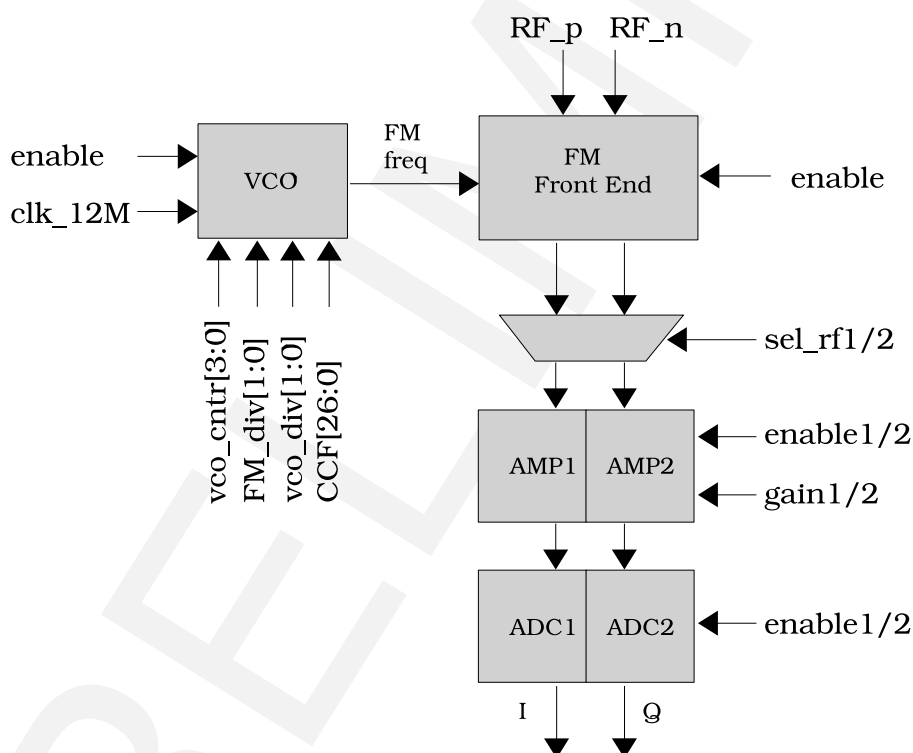


Figure 17: Block diagram of FM receiver, RF and analog section

To see how to configure the A/D converters for FM reception usage, see Chapter 10.15.1 on Page 85.

## 10.16.2 Configuring FM Demodulator

The FM demodulator has several configuration registers that must be initialized in order to receive an FM channel. FM demodulator's control and data registers are listed in next table.

FM Control and Data Registers				
Reg	Type	Reset	Abbrev	Description
0xFE40	r/w	0	FM_CF	FM demodulator control
0xFE41	r/w	0	AD_CF	AD filter configuration
0xFE42	r/w	0	FMPLL_LO	FM PLL carrier frequency bits [15:0]
0xFE43	r/w	0	FMPLL_HI	FM PLL carrier frequency bits [28:16]
0xFE44	r/w	0	FMCCF_LO	Carrier center frequency bits [15:0]
0xFE45	r/w	0	FMCCF_HI[10:0]	Carrier center frequency bits [26:16]
0xFE4E	r	0	DEC6_LEFT_LSB[15:14]	FM filter left channel bits [1:0]
0xFE4F	r	0	DEC6_LEFT	FM filter left channel bits [17:2]
0xFE50	r	0	DEC6_RIGHT_LSB[15:14]	FM filter right channel bits [1:0]
0xFE51	r	0	DEC6_RIGHT	FM filter right channel bits [17:2]
0xFE52	r	0	RDS_DATA[15:0]	FM RDS data
0xFE53	r	0	RDS_CHK[12:0]	FM RDS checkwork and block status
0xFE5B	r	0	FM_PHSCL	FM I/Q phase error scaling factor

FM\_CF register is a configuration register which is used to select demodulator operation modes. The FMCCF\_HI and FMCCF\_LO are used to tune FM receiver to a certain channel. The FMPLL\_HI and FMPLL\_LO registers are used to match xtal frequency to the stereo subcarrier frequency (38 kHz).

FM_CF Bits		
Name	Bits	Description
FM_CF_UAD2	14	AD2 input selection
FM_CF_UAD1	13	AD1 input selection
FM_CF_UAD3	12	AD3 input selection
	11:8	Reserved, Use "000"
FM_CF_PHCOMP	7	Enable for FM I- and Q-signal scaling
FM_CF_ENABLE	6	Enable & synchronize AD filters and FM demodulator
FM_CF_RDSSYNC	5	FM RDS forced to keep synchronization
FM_CF_MONO	4	FM receiver mono (1) / stereo (0) selection
FM_CF_DEEMP	3	FM de-emphasis filter configuration 75 $\mu$ s or 50 $\mu$ s
FM_CF_RDSENA	2	FM RDS enable
FM_CF_CCFLCK	1	FM carrier lock enable
FM_CF_FM_ENA	0	FM demodulator enable

In FM mode the registers FM\_CF\_UAD2 and FM\_CF\_UAD1 must be reset.

FM\_CF\_PHCOMP is the enable signal for FM input scaling.

FM\_CF\_ENABLE is the global enable for FM demodulator and AD filters. This signal can also be used to resynchronize the mono/stereo AD filters. The filters are synchronized when this register bit state changes from 0 to 1.

FM\_CF\_RDSSYNC forces the RDS decoder to keep current symbol synchronization. When sync search is enabled (i.e. FM\_CF\_RDSSYNC is reset) the RDS decoder tries to find best symbol synchronization at all times, even when the FM signal is lost.

FM\_CF\_MONO register selects between mono and stereo receive modes. When set the mode is mono.

FM\_CF\_DEEMP register selects between 75 $\mu$ s (North America) or 50 $\mu$ s (Europe, Australia) de-emphasis filters. When set the de-emphasis is 75 $\mu$ s.

FM\_CF\_RDSENA register enables the rds calculation logic when set.

FM\_CF\_CCFLCK register enables automatic FM fine tuning when set. When reset the FM band frequency is always at fixed value (as defined in FMCCF register).

FM\_CF\_FM\_ENA is the FM demodulator enable. The register must be set when FM is used.

To receive in stereo mode the FM\_PLL registers must be initialized correctly. These registers (FMPLL\_HI and FMPLL\_LO) set the FM stereo carrier PLL frequency. This factor is xtal dependent and is defined as:

$$pll\_factor = \frac{(64 \times 2^{28} \times 38000Hz)}{XTAL\_freqHz}$$

PLL value examples for most typical xtals	
xtal frequency	FMPLL register
12.0 MHz	54402918 = 0x033E 1F66
12.288 MHz	53127850 = 0x032A AAAA
13.0 MHz	50218079 = 0x02FE 445F

FMCCF\_HI and FMCCF\_LO are used to set FM tuning frequency (FM Carrier Center Frequency). These registers hold a 27-bit signed value which controls the frequency inside the selected VCO center frequency range. This VCO center frequency is set with divider registers ANA\_CF3\_DIV[1:0] and ANA\_CF3\_FMDIV[1:0].

VCO Divider Register	
ANA_CF3_DIV[1:0]	VCO Divider
"00"	36
"01" = "10"	30
"11"	25

FM divider	
ANA_CF3_FMDIV[1:0]	FM divider
"00"	16
"10"	20
"11" = "01"	24

The VCO frequency is 24, 20 or 16 times the FM tuning frequency, i.e. for 95.0 MHz FM channel the VCO frequency must be set to 1.900GHz. The target VCO frequency can be calculated from

equation :

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

and the FM channel frequency can be given as:

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

For Hi-Speed USB FMCCF registers must be reset when 12.000MHz xtal is used. When 12.288MHz xtal is used the registers are initialized to 0xFF87, 0xFFFF (-7864321). FM\_CF register is initialized to value 0x0041. This makes VCO frequency of 1.92GHz which results to 480 MHz USB clock.

FM_PHSCL Bits		
Name	Bits	Description
FM_PHSCL_I[7:0]	15:8	I scaling factor
FM_PHSCL_Q[7:0]	7:0	Q scaling factor

FM\_PHSCL register is used to compensate I and Q signal's phase and amplitude error. This error depends from several factors and values should be calculated for each FM band. The compensation logic is enabled when FM\_CF\_PHCOMP register is set. Typical values are 111 for I-scaling and 137 for Q-scaling.

DEC6\_LEFT, DEC6\_LEFT\_LSB, DEC6\_RIGHT and DEC6\_RIGHT\_LSB are the FM demodulator output data registers. Sample rate @12.288 MHz is 32 kHz (xtal / 384).

### 10.16.3 Radio Data System (RDS)

FM demodulator includes an RDS module. This module decodes the RDS bits from baseband signal to form bit groups. When a full block is decoded the 16-bit data and 10-bit checkword are stored to registers and an RDS-interrupt is generated. RDS data structure is shown in Figure 18. RDS data rate is 1187.5 bits per second.

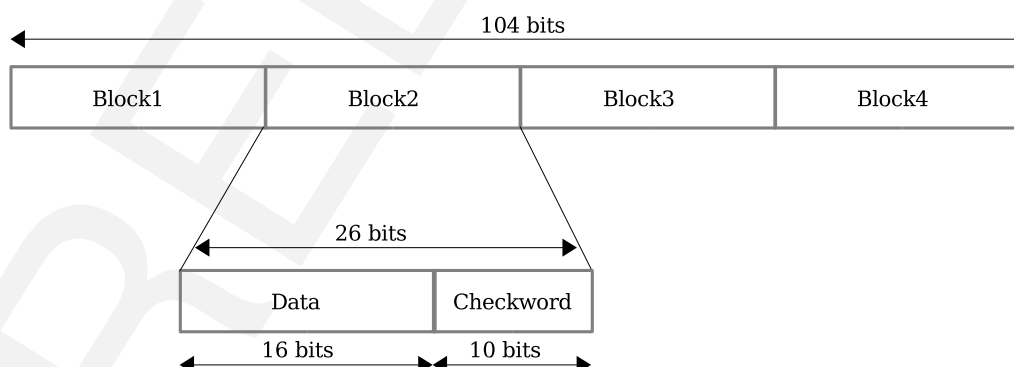


Figure 18: RDS data structure

RDS module's control bits are in register FM\_CF.



RDS Control Bits in Register FM_CF (address 0xFE40)		
Name	Bit Index	Description
FM_CF_RDSSYNC	5	FM RDS forced to keep current bit synchronization
FM_CF_RDSENA	2	FM RDS enable

FM Control and Data Registers				
Reg	Type	Reset	Abbrev	Description
0xFE52	r	0	RDS_DATA[15:0]	FM RDS data register
0xFE53	r	0	RDS_CHK[12:0]	FM RDS checkwork and block status

RDS\_DATA and RDS\_CHK registers store the last decoded RDS block. RDS module finds automatically bit and block synhronization but it does not do any data correction if crc errors exist. This must be done by software.

RDS_CHK Register Bits			
Reg	Bit index	Name	Description
RDS_CHK_CHKW	12:3	CHECKWORD	Checksum of the last received block data
RDS_CHK_ST	2	STATUS	Validity of the last received block
RDS_CHK_BLK	1:0	BLOCK	Index of the last received block, 0-3

## 10.17 I2S Peripheral

VS1005g 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 19.

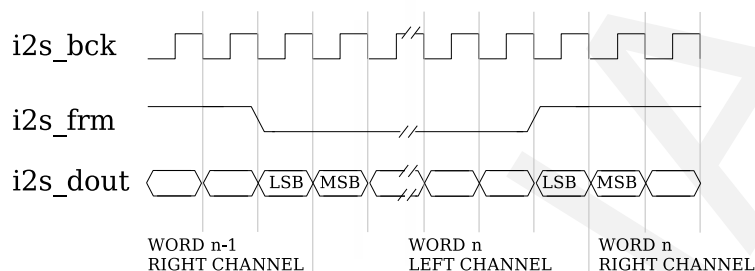


Figure 19: I2S frame format.

### 10.17.1 I2S Peripheral Registers

I2S Registers				
Reg	Type	Reset	Abbrev	Description
0xFE60	r/w	0	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[32:16]
0xFE63	r/w	0	I2S_RIGHT_LSB	Right data bits[15:0]
0xFE64	r/w	0	I2S_RIGHT	Right data bits[32:16]

I2S_CF Bits		
Name	Bits	Description
I2S_CF_32B	13	32-bit mode (1) / 16-bit mode (0) select
I2S_CF_INTENA	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	5	I2S peripheral mode select (dsp or SRC out)
I2S_CF_FS[1:0]	4:3	I2S sample rate selection
I2S_CF_ENA	2	I2S peripheral enable
I2S_CF_ENAMCK	1	I2S master clock (12 MHz) pad driver enable
I2S_CF_MASTER	0	I2S master (1) / slave (0) mode select

I2S\_CF\_MASTER bit is used to select between master (1) and slave (0) modes. In master mode the VS1005g 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 xtal oscillator clock of VS1005g.

I2S\_CF\_ENA is the transmitter and receiver enable signal. When set the receiver and transmitter enter the active state.

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 12.288 MHz xtal 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 and SRC mode. In dsp mode the data is transferred from left and right data registers. In SRC mode the 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-lsb bits of data.

## 10.18 Timer Peripheral

VS1005g 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.

### 10.18.1 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[2:0]	Timer enable
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 12MHz 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_T2	2	Enable timer 2
TIMER_ENA_T1	1	Enable timer 1
TIMER_ENA_T0	0	Enable timer 0

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.

## 10.19 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 VS1005g 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.

### 10.19.1 RTC Peripheral Registers

RTC Interface Registers				
Reg	Type	Reset	Abbrev	Description
0xFEAO	r/w	0	RTC_LOW	RTC data register bits [15:0]
0xFEAl	r/w	0	RTC_HIGH	RTC data register bits [31:16]
0xFEAl2	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 executed. The user must reset the RTC\_EXEC register after this time has elapsed.

RTC\_GSCK is used to generate RTC memory clock. When RTC\_GSCK is set the rtc\_if generates one clock pulse for memory store. RTC\_EXEC must be set during this operation. Rtc\_if resets this register automatically.

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
RTC_I_MEM_WR	35	1/12 MHz	Write to rtc memory
RTC_I_MEM_RD	3A	1/12 MHz	Read from RTC memory



## 10.20 10-Bit Successive Approximation Register Analog-to-Digital Converter (SAR)

VS1005g has a 10-bit ADC with following features:

- Successive Approximation Register conversion (SAR)
- Up to 5 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	Address	Value	Description
ANA_CF2	0xFED2	0x0008	Analog reference
ANA_CF1	0xFECD	0x0100	SAR power down

SAR operation is controlled with configuration register and the 10-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[9:0]	10-bit SAR data register

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 10-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[1:0] is used to select interface clock speed. The SAR clock is xtal clock / (2 x divider x 16).

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	Package Pin	Description
12	0xC	67	aux0
10	0xA	Internal	Divided VHIGH
7	0x7	68	aux1
6	0x6	Internal / pin 64 RTCVDD <sup>1</sup>	RTC voltage
5	0x5	Internal / pin 15 CVDD <sup>1</sup>	Core voltage CVDD
4	0x4	71	aux4
2	0x2	70	aux3
0	0x0	69	aux2

<sup>1</sup> Maximum allowed voltage is 1.95V

Other values are not allowed.

## 10.21 Pulse Width Modulation Unit

VS1005g has a PWM output which can be programmed to generate any pulse width within 256 xtal clock periods.

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

PWM\_FRAME[7:0] 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\_PULSE[7:0]. If PWM\_PULSE is zero the output is always zero. If PWM\_PULSE > PWM\_FRAME the output is always at logic high state.

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

In VS1005g power-up as the PWRBRTN pin is high the pwm output generates an oscillation for external powering circuitry. The oscillation requires 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 k $\Omega$		Value of external pull-up resistor
Start-up frequency		370 kHz		Start-up oscillation frequency

## 10.22 Special Features

### 10.22.1 Software Protection

VS1005g hardware supports software protection. Two registers control the hardware debugging and serial flash access. To use on-circuit debugging it must first be enabled by a software register. After power-up the debugger is disabled. The register can be set and reset by software. The access to serial flash can be disabled with a software register after boot-up. When serial flash is disabled it can not be enabled by software. To re-enable it requires system reset (SYSRST).



VS1005g JTAG instruction codes		
Instruction	IR register	Description
BYPASS	"0000"	Places jtag to bypass mode. In bypass mode there is one clock cycle delay between tdi and tdo.
IDCODE	"1111"	Places jtag's 32-bit ID code register between tdi and tdo.

The snoop module supports up to 8 breakpoints which can be programmed to trigger at data/address events. Snooper's control and status registers are

- Enable register (SENA)
- 16-bit event count register (ECNT)
- 16-bit instruction address register (BADDR)

SENA register enables the snoop module when set. The register is reset when breakpoint interrupt is triggered and all snoop logic is halted. ECNT register is a decrementing counter which is decremented by one at each breakpoint event. When register is zero and a breakpoint event occurs, a breakpoint interrupt is generated. BADDR register stores the instruction address when the breakpoint interrupt is generated.

Each breakpoint has three configuration registers:

- Configuration register
- Address register
- Data register

Breakpoint configuration register is used to set-up a breakpoint.

Breakpoint Configuration Register Bits		
Name	Register Bit	Description
Status	7	Breakpoint triggered flag
Bus Type	6:5	X/Y/I bus selection
Access Type	4:3	Fetch/Read/Write access type selection
Condition Type	2:0	Breakpoint condition selection

Breakpoint status bit is set when the breakpoint triggers an interrupt.

Breakpoint Bus Type Bit Configuration		
Value	Bus	Description
'00'	I	Breakpoint at I-bus
'01'	X	Breakpoint at X-bus
'10'	Y	Breakpoint at Y-bus
'11'	Illegal	Don't use

Breakpoint Access Type Bit Configuration		
Value	Register Bit	Description
'00'	Disabled	Breakpoint is disabled
'01'	Fetch/Read	Breakpoint set to snoop read accesses
'10'	Write	Breakpoint set to snoop write accesses
'11'	Read or Write	Breakpoint set to snoop both the read and write accesses

Breakpoint Condition Type Bit Configuration		
Value	Bus	Description
'000'	Disabled	Breakpoint disabled
'001'	Any	Match only address
'010'	=	X/Y-bus data EQUAL to snoop breakpoint data
'011'	≠	X/Y-bus data NOT EQUAL to snoop breakpoint data
'100'	< signed	Signed comparison of X/Y-bus data LESS THAN snoop breakpoint data
'101'	≥ signed	Signed comparison of X/Y-bus data GREATER THAN OR EQUAL to snoop breakpoint data
'110'	Illegal	Don't use
'111'	Illegal	Don't use

The hardware debugger requires the VLSI JTAG connector and Integrated Development Environment VSIDE for full debug capabilities. For further information about the hardware debugger connect VLSI technical support.



## 12 Document Version Changes

This chapter describes the most important changes to this document.

### Version 0.42 & 0.41, 2012-05-31

- Updated figures in Chapter 5.1, *Packages*. Package thickness has changed from 0.9 mm to 0.8 mm.
- Changed function descriptions for pins 67-68 and 70-73 in Chapter 5.3, *VS1005g Pin Descriptions*.

### Version 0.4, 2012-05-30

- Added Chapter 4.10, *Power Button Characteristics*.
- Renamed Chapter 10.15 from *Line and Mic Inputs* to *24-bit Analog to Digital Converters (ADC)* and added documentation to it. Among other things, added Chapter 10.15.1, *Configuring Signal Paths for ADC1 and ADC2*.
- Redrawn Figure 16, VS1005g recording (AD and FM) signal paths, on page 84, and Figure 9, VS1005g playback (DA) audio paths, on page 44. Also updated accompanying documentation.
- Some register names still adjusted.

### Version 0.3, 2012-04-04

- Checked and corrected register names.
- Modifications to Chapter 9, *Firmware Operation*.

### Version 0.2, 2012-03-16

- First release version.

## 13 Contact Information

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

Fax: +358-3-3140-8288  
Phone: +358-3-3140-8200  
Email: [sales@vlsi.fi](mailto:sales@vlsi.fi)  
URL: <http://www.vlsi.fi/>

