DATASHEET

SGTL5000

Low Power Stereo Codec with Headphone Amp

DESCRIPTION

The Low Power Stereo Codec with Headphone Amp from Freescale is designed to provide a complete audio solution for portable products needing line-in, mic-in, line-out, headphone-out, and digital I/O. Deriving it's architecture from best in class Freescale integrated products that are currently on the market, the SGTL5000 is able to achieve ultra low power with very high performance and functionality, all in one of the smallest footprints available. Target markets include portable media players, GPS units and smart phones. Features such as capless headphone design and an internal PLL help lower overall system cost.

BENEFITS AND ADVANTAGES

- High performance at low power
  - 100dB SNR (-60dB input) @ < 9.3mW
- Extremely low power modes
  - 98dB SNR (-60dB input) @ < 4mW (1.62V VDDA, 3.0V VDDIO, externally driven 1.2V VDDD)
- Small PCB Footprint
  - 3mmx3mm QFN
- Audio Processing
  - Allows for no cost system customization

FEATURES

Analog Inputs

- **Stereo Line In**
  - Support for external analog input
  - Codec bypass for low power
- **MIC**
  - MIC bias provided (5x5mm QFN, 3x3mm QFN TA2)
  - Programmable MIC gain
- **ADC**
  - 85dB SNR (-60dB input) and -73dB THD+N (VDDA=1.8V)

Analog Outputs

- **Line Out**
  - 100dB SNR (-60dB input) and -85dB THD+N (VDDIO=3.3V)
- **HP Output**
  - 100dB SNR (-60dB input) and -80dB THD+N (VDDA=1.8V, 16 ohm load, DAC to headphone)
  - 45mW max into 16 ohm load @ 3.3V
  - Capless design

Digital I/O

- I2S port to allow routing to Application Processor

Integrated Digital Processing

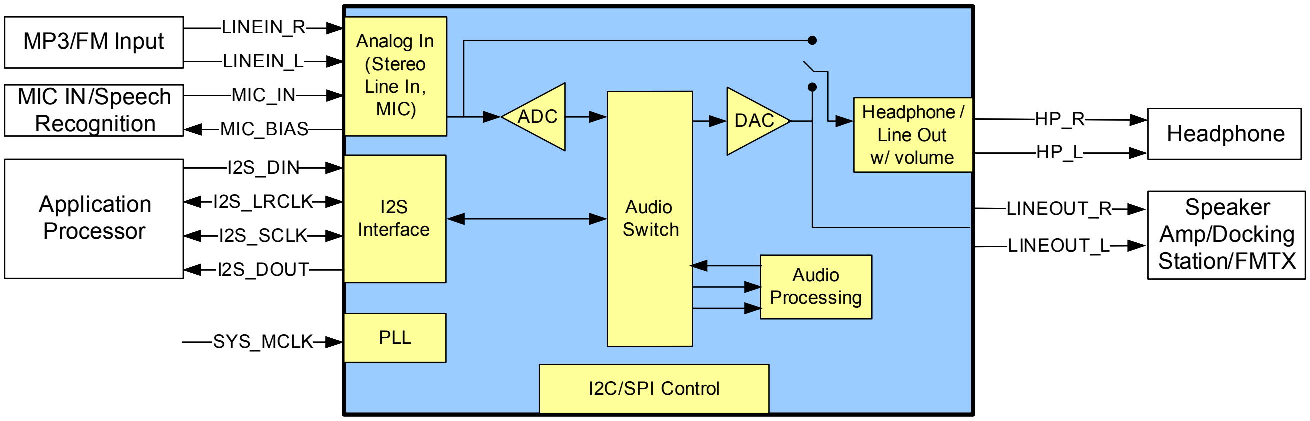
- SigmaTel Surround, SigmaTel Bass, tone control/ parametric equalizer/graphic equalizer

Clocking/Control

- PLL allows input of 8MHz to 27Mhz system clock - Standard audio clocks are derived from PLL

Power Supplies

- Designed to operate from 1.62 to 3.6 volts



Note: Only I^2C is supported in the 3 mm x 3 mm

Package

- 3mm x 3mm 20 pin QFN
- 5mm x 5mm 32 pin QFN

Freescale, Inc. makes no warranty for the use of its products, assumes no responsibility for any errors which may appear in this document, and makes no commitment to update the information contained herein. Freescale reserves the right to change or discontinue this product at any time, without notice. There are no express or implied licenses granted hereunder to design or fabricate any integrated circuits based on information in this document.

SigmaTel and the SigmaTel logo are trademarks of Freescale, Inc. and may be used to identify Freescale products only. Windows Media and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. Other product and company names contained herein may be trademarks of their respective owners.

1. ELECTRICAL SPECIFICATIONS

1.1. Absolute Maximum Ratings

Exceeding the absolute maximum ratings shown in Table 1 could cause permanent damage to SGTL5000 and is not recommended. Normal operation is not guaranteed at the absolute maximum ratings and extended exposure could affect long term reliability.

Table 1. Absolute Maximum Ratings

Parameter	Min	Max	Unit
Storage Temperature	-55	125	C
Maximum Digital Voltage VDDD		1.98	V
Maximum Digital I/O Voltage - VDDIO		3.6	V
Maximum Analog Supply Voltage - VDDA		3.6	V
Maximum voltage on any digital input	GND-0.3	VDDIO+0.3	V
Maximum voltage on any analog input	GND-0.3	VDDA+0.3	V

1.2. Recommended Operating Conditions

.

Table 2. Recommended Operating Conditions

Parameter	Symbol/pin(S)	Min	Max	Unit
Ambient Operating Temperature	Ta	-40	85	C
Digital Voltage (if supplied externally)	VDDD	1.1	2.0	V
Digital I/O Voltage	VDDIO	1.62	3.6	V
Analog Output Supply	VDDA	1.62	3.6	V

Operational Specifications

Table 3. Audio Performance

Test Conditions unless otherwise noted: VDDIO=1.8V, VDDA = 1.8V, Ta=25C, Slave mode, Fs = 48kHz, MCLK = 256Fs, 24 bit input.

Parameter	Min	Typical	Max	Unit
Line In Input Level		.75		Vrms
Line In Input Impedance	10k			Ohm
Line In -> ADC -> I2S Out				
SNR (-60dB input)		85		dB

Table 3. Audio Performance

Test Conditions unless otherwise noted: VDDIO=1.8V, VDDA = 1.8V, Ta=25C, Slave mode, Fs = 48kHz, MCLK = 256Fs, 24 bit input.

Parameter	Min	Typical	Max	Unit
THD+N		-70		dB
Frequency Response		+/-.11		dB
Channel Separation		79		dB
Line In -> Headphone_Lineout (CODEC Bypass Mode)				
SNR (-60dB input)		98		dB
THD+N (10k ohm load)		-87		dB
THD+N (16 ohm load)		-87		dB
Frequency Response		+/-.05		dB
Channel Separation (1kHz)		82		dB
I2S In -> DAC -> Line Out				
Output Level		.6		Vrms
SNR (-60dB input)		95		dB
THD+N		-85		dB
Frequency Response		+/-.12		dB
I2S In -> DAC -> Headphone Out - 16 Ohm load				
Output Power		17		mW
SNR (-60dB input)		100		dB
THD+N		-80		dB
Frequency Response		+/-.12		dB
I2S In -> DAC -> Headphone Out - 32 Ohm load				
Output Power		10		mW
SNR (-60dB input)		95		dB
THD+N		-86		dB
Frequency Response		+/-.11		dB
I2S In -> DAC -> Headphone Out - 10k Ohm load				
SNR (-60dB input)		96		dB
THD+N		-84		dB
Frequency Response		+/-.11		dB
PSRR (200mVp-p @ 1kHz on VDDA)		85		dB

Table 4. Audio Performance

Test Conditions unless otherwise noted: VDDIO=3.3V, VDDA =3.3V, Ta=25C, Slave mode, Fs = 48kHz, MCLK = 256Fs, 24 bit input. ADC tests were conducted with refbias = -37.5%, all other tests conducted with refbias = -50%

Parameter	Min	Typical	Max	Unit
Line In Input Level		1		Vrms
Line In Input Impedance	10k			Ohm
Line In -> ADC -> I2S Out				
SNR (-60dB input)		90		dB
THD+N		-72		dB
Frequency Response		+/-.11		dB
Channel Separation		80		dB
Line In -> Headphone_Lineout (CODEC Bypass Mode)				
SNR (-60dB input)		102		dB
THD+N (10k ohm load)		-89		dB
THD+N (16 ohm load)		-87		dB
Frequency Response		+/-.05		dB
Channel Separation (1kHz)		81		dB
I2S In -> DAC -> Line Out				
Output Level		1		Vrms
SNR (-60dB input)		100		dB
THD+N		-88		dB
Frequency Response		+/-.12		dB
I2S In -> DAC -> Headphone Out - 16 Ohm load				
Output Power		58		mW
SNR (-60dB input)		98		dB
THD+N		-86		dB
Frequency Response		+/-.12		dB
I2S In -> DAC -> Headphone Out - 32 Ohm load				
Output Power		30		mW
SNR (-60dB input)		100		dB
THD+N		-88		dB
Frequency Response		+/-.11		dB
I2S In -> DAC -> Headphone Out - 10k Ohm load				

Table 4. Audio Performance

Test Conditions unless otherwise noted: VDDIO=3.3V, VDDA =3.3V, Ta=25C, Slave mode, Fs = 48kHz, MCLK = 256Fs, 24 bit input. ADC tests were conducted with refbias = -37.5%, all other tests conducted with refbias = -50%

Parameter	Min	Typical	Max	Unit
SNR (-60dB input)		97		dB
THD+N		-85		dB
Frequency Response		+/-.11		dB
PSRR (200mVp-p @ 1kHz on VDDA)		89		dB

1.3. Timing Specifications

1.3.1. Power Up Timing

The SGTL5000 has an internal reset that is deasserted 8 SYS_MCLK cycles after all power rails have been brought up. After this time communication can start..

Table 5. Power Up Timing

Symbol	Parameter	Min	Typical	Max	Unit
Tpc	Time from all supplies powered up and SYS_MCLK present to initial communication	1*			uS

* 1uS represents 8 SYS_MCLK cycles at the minimum 8MHz SYS_MCLK.

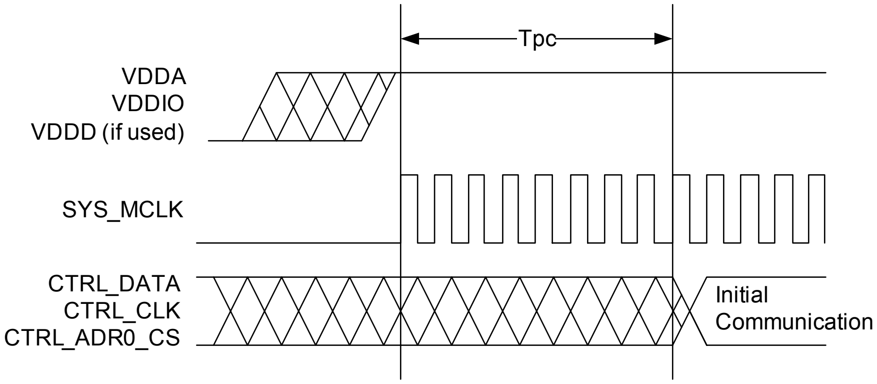


Figure 1. Power Up Timing

1.3.2. I2C

This section provides timing for the SGTL5000 while in I2C mode (CTRL_MODE = =0).

Table 6. I2C Bus Timing

Symbol	Parameter	Min	Typical	Max	Unit
Fi2c_clk	I2C Serial Clock Frequency			400	kHz
Ti2csh	I2C Start condition hold time	150			nS
Ti2cstsu	I2C Stop condition setup time	150			nS
Ti2cdsu	I2C Data input setup time to rising edge of CTRL_CLK	125			nS
Ti2cdh	I2C Data input hold time from falling edge of CTRL_CLK (SGTL5000 receiving data)	5			nS
Ti2cdh	I2C Data input hold time from falling edge of CTRL_CLK (SGTL5000 driving data)	360			nS
Ti2cclkl	I2C CTRL_CLK low time	300			nS
Ti2cclkh	I2C CTRL_CLK high time	100			nS

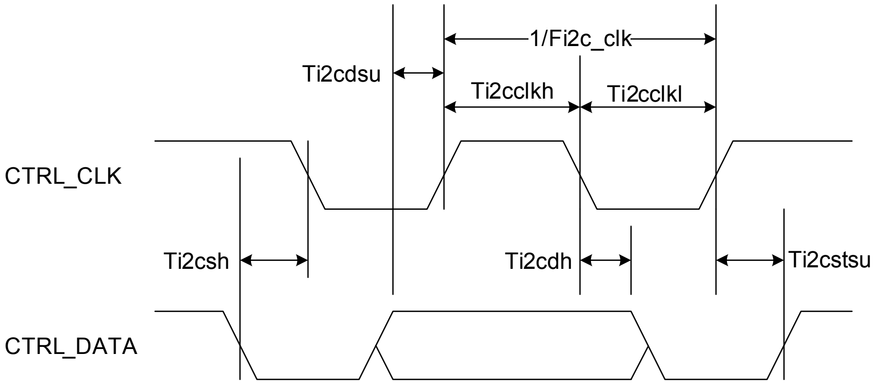


Figure 2. I2C Timing (CTRL_MODE == 0)

1.3.3. SPI

This section provides timing for the SGTL5000 while in SPI mode (CTRL_MODE = =1).

Table 7. SPI Bus Timing

Symbol	Parameter	Min	Typical	Max	Unit
Fspi_clk	SPI Serial Clock Frequency			???	MHz
Tspidsu	SPI data input setup time	10			nS

Table 7. SPI Bus Timing

Symbol	Parameter	Min	Typical	Max	Unit
Tspidh	SPI data input hold time	10			nS
Tspiclkl	SPI CTRL_CLK low time	???			nS
Tspiclkh	SPI CTRL_CLK high time	???			nS
Tccs	SPI clock to chip select	60			nS
Tcsc	SPI chip select to clock	20			nS
Tcsl	SPI chip select low	20			nS
Tcsh	SPI chip select high	20			nS

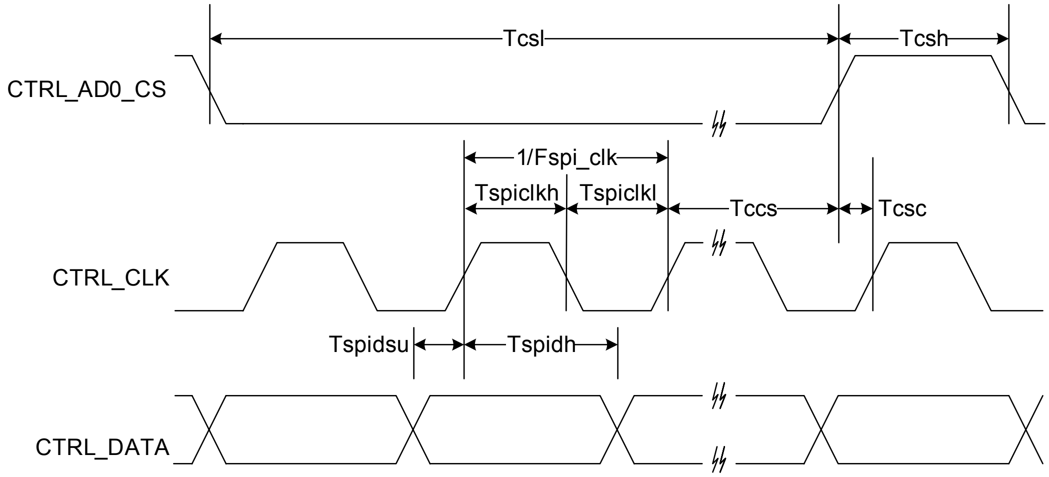


Figure 3. SPI Timing

1.3.4. I2S

The following are the specifications and timing for I2S port. The timing applies to all formats.

Table 1-1.

Symbol	Parameter	Min	Typical	Max	Unit
Flrclk	Frequency of I2S_LRCLK	???		96	kHz
Fsclk	Frequency of I2S_SCLK		32*Flrclk, 64*Flrclk		kHz
Ti2s_d	I2S delay			10	ns
Ti2s_s	I2S setup time	10			ns

Table 1-1.

Symbol	Parameter	Min	Typical	Max	Unit
Ti2s_h	I2S hold time	10			ns

.

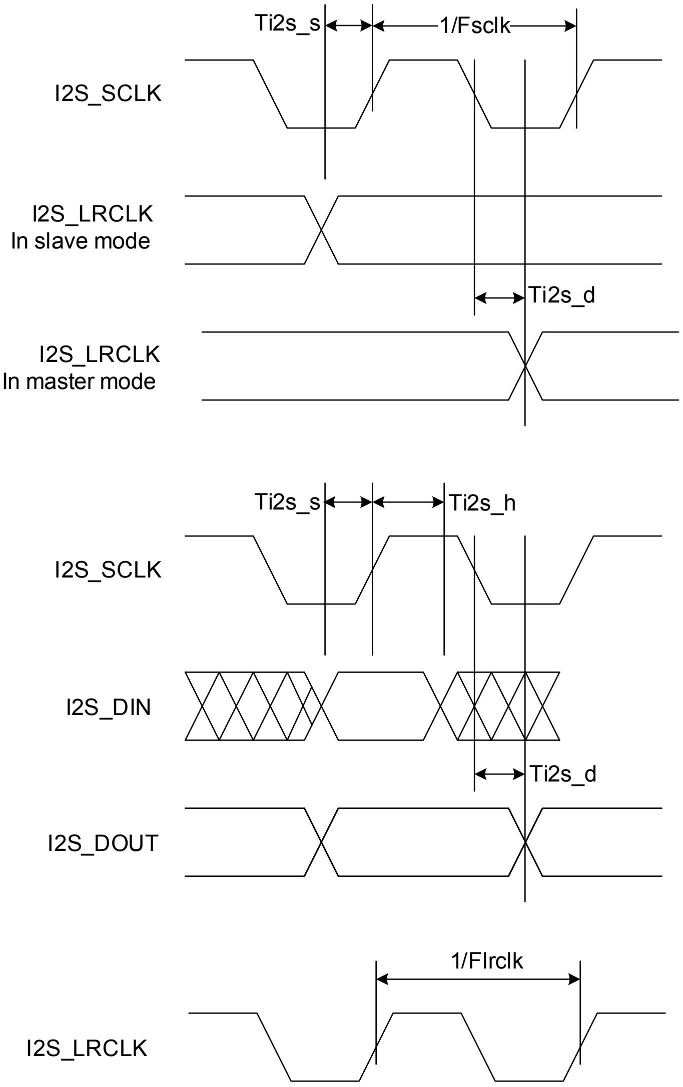


Figure 4. I2S Interface Timing

2. POWER CONSUMPTION

Table 8: Power Consumption: VDDA=1.8V, VDDIO=1.8V

Mode	Current Consumption (mA)			Power(mW)
	VDDD	VDDA	VDDIO	
Playback (I2S->DAC->Headphone)		2.54	.9	6.19
Playback with DAP ((I2S->DAP->DAC->Headphone)		3.59	.9	8.08
Playback/Record (I2S->DAC->Headphone, ADC->I2S)		3.71	1.10	8.67
Record (ADC->I2S)		2.29	1.06	6.02
Analog playback, CODEC bypassed (LINEIN->HP)		1.48	.89	4.27
Standby, all analog power off		.019	.002	.038
Playback with PLL (I2S->DAC->HP)		3.01	2.17	9.31

VDDD derived internally @ 1.2V, slave mode except for PLL case, 32 ohm load on HP, Conditions: -100dBFs signal input, slave mode unless otherwise noted, paths tested as indicated, unused paths turned off.

A further 0.5-1.0mW reduction in power is expected with TA2 silicon.

Table 9: Power Consumption: VDDA=3.3V, VDDIO=3.3V

Mode	Current Consumption (mA)			Power(mW)
	VDDD	VDDA	VDDIO	
Playback (I2S->DAC->Headphone)		3.45	.067	11.60
Playback with DAP ((I2S->DAP->DAC->Headphone)		4.49	.067	15.03
Playback/Record (I2S->DAC->Headphone, ADC->I2S)		4.67	.343	16.53
Record (ADC->I2S)		2.90	.296	10.56
Analog playback, CODEC bypassed (LINEIN->HP)		1.91	.039	6.43
Standby, all analog power off		.04	.002	.139

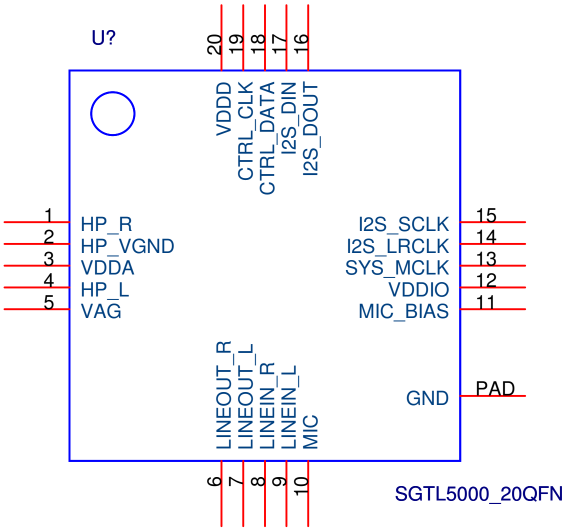
Table 9: Power Consumption: VDDA=3.3V, VDDIO=3.3V

Mode	Current Consumption (mA)			Power(mW)
	VDDD	VDDA	VDDIO	
Playback with PLL (I2S->DAC->HP)		3.92	2.76	22.05

3. PINOUT & PACKAGE INFO

3.1. Pinout

20QFN Pinout



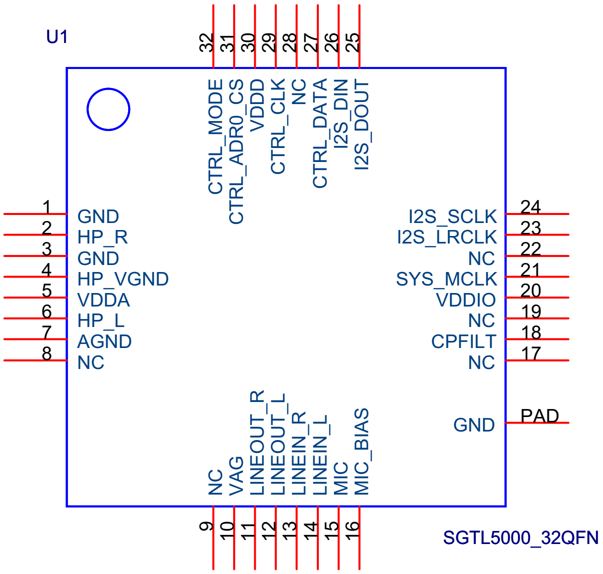


Figure 6. SGTL5000 32QFN Pinout

3.2. Pin Description

Table 10. 20 pin QFN pinout

Pin Count	Pin Name	Description	Notes
1	HP_R	Right headphone output	ANALOG
2	HP_VGND	Headphone virtual ground	ANALOG
3	VDDA	Analog voltage	POWER
4	HP_L	Left headphone output	ANALOG
5	VAG	DAC VAG filter	ANALOG
6	LINEOUT_R	Right line out	ANALOG
7	LINEOUT_L	Left line out	ANALOG
8	LINEIN_R	Right line in	ANALOG
9	LINEIN_L	Left line in	ANALOG
10	MIC	Microphone input	ANALOG
11	CPFILT	Charge Pump Filter	ANALOG
12	VDDIO	Digital I/O voltage	POWER
13	SYS_MCLK	System master clock	DIGITAL
14	I2S_LRCLK	I2S frame clock	DIGITAL
15	I2S_SCLK	I2S bit clock	DIGITAL
16	I2S_DOUT	I2S data output	DIGITAL
17	I2S_DIN	I2S data input	DIGITAL
18	CTRL_DATA	I2C Mode: Serial Data (SDA); SPI Mode: Serial Data Input (MOSI)	DIGITAL
19	CTRL_CLK	I2C Mode: Serial Clock (SCL); SPI Mode: Serial Clock (SCK)	DIGITAL
20	VDDD	Digital voltage	POWER
PAD	GND	Ground. Center PAD of package is ground connection for part and must be connected to board ground.	GROUND

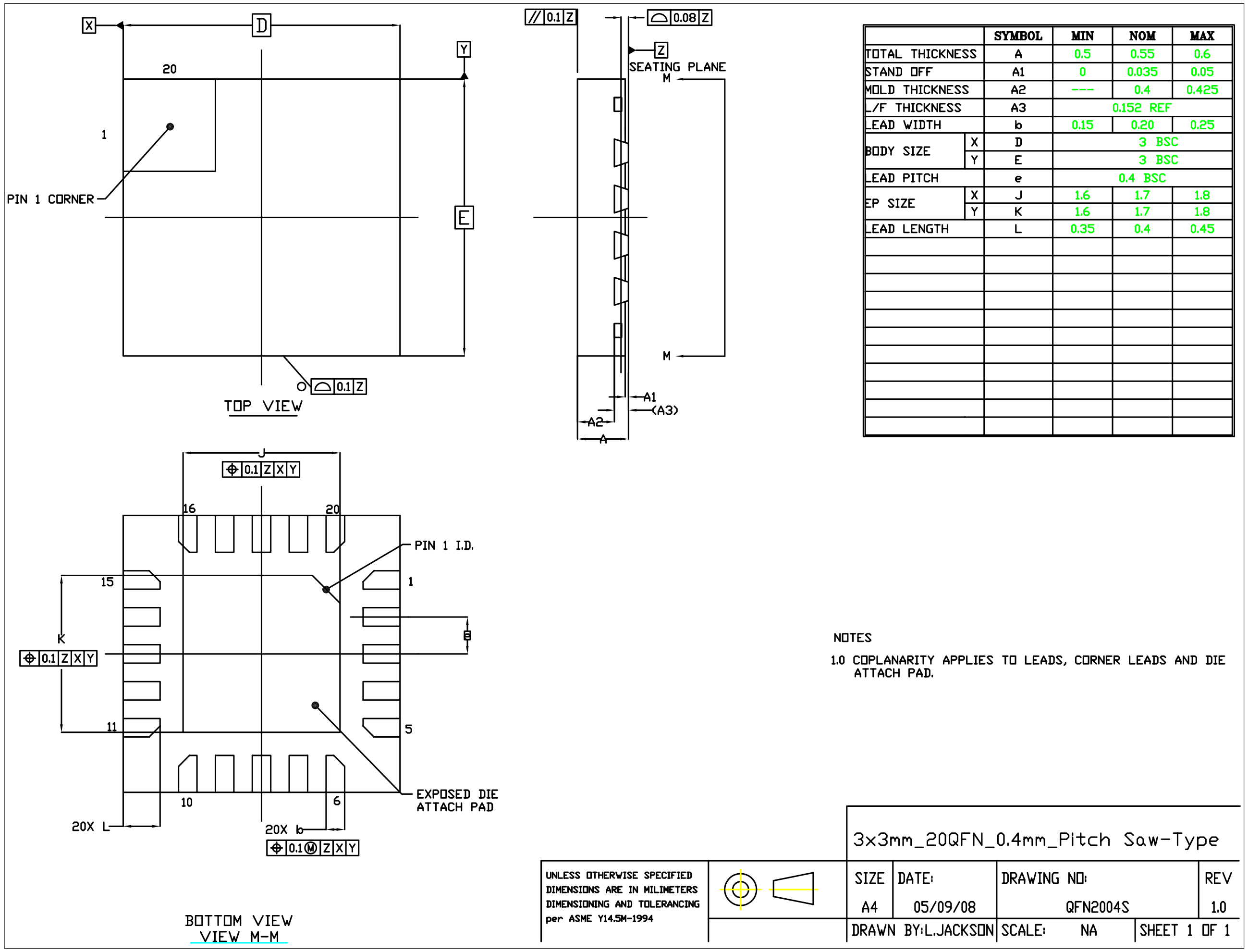
Table 11. 32 pin QFN pinout

Pin Count	Pin Name	Description	Notes
1	GND	Ground	GROUND
2	HP_R	Right headphone output	ANALOG
3	GND	Ground	GROUND
4	HP_VGND	Headphone virtual ground	ANALOG
5	VDDA	Analog voltage	POWER
6	HP_L	Left headphone output	ANALOG
7	AGND	Analog ground	GROUND
8	NC	No connect	DIGITAL
9	NC	No connect	DIGITAL
10	VAG	DAC VAG filter	ANALOG
11	LINEOUT_R	Right line output	ANALOG
12	LINEOUT_L	Left line output	ANALOG
13	LINEIN_R	Right line input	ANALOG
14	LINEIN_L	Left line input	ANALOG
15	MIC	Microphone input	ANALOG
16	MIC_BIAS	Mic bias	ANALOG
17	NC	No connect	
18	CPFILT	Charge pump filter	ANALOG
19	NC	No connect	
20	VDDIO	Digital I/O voltage	POWER
21	SYS_MCLK	System master clock	DIGITAL
22	NC	No Connect	
23	I2S_LRCLK	I2S frame clock	DIGITAL
24	I2S_SCLK	I2S bit clock	DIGITAL
25	I2S_DOUT	I2S data output	DIGITAL
26	I2S_DIN	I2S data input	DIGITAL

Table 11. 32 pin QFN pinout

Pin Count	Pin Name	Description	Notes
27	CTRL_DATA	I2C Mode: Serial Data (SDA); SPI Mode: Serial Data Input (MOSI)	DIGITAL
28	NC	No connect	
29	CTRL_CLK	I2C Mode: Serial Clock (SCL); SPI Mode: Serial Clock (SCK)	DIGITAL
30	VDDD	Digital voltage	POWER
31	CTRL_ADR0_CS	I2C Mode: I2C Address Select 0; SPI Mode: SPI Chip Select	DIGITAL
32	CTRL_MODE	Mode select for I2C or SPI; When pulled low the control mode is I2C, when pulled high the control mode is SPI	DIGITAL
PAD	GND	This PAD should be soldered to ground. This is a suggestion for mechanical stability but is not required electrically.	GROUND

3.3. Package



	SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS	A	0.5	0.55	0.6
STAND OFF	A1	0	0.035	0.05
MOLD THICKNESS	A2	---	0.4	0.425
L/F THICKNESS	A3		0.152 REF	
LEAD WIDTH	b	0.15	0.20	0.25
BODY SIZE X	D		3 BSC	
BODY SIZE Y	E		3 BSC	
LEAD PITCH	e		0.4 BSC	
EP SIZE X	J	1.6	1.7	1.8
EP SIZE Y	K	1.6	1.7	1.8
LEAD LENGTH	L	0.35	0.4	0.45

NOTES

1.0 COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH PAD.

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILIMETERS DIMENSIONING AND TOLERANCING per ASME Y14.5M-1994

3x3mm_20QFN_0.4mm_Pitch Saw-Type

SIZE	DATE:	DRAWING NO:	REV
A4	05/09/08	QFN2004S	1.0
DRAWN BY:L.JACKSON	SCALE: NA	SHEET 1 OF 1	

Figure 7. SGTL5000 3mmx3mm 20QFN Package

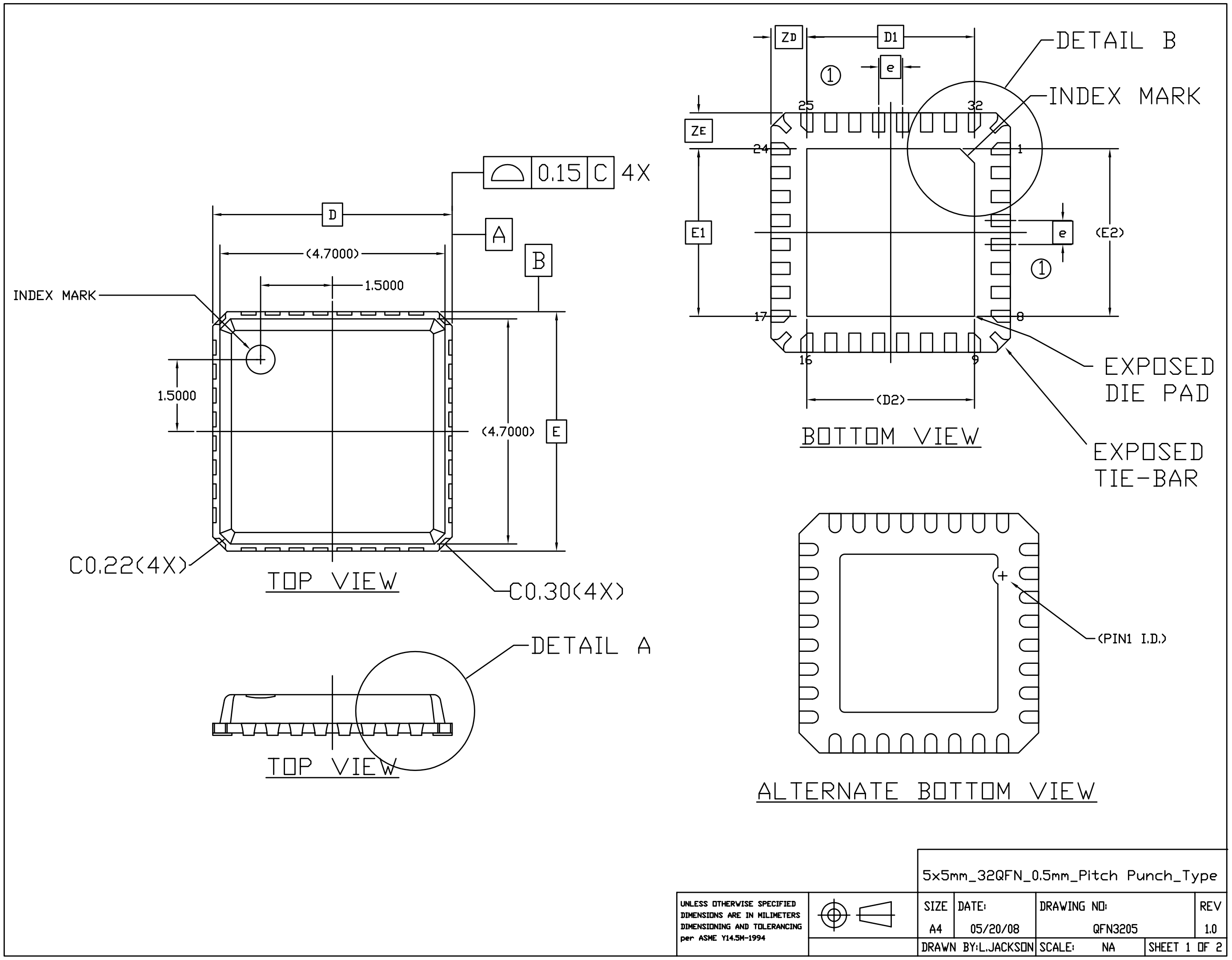
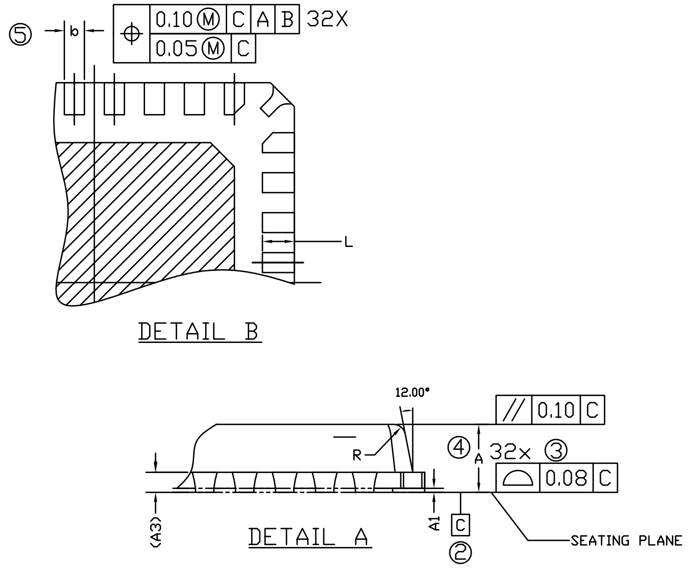


Figure 8. SGTL5000 5mmx5mm 32QFN Package (sheet 1)



in mm

Symbol	Min	Nom	Max
A	–	0.80	0.90
A1	0.00	0.02	0.05
A3	0.20 Ref		
D	5.00 BSC		
D1	3.50 BSC		
E	5.00 BSC		
E1	3.50 BSC		
e	0.50 BSC		
R	0.20~0.25		
ZD	0.75 BSC		
ZE	0.75 BSC		
b	0.18	0.25	0.30
L	0.30	0.40	0.50
D2	3.20	3.50	3.60
E2	3.20	3.50	3.60

Note:

① 'e' REPRESENTS THE BASIC TERMINAL PITCH. SPECIFIES THE TRUE GEOMETRIC POSITION OF THE TERMINAL AXIS.

② DATUM 'C' IS THE MOUNTING SURFACE, WITH WHICH THE PACKAGE IS IN CONTACT.

③ SPECIFIES THE VERTICAL SHIFT OF THE FLAT PART OF EACH TERMINAL FROM THE MOUNTING SURFACE.

④ DIMENSION 'A' INCLUDES PACKAGE WARPAGE.

⑤ DIMENSION 'b' APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURE IN THE RADIUS AREA.

6 PACKAGE DIMENSIONS CONFORM TO JEDEC MO-220 REV.J, VARIATIONS VHHD-5, EXCEPT FOR D2 & E2.

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILIMETERS
DIMENSIONING AND TOLERANCING
per ASME Y14.5M-1994

5x5mm_32QFN_0.5mm_Pitch Punch_Type

SIZE	DATE:	DRAWING NO:		REV
A4	05/20/08	QFN3205		1.0
DRAWN BY:L.JACKSON		SCALE: NA	SHEET 2 OF 2	

Figure 9. SGTL5000 5mmx5mm 32QFN Package (sheet 2)

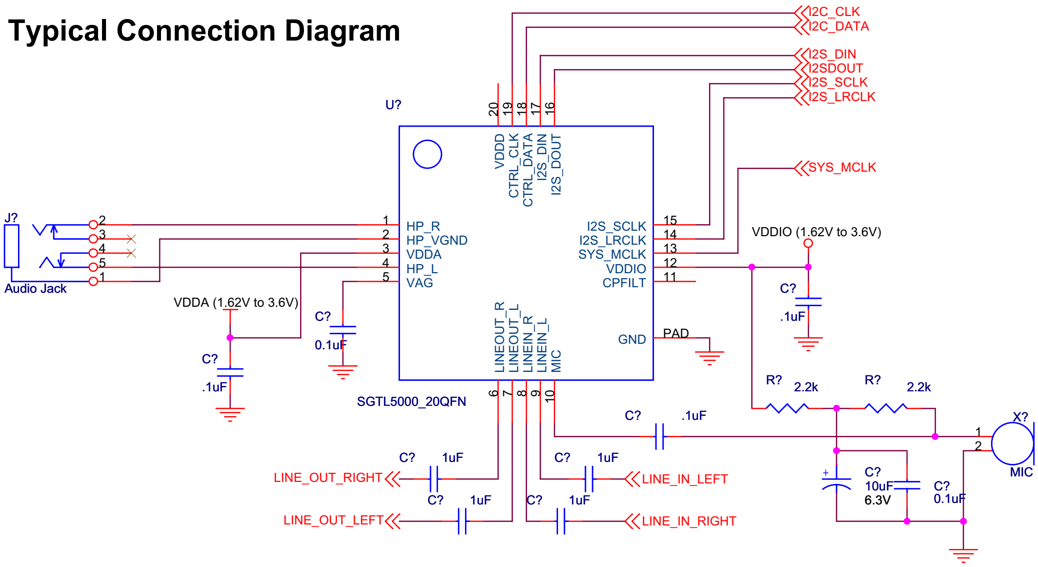
4. TYPICAL CONNECTION DIAGRAMS

Typical connection diagrams are shown in this section that demonstrate the flexibility of the SGTL5000. Both low cost and low power configurations are presented although it should be noted that all configurations offer a low cost design with high performance and low power.

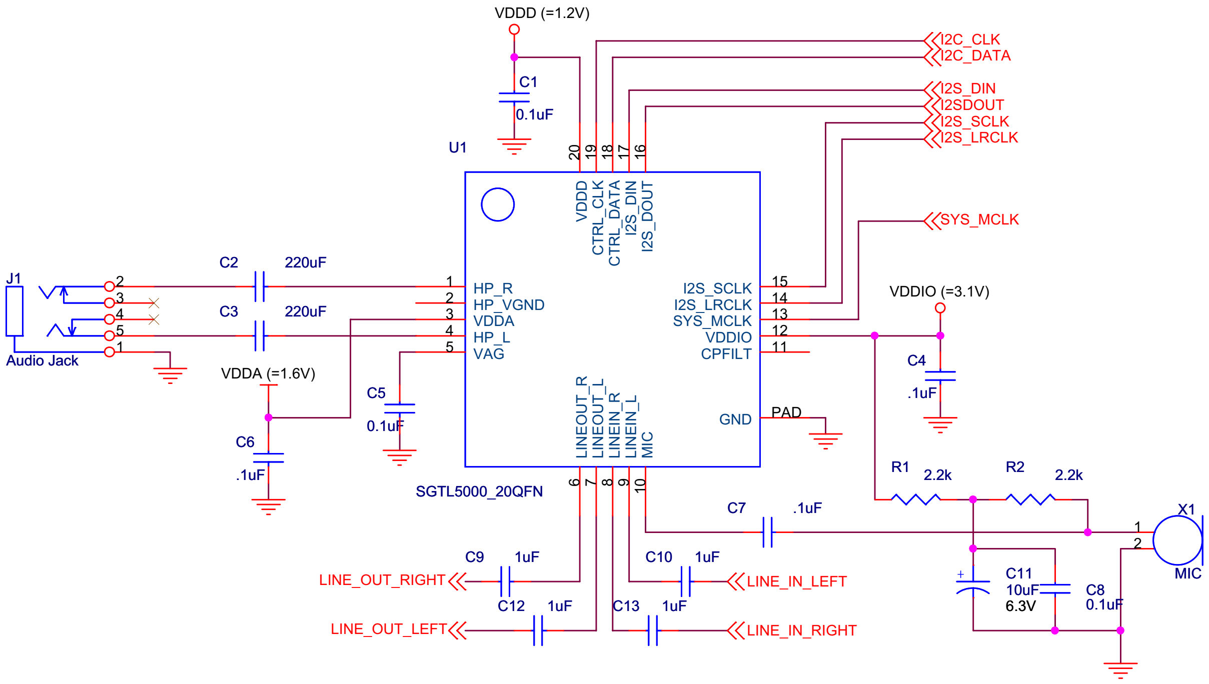
Some design considerations for SGTL5000 are as follows:

- Star the ground pins of the chip, VAG ground, and all analog inputs/outputs to a single point, then to the ground plane
- Use the widest, shortest trace possible for the HP_VGND

20QFN Typical Connection Diagram

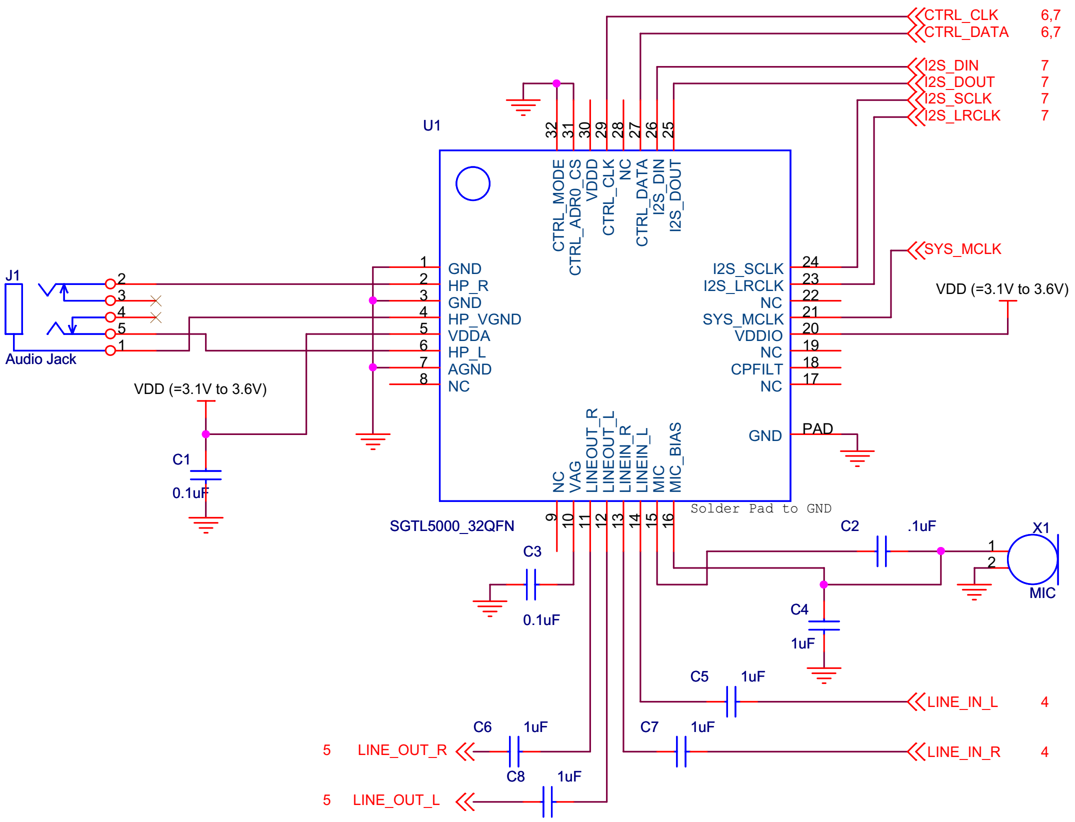


```
Notes:
1.  The above circuit shows VDDD (pin 20) being derived internally. For lowest power
operation VDDD can be driven from an external 1.2V supply with .1uF of decoupling to
ground.
2.  The above circuit shows a mic bias circuit derived from an external supply (VDDIO).
```



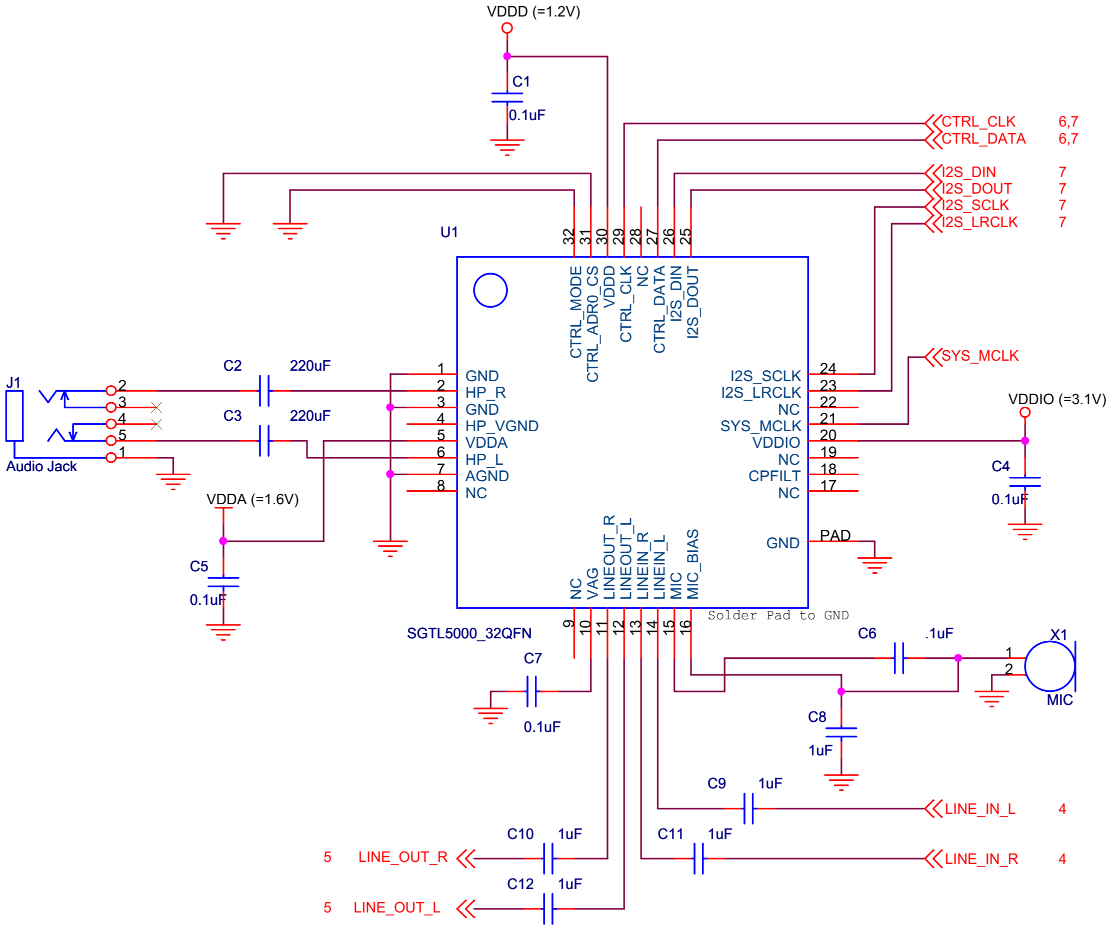
```
1.  VDDD is driven externally by 1.2V supply.
2.  VDDA is driven at 1.6V
3.  VDDIO is driven at 3.1V
```

20QFN Typical Connection Diagram - Lowest Power Configuration



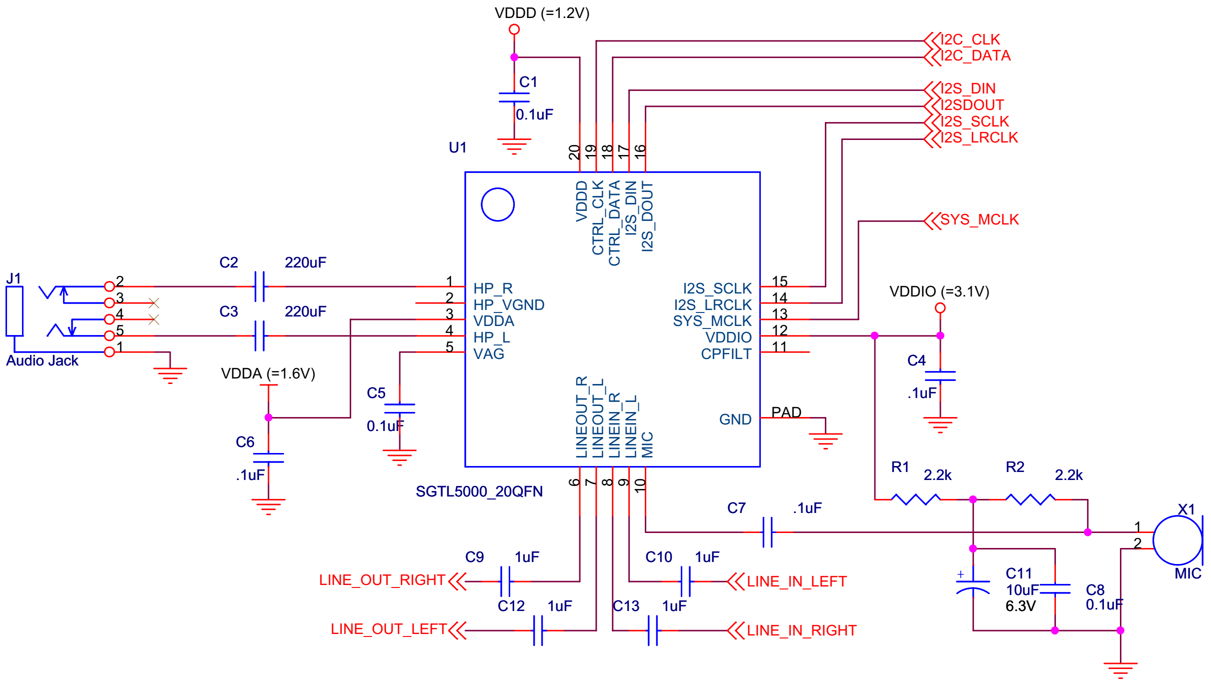
```
Notes:

1.  VDDD is derived internally (no need for external cap)
2.  VDDA and VDDIO are supplied from same voltage that is between 3.1V
and 3.6V.  By using the same voltage this allows removal of power
decoupling cap.  By using a voltage above 3.1V the CAP connected to
CPFILT can be removed.
```

32QFN Typical Connection Diagram - Lowest Cost Configuration



1. VDDD is driven externally by 1.2V supply.
2. VDDA is driven at 1.6V
3. VDDIO is driven at 3.1V

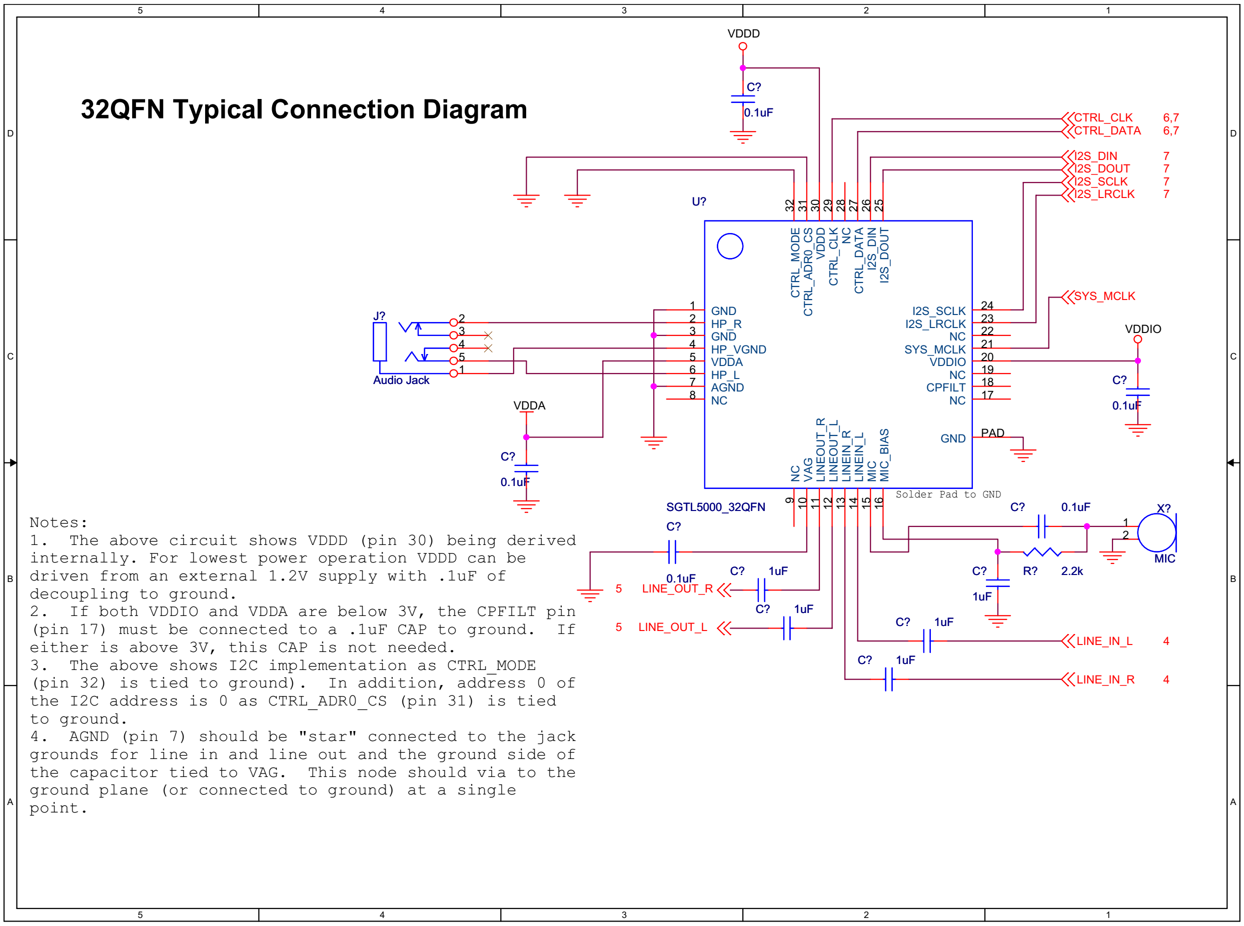
32QFN Typical Connection Diagram - Lowest Power Configuration



1. VDDD is driven externally by 1.2V supply.
2. VDDA is driven at 1.6V
3. VDDIO is driven at 3.1V

20QFN Typical Connection Diagram - Lowest Power Configuration

32QFN Typical Connection Diagram



Notes:

1. The above circuit shows VDDD (pin 30) being derived internally. For lowest power operation VDDD can be driven from an external 1.2V supply with .1uF of decoupling to ground.

2. If both VDDIO and VDDA are below 3V, the CPFILT pin (pin 17) must be connected to a .1uF CAP to ground. If either is above 3V, this CAP is not needed.

3. The above shows I2C implementation as CTRL_MODE (pin 32) is tied to ground). In addition, address 0 of the I2C address is 0 as CTRL_ADR0_CS (pin 31) is tied to ground.

4. AGND (pin 7) should be "star" connected to the jack grounds for line in and line out and the ground side of the capacitor tied to VAG. This node should via to the ground plane (or connected to ground) at a single point.

5. DEVICE DESCRIPTION

The SGTL5000 is a low power stereo codec with integrated headphone amplifier. It is designed to provide a complete audio solution for portable products needing line-in, mic-in, line-out, headphone-out, and digital I/O. Deriving it's architecture from best in class Freescale integrated products that are currently on the market, the SGTL5000 is able to achieve ultra low power with very high performance and functionality, all in one of the smallest footprints available. Target markets include portable media players, GPS units and smart phones. Features such as capless headphone design and USB clocking mode (12MHz SYS_MCLK input) help lower overall system cost.

In summary, SGTL5000 accepts the following inputs:

- Line input
- Microphone input, with mic bias (mic bias only available in 32QFN version)
- Digital I2S input

In addition, SGTL5000 supports the following outputs:

- Line output
- Headphone output
- Digital I2S output

The following digital audio processing is included to allow for product differentiation:

- Digital mixer
- SigmaTel Surround
- SigmaTel Bass Enhancement
- Tone Control, parametric equalizer, and graphic equalizer

The SGTL5000 can accept an external standard master clock at a multiple of the sampling frequency (i.e. 256*Fs, 385*Fs, 512*Fs). In addition it can take non standard frequencies and use the internal PLL to derive the audio clocks. The device supports 8kHz, 11.025kHz, 16kHz, 22.5kHz, 24kHz, 32KHz, 44.1kHz, 48kHz, 96kHz sampling frequencies.

5.1. System Block Diagram w/ Signal Flow and Gain Map

Figure 10 below shows a block diagram that highlights the signal flow and gain map for the SGTL5000.

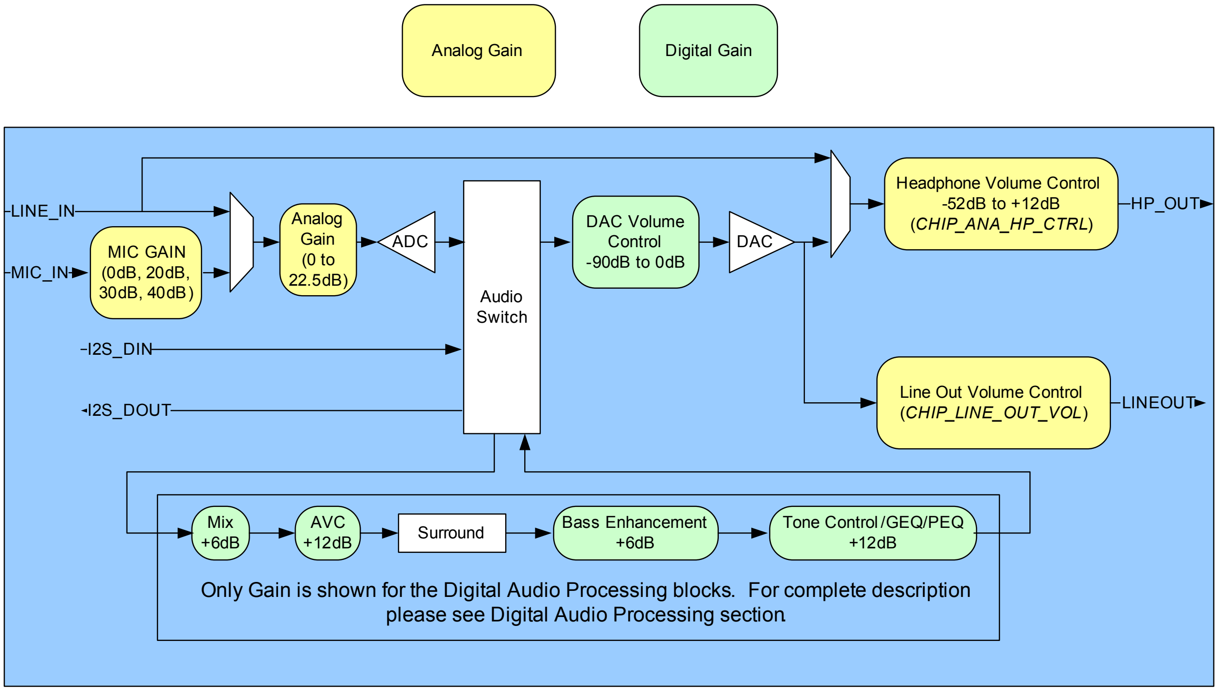

Analog Gain
Digital Gain
LINE_IN
MIC_IN
MIC GAIN (0dB, 20dB, 30dB, 40dB)
Analog Gain (0 to 22.5dB)
ADC
Audio Switch
DAC Volume Control -90dB to 0dB
DAC
Headphone Volume Control -52dB to +12dB (CHIP_ANA_HP_CTRL)
HP_OUT
Line Out Volume Control (CHIP_LINE_OUT_VOL)
LINEOUT
I2S_DIN
I2S_DOUT
Mix +6dB
AVC +12dB
Surround
Bass Enhancement +6dB
Tone Control/GEQ/PEQ +12dB
Only Gain is shown for the Digital Audio Processing blocks. For complete description please see Digital Audio Processing section

Figure 10. System Block Diagram, signal flow and gain

To guarantee against clipping it is important that the gain in a signal path in addition to the signal level does not exceed 0dB at any point.

5.2. Power

The SGTL5000 has a flexible power architecture to allow the system designer to minimize power consumption and maximize performance at the lowest cost.

5.2.1. External Power Supplies

The SGTL5000 requires 2 external power supplies: VDDA and VDDIO. An optional third external power supply VDDD may be provided externally to achieve lower power. A description for the different power supplies is as follows:

- VDDA: This external power supply is used for the internal analog circuitry including ADC, DAC, LINE inputs, MIC inputs, headphone outputs and reference voltages. VDDA supply ranges are shown in section 1.2. A decoupling cap should be used on VDDA as shown in the typical connection diagram in section 4.
- VDDIO: This external power supply controls the digital I/O levels as well as the output level of LINE outputs. VDDIO supply ranges are shown in section 1.2. A decoupling cap should be used on VDDIO as shown in the typical connection diagram in section 4.

Note that if VDDA and VDDIO are derived from the same voltage, a single decoupling capacitor can be used to minimize cost. This capacitor should be placed closest to VDDA.

- VDDD: This is a digital power supply that is used for internal digital circuitry. For a low cost design, this supply can be derived from an internal regulator and no external components are required. If no external supply is applied to VDDD, the internal regulator will automatically be used. For lowest power, this supply can be driven at the lowest specified voltage given in section 1.2. If an external supply is used for VDDD, a decoupling capacitor is recommended. VDDD supply ranges are shown in section 1.2 for when externally driven. If the system drives VDDD externally, an efficient switching supply should be used or or no system power savings will be realized.

5.2.2. Internal Power Supplies

The SGTL5000 has two exposed internal power supplies, VAG and chargepump.

- VAG is the internal voltage reference for the ADC and DAC. After startup the voltage of VAG should be set to VDDA/2 by writing *CHIP_REF_CTRL->VAG_VAL*. Refer to programming section 6.2.1.1. The VAG pin should have an external filter capacitor as shown in the typical connection diagram.
- Chargepump: This power supply is used for internal analog switches. If VDDA or VDDIO is greater than 2.7V, this supply is automatically driven from the highest of VDDIO and VDDA. If both VDDIO and VDDA are less than 3.1V, then the user should turn on the charge pump function to create the chargepump rail from VDDIO by writing *CHIP_ANA_POWER->VDDC_CHRGPMP_POWERUP* register. Refer to programming section 6.2.1.1.
- LINE_OUT_VAG is the line output voltage reference. It should be set to VDDIO/2 by writing *CHIP_LINE_OUT_CTRL->LO_VAGCNTRL*.

5.2.3. Power Schemes

The SGTL supports a flexible architecture and allows the system designer to minimize power or maximize BOM savings.

- For maximum cost savings, all supplies can be run at the same voltage.
- Alternatively for minimum power, the analog and digital supplies can be run at minimum voltage while driving the digital I/O voltage at the voltage needed by the system.
- To save power, independent supplies are provided for line outputs and headphone outputs. This allows for 1VRMS line outputs while using minimal headphone power.
- For best power, VDDA should be run at the lowest possible voltage required for the maximum headphone output level. For highest performance, VDDA should be run at 3.3V. For most applications a lower voltage can be used for the best performance/power combination.

5.3. Reset

The SGTL5000 has an internal reset that is deasserted 8 SYS_MCLKs after all power rails have been brought up. After this time communication can start. See section 1.3 for timing specification.

5.4. Clocking

Clocking for the SGTL5000 is provided by a system master clock input (SYS_MCLK). SYS_MCLK should be synchronous to the sampling rate (Fs) of the I2S port. Alternatively any clock between 8Mhz and 27Mhz can be provided on SYS_MCLK and the SGTL5000 can use an internal PLL to derive all internal and I2S clocks. This allows the system to use an available clock such as 12MHz (common USB clock) for SYS_MCLK to reduce overall system costs.

5.4.1. Synchronous SYS_MCLK input

The SGTL5000 supports various combinations of SYS_MCLK frequency and sampling frequency as shown in Table 12. Using a synchronous SYS_MCLK allows for lower power as the internal PLL is not used.

Table 12. Synchronous MCLK Rates

Clock	Supported rates	Units
System Master Clock (SYS_MCLK)	256, 384, 512	Fs
Sampling Frequency (Fs)	8, 11.025, 16, 22.5, 32, 44.1, 48, 96(note 1)	kHz

note 1. For a sampling frequency of 96kHz, only 256Fs SYS_MCLK is supported

5.4.2. Using the PLL - Asynchronous SYS_MCLK input

An integrated PLL is provided in the SGTL5000 that allows any clock from 8MHz to 27MHz to be connected to SYS_MCLK. This can help save system costs as a clock available elsewhere in the system can be used to derive all audio clocks using the internal PLL. In this case the clock input to SYS_MCLK can be asynchronous with the sampling frequency needed in the system. For example a 12MHz clock from the system processor could be used as the clock input to the SGTL5000.

Three register fields need to be configured to properly use the PLL. They are *CHIP_PLL_CTRL->INT_DIVISOR*, *CHIP_PLL_CTRL->FRAC_DIVISOR* and *CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2*. Figure 11 shows a flowchart that shows how to determine the values to program in the register fields.

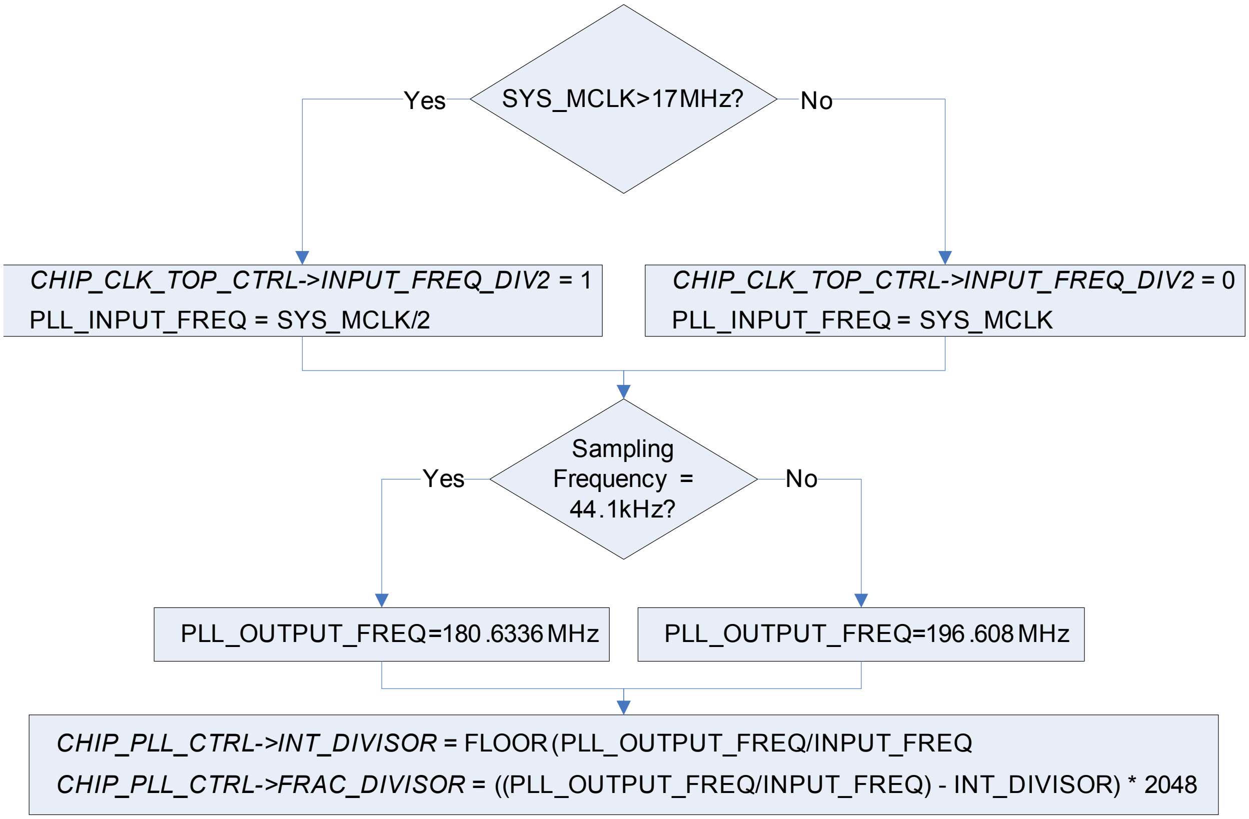

SYS_MCLK>17MHz?
Yes
No
CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 1
PLL_INPUT_FREQ = SYS_MCLK/2
CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0
PLL_INPUT_FREQ = SYS_MCLK
Sampling Frequency = 44.1kHz?
Yes
No
PLL_OUTPUT_FREQ=180.6336MHz
PLL_OUTPUT_FREQ=196.608MHz
CHIP_PLL_CTRL->INT_DIVISOR = FLOOR(PLL_OUTPUT_FREQ/INPUT_FREQ
CHIP_PLL_CTRL->FRAC_DIVISOR = ((PLL_OUTPUT_FREQ/INPUT_FREQ) - INT_DIVISOR) * 2048

Figure 11. PLL Programming Flowchart

For example, when a 12MHz digital signal is placed on MCLK, for a 48kHz frame clock

CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0 // SYS_MCLK<17MHz

CHIP_PLL_CTRL->INT_DIVISOR = FLOOR(196.608MHz/12MHz) = 16 (decimal)

CHIP_PLL_CTRL->FRAC_DIVISOR = ((196.608MHz/12MHz) - 16) * 2048 = 786 (decimal)

Refer to PLL programming note 6.2.2.

5.5. Audio Switch (Source Select Switch)

The audio switch is the central routing block that controls the signal flow from input to output. Any single input can be routed to any single or multiple outputs.

Any signal can be routed to the Digital Audio Processor (DAP). The output of the DAP (an input to the audio switch) can in turn be routed to any physical output. The output of the DAP can not be routed into itself. Refer to section 5.9, Digital Audio Processing, for DAP information and configuration.

It should be noted that the analog bypass from Line input to headphone output does not go through the audio switch.

To configure a route, the *CHIP_SSS_CTRL* register is used. Each output from the source select switch has its own register field that is used to select what input is routed to that output.

For example, to route the I2S digital input through the DAP and then out to the DAC (headphone) outputs write *SSS_CTRL->DAP_SELECT* to 0x1 (selects I2S_IN) and *SSS_CTRL->DAC_SELECT* to 0x3 (selects DAP output).

5.6. Analog Input Block

The analog input block contains a stereo line input and a microphone input with mic bias (in the 32QFN package). Either input can be routed to the ADC. The line input can also configured to bypass the CODEC and be routed the analog input directly to the headphone output.

5.6.1. Line Inputs

One stereo line input is provided for connection to line sources such as an FM radio or MP3 input.

The source should be connected to the left and right line inputs through series coupling capacitors. The suggested value is shown in the typical connection diagram in section 4.

As detailed in section 5.6.3, the line input can be routed to the ADC.

The line input can also be routed to the headphone output by writing *CHIP_ANA_CTRL->SELECT_HP*. This selection bypasses the ADC and audio switch and routes the line input directly to the headphone output to enable a very low power pass through.

5.6.2. Microphone Input

One mono microphone input is provided for uses such as voice recording.

Mic bias is provided in the 32QFN package. The mic bias is can be programmed with the *CHIP_MIC_CTRL->BIAS_VOLT* registor field. Values from 1.25V to 3.00V are supported in 0.25V steps. Mic bias should be set less than 200mV from VDDA, e.g. with VDDA at 1.70V, Mic bias should be set no greater than 1.50V.

The microphone should be connected through a series coupling capacitor. The suggested value is shown in the typical connection diagram.

The microphone has programmable gain through the *CHIP_MIC_CTRL->GAIN* register field. Values of 0dB, +20dB, +30dB and +40dB are available.

5.6.3. ADC

The SGTL5000 contains an ADC who takes its input from either the line input or a microphone. The register field *CHIP_ANA_CTRL->SELECT_ADC* controls this selection. The output of the ADC feeds the audio switch.

The ADC has its own analog gain stage that provides 0 to +22.5dB of gain in 1.5dB steps. A bit is available that shifts this range down by 6dB to effectively provide -6dB to +16.5dB of gain. The ADC gain is controlled in the *CHIP_ANA_ADC_CTRL* register.

The ADC has an available Zero-Cross Detect (ZCD) that will prevent any volume change until a zero-volt crossing of the audio signal is detected. This helps in eliminating pop or other audio anomalies. If the ADC is to be used, the chip reference bias current should not be set to -50% when in 3V mode.

5.7. Analog Outputs

The SGTL5000 contains a single stereo DAC that can be used to drive a heapdhone output and a line output. The DAC receives its input from the audio switch. The headphone output and the line output can be driven at the same time from the DAC.

The headphone output can also be driven directly by the line input bypassing the ADC and DAC for a very low power mode of operation.

The headphone output is powered by VDDA while the line output is powered by VDDIO. This allows the headphone output to be run at the lowest possible voltage while the line output can still meet line output level requirements.

5.7.1. DAC

The DAC output is routed to the headphone and the dedicated line output.

The DAC output has a digital volume control from -90dB to 0dB in ~.5dB step sizes. This volume is shared among headphone output and line output. The register *CHIP_DAC_VOL* controls the DAC volume.

5.7.2. Headphone

Stereo headphone outputs are provided which can be used to drive a headphone load or a line level output. The headphone output has its own independent analog volume control with a volume range of -52dB to +12dB in .5dB step sizes. This volume control can be used in addition to the DAC volume control. For best performance the DAC volume control should be left at 0dB until the headphone is brought to its lowest setting of -52dB. The register *CHIP_ANA_HP_CTRL* is used to control the headphone volume.

The headphone output has an independent mute that is controlled by the register field *CHIP_ANA_CTRL->MUTE_HP*.

The line input is routed to the headphone output by writing *CHIP_ANA_CTRL->SELECT_HP*. This selection bypasses the ADC and audio switch and routes the line input directly to the headphone output to enable a very low power pass through. When the line input is routed to the headphone output, only the headphone analog volume and mute will affect the headphone output.

The headphone has an available zero cross detect (ZCD) which, as previously described, will prevent any volume change until a zero-volt crossing of the audio signal is detected. This helps in eliminating pop or other audio anomalies.

5.7.3. Line Outputs

The SGTL5000 contains a stereo line output. The line output has a dedicated gain stage that can be used to adjust the output level. The *CHIP_LINE_OUT_VOL* controls the line level output gain.

The line outputs also have a dedicated mute that is controlled by the register field *CHIP_ANA_CTRL->MUTE_LO.*

The lineout volume is intended as maximum output level adjustment. It is intended to be used to set the maximum output swing. It does not have the range of a typical volume control and does not have a zero cross detect (ZCD). However the dac digital volume could be used if volume control is desired

5.8. Digital Input & Output

One I2S (Digital Audio) Port is provided which supports the following formats: I2S, Left Justified, Right Justified and PCM mode.

5.8.1. I2S, Left Justified and Right Justified Modes

I2S, Left Justified and Right Justified modes are stereo interface formats. The I2S_SCLK frequency, I2S_SCLK polarity, I2S_DIN/DOUT data length, and I2S_LRCLK polarity can all be change through the *CHIP_I2S_CTRL* register. For I2S, Left Justified and Right Justified formats the left subframe should always be presented first regardless of the *CHIP_I2S_CTRL->LRPOL* setting.

The I2S_LRCLK and I2S_SCLK can be programmed as master (driven to an external target) or slave (driven from an external source). When the clocks are in slave mode, they must be synchronous to SYS_MCLK. For this reason the SGTL5000 can only operate in synchronous mode (see section 5.4) while in I2S slave mode.

In master mode, the clocks will be synchronous to SYS_MCLK or the output of the PLL when the part is running in asynchronous mode.

Figure 12 shows functional examples of different common digital interface formats and their associated register settings.

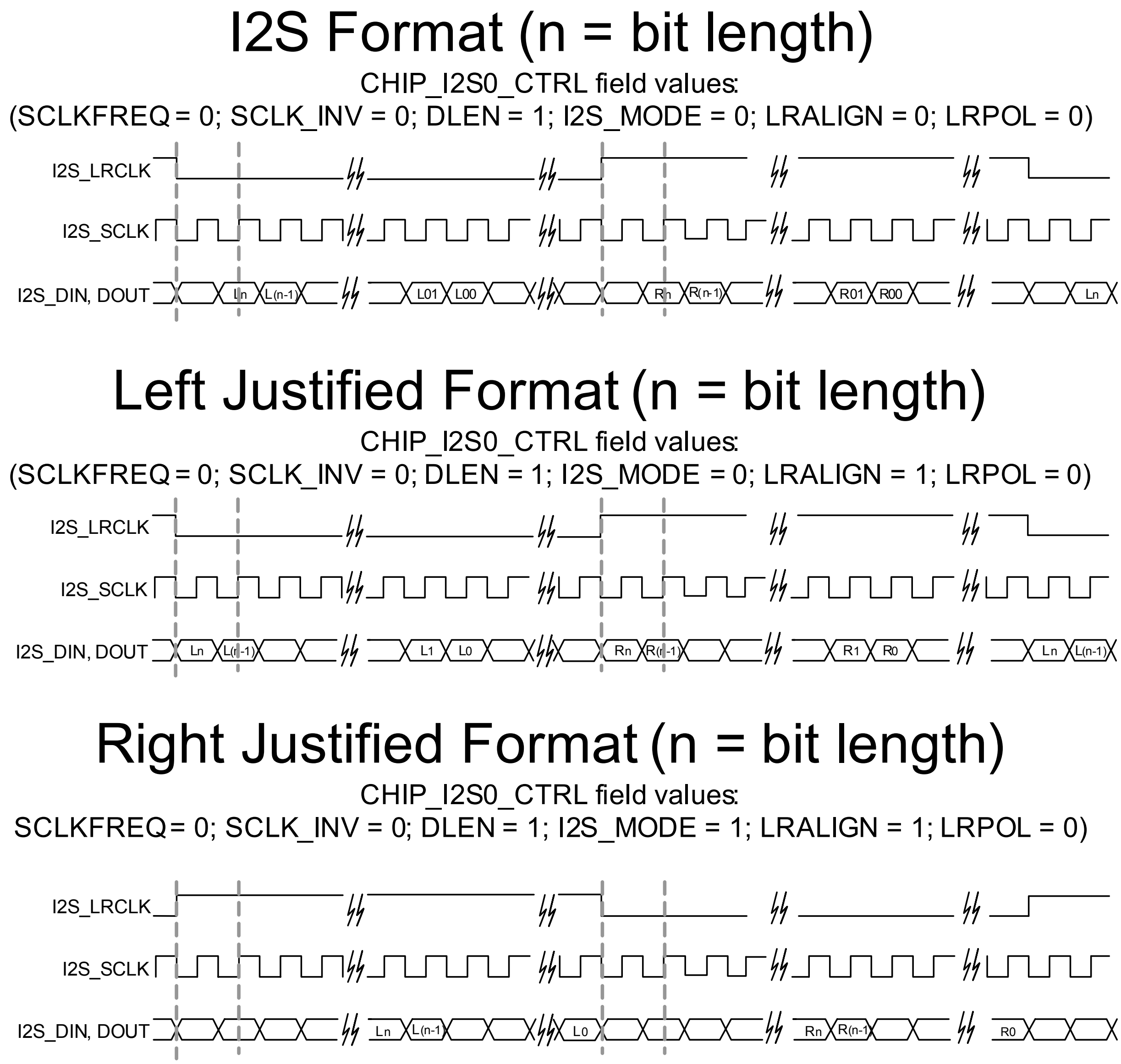

I2S Format (n = bit length)
CHIP_I2S0_CTRL field values:
(SCLKFREQ = 0; SCLK_INV = 0; DLEN = 1; I2S_MODE = 0; LRALIGN = 0; LRPOL = 0)
I2S_LRCLK
I2S_SCLK
I2S_DIN, DOUT
Ln
L(n-1)
L01
L00
Rn
R(n-1)
R01
R00
Ln
Left Justified Format (n = bit length)
CHIP_I2S0_CTRL field values:
(SCLKFREQ = 0; SCLK_INV = 0; DLEN = 1; I2S_MODE = 0; LRALIGN = 1; LRPOL = 0)
I2S_LRCLK
I2S_SCLK
I2S_DIN, DOUT
Ln
L(n-1)
L1
L0
Rn
R(n-1)
R1
R0
Ln
L(n-1)
Right Justified Format (n = bit length)
CHIP_I2S0_CTRL field values:
SCLKFREQ = 0; SCLK_INV = 0; DLEN = 1; I2S_MODE = 1; LRALIGN = 1; LRPOL = 0)
I2S_LRCLK
I2S_SCLK
I2S_DIN, DOUT
Ln
L(n-1)
L0
Rn
R(n-1)
R0

Figure 12. I2S Port Supported Formats

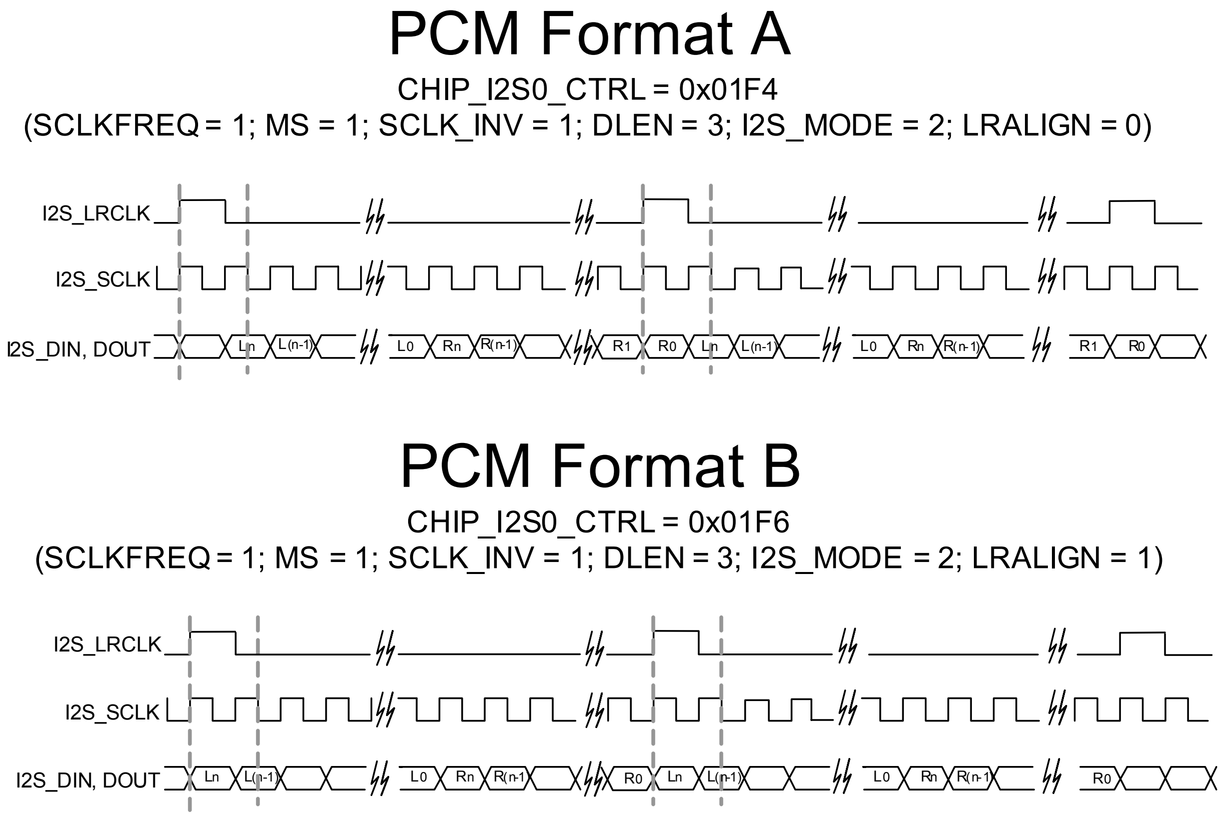
5.8.2. PCM Mode

The I2S port can also be configured into a PCM mode (also known as DSP mode). This mode is provided to allow connectivity to to external devices such as Bluetooth modules. PCM mode differs from other interface formats presented in section 5.8.1 in that the frame clock (I2S_LRCLK) does not represent a different channel when high or low, but is a bit-wide pulse that marks the start of a frame. Data is aligned such that the left channel data is immediately followed by right channel data. Zero padding is filled in for the remaining bits. The data and frame clock may be configured to clock in on the rising or falling edge of Bit Clock.

PCM Format A signifies the data word beginning one SCLK bit following the I2S_LRCLK transition, as in I2S Mode. PCM Format B signifies the data word beginning after the I2S_LRCLK transition, as in Left Justified.

In Slave mode, the pulse width of the I2S_LRCLK does not matter. The pulse can range from one cycle high to all but one cycle high. In Master mode, it will be driven one cycle high.

Figure 13 shows a functional drawing of the different formats in master mode.



PCM Format A

CHIP_I2S0_CTRL = 0x01F4
(SCLKFREQ = 1; MS = 1; SCLK_INV = 1; DLEN = 3; I2S_MODE = 2; LRALIGN = 0)

PCM Format B

CHIP_I2S0_CTRL = 0x01F6
(SCLKFREQ = 1; MS = 1; SCLK_INV = 1; DLEN = 3; I2S_MODE = 2; LRALIGN = 1)

Figure 13. PCM Formats

5.9. Digital Audio Processing

The SGTL5000 contains a digital audio processing block (DAP) attached to the source select switch. The digitized signal from the source select switch can be routed into the DAP block for audio processing. The DAP has the following 5 sub blocks:

- Dual Input Mixer
- SigmaTel Surround
- SigmaTel Bass Enhancement
- 7-Band Paramter EQ / 5-Band Graphic EQ / Tone Control (only one can be used at a time)
- Automatic Volume Control (AVC)

The block diagram in Figure 14 shows the sequence in which the signal passes through these blocks.

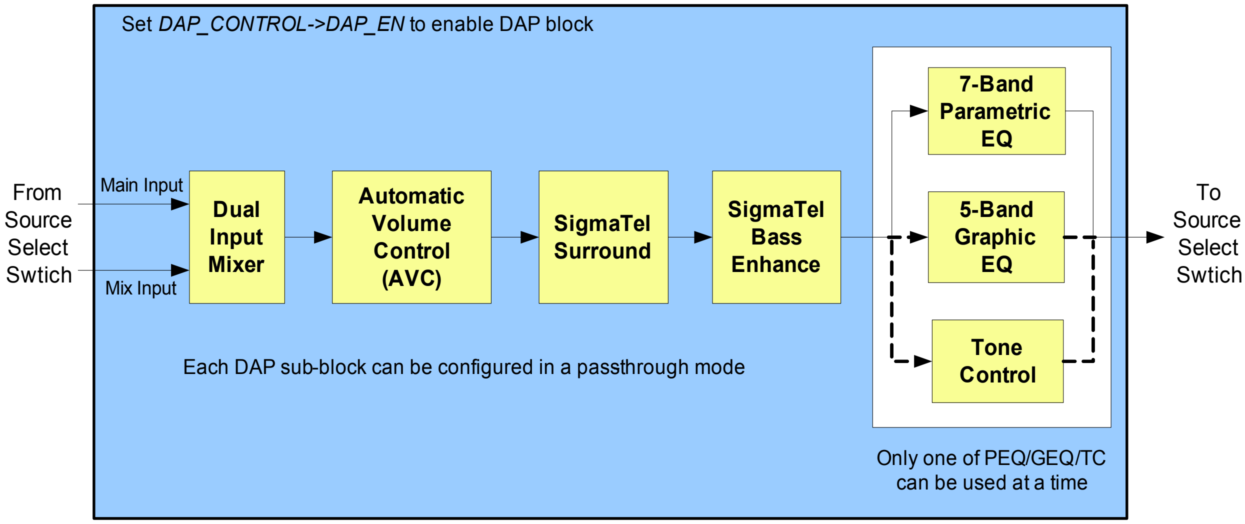

Set *DAP_CONTROL->DAP_EN* to enable DAP block
From Source Select Swtich
Main Input
Mix Input
Dual Input Mixer
Automatic Volume Control (AVC)
SigmaTel Surround
SigmaTel Bass Enhance
7-Band Parametric EQ
5-Band Graphic EQ
Tone Control
To Source Select Swtich
Each DAP sub-block can be configured in a passthrough mode
Only one of PEQ/GEQ/TC can be used at a time

Figure 14. Digital Audio Processing Block Diagram

When the DAP block is added in the route, it must be enabled separately to get audio through. It is recommended to mute the outputs before enabling/disabling the DAP block to avoid any pops or clicks due to discontinuities in the output.

Refer to section 6.2.4 for programming examples on how to enable/disable the DAP block.

Each sub-block of the DAP can be individually disabled if its processing is not required. The sections below describes the DAP sub-blocks and how to configure them.

5.9.1. Dual Input Mixer

The dual input digital mixer allows for two incoming streams from the source select switch as shown in Figure 15.

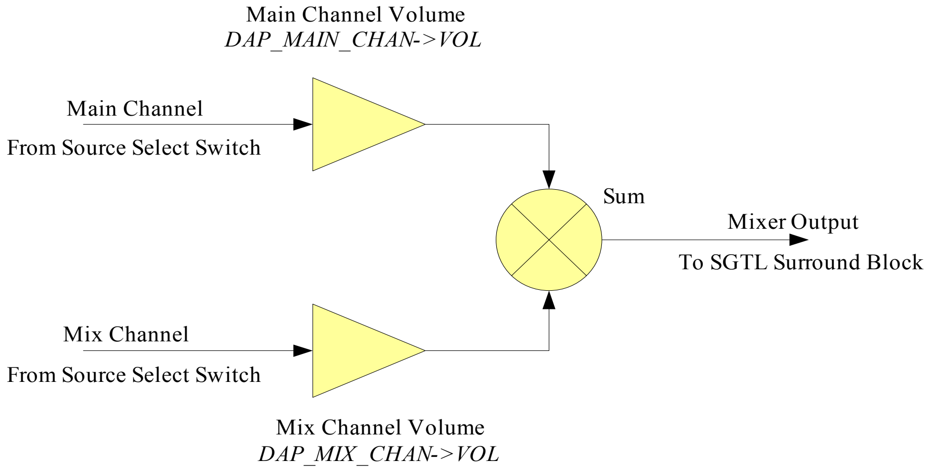


Figure 15. DAP - Dual Input Mixer

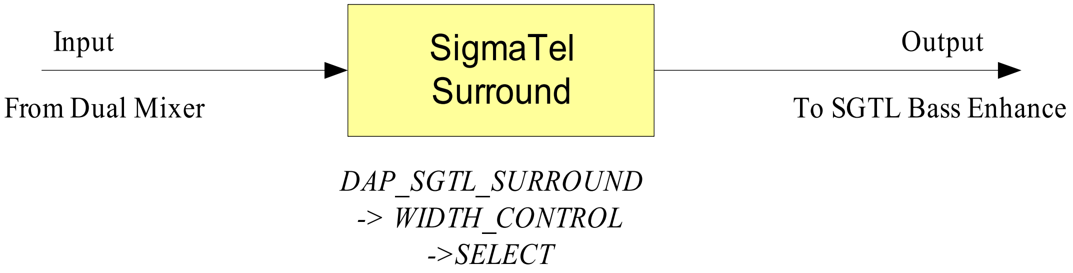
The Dual Input Mixer can be enabled or configured in a pass-through mode (Main channel will be passed through without any mixing). When enabled, the volume of the main and mix channels can be independently controlled before they are mixed together.

The volume range allowed on each channel is 0% to 200% of the incoming signal level. The default is 100% (same as input signal level) volume on the main input and 0% (muted) on the mix input.

Please refer to section 6.2.4.1 for programming examples on how to enable/disable the mixer and also to set the main and mix channel volume.

5.9.2. *SigmaTel Surround*

SigmaTel Surround is a royalty free virtual surround algorithm for stereo or mono inputs. It widens and deepens sound stage for music input.



The SGTL Surround can be enabled or configured in pass-through mode (input will be passed through without any processing). When enabling the Surround, mono or stereo input type must be selected based on the input signal. Surround width may be adjusted for the size of the sound stage.

Please refer to section 6.2.4.2 and section 6.3.5 for a programming example on how to configure Surround width and how to enable/disable Surround.

5.9.3. SigmaTel Bass Enhance

SigmaTel Bass Enhance is a royalty-free algorithm that enhances natural bass response of the audio. Bass Enhance extracts bass content from right and left channels, adds bass and mixes this back up with the original signal. An optional complementary high pass filter is provided after the mixer.

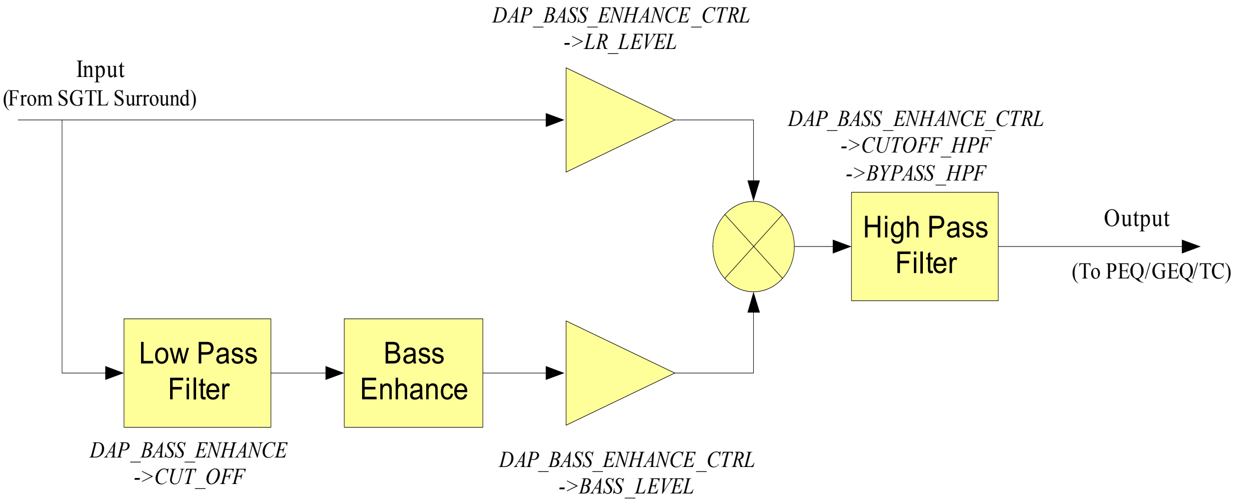


Figure 16. DAP- SigmaTel Bass Enhance

The SGTL Bass Enhance can be enabled or configured in pass-through mode (input will be passed through without any processing).

The cut-off frequency of the low-pass filter (LPF) can be selected based on the speakers freqeuncy response. The cut-off frequency of the low-pass and high-pass filters are selectable between 80Hz to 225Hz. Also, the input signal and bass enhanced signal can be individually adjusted for level before the two signals are mixed.

Please refer to section 6.2.4.3 and section 6.3.6 for a programming example on how to configure Bass Enhance and how to enable/disable this feature.

5.9.4. 7-Band Parametric EQ / 5-Band Graphic EQ / Tone Control

One 7-band parametric equalizer (PEQ) and one 5-band graphic equalizer (GEQ) and a Tone Control (Bass and Treble control) blocks are implemented as mutually exclusive blocks. Only one block can be used at a given time.

Please refer to section 6.2.4.4 for a programming example that shows how to select the desired EQ mode.

5.9.4.1. 7-Band Parametric EQ

The 7-band PEQ allows the designer to compensate for speaker response and to provide the ability to filter out resonant frequencies caused by the physical system design. The system designer can create custom EQ presets such Rock, Speech, Classical etc that allows the users the flexibility in customizing their audio.

The 7-band PEQ is implemented using 7 cascaded second order IIR filters. All filters are implemented using programmable biquad filters. Figure 17 shows the transfer function and Direct Form 1 of the five coefficient biquadratic filter.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

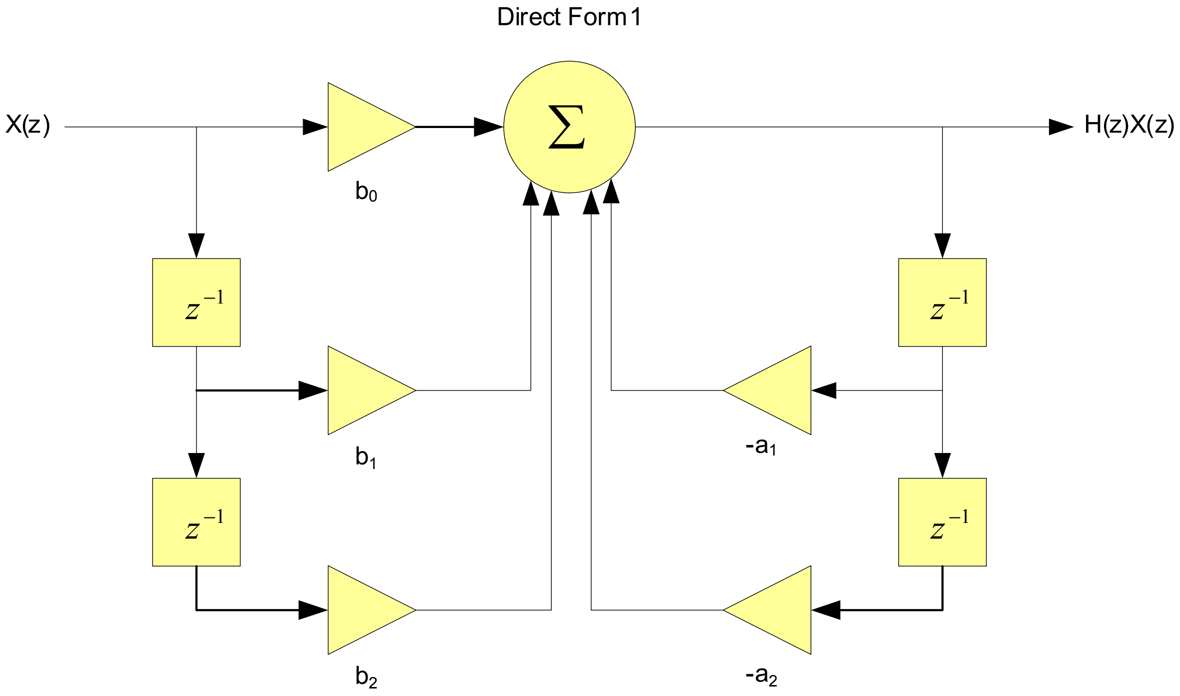


Figure 17. 5-Coefficient Biquad Filter and Transfer Function

If a band is enabled but is not being used (flat response), then a value of 0.5 should be put in b_0 and all other coefficients should be set to 0.0. Please note that the coefficients must be converted to hex values before writing to the registers. By default, all the filters are loaded with coefficients to give a flat response.

In order to create EQ presets such as Rock, Speech, Classical etc, the coefficients must be calculated, converted to 20-bit hex values and written to the registers. Note that coefficients are sample-rate dependent and separate coefficients must be generated for different sample rates. Please contact Freescale for assistance with generating the coefficients.

Please refer to section 6.3.2 for a programming example that shows how load the filter coefficients when the end-user changes the preset.

PEQ can be disabled (pass-through mode) by writing 0 to DAP_AUDIO_EQ->EN bits.

5.9.4.2. 5-Band Graphic EQ

The 5-band graphic equalizer is implemented using 5 parallel second order IIR filters. All filters are implemented using biquad filters whose coefficients are programmed to set the bands at specific frequency. The GEQ bands are fixed at 115Hz, 330Hz, 990Hz, 3000Hz, and 9900Hz. The volume on each band is independently adjustable in the range of +12dB to -11.75 dB in 0.25dB steps.

Please refer to section 6.3.3 for a programming example that shows how to change the GEQ volume

5.9.4.3. Tone Control

Tone control comprises treble and bass controls. The tone control is implemented as one 2nd order low pass filter (bass) and one 2nd order high pass filter (treble).

Please refer to section section 6.3.4 for a programming example that shows how to change Bass and Treble values.

5.9.5. Automatic Volume Control (AVC)

An Automatic Volume Control (AVC) block is provided to reduce loud signals and amplify low level signals for easier listening. The AVC is designed to compress audio when the measured level is above the programmed threshold or to expand the audio to the programmed threshold when the measured audio is below the threshold. The threshold level is programmable with allowed range of 0dB to -96 dB. Figure 18 shows the AVC block diagram and controls.

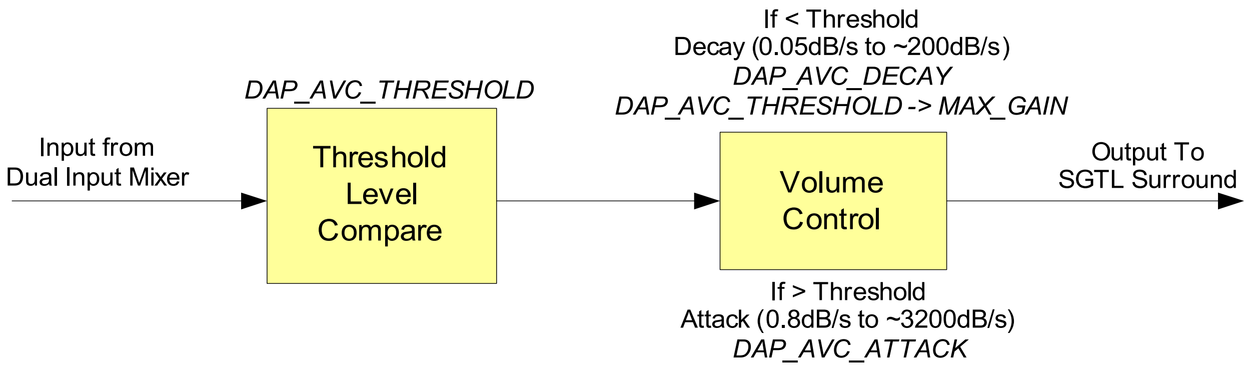


Figure 18. DAP AVC Block Diagram

When the measured audio level is below threshold, the AVC can apply a maximum gain of up to 12dB. The maximum gain can be selected, either 0, 6 or 12dB. When the maximum gain is set to 0dB the AVC acts as a limiter. In this case the AVC will only take effect when the signal level is above the threshold.

The rate at which the incoming signal is attenuated down to the threshold is called the attack rate. Too high of an attack will cause an unnatural sound as the input signal is distorted. Too low of an attack may cause saturation of the output as the incoming signal will not be compressed quickly enough. The attack rate is programmable with allowed range of 0.05dB/s to 200dB/s.

When the signal is below the threshold, AVC will adjust the volume up until either the threshold or the maximum gain is reached. The rate at which this volume is changed is called the decay rate. The decay rate is programmable with allowed range of 0.8dB/s to 3200dB/s. It is desirable to use very slow decay rate to avoid any distortion in the signal and prevent the AVC from entering a contiuous attack-decay loop.

Please refer to section 6.2.4.5 and section 6.3.7 for a programming example that shows how to configure AVC and how to enable/disable AVC respectively.

5.10. Control

The SGTL5000 supports both I2C and SPI control modes. the CTRL_MODE pin chooses which mode will be used. When CTRL_MODE is tied to ground, the control mode is I2C. When CTRL_MODE is tied to VDDIO, the control mode is SPI.

Regardless of the mode, the control interface is used for all communication with the SGTL5000 including startup configuration, routing, volume, etc.

5.10.1. I2C

The I2C port is implemented according to the I2C specification v2.0. The I2C interface is used to read and write all registers.

For the 32QFN version of the SGTL5000, the I2C device address is 0n01010(R/W) where n is determined by I2C_ADR0_CS and R/W is the read/write bit from the I2C protocol.

For the 20QFN version of the SGTL5000 the I2C address is always 0001010(R/W).

The SGTL5000 is always the slave on all transactions which means that an external master will always drive CTRL_CLK.

In general an I2C transaction looks as follows.

All locations are accessed with a 16 bit address. Each location is 16 bits wide.

An example I2C write transaction follows:

- Start condition
- Device address with the R/W bit cleared to indicate write
- Send two bytes for the 16 bit register address (most significant byte first)

- Send two bytes for the 16 bits of data to be written to the register (most significant byte first)
- Stop condition

An I2C read transaction is defined as follows:

- Start condition
- Device address with the R/W bit cleared to indicate write
- Send two bytes for the 16 bit register address (most significant byte first)
- Stop Condition followed by start condition (or a single restart condition)
- Device address with the R/W bit set to indicate read
- Read two bytes from the addressed register (most significant byte first)
- Stop condition

Figure 19 shows the functional I2C timing diagram.

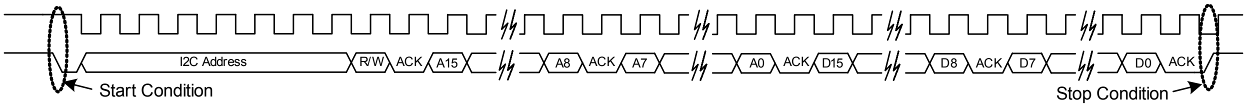


Figure 19. Functional I2C Diagram

The protocol has an auto increment feature. Instead of sending the stop condition after two bytes of data, the master may continue to send data byte pairs for writing, or it may send extra clocks for reading data byte pairs. In either case, the access address is incremented after every two bytes of data. A start or stop condition from the I2C master interrupts the current command. For reads, unless a new address is written, a new start condition with R/W=0 reads from the current address and continues to auto increment.

The following diagrams describe the different access formats. The gray fields are from the I2C master, and the white fields are the SGTL5000 responses. Data[n] corresponds to the data read from the address sent, data[n+1] is the data from the next register, and so on.

S = Start Condition

Sr = Restart Condition

A = Ack

N = Nack

P = Stop Condition

TA2 silicon will allow for up to a 3.6V I2C signal level, regardless of the VDDIO level.

Table 13. Write Single Location

S	Device Address	W (0)	A	ADDR byte 1	A	ADDR byte 0	A	DATA byte 1	A	DATA byte 0	A	P

Table 14. Write Auto increment

S	Device Address	W (0)	A	start ADDR byte 1	A	start ADDR byte 0	A	DATA [n] byte 1	A	DATA [n] byte 0	A	DATA [n+1] byte 1	A	DATA [n+1] byte 0	A	P

Table 15. Read Single Location

S	Device Address	W (0)	A	ADDR byte 1	A	ADDR byte 0	A	Sr	Device Address	R (1)	A	DATA byte 1	A	DATA byte 0	N	P

Table 16. Read Auto increment

S	Device Address	W (0)	A	start ADDR byte 1	A	start ADDR byte 0	A	Sr	Device Address	R (1)	A	DATA [n] byte 1	A	DATA [n] byte 0	A	DATA [n+1] byte 1	A	DATA [n+1] byte 0	N	P

Table 17. Read Continuing Auto increment

S	Device Address	R	A	DATA [n+2] byte 1	A	DATA [n+2] byte 0	A	DATA [n+3] byte 1	A	DATA [n+3] byte 0	N	P

5.10.2. *SPI*

Serial Peripheral Interface (SPI) is a communications protocol supported by the SGTL5000. The SGTL5000 is always a slave. The CTRL_AD0_CS is used as the slave select (SS) when the master wants to select the SGTL5000 for communication. CTRL_CLK is connected to master's SCLK and CTRL_DATA is connected to master's MOSI line. The part only supports allows SPI write operations and does not support read operations.

Figure 20 below shows the functional timing diagram of the SPI communication protocol as supported by SGTL5000 chip. Note that on the rising edge of the SS, the chip latches to previous 32 bits of data. It interprets the latest 16-bits as register value and 16-bits preceding it as register address.

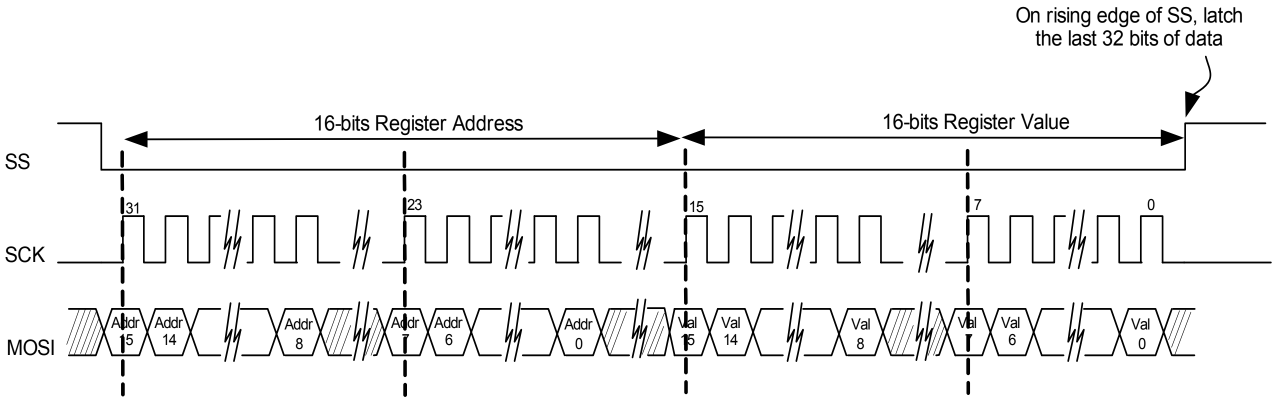


Figure 20. Functional Timing Diagram of SPI Protocol

6. PROGRAMMING EXAMPLES

This section provides programming examples that show how to configure the chip. The registers can be written/read by using I2C communication protocol. The chip also supports SPI communication protocol but only register write operation is supported.

6.1. Prototype for Reading and Writing a Register

The generic register read write prototype will be used throughout this section as shown below. The I2C or SPI implementation will be specific to the I2C/SPI hardware used in the system.

```
// This prototype writes a value to the entire register.  All
// bit-fields of the register will be written.
Write  REGISTER  REGISTERVALUE

// This prototype writes a value only to the bit-field specified.
// In the actual implemention, the other bit-fields should be
// masked to prevent them from being written.  Also, the
// actual implementation should left-shift the BITFIELDVALUE
// by appropriate number to match the starting bit location of
// the BITFIELD.
Modify  REGISTER -> BITFIELD,   BITFIELDVALUE //Bitfield Location


// Example implementation
// Modify DAP_EN (bit 0) bit to value 1 to enable DAP block
Modify( DAP_CONTROL_REG, 0xFFFE, 1 << DAP_EN_STARTBIT );


// Example Implementation of Modify
void Modify( unsigned short usRegister,
               unsigned short usClearMask,
               unsigned short usSetValue )
{
  unsigned short usData;

  // 1) Read current value
  ReadRegister( usRegister, &usData );

  // 2) Clear out old bits
  usData = usData & usClearMask;

  // 3) set new bit values
  usData = usData | usSetValue;

  // 4) Write out new value created
  WriteRegister( usRegister, usData );
}
```

6.2. Chip Configuration

All outputs (LINEOUT, HP_OUT, I2S_OUT) are muted by default on powerup. To avoid any pops/clicks, the outputs should remain muted during these chip configuration steps. Refer to section 6.2.6 for volume and mute control.

6.2.1. *Initialization*

6.2.1.1. Chip Powerup and Supply Configurations

After the power supplies for chip is turned on, following initialization sequence should be followed. Please note that certain steps may be optional or different values may need to be written based on the power supply voltage used and desired configuration. The initialization sequence below assumes VDDIO = 3.3V and VDDA = 1.8V.

```
//--------------- Power Supply Configuration----------------
// NOTE:  This next 2 Write calls is needed ONLY if VDDD is
// internally driven by the chip
// Configure VDDD level to 1.2V (bits 3:0)
Write CHIP_LINREG_CTRL    0x0008
// Power up internal linear regulator (Set bit 9)
Write CHIP_ANA_POWER      0x7260

// NOTE:  This next Write call is needed ONLY if VDDD is
// externally driven
// Turn off startup power supplies to save power (Clear bit 12 and 13)
Write CHIP_ANA_POWER      0x4260

// NOTE:  The next 2 Write calls is needed only if both VDDA and
// VDDIO power supplies are less than 3.1V.
// Enable the internal oscillator for the charge pump (Set bit 11)
Write CHIP_CLK_TOP_CTRL   0x0800
// Enable charge pump (Set bit 11)
Write CHIP_ANA_POWER      0x4A60

// NOTE:  The next 2 modify calls is only needed if both VDDA and
// VDDIO are greater than 3.1V
// Configure the chargepump to use the VDDIO rail (set bit 5 and bit 6)
Write CHIP_LINREG_CTRL    0x006C

//------ Reference Voltage and Bias Current Configuration----------
// NOTE:  The value written in the next 2 Write calls is dependent
// on the VDDA voltage value.
// Set ground, ADC, DAC reference voltage (bits 8:4).  The value should
// be set to VDDA/2.   This example assumes VDDA = 1.8V. VDDA/2 =  0.9V.
// The bias current should be set to 50% of the nominal value (bits 3:1)
Write CHIP_REF_CTRL       0x004E
// Set LINEOUT reference voltage to VDDIO/2 (1.65V) (bits 5:0) and bias cur-
rent (bits 11:8) to the recommended value of 0.36mA for 10kOhm load with 1nF
capacitance
Write CHIP_LINE_OUT_CTRL  0x0322

//----------------Other Analog Block Configurations------------------
// Configure slow ramp up rate to minimize pop (bit 0)
Write CHIP_REF_CTRL       0x004F

// Enable short detect mode for headphone left/right
// and center channel and set short detect current trip level
// to 75mA
Write CHIP_SHORT_CTRL      0x1106

// Enable Zero-cross detect if needed for HP_OUT (bit 5) and ADC (bit 1)
Write CHIP_ANA_CTRL         0x0133

//----------------Power up Inputs/Outputs/Digital Blocks-------------
// Power up LINEOUT, HP, ADC, DAC
Write CHIP_ANA_POWER        0x6AFF

// Power up desired digital blocks
// I2S_IN (bit 0), I2S_OUT (bit 1), DAP (bit 4), DAC (bit 5),
// ADC (bit 6) are powered on
Write CHIP_DIG_POWER        0x0073

//--------------------Set LINEOUT Volume Level-----------------------
// Set the LINEOUT volume level based on voltage reference (VAG)
// values using this formula
// Value = (int)(40*log(VAG_VAL/LO_VAGCNTRL) + 15)
// Assuming VAG_VAL and LO_VAGCNTRL is set to 0.9V and 1.65V respectively, the
// left LO vol (bits 12:8) and right LO volume (bits 4:0) value should be set
// to 5
Write CHIP_LINE_OUT_VOL     0x0505
```

6.2.1.2. System MCLK and Sample Clock

```
// Configure SYS_FS clock to 48kHz
// Configure MCLK_FREQ to 256*Fs
Modify CHIP_CLK_CTRL->SYS_FS 0x0002  // bits 3:2
Modify CHIP_CLK_CTRL->MCLK_FREQ 0x0000  // bits 1:0

// Configure the I2S clocks in master mode
// NOTE: I2S LRCLK is same as the system sample clock
Modify  CHIP_I2S_CTRL->MS  0x0001  // bit 7
```

6.2.2. PLL Configuration

These programming steps are needed only when the PLL is used. Please refer to section 5.4.2 for details on when to use the PLL.

To avoid any pops/clicks, the outputs should be muted during these chip configuration steps. Refer to section 6.2.6 for volume and mute control.

```
// Power up the PLL
Modify  CHIP_ANA_POWER->PLL_POWERUP  0x0001  // bit 10
Modify  CHIP_ANA_POWER->VCOAMP_POWERUP  0x0001  // bit 8

// NOTE:  This step is required only when the external SYS_MCLK
// is above 17MHz. In this case the external SYS_MCLK clock
// must be divided by 2
Modify  CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2  0x0001  // bit 3
Sys_MCLK_Input_Freq = Sys_MCLK_Input_Freq/2;

// PLL output frequency is different based on the sample clock
// rate used.
if (Sys_Fs_Rate == 44.1kHz)
   PLL_Output_Freq = 180.6336MHz
else
   PLL_Output_Freq = 196.608MHz
```

```
// Set the PLL dividers
Int_Divisor = floor(PLL_Output_Freq/Sys_MCLK_Input_Freq)
Frac_Divisor = ((PLL_Output_Freq/Sys_MCLK_Input_Freq) - Int_Divisor)*2048
Modify CHIP_PLL_CTRL->INT_DIVISOR Int_Divisor  // bits 15:11
Modify CHIP_PLL_CTRL->FRAC_DIVISOR Frac_Divisor  // bits 10:0
```

6.2.3. *Input/Output Routing*

To avoid any pops/clicks, the outputs should be muted during these chip configuration steps. Refer to section 6.2.6 for volume and mute control.

A few example routes are shown below:

```
// Example 1:  I2S_IN -> DAP -> DAC -> LINEOUT, HP_OUT

// Route I2S_IN to DAP
Modify CHIP_SSS_CTRL->DAP_SELECT 0x0001  // bits 7:6
// Route DAP to DAC
Modify CHIP_SSS_CTRL->DAC_SELECT 0x0003  // bits 5:4
// Select DAC as the input to HP_OUT
Modify CHIP_ANA_CTRL->SELECT_HP 0x0000  // bit 6


// Example 2:  MIC_IN -> ADC -> I2S_OUT

// Set ADC input to MIC_IN
Modify CHIP_ANA_CTRL->SELECT_ADC 0x0000  // bit 2
// Route ADC to I2S_OUT
Modify CHIP_SSS_CTRL->I2S_SELECT 0x0000  // bits 1:0

// Example 3:  LINEIN -> HP_OUT

// Select LINEIN as the input to HP_OUT
Modify CHIP_ANA_CTRL->SELECT_HP 0x0001  // bit 6
```

6.2.4. *Digital Audio Processor Configuration*

To avoid any pops/clicks, the outputs should be muted during these chip configuration steps. Refer to section 6.2.6 for volume and mute control.

```
// Enable DAP block
// NOTE:  DAP will be in a pass-through mode if none of DAP
// sub-blocks are enabled.
Modify DAP_CONTROL->DAP_EN 0x0001  // bit 0
```

6.2.4.1. *Dual Input Mixer*

These programming steps are needed only if dual input mixer feature is used.

```
// Enable Dual Input Mixer
Modify DAP_CONTROL->MIX_EN 0x0001  // bit 4

// NOTE: This example assumes mix level of main and mix
// channels as 100% and 50% respectively

// Configure main channel volume to 100% (No change from input
// level)
```

```
Write DAP_MAIN_CHAN 0x4000

// Configure mix channel volume to 50% (attenuate the mix
// input level by half)
Write DAP_MIX_CHAN 0x4000
```

6.2.4.2. SigmaTel Surround

The SigmaTel Surround on/off function will be typically controlled by the end-user. End-user driven programming steps are shown in section 6.3.

The default WIDTH_CONTROL of 4 should be appropriate for most applications. This optional programming step shows how to configure a different width value.

```
// Configure the surround width
// (0x0 = Least width, 0x7 = Most width).  This example shows
// a width setting of 5
Modify DAP_SGTL_SURROUND->WIDTH_CONTROL 0x0005  // bits 6:4
```

6.2.4.3. SigmaTel Bass Enhance

The SigmaTel Bass Enhance on/off function will be typically controlled by the end-user. End-user driven programming steps are shown in section 6.3.

The default LR_LEVEL value of 0x0005 results in no change in the input signal level and BASS_LEVEL value of 0x001F adds some harmonic boost to the main signal. The default settings should work for most applications. This optional programming step shows how to configure a different value.

```
// Gain up the input signal level
Modify DAP_BASS_ENHANCE_CTRL->LR_LEVEL 0x0002  // bits 7:4

// Add harmonic boost
Modify DAP_BASS_ENHANCE_CTRL->BASS_LEVEL 0x003F);  // bits 6:0
```

6.2.4.4. 7-Band Parametric EQ / 5-Band Graphic EQ / Tone Control

Only one audio EQ block can be used at a given time. The psuedocode in this section shows how to select each block.

Some parameters of the audio EQ will typically be controlled by end-user. End-user driven programming steps are shown in section 6.3.

```
// 7-Band PEQ Mode
// Select 7-Band PEQ mode and enable 7 PEQ filters
Write DAP_AUDIO_EQ 0x0001
Write DAP_PEQ 0x0007

// Tone Control mode
Write DAP_AUDIO_EQ 0x0002

// 5-Band GEQ Mode
Write DAP_AUDIO_EQ 0x0003
```

6.2.4.5. *Automatic Volume Control (AVC)*

The AVC on/off function will be typically controlled by the end-user. End-user driven programming steps are shown in section 6.3.

The default configuration of the AVC should work for most applications. However, the following example shows how to change the configuration if needed.

```
// Configure threshold to -18dB
Write DAP_AVC_THRESHOLD 0x0A40

// Configure attack rate to 16dB/s
Write DAP_AVC_ATTACK 0x0014

// Configure decay rate to 2dB/s
Write DAP_AVC_DECAY 0x0028
```

6.2.5. *I2S Configuration*

By default the I2S port on the chip is configured for 24-bits of data in I2S format with SCLK set for 64*Fs. This can be modified by setting various bit-fields in CHIP_I2S_CTRL register.

6.2.6. *Volume Control*

The outputs should be unmuted after all the configuration is complete.

```
//---------------- Input Volume Control---------------------
// Configure ADC left and right analog volume to desired default.
// Example shows volume of 0dB
Write CHIP_ANA_ADC_CTRL 0x0000

// Configure MIC gain if needed.  Example shows gain of 20dB
Modify CHIP_MIC_CTRL->GAIN 0x0001  // bits 1:0

//---------------- Volume and Mute Control---------------------
// Configure HP_OUT left and right volume to minimum, unmute
// HP_OUT and ramp the volume up to desired volume.
Write CHIP_ANA_HP_CTRL 0x7F7F
Modify CHIP_ANA_CTRL->MUTE_HP 0x0000  // bit 5

// Code assumes that left and right volumes are set to same value
// So it only uses the left volume for the calculations
usCurrentVolLeft = 0x7F;
usNewVolLeft = usNewVol & 0xFF;
usNumSteps = usNewVolLeft - usCurrentVolLeft;
if (usNumSteps == 0) return;

// Ramp up
for (int i = 0; i < usNumSteps; i++ )
{
   ++usCurrentVolLeft;
   usCurrentVol = (usCurrentVolLeft << 8) | (usCurrentVolLeft);
   Write CHIP_ANA_HP_CTRL usCurrentVol;
}

// LINEOUT and DAC volume control
Modify CHIP_ANA_CTRL->MUTE_LO 0x0000  // bit 8
// Configure DAC left and right digital volume.  Example shows
// volume of 0dB
```

```
Write CHIP_DAC_VOL 0x3C3C
Modify CHIP_ADCDAC_CTRL->DAC_MUTE_LEFT 0x0000  // bit 2
Modify CHIP_ADCDAC_CTRL->DAC_MUTE_RIGHT 0x0000  // bit 3

// Unmute ADC
Modify CHIP_ANA_CTRL->MUTE_ADC 0x0000  // bit 0
```

6.3. End-user Driven Chip Configuration

End-users will control features like volume up/down, audio EQ parameters such as Bass and Treble. This will require programming the chip without introducing any pops/clicks or any disturbance to the output. This section shows examples on how to program these features.

6.3.1. Volume and Mute Control

Refer to section 6.2.6 for examples on how to program volume when end-user changes the volume or mutes/unmutes output. Note that the DAC volume ramp is automatically handled by the chip.

6.3.2. 7-Band PEQ Preset Selection

This programming example shows how to load the filter coefficients when the end-user changes PEQ presets such as Rock, Speech, Classical etc.

```
// Load the 5 coefficients for each band and write them to
// appropriate filter address.  Repeat this for all enabled
// filters (this example shows 7 filters)
for (i = 0; i < 7; i++)
{
   // Note that each 20-bit coefficient is broken into 16-bit MSB
   // (unsigned short usXXMSB) and 4-bit LSB (unsigned short
   // usXXLSB)
   Write DAP_COEF_WR_B0_LSB  usB0MSB[i]
   Write DAP_COEF_WR_B0_MSB  usB0LSB[i]
   Write DAP_COEF_WR_B1_LSB  usB1MSB[i]
   Write DAP_COEF_WR_B1_MSB  usB1LSB[i]
   Write DAP_COEF_WR_B2_LSB  usB2MSB[i]
   Write DAP_COEF_WR_B2_MSB  usB2LSB[i]
   Write DAP_COEF_WR_A1_LSB  usA1MSB[i]
   Write DAP_COEF_WR_A1_MSB  usA1LSB[i]
   Write DAP_COEF_WR_A2_LSB  usA2MSB[i]
   Write DAP_COEF_WR_A2_MSB  usA2LSB[i]
   // Set the index of the filter (bits 7:0) and load the
   // coeffiecents
   Modify DAP_FILTER_COEF_ACCESS->INDEX (0x0101 + i) // bit 8
}
```

6.3.3. 5-Band GEQ Volume Change

This programming example shows how to program the GEQ volume when end-user changes the volume on any of the 5 bands.

GEQ volume should be ramped in 0.5 dB steps in order to avoid any pops. The example assumes that volume is ramped on Band 0. Other bands can be programmed similiarly.

```
// Read current volume set on Band 0
usCurrentVol = Read DAP_AUDIO_EQ_BASS_BAND0

// Convert the new volume to hex value
usNewVol = 4*dNewVolDb + 47;

// Calculate the number of steps
usNumSteps = abs(usNewVol - usCurrentVol);
if (usNumSteps == 0) return;
for (int i = 0; i++; usNumSteps )
{
  if (usNewVol > usCurrentVol)
    ++usCurrentVol;
  else
    --usCurrentVol;

  Write DAP_AUDIO_EQ_BASS_BAND0 usCurrentVol;
}
```

6.3.4. *Tone Control - Bass and Treble Change*

This programming example shows how to program the Tone Control Bass and Treble when end-user changes it on the fly.

Tone Control Bass and Treble volume should be ramped in 0.5 dB steps in order to avoid any pops. The example assumes that Treble is changed to a new value. Bass can be programmed similarly.

```
// Read current Treble value
usCurrentVal = Read DAP_AUDIO_EQ_TREBLE_BAND4

// Convert the new Treble value to hex value
usNewVal = 4*dNewValDb + 47;

// Calculate the number of steps
usNumSteps = abs(usNewVal - usCurrentVal);
if (usNumSteps == 0) return;
for (int i = 0; i++; usNumSteps )
{
  if (usNewVal > usCurrentVal)
    ++usCurrentVal;
  else
    --usCurrentVal;

  Write DAP_AUDIO_EQ_TREBLE_BAND4 usCurrentVal;
}
```

6.3.5. *SigmaTel Surround On/Off*

This programming example shows how to program the Surround when end-user turns it on/off on his device.

The Surround width should be ramped up to highest value before enabling/disabling the Surround to avoid any pops.

```
// Read current Surround width value
// WIDTH_CONTROL bits 6:4
usOriginalVal = (Read DAP_SGTL_SURROUND >> 4) && 0x0003;
usNextVal = usOriginalVal;

// Ramp up the width to maximum value of 7
for (int i = 0; i++; (7 - usOriginalVal)
{
  ++usNextVal;
  Modify DAP_SGTL_SURROUND->WIDTH_CONTROL usNextVal;
}

// Enable (To disable, write 0x0000) Surround
// SELECT bits 1:0
Modify DAP_SGTL_SURROUND->SELECT 0x0003;

// Ramp down the width to original value
for (int i = 0; i++; (7 - usOriginalVal)
{
  --usNextVal;
  Modify DAP_SGTL_SURROUND->WIDTH_CONTROL usNextVal;
}
```

6.3.6. *Bass Enhace On/Off*

This programming example shows how to program the Bass Enhance on/off when end-user turns it on/off on his device.

The Bass level should be ramped down to the lowest Bass before Bass Enhance feature is turned on/off.

```
// Read current Bass level value
// BASS_LEVEL bits 6:0
usOriginalVal = Read DAP_BASS_ENHANCE_CTRL && 0x007F;
usNextVal = usOriginalVal;

// Ramp Bass level to lowest bass (lowest bass = 0x007F)
usNumSteps = abs(0x007F - usOriginalVal);
for (int i = 0; i++; usNumSteps )
{
  ++usNextVal;
  Modify DAP_BASS_ENHANCE_CTRL->BASS_LEVEL usNextVal;
}

// Enable (To disable, write 0x0000) Bass Enhance
// EN bit 0
Modify DAP_BASS_ENHANCE->EN 0x0001;

// Ramp Bass level back to original value
for (int i = 0; i++; usNumSteps )
{
  --usNextVal;
  Modify DAP_BASS_ENHANCE_CTRL->BASS_LEVEL usNextVal;
}
```

6.3.7. Automatic Volume Control (AVC) On/Off

This programming example shows how to program the AVC on/off when end-user turns it on/off on his device.

```
// Enable AVC (To disable, write 0x0000)
Modify DAP_AVC_CTRL->EN 0x0001  // bit 0
```

7. REGISTER DESCRIPTION

7.0.0.1. CHIP_ID 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTID								REVID							

BITS	FIELD	RW	RESET	DEFINITION
15:8	PARTID	RO	0xA0	SGTL5000 Part ID 0xA0 - 8 bit identifier for SGTL5000
7:0	REVID	RO	0x00	SGTL5000 Revision ID 0xHH - revision number for SGTL5000.

7.0.0.2. CHIP_DIG_POWER 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									ADC_POWERUP	DAC_POWERUP	DAP_POWERUP	RSVD		I2S_OUT_POWERUP	I2S_IN_POWERUP

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6	ADC_POWERUP	RW	0x0	Enable/disable the ADC block, both digital and analog 0x0 = Disable 0x1 = Enable
5	DAC_POWERUP	RW	0x0	Enable/disable the DAC block, both analog and digital 0x0 = Disable 0x1 = Enable
4	DAP_POWERUP	RW	0x0	Enable/disable the DAP block 0x0 = Disable 0x1 = Enable
3:2	RSVD	RW	0x0	Reserved
1	I2S_OUT_POWERUP	RW	0x0	Enable/disable the I2S data output 0x0 = Disable 0x1 = Enable
0	I2S_IN_POWERUP	RW	0x0	Enable/disable the I2S data input 0x0 = Disable 0x1 = Enable

7.0.0.3. CHIP_CLK_CTRL 0x0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										RATE_MODE		SYS_FS		MCLK_FREQ	

BITS	FIELD	RW	RESET	DEFINITION
15:6	RSVD	RO	0x0	Reserved
5:4	RATE_MODE	RW	0x0	Sets the sample rate mode. MCLK_FREQ is still specified relative to the rate in SYS_FS 0x0 = SYS_FS specifies the rate 0x1 = Rate is 1/2 of the SYS_FS rate 0x2 = Rate is 1/4 of the SYS_FS rate 0x3 = Rate is 1/6 of the SYS_FS rate
3:2	SYS_FS	RW	0x2	Sets the internal system sample rate 0x0 = 32 kHz 0x1 = 44.1 kHz 0x2 = 48 kHz 0x3 = 96 kHz
1:0	MCLK_FREQ	RW	0x0	Identifies incoming SYS_MCLK frequency and if the PLL should be used 0x0 = 256*Fs 0x1 = 384*Fs 0x2 = 512*Fs 0x3 = Use PLL The 0x3 (Use PLL) setting must be used if the SYS_MCLK is not a standard multiple of Fs (256, 384 or 512). This setting can also be used if SYS_MCLK is a standard multiple of Fs. Before this field is set to 0x3 (Use PLL), the PLL must be powered up by setting CHIP_ANA_POWER->PLL_POWERUP and CHIP_ANA_POWER->VCOAMP_POWERUP. Also, the PLL dividers must be calculated based on the external MCLK rate and CHIP_PLL_CTRL register must be set (see CHIP_PLL_CTRL register description details on how to calculate the divisors).

7.0.0.4. CHIP_I2S_CTRL 0x0006

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							SCLKFREQ	MS	SCLK_INV	DLEN		I2S_MODE		LRALIGN	LRPOL

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
8	SCLKFREQ	RW	0x0	Sets frequency of I2S_SCLK when in master mode (MS=1). When in slave mode (MS=0), this field must be set appropriately to match SCLK input rate. 0x0 = 64Fs 0x1 = 32Fs - Not supported for RJ mode (I2S_MODE = 1)
7	MS	RW	0x0	Configures master or slave of I2S_LRCLK and I2S_SCLK. 0x0 = Slave: I2S_LRCLK and I2S_SCLK are inputs 0x1 = Master: I2S_LRCLK and I2S_SCLK are outputs NOTE: If the PLL is used (CHIP_CLK_CTRL->MCLK_FREQ==0x3), the SGTL5000 must be a master of the I2S port (MS==1)
6	SCLK_INV	RW	0x0	Sets the edge that data (input and output) is clocked in on for I2S_SCLK 0x0 = data is valid on rising edge of I2S_SCLK 0x1 = data is valid on falling edge of I2S_SCLK
5:4	DLEN	RW	0x1	I2S data length 0x0 = 32 bits (only valid when SCLKFREQ=0), not valid for Right Justified Mode 0x1 = 24 bits (only valid when SCLKFREQ=0) 0x2 = 20 bits 0x3 = 16 bits
3:2	I2S_MODE	RW	0x0	Sets the mode for the I2S port 0x0 = I2S mode or Left Justified (Use LRALIGN to select) 0x1 = Right Justified Mode 0x2 = PCM Format A/B 0x3 = RESERVED
1	LRALIGN	RW	0x0	I2S_LRCLK Alignment to data word. Not used for Right Justified mode 0x0 = Data word starts 1 I2S_SCLK delay after I2S_LRCLK transition (I2S format, PCM format A) 0x1 = Data word starts after I2S_LRCLK transition (left justified format, PCM format B)
0	LRPOL	RW	0x0	I2S_LRCLK Polarity when data is presented. 0x0 = I2S_LRCLK = 0 - Left, 1 - Right 1x0 = I2S_LRCLK = 0 - Right, 1 - Left The left subframe should be presented first regardless of the setting of LRPOL.

7.0.0.5. CHIP_SSS_CTRL 0x000A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	DAP_MIX_LRSWAP	DAP_LRSWAP	DAC_LRSWAP	RSVD	I2S_LRSWAP	DAP_MIX_SELECT		DAP_SELECT		DAC_SELECT		RSVD		I2S_SELECT	

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RW	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
14	DAP_MIX_LRSWAP	RW	0x0	DAP Mixer Input Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the DAP MIXER Input will be swapped.
13	DAP_LRSWAP	RW	0x0	DAP Input Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the DAP Input will be swapped
12	DAC_LRSWAP	RW	0x0	DAC Input Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the DAC will be swapped
11	RSVD	RW	0x0	Reserved
10	I2S_LRSWAP	RW	0x0	I2S_DOUT Swap 0x0 = Normal Operation 0x1 = Left and Right channels for the I2S_DOUT will be swapped
9:8	DAP_MIX_SELECT	RW	0x0	Select data source for DAP mixer 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = Reserved
7:6	DAP_SELECT	RW	0x0	Select data source for DAP 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = Reserved
5:4	DAC_SELECT	RW	0x1	Select data source for DAC 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = DAP
3:2	RSVD	RW	0x0	Reserved
1:0	I2S_SELECT	WO	0x0	Select data source for I2S_DOUT 0x0 = ADC 0x1 = I2S_IN 0x2 = Reserved 0x3 = DAP

7.0.0.6. CHIP_ADCDAC_CTRL 0x000E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		VOL_BUSY_DAC_RIGHT	VOL_BUSY_DAC_LEFT	RSVD		VOL_RAMP_EN	VOL_EXPO_RAMP	RSVD				DAC_MUTE_RIGHT	DAC_MUTE_LEFT	ADC_HPF_FREEZE	ADC_HPF_BYPASS

BITS	FIELD	RW	RESET	DEFINITION
15:14	RSVD	RO	0x0	Reserved
13	VOL_BUSY_DAC_RIGHT	RO	0x0	Volume Busy DAC Right 0x0 = Ready 0x1 = Busy - This indicates the channel has not reached its programmed volume/mute level
12	VOL_BUSY_DAC_LEFT	RO	0x0	Volume Busy DAC Left 0x0 = Ready 0x1 = Busy - This indicates the channel has not reached its programmed volume/mute level
11:10	RSVD	RO	0x0	Reserved
9	VOL_RAMP_EN	RW	0x1	Volume Ramp Enable 0x0 = Disables volume ramp. New volume settings will take immediate effect without a ramp 0x1 = Enables volume ramp This field affects DAC_VOL. The volume ramp effects both volume settings and mute. When set to 1 a soft mute is enabled.
8	VOL_EXPO_RAMP	RW	0x0	Exponential Volume Ramp Enable 0x0 = Linear ramp over top 4 volume octaves 0x1 = Exponential ramp over full volume range This bit only takes effect if *VOL_RAMP_EN* is 1.
7:4	RSVD	RW	0x0	Reserved
3	DAC_MUTE_RIGHT	RW	0x1	DAC Right Mute 0x0 = Unmute 0x1 = Muted If *VOL_RAMP_EN* = 1, this is a soft mute.
2	DAC_MUTE_LEFT	RW	0x1	DAC Left Mute 0x0 = Unmute 0x1 = Muted If *VOL_RAMP_EN* = 1, this is a soft mute.
1	ADC_HPF_FREEZE	RW	0x0	ADC High Pass Filter Freeze 0x0 = Normal operation 0x1 = Freeze the ADC high-pass filter offset register. The offset will continue to be subtracted from the ADC data stream.
0	ADC_HPF_BYPASS	RW	0x0	ADC High Pass Filter Bypass 0x0 = Normal operation 0x1 = Bypassed and offset not updated

7.0.0.7. CHIP_DAC_VOL 0x0010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC_VOL_RIGHT								DAC_VOL_LEFT							

BITS	FIELD	RW	RESET	DEFINITION
15:8	DAC_VOL_RIGHT	RW	0x3C	DAC Right Channel Volume Set the Right channel DAC volume with 0.5017 dB steps from 0 to -90 dB 0x3B and less = Reserved 0x3C = 0 dB 0x3D = -0.5 dB 0xF0 = -90 dB 0xFC and greater = Muted If *VOL_RAMP_EN* = 1, there will be an automatic ramp to the new volume setting.
7:0	DAC_VOL_LEFT	RW	0x3C	DAC Left Channel Volume Set the Left channel DAC volume with 0.5017 dB steps from 0 to -90 dB 0x3B and less = Reserved 0x3C = 0 dB 0x3D = -0.5 dB 0xF0 = -90 dB 0xFC and greater = Muted If *VOL_RAMP_EN* = 1, there will be an automatic ramp to the new volume setting.

7.0.0.8. CHIP_PAD_STRENGTH 0x0014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						I2S_LRCLK		I2S_SCLK		I2S_DOUT		CTRL_DATA		CTRL_CLK	

BITS	FIELD	RW	RESET	DEFINITION
15:14	RSVD	RW	0x0	Reserved
9:8	I2S_LRCLK	RW	0x1	I2S LRCLK Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8V 2.5V 3.3V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA
7:6	I2S_SCLK	RW	0x1	I2S SCLK Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8V 2.5V 3.3V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA

BITS	FIELD	RW	RESET	DEFINITION
5:4	I2S_DOUT	RW	0x1	I2C DOUT Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8V 2.5V 3.3V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA
3:2	CTRL_DATA	RW	0x3	I2C DATA Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8V 2.5V 3.3V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA
1:0	CTRL_CLK	RW	0x3	I2C CLK Pad Drive Strength Sets drive strength for output pads per the table below. VDDIO 1.8V 2.5V 3.3V 0x0 = Disable 0x1 = 1.66 mA 2.87 mA 4.02 mA 0x2 = 3.33 mA 5.74 mA 8.03 mA 0x3 = 4.99 mA 8.61 mA 12.05 mA

7.0.0.9. CHIP_ANA_ADC_CTRL 0x0020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							ADC_VOL_M6DB	ADC_VOL_RIGHT				ADC_VOL_LEFT			

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	ADC_VOL_M6DB	RW	0x0	ADC Volume Range Reduction This bit shifts both right and left analog ADC volume range down by 6dB. 0x0 = No change in ADC range 0x1 = ADC range reduced by 6dB
7:4	ADC_VOL_RIGHT	RW	0x0	ADC Right Channel Volume Right channel analog ADC volume control in 1.5dB steps. 0x0 = 0dB 0x1 = +1.5dB ... 0xF = +22.5dB This range will be -6dB to +16.5dB if *ADC_VOL_M6DB* is set to 1.

BITS	FIELD	RW	RESET	DEFINITION
3:0	ADC_VOL_LEFT	RW	0x0	ADC Left Channel Volume Left channel analog ADC volume control in 1.5dB steps. 0x0 = 0dB 0x1 = +1.5dB ... 0xF = +22.5dB This range will be -6dB to +16.5dB if *ADC_VOL_M6DB* is set to 1.

7.0.0.10. CHIP_ANA_HP_CTRL 0x0022

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	HP_VOL_RIGHT							RSVD	HP_VOL_LEFT						

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14:8	HP_VOL_RIGHT	RW	0x18	Headphone Right Channel Volume Right channel headphone volume control with 0.5dB steps. 0x00 = +12dB 0x01 = +11.5dB 0x18 = 0dB ... 0x7F = -51.5dB
7	RSVD	RO	0x0	Reserved
6:0	HP_VOL_LEFT	RW	0x18	Headphone Left Channel Volume Left channel headphone volume control with 0.5dB steps. 0x00 = +12dB 0x01 = +11.5dB 0x18 = 0dB ... 0x7F = -51.5dB

7.0.0.11. CHIP_ANA_CTRL 0x0024

This is an analog control register that includes mutes, input selects, and zero-cross-detectors for the ADC, headphone, and lineout.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							MUTE_LO	RSVD	SELECT_HP	EN_ZCD_HP	MUTE_HP	RSVD	SELECT_ADC	EN_ZCD_ADC	MUTE_ADC

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	MUTE_LO	RW	0x1	LineOut Mute 0x0 = Unmute 0x1 = Mute
7	RSVD	RO	0x0	Reserved
6	SELECT_HP	RW	0x0	Select the headphone input. 0x0 = DAC 0x1 = Line in
5	EN_ZCD_HP	RW	0x0	Enable the headphone zero cross detector (ZCD) 0x0 = HP ZCD disabled 0x1 = HP ZCD enabled
4	MUTE_HP	RW	0x1	Mute the headphone outputs 0x0 = Unmute 0x1 = Mute
3	RSVD	RO	0x0	Reserved
2	SELECT_ADC	RW	0x0	Select the ADC input. 0x0 = Microphone 0x1 = Line in
1	EN_ZCD_ADC	RW	0x0	Enable the ADC analog zero cross detector (ZCD) 0x0 = ADC ZCD disabled 0x1 = ADC ZCD enabled
0	MUTE_ADC	RW	0x1	Mute the ADC analog volume 0x0 = Unmute 0x1 = Mute

7.0.0.12. CHIP_LINREG_CTRL 0x0026

This register controls the VDDD linear regulator and the charge pump.

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="9">RSVD</td><td>VDDC_MAN_ASSN</td><td>VDDC_ASSN_OVRD</td><td>RSVD</td><td colspan="4">D_PROGRAMMING</td></tr>
</table>

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6	VDDC_MAN_ASSN	RW	0x0	Determines chargepump source when VDDC_ASSN_OVRD is set. 0x0 = VDDA 0x1 = VDDIO

BITS	FIELD	RW	RESET	DEFINITION
5	VDDC_ASSN_OVRD	RW	0x0	Chargepump Source Assignment Override 0x0 = Chargepump source is automatically assigned based on higher of VDDA and VDDIO 0x1 = the source of chargepump is manually assigned by VDDC_MAN_ASSN If VDDIO and VDDA are both the same and greater than 3.1V, VDDC_ASSN_OVRD and VDDC_MAN_ASSN should be used to manually assign VDDIO as the source for chargepump.
4	RSVD	RW	0x0	Reserved
3:0	D_PROGRAMMING	RW	0x0	Sets the VDDD lin. regulator output voltage in 50mV steps. Must clear pwd_linreg_d bit to enable this lin reg. 0x0=1.60 0xF=0.85

7.0.0.13. CHIP_REF_CTRL 0x0028

This register controls the bandgap reference bias voltage and currents.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							VAG_VAL					BIAS_CTRL			SMALL_POP

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8:4	VAG_VAL	RW	0x0	Analog Ground Voltage Control These bits control the analog ground voltage in 25mV steps. This should usually be set to VDDA/2 or lower for best performance (maximum output swing at minimum THD). This VAG reference is also used for the DAC and ADC voltage reference. So changing this voltage scales the output swing of the DAC and the output signal of the ADC. 0x00 = 0.800V 0x1F = 1.575V
3:1	BIAS_CTRL	RW	0x0	Bias control These bits adjust the bias currents for all of the analog blocks. By lowering the bias current a lower quiescent power is achieved. It should be noted that this mode can affect perfomance by 3-4dB. 0x0 = Nominal 0x1-0x3=+12.5% 0x4=-12.5% 0x5=-25% 0x6=-37.5% 0x7=-50%
0	SMALL_POP	RW	0x0	VAG Ramp Control Setting this bit slows down the VAG ramp from ~200ms to ~400ms to reduce the startup pop, but increases the turn on/off time. 0x0 = Normal VAG ramp 0x1 = Slowdown VAG ramp

7.0.0.14. CHIP_MIC_CTRL 0x002A

This register controls the microphone gain and the internal microphone biasing circuitry.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						BIAS_RESISTOR		RSVD	BIAS_VOLT			RSVD		GAIN	

BITS	FIELD	RW	RESET	DEFINITION
15:10	RSVD	RO	0x0	Reserved
9:8	BIAS_RESISTOR	RW	0x0	MIC Bias Output Impedance Adjustment Controls an adjustable output impedance for the microphone bias. If this is set to zero the micbias block is powered off and the output is highZ. 0x0 = Powered off 0x1 = 2Kohm 0x2 = 4Kohm 0x3 = 8Kohm
7	RSVD	RO	0x0	Reserved
6:4	BIAS_VOLT	RW	0x0	MIC Bias Voltage Adjustment Controls an adjustable bias voltage for the microphone bias amp in 250mV steps. This bias voltage setting should be no more than VDDA-200mV for adequate power supply rejection. 0x0 = 1.25V ... 0x7 = 3.00V
3:2	RSVD	RO	0x0	Reserved
1:0	GAIN	RW	0x0	MIC Amplifier Gain Sets the microphone amplifier gain. At 0dB setting the THD can be slightly higher than other paths- typically around ~65dB. At other gain settings the THD will be better. 0x0 = 0dB 0x1 = +20dB 0x2 = +30dB 0x3 = +40dB

7.0.0.15. CHIP_LINE_OUT_CTRL 0x002C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				OUT_CURRENT				RSVD		LO_VAGCNTRL					

BITS	FIELD	RW	RESET	DEFINITION
15:12	RSVD	RO	0x0	Reserved
11:8	OUT_CURRENT	RW	0x0	Controls the output bias current for the lineout amplifiers. The nominal recommended setting for a 10kohm load with 1nF load cap is 0x3. There are only 5 valid settings. 0x0=0.18mA, 0x1=0.27mA, 0x3=0.36mA, 0x7=0.45mA, 0xF=0.54mA
7:6	RSVD	RO	0x0	Reserved
5:0	LO_VAGCNTRL	RW	0x0	Lineout Amplifier Analog Ground Voltage Controls the analog ground voltage for the lineout amplifiers in 25mV steps. This should usually be set to VDDIO/2. 0x00 = 0.800V ... 0x1F = 1.575V ... 0x23 = 1.675 0x24-0x3F are invalid

7.0.0.16. CHIP_LINE_OUT_VOL 0x002E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			LO_VOL_RIGHT					RSVD			LO_VOL_LEFT				

BITS	FIELD	RW	RESET	DEFINITION
15:13	RSVD	RO	0x0	Reserved
12:8	LO_VOL_RIGHT	RW	0x4	Lineout Right Channel Volume Controls the right channel lineout volume in 0.5dB steps. Higher codes have more attenuation. See programming information for Left channel.
7:5	RSVD	RO	0x0	Reserved
4:0	LO_VOL_LEFT	RW	0x4	Lineout Left Channel Output Level The LO_VOL_LEFT is used to normalize the output level of the left line output to full scale based on the values used to set LINE_OUT_CTRL -> LO_VAGCNTRL and CHIP_REF_CTRL -> VAG_VAL. In general this field should be set to: $40*\log((\mathrm{VAG\_VAL})/(\mathrm{LO\_VAGCNTRL})) + 15$ Table 18 shows suggested values based on typical VDDIO and VDDA voltages. After setting to the nominal voltage, this field can be used to adjust the output level in +/-.5dB increments by using values higher or lower than the nominal setting.

Table 18. Line Out Output Level Values

VDDA	VAG_VAL	VDDIO	LO_VAGCNTRL	LO_VOL_*
1.8V	0.9	3.3V	1.55	0x06
1.8V	0.9	1.8V	0.9	0x0F
3.3V	1.55	1.8V	0.9	0x19
3.3V	1.55	3.3V	1.55	0x0F

7.0.0.17. CHIP_ANA_POWER 0x0030

This register contains all of the powerdown controls for the analog blocks. The only other powerdown controls are BIAS_RESISTOR in the MIC_CTRL register and the EN_ZCD control bits in ANA_CTRL.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	DAC_MONO	LINREG_SIMPLE_POWERUP	STARTUP_POWERUP	VDDC_CHRGPMP_POWERUP	PLL_POWERUP	LINREG_D_POWERUP	VCOAMP_POWERUP	VAG_POWERUP	ADC_MONO	REFTOP_POWERUP	HEADPHONE_POWERUP	DAC_POWERUP	CAPLESS_HEADPHONE_POWERUP	ADC_POWERUP	LINEOUT_POWERUP

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RW	0x0	Reserved
14	DAC_MONO	RW	0x1	While DAC_POWERUP is set, this allows the DAC to be put into left only mono operation for power savings. 0x0 = Mono (left only) 0x1 = Stereo
13	LINREG_SIMPLE _POWERUP	RW	0x1	Power up the simple (low power) digital supply regulator. After reset, this bit can be cleared IF VDDD is driven externally OR the primary digital linreg is enabled with LINREG_D_POWERUP 0x0 = Power down 0x1 = Power up
12	STARTUP_POWE RUP	RW	0x1	Power up the circuitry needed during the power up ramp and reset. After reset this bit can be cleared if VDDD is coming from an external source. 0x0 = Power down 0x1 = Power up

BITS	FIELD	RW	RESET	DEFINITION
11	VDDC_CHRGPMP_POWERUP	RW	0x0	Power up the VDDC chargepump block. If neither VDDA or VDDIO is 3V or larger this bit should be cleared before analog blocks are powered up. 0x0 = Power down 0x1 = Power up Note that for charge pump to function, either the PLL must be powered on and programmed correctly (refer to CHIP_CLK_CTRL->MCLK_FREQ description) or the internal oscillator (set CLK_TOP_CTRL->ENABLE_INT_OSC) must be enabled
10	PLL_POWERUP	RW	0x0	PLL Power Up 0x0 = Power down 0x1 = Power up When cleared, the PLL will be turned off. This must be set before CHIP_CLK_CTRL -> MCLK_FREQ is programmed to 0x3. The CHIP_PLL_CTRL register must be configured correctly before setting this bit.
9	LINREG_D_POWERUP	RW	0x0	Power up the primary VDDD linear regulator. 0x0 = Power down 0x1 = Power up
8	VCOAMP_POWERUP	RW	0x0	Power up the PLL VCO amplifier. 0x0 = Power down 0x1 = Power up
7	VAG_POWERUP	RW	0x0	Power up the VAG reference buffer. Setting this bit starts the power up ramp for the headphone and lineout. The headphone (and/or lineout) powerup should be set BEFORE clearing this bit. When this bit is cleared the powerdown ramp is started. The headphone (and/or lineout) powerup should stay set until the VAG is fully ramped down (200-400ms after clearing this bit). 0x0 = Power down 0x1 = Power up
6	ADC_MONO	RW	0x1	While ADC_POWERUP is set, this allows the ADC to be put into left only mono operation for power savings. This mode is useful when only using the microphone input. 0x0 = Mono (left only) 0x1 = Stereo
5	REFTOP_POWERUP	RW	0x1	Power up the reference bias currents 0x0 = Power down 0x1 = Power up This bit can be cleared when the part is is a sleep state to minimize analog power.
4	HEADPHONE_POWERUP	RW	0x0	Power up the headphone amplifiers 0x0 = Power down 0x1 = Power up
3	DAC_POWERUP	RW	0x0	Power up the DACs 0x0 = Power down 0x1 = Power up
2	CAPLESS_HEADPHONE_POWERUP	RW	0x0	Power up the capless headphone mode 0x0 = Power down 0x1 = Power up

BITS	FIELD	RW	RESET	DEFINITION
1	ADC_POWERUP	RW	0x0	Power up the ADCs 0x0 = Power down 0x1 = Power up
0	LINEOUT_POWERUP	RW	0x0	Power up the lineout amplifiers 0x0 = Power down 0x1 = Power up

7.0.0.18. CHIP_PLL_CTRL 0x0032

This register may only be changed after reset, and before PLL_POWERUP is set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_DIVISOR					FRAC_DIVISOR										

BITS	FIELD	RW	RESET	DEFINITION

15:11	INT_DIVISOR	RW	0xA	This is the integer portion of the PLL divisor. To determine the value of this field, use the following calculation:
				INT_DIVISOR = FLOOR(PLL_OUTPUT_FREQ/ INPUT_FREQ)
				
				PLL_OUTPUT_FREQ = 180.6336 MHz if System sample rate = 44.1 KHz
				else
				PLL_OUTPUT_FREQ = 196.608 MHz if System sample rate != 44.1 KHz
				
				INPUT_FREQ = Frequency of the external MCLK provided if CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x0
				else
				INPUT_FREQ = (Frequency of the external MCLK provided/2) If CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x1

10:0	FRAC_DIVISOR	RW	0x0	This is the fractional portion of the PLL divisor. To determine the value of this field, use the following calculation:
				FRAC_DIVISOR = ((PLL_OUTPUT_FREQ/INPUT_FREQ) - INT_DIVISOR)*2048
				
				PLL_OUTPUT_FREQ = 180.6336 MHz if System sample rate = 44.1 KHz
				else
				PLL_OUTPUT_FREQ = 196.608 MHz if System sample rate != 44.1 KHz
				
				INPUT_FREQ = Frequency of the external MCLK provided if CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x0
				else
				INPUT_FREQ = (Frequency of the external MCLK provided/2) If CHIP_CLK_TOP_CTRL->INPUT_FREQ_DIV2 = 0x1

7.0.0.19. CHIP_CLK_TOP_CTRL 0x0034

Miscellaneous controls for the clock block.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				ENABLE_INT_OSC	RSVD							INPUT_FREQ_DIV2	RSVD		

BITS	FIELD	RW	RESET	DEFINITION
15:12	RESERVED	RO	0x0	Reserved
11	ENABLE_INT_OSC	RW	0x0	Setting this bit enables an internal oscillator to be used for the zero cross detectors, the short detect recovery, and the charge pump. This will allow the I2S clock to be shut off while still operating an analog signal path. This bit can be kept on when the I2S clock is enabled, but the I2S clock is more accurate so it is preferred to clear this bit when I2S is present.
10:4	RSVD	RW	0x0	Reserved
3	INPUT_FREQ_DIV2	RW	0x0	SYS_MCLK divider before PLL input 0x0 = pass through 0x1 = SYS_MCLK is divided by 2 before entering PLL This must be set when the input clock is above 17Mhz. This has no effect when the PLL is powered down.
2:0	RSVD	RW	0x0	Reserved

7.0.0.20. CHIP_ANA_STATUS 0x0036

Status bits for analog blocks.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						LRSHORT_STS	CSHORT_STS	RSVD			PLL_IS_LOCKED	RSVD			

BITS	FIELD	RW	RESET	DEFINITION
15:10	RSVD	RO	0x0	Reserved
9	LRSHORT_STS	RO	0x0	This bit is high whenever a short is detected on the left or right channel headphone drivers. 0x0 = Normal 0x1 = Short detected
8	CSHORT_STS	RO	0x0	This bit is high whenever a short is detected on the capless headphone common/center channel driver. 0x0 = Normal 0x1 = Short detected
7:5	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
4	PLL_IS_LOCKED	RO	0x0	This bit goes high after the PLL is locked. 0x0 = PLL is not locked 0x1 = PLL is locked
3:0	RSVD	RO	0x0	Reserved

7.0.0.21. CHIP_ANA_TEST1 0x0038

These register controls are intended only for debug.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HP_IALL_ADJ		HP_I1_ADJ		HP_ANTIPOP			HP_CLASSAB	HP_HOLD_GND_CENTER	HP_HOLD_GND	VAG_DOUB_CURRENT	VAG_CLASSA	TM_ADCIN_TOHP	TM_HPCOMMON	TM_SELECT_MIC	TESTMODE

BITS	FIELD	RW	RESET	DEFINITION
15:14	HP_IALL_ADJ	RW	0x0	These bits control the overall bias current of the headphone amplifier (all stages including first and output stage). 0x0=nominal, 0x1=-50%, 0x2=+50%, 0x3=-40%
13:12	HP_I1_ADJ	RW	0x0	These bits control the bias current for the first stage of the headphone amplifier. 0x0=nominal, 0x1=-50%, 0x2=+100%, 0x3=+50%
11:9	HP_ANTIPOP	RW	0x0	These bits control the headphone output current in classA mode and also the pulldown strength while powering off. These bits will normally not be needed.
8	HP_CLASSAB	RW	0x1	This defaults high. When this bit is high the headphone is in classAB mode. ClassA mode would normally not be used.
7	HP_HOLD_GND_CENTER	RW	0x1	This defaults high. When this bit is high and the capless headphone center channel is powered off the output will be tied to ground. This is the preferred mode of operation for best antipop performance.
6	HP_HOLD_GND	RW	0x1	This defaults high. When this bit is high and the headphone is powered off the output will be tied to ground. This is the preferred mode of operation for best antipop performance.
5	VAG_DOUB_CURRENT	RW	0x0	Double the VAG output current when in classA mode.
4	VAG_CLASSA	RW	0x0	Turn off the classAB output current for the VAG buffer. The classA current is limited so this may cause clipping in some modes.
3	TM_ADCIN_TOHP	RW	0x0	Put ADCmux output onto the headphone output pin. Must remove headphone load and any external headphone compensation for this mode.
2	TM_HPCOMMON	RW	0x0	Enable headphone common to be used in ADCmux for testing
1	TM_SELECT_MIC	RW	0x0	Enable the mic-adc-dac-HP path

BITS	FIELD	RW	RESET	DEFINITION
0	TESTMODE	RW	0x0	Enable the analog testmode paths

7.0.0.22. CHIP_ANA_TEST2 0x003A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	LINEOUT_TO_VDDA	SPARE	MONOMODE_DAC	VCO_TUNE_AGAIN	LO_PASS_MASTERVAG	INVERT_DAC_SAMPLE_CLOCK	INVERT_DAC_DATA_TIMING	DAC_EXTEND_RTZ	DAC_DOUBLE_I	DAC_DIS_RTZ	DAC_CLASSA	INVERT_ADC_SAMPLE_CLOCK	INVERT_ADC_DATA_TIMING	ADC_LESSI	ADC_DITHEROFF

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14	LINEOUT_TO_VDDA	RW	0x0	Changes the lineout amplifier power supply from VDDIO to VDDA. Typically lineout should be on the higher power supply. This bit is useful when VDDA is ~3.3V and VDDIO is ~1.8V.
13	SPARE	RW	0x0	Spare registers to analog.
12	MONOMODE_DAC	RW	0x0	Copy the left channel DAC data to the right channel. This allows both left and right to play from MONO dac data.
11	VCO_TUNE_AGAIN	RW	0x0	When toggled high then low forces the PLL VCO to retune the number of inverters in the ring oscillator loop.
10	LO_PASS_MASTERVAG	RW	0x0	Tie the main analog VAG to the lineout VAG. This can improve SNR for the lineout when both are the same voltage.
9	INVERT_DAC_SAMPLE_CLOCK	RW	0x0	Change the clock edge used for the DAC output sampling.
8	INVERT_DAC_DATA_TIMING	RW	0x0	Change the clock edge used for the digital to analog DAC data crossing.
7	DAC_EXTEND_RTZ	RW	0x0	Extend the return-to-zero time for the DAC.
6	DAC_DOUBLE_I	RW	0x0	Double the output current of the DAC amplifier when it is in classA mode.
5	DAC_DIS_RTZ	RW	0x0	Turn off the return-to-zero in the DAC. In mode cases this will hurt the SNDR of the DAC.
4	DAC_CLASSA	RW	0x0	Turn off the classAB mode in the DAC amplifier. This mode should normally not be used. The output current will not be high enough to support a full scale signal in this mode.
3	INVERT_ADC_SAMPLE_CLOCK	RW	0x0	Change the clock edge used for the ADC sampling.
2	INVERT_ADC_DATA_TIMING	RW	0x0	Change the clock edge used for the analog to digital ADC data crossing
1	ADC_LESSI	RW	0x0	Drops ADC bias currents by 20%

BITS	FIELD	RW	RESET	DEFINITION
0	ADC_DITHEROFF	RW	0x0	Turns off the ADC dithering.

7.0.0.23. CHIP_SHORT_CTRL 0x003C

This register contains controls for the headphone short detectors.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	LVLADJR			RSVD	LVLADJL			RSVD	LVLADJC			MODE_LR		MODE_CM	

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14:12	LVLADJR	RW	0x0	These bits adjust the sensitivity of the right channel headphone short detector in 25mA steps.This trip point can vary by ~30% over process so leave plenty of guardband to avoid false trips. This short detect trip point is also effected by the bias current adjustments made by CHIP_REF_CTRL -> BIAS_CTRL and by CHIP_ANA_TEST1 -> HP_IALL_ADJ. 0x3=25mA 0x2=50mA 0x1=75mA 0x0=100mA 0x4=125mA 0x5=150mA 0x6=175mA 0x7=200mA
11	RSVD	RO	0x0	Reserved
10:8	LVLADJL	RW	0x0	These bits adjust the sensitivity of the left channel headphone short detector in 25mA steps.This trip point can vary by ~30% over process so leave plenty of guardband to avoid false trips. This short detect trip point is also effected by the bias current adjustments made by CHIP_REF_CTRL -> BIAS_CTRL and by CHIP_ANA_TEST1 -> HP_IALL_ADJ. 0x3=25mA 0x2=50mA 0x1=75mA 0x0=100mA 0x4=125mA 0x5=150mA 0x6=175mA 0x7=200mA
7	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
6:4	LVLADJC	RW	0x0	These bits adjust the sensitivity of the capless headphone center channel short detector in 50mA steps. This trip point can vary by ~30% over process so leave plenty of guardband to avoid false trips. This short detect trip point is also effected by the bias current adjustments CHIP_REF_CTRL -> BIAS_CTRL and by CHIP_ANA_TEST1 -> HP_IALL_ADJ. 0x3=50mA 0x2=100mA 0x1=150mA 0x0=200mA 0x4=250mA 0x5=300mA 0x6=350mA 0x7=400mA
3:2	MODE_LR	RW	0x0	These bits control the behavior of the short detector for the capless headphone central channel driver. This mode should be set prior to powering up the headphone amplifier. When a short is detected the amplifier output switches to classA mode interally to avoid excessive currents. 0x0 = Disable short detector, reset short detect latch, software view non-latched short signal 0x1 = Enable short detector and reset the latch at timeout (every ~50ms) 0x2 = This mode is not used/invalid 0x3 = Enable short detector with only manual reset (have to return to 0x0 to reset the latch)
1:0	MODE_CM	RW	0x0	These bits control the behavior of the short detector for the capless headphone central channel driver. This mode should be set prior to powering up the headphone amplifier. When a short is detected the amplifier output switches to classA mode interally to avoid excessive currents. 0x0 = Disable short detector, reset short detect latch, software view non-latched short signal 0x1 = Enable short detector and reset the latch at timeout (every ~50ms) 0x2 = Enable short detector and auto reset when output voltage rises (preferred mode) 0x3 = Enable short detector with only manual reset (have to return to 0x0 to reset the latch)

7.0.0.24. DAP_CONTROL 0x0100

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											MIX_EN	RSVD			DAP_EN

BITS	FIELD	RW	RESET	DEFINITION
15:5	RSVD	RO	0x0	Reserved
4	MIX_EN	RW	0x0	Enable/Disable the DAP mixer path 0x0 = Disable 0x1 = Enable When enabled, DAP_EN must also be enabled to use the mixer.
3:1	RSVD	RO	0x0	Reserved
0	DAP_EN	RW	0x0	Enable/Disable digital audio processing (DAP) 0x0 = Disable. When disabled, no audio will pass-through. 0x1 = Enable. When enabled, audio can pass-through DAP even if none of the DAP functions are enabled.

7.0.0.25. DAP_PEQ 0x0102

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													EN		

BITS	FIELD	RW	RESET	DEFINITION
15:3	RSVD	RO	0x0	Reserved
2:0	EN	RW	0x0	Set to Enable the PEQ filters 0x0 = Disabled 0x1 = 1 Filter Enabled 0x2 = 2 Filters Enabled ..... 0x7 = Cascaded 7 Filters DAP_AUDIO_EQ->EN bit must be set to 1 in order to enable the PEQ

7.0.0.26. DAP_BASS_ENHANCE 0x0104

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							BYPASS_HPF	RSVD	CUTOFF			RSVD			EN

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
8	BYPASS_HPF	RW	0x0	Bypass high pass filter 0x0 = Enable high pass filter 0x1 = Bypass high pass filter
7	RSVD	RO	0x0	Reserved
6:4	CUTOFF	RW	0x4	Set cut-off frequency 0x0 = 80 Hz 0x1 = 100 Hz 0x2 = 125 Hz 0x3 = 150 Hz 0x4 = 175 Hz 0x5 = 200 Hz 0x6 = 225 Hz
3:1	RSVD	RO	0x0	Reserved
0	EN	RW	0x0	Enable/Disable Bass Enhance 0x0 = Disable 0x1 = Enable

7.0.0.27. DAP_BASS_ENHANCE_CTRL 0x0106

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		LR_LEVEL						RSVD	BASS_LEVEL						

BITS	FIELD	RW	RESET	DEFINITION
15:14	RSVD	RO	0x0	Reserved
13:8	LR_LEVEL	RW	0x5	Left/Right Mix Level Control 0x00= +6dB for Main Channel ...... 0x3F= Least L/R Channel Level
7	RSVD	RO	0x0	
6:0	BASS_LEVEL	RW	0x1f	Bass Harmonic Level Control 0x00= Most Harmonic Boost ...... 0x7F=Least Harmonic Boost

7.0.0.28. DAP_AUDIO_EQ 0x0108

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														EN	

BITS	FIELD	RW	RESET	DEFINITION
15:2	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
1:0	EN	RW	0x0	Selects between PEQ/GEQ/Tone Control and Enables it. 0x0 = Disabled. 0x1 = Enable PEQ. NOTE: DAP_PEQ->EN bit must also be set to the desired number of filters (bands) in order for the PEQ to be enabled. 0x2 = Enable Tone Control 0x3 = Enable 5 Band GEQ

7.0.0.29. DAP_SGTL_SURROUND 0x010A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									WIDTH_CONTROL			RSVD		SELECT	

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:4	WIDTH_CONTROL	RW	0x4	SGTL Surround Width Control - The width control changes the perceived width of the sound field. 0x0 = Least Width ...... 0x7 = Most Width
3:2	RSVD	RO	0x0	Reserved
1:0	SELECT	RW	0x0	SGTL Surround Selection 0x0 = Disabled 0x1 = Disabled 0x2 = Mono input Enable 0x3 = Stereo input Enable

7.0.0.30. DAP_FILTER_COEF_ACCESS 0x010C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							WR	INDEX							

BITS	FIELD	RW	RESET	DEFINITION
15:9	RSVD	RO	0x0	Reserved
8	WR	WO	0x0	When set, the coefficients written in the ten coefficient data registers will be loaded into the filter specified by INDEX

BITS	FIELD	RW	RESET	DEFINITION
7:0	INDEX	RW	0x0	Specifies the index for each of the seven bands of the filter coefficient that needs to be written to. Each filter has 5 coefficients that need to be loaded into the 10 coefficient registers (MSB,LSB) before setting the index and WR bit. Steps to write coefficients: 1. Write the five 20-bit coefficient values to DAP_COEF_WR_XX_MSB and DAP_COEF_WR_XX_LSB registers (XX= B0,B1,B2,A1,A2) 2. Set INDEX of the coefficient from the table below. 3. Set the WR bit to load the coefficient. NOTE: Steps 2 and 3 can be performed with a single write to DAP_FILTER_COEF_ACCESS register. Coefficient address: Band 0 = 0x00 Band 1 = 0x01 Band 2 = 0x02 Band 3 = 0x03 Band 4 = 0x04 ... Band 7 = 0x06

7.0.0.31. DAP_COEF_WR_B0_MSB 0x010E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIT_19	BIT_18	BIT_17	BIT_16	BIT_15	BIT_14	BIT_13	BIT_12	BIT_11	BIT_10	BIT_9	BIT_8	BIT_7	BIT_6	BIT_5	BIT_4

BITS	FIELD	RW	RESET	DEFINITION
15	BIT_19	WO	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written
14	BIT_18	WO	0x0	
13	BIT_17	WO	0x0	
12	BIT_16	WO	0x0	
11	BIT_15	WO	0x0	
10	BIT_14	WO	0x0	
9	BIT_13	WO	0x0	
8	BIT_12	WO	0x0	
7	BIT_11	WO	0x0	
6	BIT_10	WO	0x0	
5	BIT_9	WO	0x0	
4	BIT_8	WO	0x0	
3	BIT_7	WO	0x0	
2	BIT_6	WO	0x0	
1	BIT_5	WO	0x0	

BITS	FIELD	RW	RESET	DEFINITION
0	BIT_4	WO	0x0	

7.0.0.32. DAP_COEF_WR_B0_LSB 0x0110

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												BIT_3	BIT_2	BIT_1	BIT_0

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	
3	BIT_3	WO	0x0	
2	BIT_2	WO	0x0	
1	BIT_1	WO	0x0	
0	BIT_0	WO	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

7.0.0.33. DAP_AUDIO_EQ_BASS_BAND0 0x0116

115Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets Tone Control Bass/GEQ Band0 0x5F = sets to 12dB 0x2F = sets to 0dB 0x00 = sets to -12dB Each LSB is 0.25dB. To convert dB to hex value, use: Hex Value = 4*dBValue + 47

7.0.0.34. DAP_AUDIO_EQ_BAND1 0x0118

330Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
6:0	VOLUME	RW	0x2F	Sets GEQ Band1 0x5F = sets to 12dB 0x2F = sets to 0dB 0x00 = sets to -12dB Each LSB is 0.25dB. To convert dB to hex value, use: Hex Value = 4*dBValue + 47

7.0.0.35. DAP_AUDIO_EQ_BAND2 0x011A

990Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets GEQ Band2 0x5F = sets to 12dB 0x2F = sets to 0dB 0x00 = sets to -12dB Each LSB is 0.25dB. To convert dB to hex value, use: Hex Value = 4*dBValue + 47

7.0.0.36. DAP_AUDIO_EQ_BAND3 0x011C

3000Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets GEQ Band3 0x5F = sets to 12dB 0x2F = sets to 0dB 0x00 = sets to -12dB Each LSB is 0.25dB. To convert dB to hex value, use: Hex Value = 4*dBValue + 47

7.0.0.37. DAP_AUDIO_EQ_TREBLE_BAND4 0x011E

9900Hz

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									VOLUME						

BITS	FIELD	RW	RESET	DEFINITION
15:7	RSVD	RO	0x0	Reserved
6:0	VOLUME	RW	0x2F	Sets Tone Control Treble/GEQ Band4 0x5F = sets to 12dB 0x2F = sets to 0dB 0x00 = sets to -12dB Each LSB is 0.25dB. To convert dB to hex value, use: Hex Value = 4*dBValue + 47

7.0.0.38. DAP_MAIN_CHAN 0x0120

Sets the main channel volume level.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOL															

BITS	FIELD	RW	RESET	DEFINITION
15:0	VOL	RW	0x8000	DAP Main Channel Volume 0xFFFF = 200% 0x8000 (default) = 100% 0x0000 = 0%

7.0.0.39. DAP_MIX_CHAN 0x0122

Sets the mix channel volume level.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOL															

BITS	FIELD	RW	RESET	DEFINITION
15:0	VOL	RW	0x0000	DAP Mix Channel Volume 0xFFFF = 200% 0x8000 = 100% 0x0000 (default) = 0%

7.0.0.40. DAP_AVC_CTRL 0x0124

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	MAX_GAIN		RSVD		LBI_RESPONSE		RSVD		HARD_LIMIT_EN	RSVD				EN

BITS	FIELD	RW	RESET	DEFINITION
15	RSVD	RO	0x0	Reserved
14	RSVD	RW	0x1	Reserved.

BITS	FIELD	RW	RESET	DEFINITION
13:12	MAX_GAIN	RW	0x1	Maximum gain that can be applied by the AVC in expander mode. 0x0 = 0dB gain 0x1 = 6dB of gain 0x2 = 12dB of gain
11:10	RSVD	RO	0x0	Reserved
9:8	LBI_RESPONSE	RW	0x1	Integrator Response 0x0 = 0mS LBI 0x1 = 25mS LBI 0x2 = 50mS LBI 0x3 = 100mS LBI
7:6	RSVD	RO	0x0	Reserved
5	HARD_LIMIT_EN	RW	0x0	Enable Hard Limiter Mode 0x0 = Hard limit disabled. AVC Compressor/Expander is enabled. 0x1 = Hard limit enabled. The signal is limited to the programmed threshold. (Signal saturates at the threshold)
4:1	RSVD	RO	0x0	Reserved
0	EN	RW	0x0	Enable/disable AVC 0x0 = Disable 0x1 = Enable

7.0.0.41. DAP_AVC_THRESHOLD 0x0126

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRESH															

BITS	FIELD	RW	RESET	DEFINITION
15:0	THRESH	RW	0x1473	AVC Threshold Value Threshold is programmable. Use the following formula to calculate hex value: Hex Value = ((10^(THRESHOLD_dB/20))*0.636)*2^15 Threshold can be set in the range of 0dB to -96 dB Example Values: 0x1473 = Set Threshold to -12dB 0x0A40 = Set Threshold to -18dB

7.0.0.42. DAP_AVC_ATTACK 0x0128

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				RATE											

BITS	FIELD	RW	RESET	DEFINITION
15:12	RSVD	RO	0x0	Reserved

BITS	FIELD	RW	RESET	DEFINITION
11:0	RATE	RW	0x28	AVC Attack Rate This is the rate at which the AVC will apply attenuation to the signal to bring it to the threshold level. AVC Attack Rate is programmable. To use a custom rate, use the formula below to convert from dB/S to hex value: Hex Value = (1 - (10^(-(Rate_dBs/(20*SYS_FS)))) * 2^19 where, SYS_FS is the system sample rate configured in CHIP_CLK_CTRL register. Example values: 0x28 = 32dB/s 0x10 = 8dB/s 0x05 = 4dB/s 0x03 = 2dB/s

7.0.0.43. DAP_AVC_DECAY 0x012A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				RATE											

BITS	FIELD	RW	RESET	DEFINITION
15:12	RSVD	RO	0x0	Reserved
11:0	RATE	RW	0x50	AVC Decay Rate This is the rate at which the AVC releases the attenuation previously applied to the signal during attack. AVC Decay Rate is programmable. To use a custom rate, use the formula below to convert from dB/S to hex value: Hex Value = (1 - (10^(-(Rate_dBs/(20*SYS_FS)))) * 2^23 where, SYS_FS is the system sample rate configured in CHIP_CLK_CTRL register. Example values: 0x284 = 32dB/s 0xA0 = 8dB/s 0x50 = 4dB/s 0x28 = 2dB/s

7.0.0.44. DAP_COEF_WR_B1_MSB 0x012C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

7.0.0.45. DAP_COEF_WR_B1_LSB 0x012E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

7.0.0.46. DAP_COEF_WR_B2_MSB 0x0130

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

7.0.0.47. DAP_COEF_WR_B2_LSB 0x0132

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

7.0.0.48. DAP_COEF_WR_A1_MSB 0x0134

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

7.0.0.49. DAP_COEF_WR_A1_LSB 0x0136

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.

7.0.0.50. DAP_COEF_WR_A2_MSB 0x0138

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB															

BITS	FIELD	RW	RESET	DEFINITION
15:0	MSB	RW	0x0	Most significant 16-bits of the 20-bit filter coefficient that needs to be written

7.0.0.51. DAP_COEF_WR_A2_LSB 0x013A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LSB			

BITS	FIELD	RW	RESET	DEFINITION
15:4	RSVD	RO	0x0	Reserved
3:0	LSB	RW	0x0	Least significant 4 bits of the 20-bit filter coefficient that needs to be written.