Application Report
SLAA404C–January 2009–Revised April 2010

Design and Configuration Guide for the TLV320AIC3204 and TLV320AIC3254 Audio Codecs

Jorge Arbona — *High-Performance Analog/Audio and Imaging Products*

ABSTRACT

This application report provides guidelines, application examples, and register programming sequence information as well as sample scripts to help the system designer and programmer with the design process and configuration of the TLV320AIC3204 and TLV320AIC3254 audio codecs.

Contents

List of Figures

PowerTune is a trademark of Texas Instruments.
SPI is a trademark of Motorola, Inc.
I^{2}C, I^{2}S are trademarks of NXP Semiconductors.
All other trademarks are the property of their respective owners.

Copyright © 2009–2010, Texas Instruments Incorporated

1 Introduction

The TLV320AIC3204 and TLV320AIC3254 are first in a new generation of audio codecs from Texas Instruments. These devices feature real-time filtering and the ability to trade off between performance and power consumption (PowerTune™), as well as dynamic range compression (DRC) and other features, and are intended for the portable audio market. Both codecs are pin-compatible with one another; the primary difference between the two units is that the TLV320AIC3254 features programmable miniDSPs. For simplicity, the abbreviation *AIC32x4* is used throughout this document to refer to both devices, unless explicitly noted otherwise.

The main components of an audio coder/decoder (codec) device are analog-to digital-converters (ADCs), digital-to-analog converters (DACs), and a data interface bus to transfer converted data between the codec and a microcontroller (MCU) or DSP. As system complexity increases and size decreases in portable applications, feature integration becomes an attractive option for designers. The AIC32x4 integrates processing capabilities that can reduce the overhead of an external DSP or simply act as a signal processor along with an MCU.

The AIC32x4 is programmed by writing to registers that can be accessed by using the I^2C™ or SPI™ communication protocols. The fact that this device has many pages with hundreds of registers may seem overwhelming at first, but in reality, many registers do not need to be configured for most typical audio applications. The purpose of this document is to guide the system designer through the process of selecting which registers must be configured as well as illustrating how the device should be connected to the rest of the system for general applications. The miniDSP function of the TLV320AIC3254 is not discussed in this document; this report is intended for processing block use. In order to keep this document as concise as possible, some important details of overall device operation may be omitted—therefore, the designer is strongly advised to read the respective product data sheet (see Section 4).

2 System-Level Considerations

Each system may have several constraints with regard to supply voltages, clock frequencies, the number of analog inputs and outputs, serial interfaces, and sampling rate, for example. This section provides information to help the system designer with these constraints and reviews other useful information related to signal processing. Application examples are also included.

Copyright © 2009–2010, Texas Instruments Incorporated

2.1 Hardware Pinout

A closer look at the AIC32x4 pinout shows that the hardware pins are classified into four different groups based on their function: power, digital, ADC channel and DAC channel pins.

The AIC32x4 features single supply operation as well as other supply configurations. The hardware connections for these pins (marked in red as Figure 1 shows) depend on the specific configuration that is used. Refer to Section 2.4 for more details.

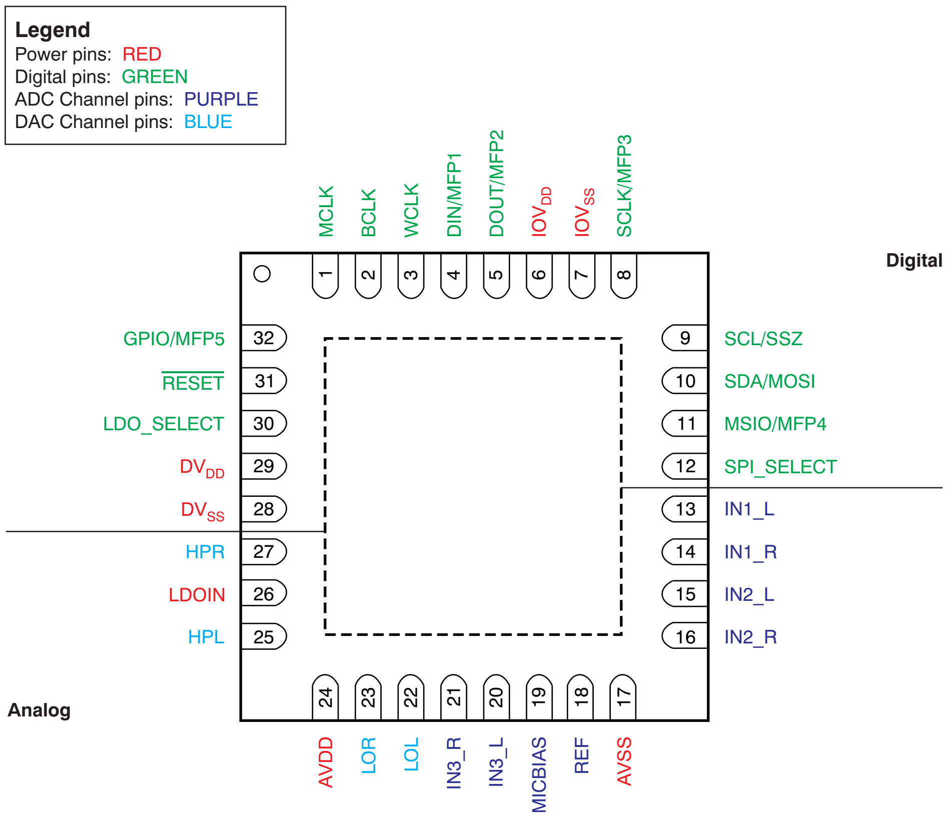


Figure 1. AIC32x4 Hardware Pinout

To achieve the best performance from the AIC32x4, care must be taken in printed circuit board (PCB) design and layout to avoid coupling external noise into the device. In particular, to avoid coupling high-frequency digital signals to the analog signals, the digital and analog sections should be separated. As shown in Figure 1, the pinout is organized to aid such a board layout. Use a separate analog ground plane that is shorted at one point, close to the AIC32x4 device itself.

Copyright © 2009–2010, Texas Instruments Incorporated

2.2 Clocks

The AIC32x4 features a flexible clocking scheme that can be used to accomplish the following:

- Derive the clocks required to operate the internal delta-sigma modulators and processing blocks;
- Generate the audio interface clocks; and
- Output a clock for an external device through multipurpose pins.

This section focuses on the clocks needed to operate the converters and processing blocks. (The miniDSP clocking scheme for the TLV320AIC3254 is not discussed in this document; for more details, refer to the product data sheet.) Figure 2 depicts the clock distribution tree of the AIC32x4 codec.

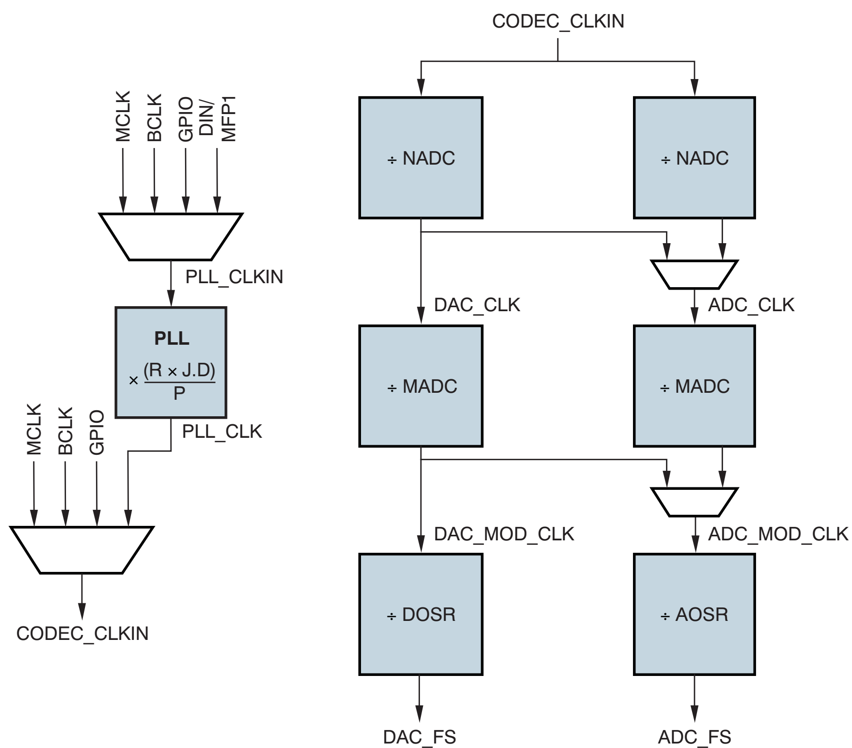


Figure 2. Clock Distribution Tree

A master clock can be provided directly to the CODEC_CLKIN node via the MCLK, BCLK, or GPIO pins; or, alternatively, use the internal PLL to provide the appropriate frequency. To minimize power consumption, the ADC_MOD_CLK as well as the ADC_CLK nodes can be fed by the DAC_MOD_CLK and DAC_CLK nodes, respectively, through the use of internal multiplexers. The path of these multiplexers can be switched by powering the NADC and/or MADC dividers off or on. Note that even if the MADC and NADC dividers are powered down, the respective divider value must be set equal to its corresponding DAC divider when the ADCs are used.

A good strategy for selecting clock divider values is to start from the bottom up, especially if a standard master clock frequency can be provided by the system. Table 1 provides a step-by-step process for proper clock divider selection. Note that the order specified in the table is not the same order that should be followed when programming the corresponding registers.

The PLL section of the respective product data sheet gives a very thorough explanation as well as related constraints and example configurations for various PLL clock inputs.

Copyright © 2009–2010, Texas Instruments Incorporated

Table 1. Clock Divider Selection Process

Step	ADC Channel	DAC Channel
1. Select AOSR and DOSR	**Equations:**	
	ADC_MOD_CLK = AOSR × ADC_FS	DAC_MOD_CLK = DOSR × DAC_FS
	Constraints:	
	For Filter A: AOSR can be 128 or 64 For Filter B: AOSR should be 64 For Filter C: AOSR should be 32 ADC_MOD_CLK ≤ 6.758 MHz	For Filter A: DOSR must be a multiple of 8 For Filter B: DOSR must be a multiple of 4 For Filter C: DOSR must be a multiple of 2 ADC_MOD_CLK ≤ 6.758 MHz (4.2 MHz for Class D operation)
	Comments: Filter A is typically used for sampling frequencies less than or equal to 48 kHz, while Filter B and C should be used for 96 kHz and 192 kHz, respectively. For some low-power modes, Filter B can be used for lower frequencies. Refer to the PowerTune™ section of the respective data sheet for more details on AOSR and DOSR selection.	
2. Select MADC and MDAC	**Equations:**	
	ADC_CLK = MADC × ADC_MOD_CLK	DAC_CLK = MDAC × DAC_MOD_CLK
	Constraints:	
	(MADC × AOSR) / 32 ≥ RC_{PRB_Rx} For DVDD less than 1.65 V: ADC_CLK ≤ 25 MHz For DVDD greater than 1.65 V: ADC_CLK ≤ 55.296 MHz	(MDAC × DOSR) / 32 ≥ RC_{PRB_Py} For DVDD less than 1.65 V: DAC_CLK ≤ 25 MHz For DVDD greater than 1.65 V: DAC_CLK ≤ 55.296 MHz
	Comments: The AIC32x4 has various processing blocks (called *PRB_Rx* and *PRB_Py* for record and playback, respectively) that provide access to several signal processing features such as multiple biquad filters, DRC, 3D, tone synthesizer, etc. Each processing block has a resource class (RC) that relates directly to signal processing capability and power consumption. The ADC and DAC sections of the data sheet provide processing block tables that specify which features are available in each, as well as other useful information such as resource class requirements.	
3. Select NADC and NDAC	**Equations:**	
	CODEC_CLKIN = NADC × ADC_CLK = NDAC × DAC_CLK **Note:** CODEC_CLKIN can be fed by MCLK, BCLK, and GPIO pins, or by PLL_CLK node.	
	Constraints:	
	For DVDD less than 1.65 V: CODEC_CLK ≤ 50 MHz For DVDD greater than 1.65 V: CODEC_CLK ≤ 137 MHz, NADC even, NDAC even CODEC_CLK ≤ 112 MHz, NADC odd, NDAC even CODEC_CLK ≤ 110 MHz, NADC even, NDAC odd CODEC_CLK ≤ 110 MHz, NADC odd, NDAC odd	
	Comments: At this point, the clock frequency at the ADC_CLK and DAC_CLK is known and may differ between each other in cases where the sampling rates are different for the ADC and the DAC, or in cases where different oversampling rates are desired (for example, an 8-kHz sampling rate for both ADC and DAC). If ADC_CLK and DAC_CLK differ, NDAC and NADC must be chosen such that both are equal. An external master clock can be fed directly to the CODEC_CLKIN node via the MCLK, BCLK, or GPIO pins without using the internal PLL. For this case, the maximum CODEC_CLKIN frequency is 50 MHz and its minimum is 512 kHz. Alternatively, a clock to the CODEC_CLKIN node can be provided by using the internal PLL (note that other restrictions apply).	

Copyright © 2009–2010, Texas Instruments Incorporated

Table 1. Clock Divider Selection Process (continued)

Step	ADC Channel	DAC Channel
4. Select PLL values (optional)	**Comments:** The PLL is best suited for the following cases: • MADC / AOSR or MDAC / DOSR combination does not satisfy the minimum resource class requirement for a specific processing block and a higher frequency clock is needed. • MADC / NADC or MDAC / NDAC integer values do not yield the desired sampling frequency from a specified master clock. For additional details and constraints related to the PLL, refer to the PLL section of the respective product data sheet.	

2.3 Audio Interface

The AIC32x4 supports four audio interface modes: I²S™, DSP, Left-Justified, and Right-Justified. The DSP mode is commonly used for time division multiplexing (TDM) applications where more than two audio channels are transferred between cascaded codecs and an applications processor on a single 4-wire bus.

A typical audio interface bus consists of four signals: the word clock, bit clock, data in (DAC data) and data out (ADC data). The AIC32x4 has two audio buses, where the primary bus has its signals fixed to the WCLK, BCLK, DIN, and DOUT pins and the secondary bus and ADC word clock can be routed to multifunction pins. The ADC word clock (ADC_WCLK) is suitable for cases where the ADC and DAC sampling rates differ. The audio bus signals can either be supplied by an external processor or generated by the AIC32x4.

Table 2 shows all the registers related to the audio interface, as well as a description for each. Typical system configurations do not require many changes to these registers. For example, no register programming related to the audio interface is needed if a host processor provides the I²S clocks to BCLK and WCLK (AIC32x4 as slave), with a word length of 16 bits. To set BCLK and WCLK as outputs, the BCLK divider must be configured (bits D1–D0 of Page 0 / Register 29 and Page 0 / Register 30) and the direction must be set accordingly (bits D3–D2 of Page 0 / Register 27).

Table 2. AIC23x4 Audio Interface-Related Registers

Label	Page	Register(s)	Bit(s)	Description
Audio Interface Mode	0	27	D7-D6	Sets the audio interface mode for both primary and secondary interfaces. I²S (default), DSP, Left-Justified, and Right-Justified modes are supported. I²S is the default mode.
Audio Data Word Length	0	27	D5-D4	Sets the audio bit resolution to 16 (default), 20, 24, or 32 bits.
BCLK Direction	0	27	D3	Sets the BCLK pin as input (default) or output.
WCLK Direction	0	27	D2	Sets the WCLK pin as input (default) or output.
Tristate DOUT during unused time slots	0	27	D0	Sets the DOUT pin as high impedance during unused time slots.
Data Offset	0	28	D7-D0	Offsets the data by *n* amount of bit clock cycles with respect to the default value. Typically used to assign time slots in time division multiplexing (TDM) schemes. For the DSP audio interface mode, a data offset of '0' aligns with the rising edge of the word clock.

Copyright © 2009–2010, Texas Instruments Incorporated

Table 2. AIC23x4 Audio Interface-Related Registers (continued)

Label	Page	Register(s)	Bit(s)	Description
Audio Bus Loopback	0	29	D5	Connects the audio bus data in to audio bus data out, bypassing the audio converters. Typically used to diagnose the host processor audio bus. It is disabled by default.
Digital Loopback	0	29	D4	Connects the ADC output to the DAC input. Data fed into the data in pin are ignored. It is disabled by default.
Bit Clock Polarity	0	29	D3	Inverts the bit clock with respect to the default value of a particular audio interface mode.
BCLK and WCLK Power Control	0	29	D2	Powers BCLK and WCLK buffers even when the ADC or DAC are powered down.
Bit Clock Divider Source	0	29	D1-D0	Selects the BDIV_CLKIN clock source when configured as an output.
Bit Clock N Divider Power	0	30	D7	Powers bit clock N divider.
Bit Clock N Divider Value	0	30	D6-D0	Sets N divider value.
Secondary Interface Pin Assignment	0	31	D6-D0	Assigns pins for the secondary bit clock, word clock, data in, as well as the ADC word clock.
Interface Block Signal Selection	0	32	D3-D0	Assigns bit clock, ADC word clock, DAC word clock and data in signals to the audio serial interface.
Interface Output Sources	0	33	D7-D0	Selects output source for both primary and secondary bit clock, word clock and data out signals.
Multi-function Pin Configuration	0	52, 53, 54, 55, and 56	N/A	Assigns the secondary audio interface to GPIO, DOUT, DIN, MISO, and SCLK pins and ADC word clock to GPIO, MISO, or SCLK pins.

Copyright © 2009–2010, Texas Instruments Incorporated

2.4 Processing Blocks

The AIC32x4 has 18 ADC channel pre-defined processing blocks and 25 DAC channel pre-defined processing blocks. These processing blocks provide access to several features such as multiple biquad sections, 3D, DRC, and others. The ADC and DAC sections of the product data sheet provide processing block tables that specify which features are available in each, as well as other useful information such as resource class requirements. These sections also review important details related to resource class requirements.

With this codec, it is possible to change filter coefficients on the fly by using the device adaptive filtering mode. Two buffers, called Buffer A and Buffer B, provide the control interface and processing block access to the filter coefficients. These buffers are available for both the ADC and DAC channel processing blocks.

For applications where a specific fixed frequency response is desired for the DAC, adaptive filtering is not required. In this case, Buffer B is not needed. Follow this simplified procedure for such a case.

Step 1. Write filter coefficients to DAC Buffer A (starting at page 44).

Step 2. Power up DAC(s).

For applications where filter coefficients are changed on the fly, such as bass-boost and treble-boost, adaptive filtering must be used; both buffers are required. Follow this simplified procedure for such a case.

Step 1. Enable Adaptive Filtering.

Step 2. Write filter coefficients to DAC Buffer A and DAC Buffer B (exact copy). This step is not necessary if using default coefficients (all-pass).

Step 3. Power up DAC(s). At this moment, audio can start playing.

Step 4. To modify the frequency response, write new filter coefficients to the Buffer A **address** (starting at page 44).

Step 5. Switch buffers by writing a '1' to Page 44 / Register 1 / Bit D0.

Step 6. Rewrite the exact same coefficients to the Buffer A **address** (starting at page 44). This step ensures that both buffers are synchronized.

Refer to Appendix B for example scripts related to filtering. Also, refer to the *Adaptive Filtering* section of the respective product data sheet for more details on the buffer switching mechanism and coefficient memory mapping.

Copyright © 2009–2010, Texas Instruments Incorporated

2.5 Power Supplies/LDOs

Figure 3 illustrates a simplified block diagram of the power-supply scheme and the related register bits (shown as *pP_rR_bM-L*, where P, R, M, and L are page, register, MSB, and LSB, respectively). The AIC32x4 has four supply pins: AVDD, DVDD, IOVDD, and LDOin. AVDD and DVDD can be supplied externally or internally (using the internal LDOs). In either case, decoupling capacitors at each power pin are required to filter noise.

Power for both the headphone and the line output amplifier can be provided by either the internal AVDD node or by a supply connected to the LDOin pin, as shown in Figure 3.

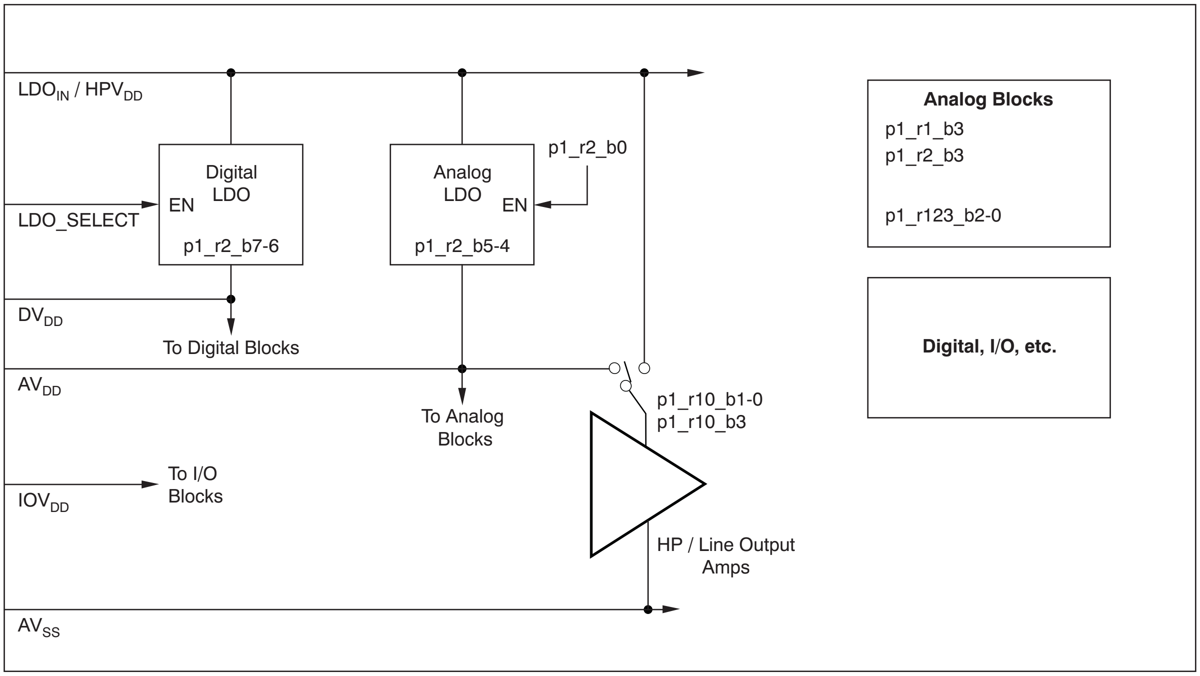


Figure 3. Power-Supply Scheme: Simplified Block Diagram

The internal low-dropout regulators (LDOs) can be used to provide power to the internal DVDD and AVDD nodes that feed the internal digital and analog blocks, respectively. A voltage supply (1.9 V to 3.6 V) must be connected to the LDOin pin in order to use either LDO. The respective output voltage can be set independently by programming Page 1/Register 2.

The Digital LDO can be enabled by connecting the LDO_SELECT pin to IOVDD through a pull-up resistor.

The Analog LDO can be enabled by setting bit D0 of Page 1/Register 2 to ‘1’.

Copyright © 2009–2010, Texas Instruments Incorporated

Figure 4 illustrates the typical power-supply circuit connections. Circuit A shows the typical connections for single-rail operation using both analog and digital internal LDOs to generate AVDD and DVDD, respectively. Note that the LDO_SELECT pin is pulled to IOVDD for this configuration. For cases in which only a low-voltage supply is available (for example, 1.8 V) and lower power consumption is desired, power can be provided directly to the AVDD and DVDD pins, as shown in circuit B (LDOin supply is optional). LDO_SELECT is tied to IOVSS in this case.

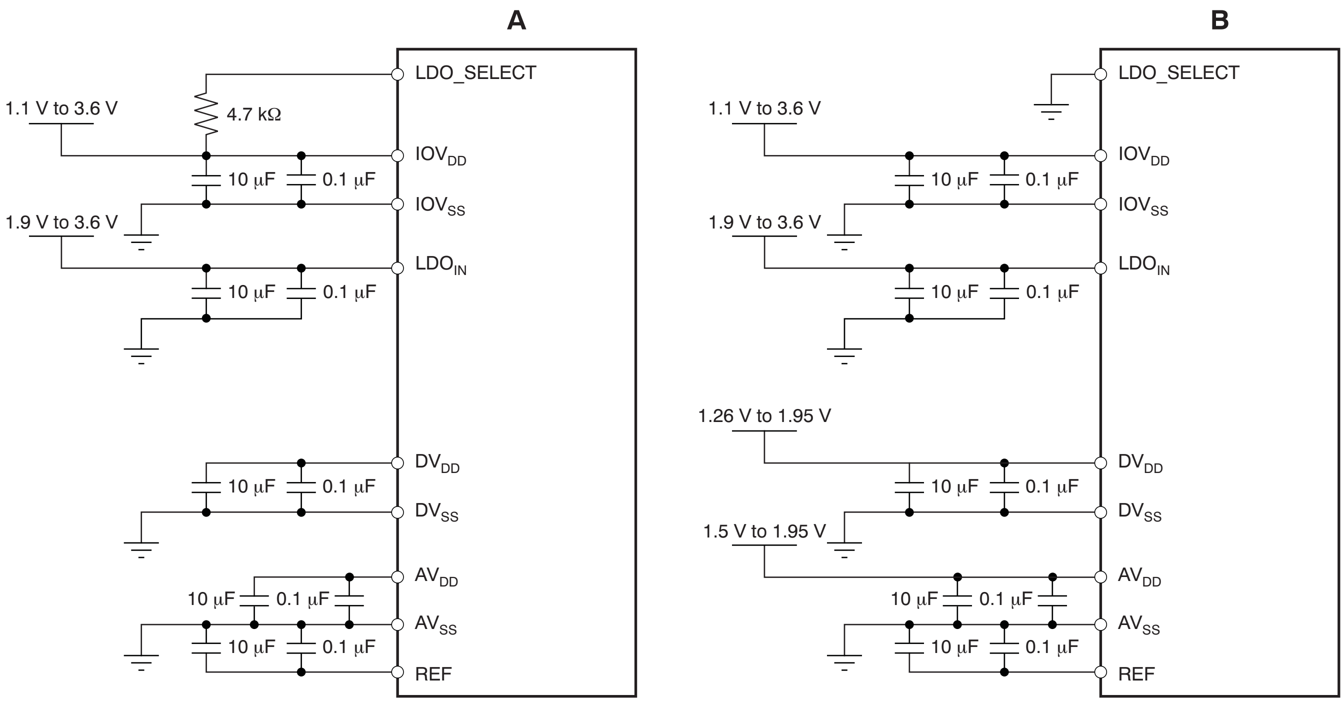


Figure 4. Typical Power-Supply Circuit Configurations

As mentioned previously, the LDOin pin can also be used as the power supply for both the headphone and line output drivers. This option allows the possibility to achieve higher output signal swings than the full-scale voltage defined for the AVDD supply. This feature is available when using the internal LDOs or providing supplies externally, as well.

It is recommended to provide the IOVDD supply before or at the same time as the other supply pins while holding the $\overline{\text{RESET}}$ pin low until all supplies stabilize. This procedure ensures that the codec boots in its lowest power consumption mode and with the correct logic level at the LDO_SELECT pin. Lastly, AVDD can be provided.

Copyright © 2009–2010, Texas Instruments Incorporated

2.6 PowerTune™

In some applications, it is desired to have the ability to trade off between power consumption and performance. PowerTune gives the AIC32x4 the ability to do such a task. Both the ADC and DAC channels have four PowerTune modes, called PTM_Rx and PTM_Py for record and playback, respectively, where x and y range independently from 1 to 4. PowerTune mode 4 provides the highest audio performance, while PowerTune mode 1 consumes less power.

As part of the PowerTune strategy, power consumption can be lowered even further by proper selection of processing blocks. Each processing block has a resource class (RC) that is proportional with power consumption; the lower the resource class, the less the power consumption. Supply voltages and configuration, common mode settings and sampling frequency also play a role in power consumption.

The ADC PowerTune mode can be selected by writing the register shown below:

	PTM_R1	PTM_R2	PTM_R3	PTM_R4
Pg 1, Reg 61, D(7:0)	0xFF	0xB6	0x64	0x00

The DAC PowerTune mode can be independently selected for each output channel by writing the registers shown below. In order to fully benefit the AIC32x4's high SNR performance, the bit resolution for PTM_P4 must be 20 bits or greater.

	PTM_R1	PTM_R2	PTM_R3	PTM_R4
Pg 1, Reg 3, D(4:2)	0x2	0x1	0x0	0x0
Pg 1, Reg 4, D(4:2)	0x2	0x1	0x0	0x0
Audio Data word length Pg 0, Reg 27, D(5:4)	16 bits 0x0	16 bits 0x0	16 bits 0x0	20 or more bits 0x1, 0x2, 0x3

PowerTune™ Example

Table 3 shows an example for stereo ADC at a 48-kHz sampling rate. An 'X' in a PowerTune mode column means that that particular mode is not available for that configuration. For this particular example, PTM_R1 with a common-mode setting of 0.75 V allows a maximum input level of –2 dB with respect to 375 mV_{RMS}. This value means that a maximum of –2 dB (or 0.298 mV_{RMS}) is allowed at the ADC inputs. The programmable input resistance for each input into the MicPGA must be chosen such that the maximum voltage out of the MicPGA and into the ADC does not exceed this voltage (see the ADC Channel section). The –2 dB difference can be then compensated by adjusting the ADC gain (Page 0 / Registers 83 and 84).

An estimated delta in power consumption (with respect to PRB_R7) is also shown for alternative processing blocks.

Table 3. ADC, Stereo, 48 kHz, Highest Performance, DV_{DD} = 1.8 V, AV_{DD} = 1.8 V(1)

	Device Common-Mode Setting = 0.75 V				Device Common-Mode Setting = 0.9 V				
	PTM_R1	PTM_R2	PTM_R3	PTM_R4	PTM_R1	PTM_R2	PTM_R3	PTM_R4	UNIT
0-dB full-scale	375	X	375	X	X	X	500	X	mV_{RMS}
Maximum allowed input level with regard to 0-dB full-scale	–2	X	0	X	X	X	0	X	dB full-scale
Effective SNR with regard to maximum allowed input level	86.0	X	88.1	X	X	X	90.4	X	dB
Power consumption	8.4	X	11.4	X	X	X	11.5	X	mW

(1) AOSR = 64, Processing Block = PRB_R7 (Decimation Filter B).

Table 4. Alternative Processing Blocks (ADC, Stereo)

Processing Block	Filter	Est. Power Change (mW)
PRB_R8	B	+0.7
PRB_R9	B	+0.7

Copyright © 2009–2010, Texas Instruments Incorporated

Table 4. Alternative Processing Blocks (ADC, Stereo) (continued)

Processing Block	Filter	Est. Power Change (mW)
PRB_R1	A	+2.0
PRB_R2	A	+3.4
PRB_R3	A	+3.4

Similarly, the output gain for DAC PowerTune modes PTM_P1 and PTM_P2 must be adjusted if an output voltage equal to 375 mV_{RMS} or 500 mV_{RMS} (for 0.75-V or 0.9-V common mode, respectively) is desired. As shown in Table 5, PTM_P1 is 14 dB below full-scale voltage. The headphone output gain (Page 1 / Registers 16 and 17) and the line output gain (Page 1 / Registers 18 and 19) can be adjusted to compensate for the –14dB difference.

Table 5. DAC, Mono, 48 kHz, Highest Performance, DV_{DD} = 1.8 V, AV_{DD} = 1.8 V(1)

		Device Common-Mode Setting = 0.75 V				Device Common-Mode Setting = 0.9 V				
		PTM_P1	PTM_P2	PTM_P3	PTM_P4	PTM_P1	PTM_P2	PTM_P3	PTM_P4	UNIT
0-dB full-scale		75	225	375	375	100	300	500	500	mV_{RMS}
HP out (32-Ω load)	Effective SNR with regard to 0-dB full-scale	88.1	96.1	98.7	99.5	90.4	96.3	99.4	100	dB
	Power consumption	5.8	6.2	6.5	6.5	5.8	6.2	6.5	6.5	mW
Line out	Effective SNR with regard to 0-dB full-scale	89.6	97.1	100.3	100.3	90.5	96.3	100	100	dB
	Power consumption	5.0	5.4	5.7	5.7	5.0	5.4	5.7	5.7	mW

(1) DOSR = 128, Processing Block = PRB_P13 (Interpolation Filter B).

Copyright © 2009–2010, Texas Instruments Incorporated

2.7 ADC Channel

Figure 5 illustrates a simplified block diagram of the ADC channel analog input internal routing. The AIC32x4 has six analog input pins that can be connected in different ways to achieve different purposes. Both single-ended and differential input configurations are supported.

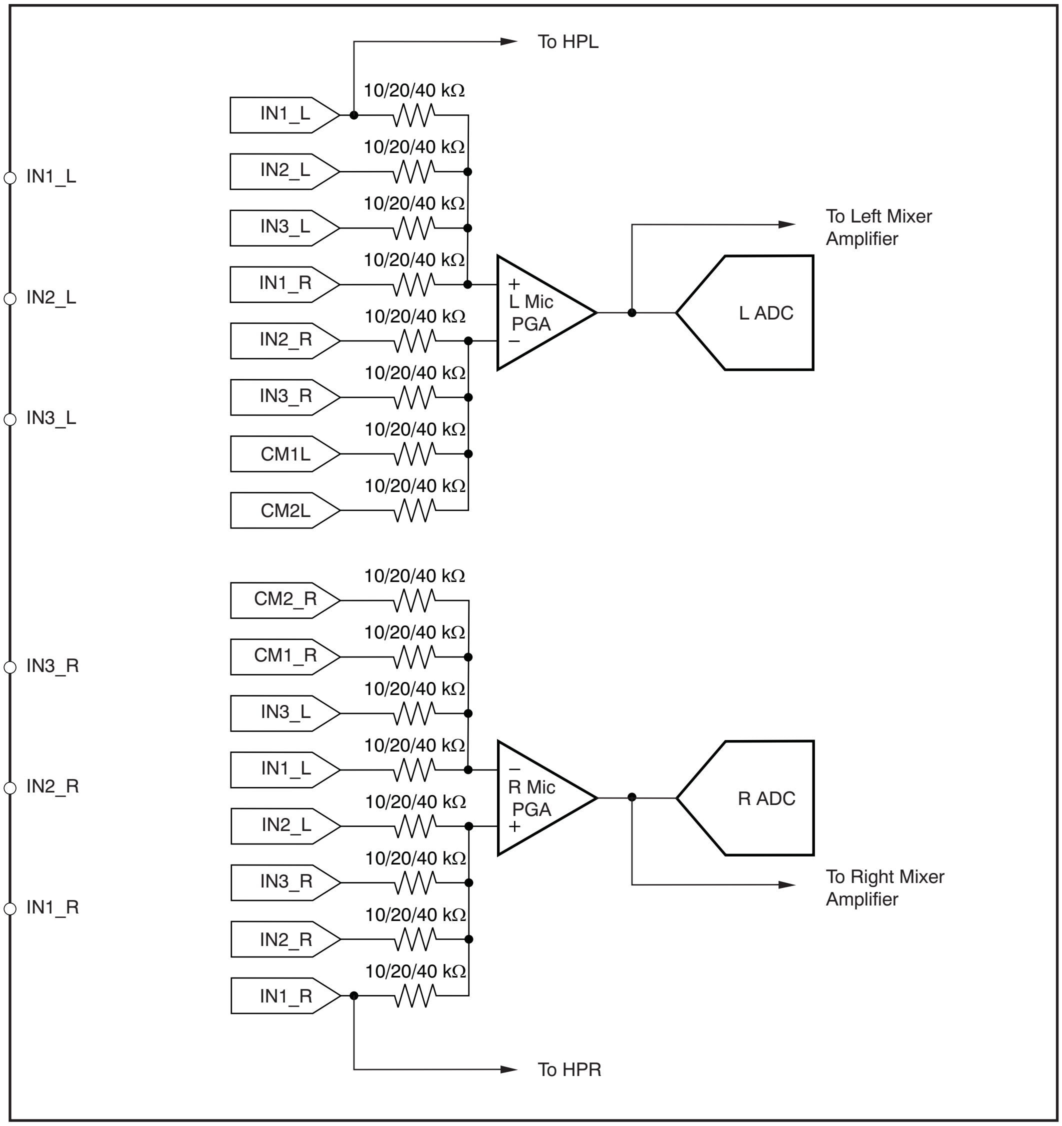


Figure 5. ADC Channel: Simplified Block Diagram

Copyright © 2009–2010, Texas Instruments Incorporated

Application Example

Suppose that a system requires three signals to be mixed into the left ADC, as shown in Figure 6. The three signals can be connected to the IN1_L, IN2_L, and IN3_L inputs and routed to the noninverting inputs of the left MicPGA amplifier. To allow more headroom, the input resistances can be set to 40 kΩ, which yield a 12-dB attenuation per single-ended channel. To balance the inverting and noninverting MicPGA inputs, CM1L can be set to 20 kΩ and CM2L to 40 kΩ.

As in the previous example, the MicPGA amplifiers require a common-mode (programmable voltage) connected to the inverting inputs for single-ended configurations. Because the connected input pins are biased to this voltage, ac-coupling capacitors are required between the input source and the pins. Unused inputs can be left floating or ac-coupled to ground (preferred).

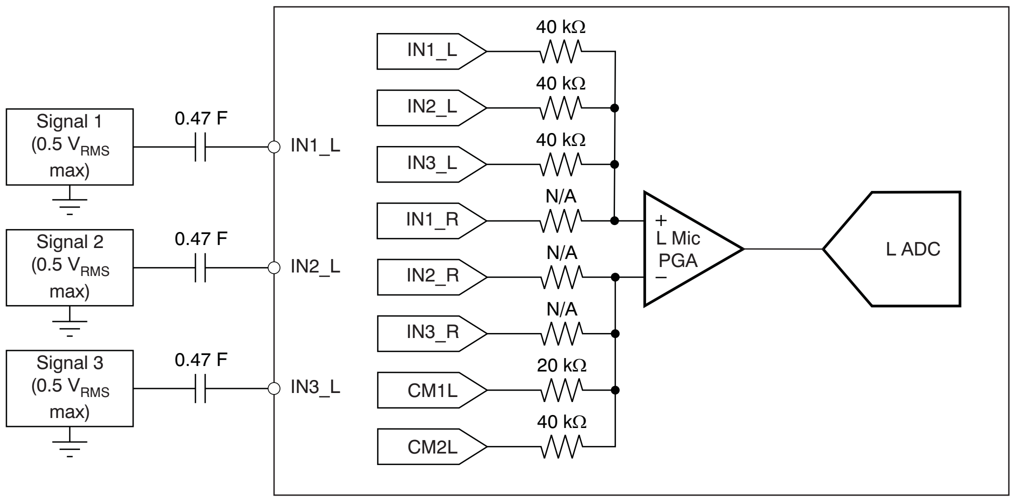


Figure 6. Application Example

Copyright © 2009–2010, Texas Instruments Incorporated

2.8 DAC Channel

The AIC32x4 features two high-power amplifier outputs and line outputs. The input for these amplifiers can be mixed from a variety of sources, such as the DAC channel outputs and analog inputs, as shown in Figure 7.

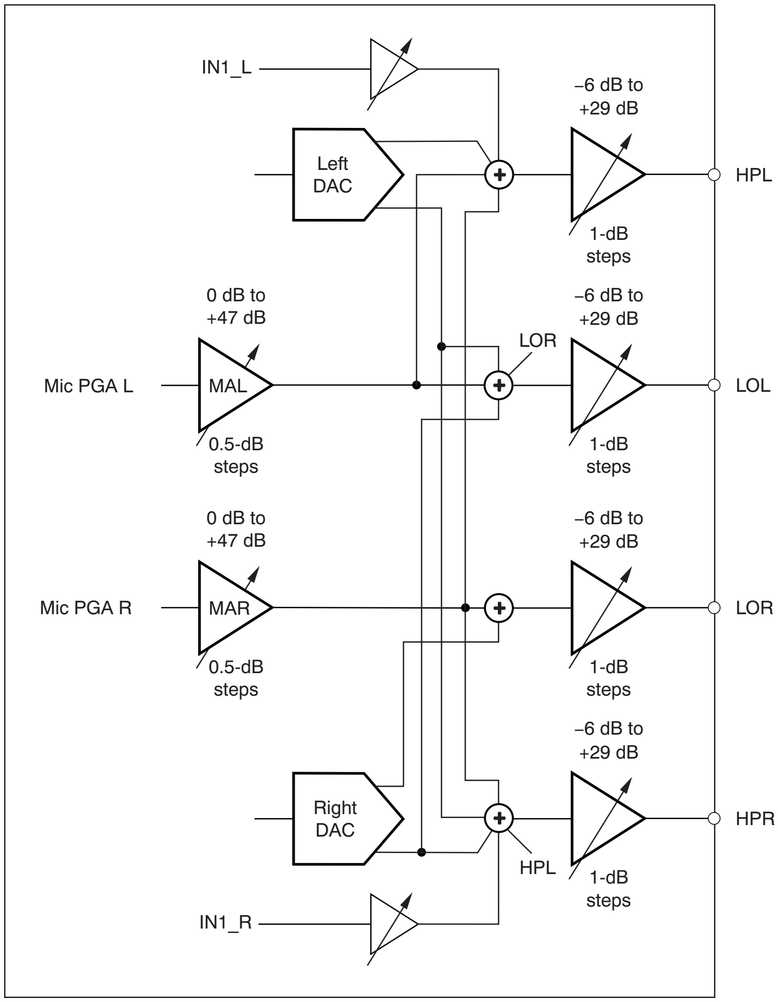


Figure 7. DAC Channel: Simplified Block Diagram

The mixer amplifiers (MAL and MAR) obtain the input signal from the MicPGA output (see previous section). Also, the IN1_L and IN1_R inputs can be mixed into the HPL and HPR outputs, respectively.

Both headphone and line outputs are referenced to a programmable common-mode voltage. A dc blocking capacitor between the output pin and the load is required for applications in which these outputs are driven in a single-ended fashion. The value of this capacitor depends on the desired cutoff frequency and the load. For portable audio applications, it is typical to use a 47-µF capacitor with a 32-Ω load for a 106-Hz corner frequency. For higher impedance loads, such as 20 kΩ, a smaller capacitor can be used.

By default, the output amplifiers are referenced to a 0.9-V common-mode voltage and have a full-scale voltage of 500 mV_{RMS}. For a higher signal swing (for example, 1 V_{RMS}), the common-mode voltage can be set to a maximum of 1.65 V and a higher voltage at the LDOin pin can be used as the amplifier supply. The full-scale voltage is increased by increasing the amplifier gain.

Copyright © 2009–2010, Texas Instruments Incorporated

3 Register Programming Sequence and Configuration

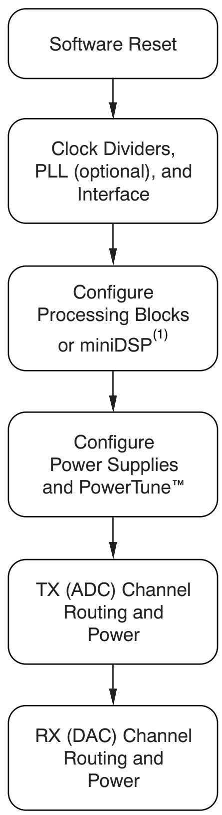
The TLV320AIC32x4 is configured by writing to 8-bit registers that can be accessed using either the I^{2}C or SPI communication protocols.

There are some functions in the device that should be executed or initialized in a certain order for proper operation. For example, the clock dividers should be initialized before powering up either the ADC or DAC. For additional information, refer to the respective product data sheet.

Before writing to any register, the device should be initialized by either a hardware or software reset. This initialization ensures that the codec boots up in its default mode. A hardware reset is accomplished by pulling the $\overline{\text{RESET}}$ pin low for at least 10 ns. A software reset can be done by writing a ‘1’ to bit ‘0’ of Page 0/Register 1.

After the AIC32x4 is initialized through a hardware or software reset, the internal memories are initialized to the respective default values. This initialization phase lasts for 1 ms. No register should be written during this period.

Clocks, processing blocks, power supplies, the ADC channel, and the DAC channel have been discussed thus far in this document. Figure 8 shows the recommended register programming flow for these after powering up the codec for the first time.



(1) TLV320AIC3254 only.

Figure 8. Register Programming Sequence

Appendix A through Appendix E contain script snippets that can be pieced together following the sequence previously described. Example 1 contains a sample script that programs the entire device to play stereo DAC data into headphones. A ‘w’ in these scripts refers to a register write; the first byte afterwards is the I^{2}C address; the second byte is the first register to write, and the following bytes are data. These scripts can be copied directly to be used with the EVM software.

Copyright © 2009–2010, Texas Instruments Incorporated

Example 1. Stereo DAC Playback to Headphones

```
################################################
# Software Reset
################################################
#
# Select Page 0
w 30 00 00
#
# Initialize the device through software reset
w 30 01 01
#
################################################

################################################
# Clock and Interface Settings
# ---------------------------------------------
# The codec receives: MCLK = 11.2896 MHz,
# BLCK = 2.8224 MHz, WCLK = 44.1 kHz
################################################
#
# Select Page 0
w 30 00 00
#
# NDAC = 1, MDAC = 2, dividers powered on
w 30 0b 81 82
#
################################################

################################################
# Configure Power Supplies
################################################
#
# Select Page 1
w 30 00 01
#
# Disable weak AVDD in presence of external
# AVDD supply
w 30 01 08
#
# Enable Master Analog Power Control
w 30 02 00
#
# Set the input power-up time to 3.1ms (for ADC)
w 30 47 32
#
# Set the REF charging time to 40ms
w 30 7b 01
#
################################################

################################################
# Configure DAC Channel
################################################
#
# Select Page 1
w 30 00 01
#
# De-pop: 5 time constants, 6k resistance
w 30 14 25
#
# Route LDAC/RDAC to HPL/HPR
w 30 0c 08 08
#
# Power up HPL/HPR
w 30 09 30
#
# Unmute HPL/HPR driver, 0dB Gain
w 30 10 00 00
#
```


Copyright © 2009–2010, Texas Instruments Incorporated

Example 1. Stereo DAC Playback to Headphones (continued)

```
#
# Select Page 0
w 30 00 00
#
# DAC => 0dB
w 30 41 00 00
#
# Power up LDAC/RDAC
w 30 3f d6
#
# Unmute LDAC/RDAC
w 30 40 00
#
################################################
```

4 References

1. TLV320AIC3204, Ultra Low-Power Stereo Audio Codec with PowerTune™ Technology (SLOS602)
2. TLV320AIC3254, Ultra Low-Power Stereo Audio Codec with miniDSP and PowerTune™ Technology(SLAS549)

Copyright © 2009–2010, Texas Instruments Incorporated

Appendix A Clocks and PLL Scripts

A.1 Clock Configuration Script without Using the PLL

The following script fragment configures the codec without use of the PLL. The AOSR and DOSR registers are not written because the default value of 128 is used. This script is only valid for processing blocks with a resource class less than or equal to 8 because MDAC and MADC are equal to 2. In order to use processing blocks with a resource class higher than 8, the PLL must be used to allow higher MADC and MDAC values.

The MADC divider is powered off; therefore, the ADC_MOD_CLK node is fed by DAC_MOD_CLK.

```
##################################################

# Clock and Interface Settings
# ---------------------------------------------
# The codec receives: MCLK = 11.2896 MHz,
# BLCK = 2.8224 MHz, WCLK = 44.1 kHz
###############################################
#
# Select Page 0
w 30 00 00
#
# NDAC = 1, MDAC = 2, dividers powered on
w 30 0b 81 82
#
# NADC = 1, MADC = 2, dividers powered off
w 30 12 01 02
#
###############################################
```

By default, BCLK and WCLK are inputs. These pins can be configured as outputs by writing to Page 0/Registers 27, 29, and 30. The last two commands in the script fragment below (highlighted in blue) program the BCLK frequency and set the pins as outputs.

```
###############################################
# Clock and Interface Settings
# ---------------------------------------------
# The codec receives: MCLK = 11.2896 MHz
# and generates: BLCK = 2.8224 MHz,
# WCLK = 44.1 kHz
###############################################
#
# Select Page 0
w 30 00 00
#
# NDAC = 1, MDAC = 2, dividers powered on w 30 0b 81 82
#
# NADC = 1, MADC = 2, dividers powered off
w 30 12 01 02
#
# BCLK frequency is generated from DAC_CLK
# and N = 4
w 30 1D 00 84
#
# Set BCLK and WCLK as outputs
w 30 1B 0C
#
###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

A.2 Clock Configuration Script Using the PLL

For cases in which a processing block with a higher resource class is desired, the PLL must be used to satisfy the M and OSR constraint. The following script fragment programs and enables the PLL, and sets the appropriate clock divider values based on the clock conditions described in the code header. This PLL and divider configuration works with any processing block that supports an OSR of 128.

```
# Clock and Interface Settings
# ---------------------------------------------
# The codec receives: MCLK = 11.2896 MHz,
# BLCK = 2.8224 MHz, WCLK = 44.1 kHz
###############################################
#
# Select Page 0
w 30 00 00
#
# PLL_clkin = MCLK, codec_clkin = PLL_CLK,
# PLL on, P=1, R=1, J=8
w 30 04 03 91 08
#
# NDAC = 2, MDAC = 8, dividers powered on
w 30 0b 82 88
#
# NADC = 2, MADC = 8, dividers powered off
w 30 12 02 08
#
###############################################
```

If an 8-kHz sampling rate is desired, DOSR can be set to 768 to push the out-of-band noise of the DAC modulator as far as possible from the audible frequency range. M and N values are different for the ADC and DAC; thus, the ADC frequency dividers must be turned on.

```
# Clock and Interface Settings
# ---------------------------------------------
# The codec receives: MCLK = 12.288 MHz,
# BLCK = 512 kHz, WCLK = 8 kHz
###############################################
#
# Select Page 0
w 30 00 00
#
# PLL_clkin = MCLK, codec_clkin = PLL_CLK,
# PLL on, P=1, R=1, J=8
w 30 04 03 91 08
#
# NDAC = 2, MDAC = 8, dividers powered on
w 30 0b 82 88
#
# DOSR = 768
w 30 0D 03 00
#
# NADC = 8, MADC = 12, dividers powered on
w 30 12 88 8C
# ###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

Appendix B Processing Blocks Scripts

B.1 Writing Filter Coefficients

The script fragment below implements a first-order, high-pass Butterworth filter with a corner frequency of 400 Hz (for a 44.1-kHz sampling rate). First, the desired processing block is selected. PRB_P2 has a resource class of 12, so MDAC and DOSR must have been previously programmed to satisfy the restriction described in Section 2.2. Second, the filter coefficients are written to Biquad A for both left and right channel and to Buffers A and B. The code highlighted in blue is not necessary if adaptive filtering is not used (such as this case, for example). This script must be executed before powering up the DAC(s).

Refer to the *User Programmable Filters* section in the respective product data sheet for details about the coefficient memory space.

```
# Configure Processing Blocks
###############################################
#
# Select Page 0
w 30 00 00
#
# PRB_P2 selected
w 30 3C 02
#
###############################################
# High-pass first order Butterworth filter,
# fc = 400 Hz
###############################################
#
# Write to Buffer A:
#
# BIQUAD A, Left Channel (Page 44, Register 12, C1-C5)
w 30 00 2c
w 30 0C 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
# BIQUAD A, Right Channel (Page 45, Register 20, C33-C37)
w 30 00 2D
w 30 14 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
# Write to Buffer B:
#
# BIQUAD A, Left Channel (Page 62, Register 12, C1-C5)
w 30 00 3E
w 30 0C 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
# BIQUAD A, Right Channel (Page 63, Register 20, C33-C37)
w 30 00 3F
w 30 14 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
# ###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

For some applications, it may be desired to change filter coefficients on the fly (that is, when the DAC is enabled). In order to do this, adaptive filtering must be enabled before powering up the DAC(s) as shown below. If it is desired to power the DAC with a filter already implemented, then both Buffer A and Buffer B must be written with the same data to avoid buffer mismatch.

```
# Configure Processing Blocks
###############################################
#
# Select Page 0
w 30 00 00
#
# PRB_P2 selected
w 30 3C 02
#
# Select Page 44, Enable Adaptive filtering for DAC
w 30 00 2c 04
# ###############################################
```

Once the DAC is enabled by executing a DAC channel script, the filter coefficients can be updated by writing to the Buffer A registers, switching buffers, and writing to the Buffer A registers again, as shown below. This write sequence ensures that both buffers are synchronized for future buffer switching.

```
# High-pass first order Butterworth filter,
# fc = 400 Hz
###############################################
#
# First, write to Buffer A's registers:
#
# BIQUAD A, Left Channel (Page 44, Register 12, C1-C5)
w 30 00 2c
w 30 0C 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
# BIQUAD A, Right Channel (Page 45, Register 20, C33-C37)
w 30 00 2D
w 30 14 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
# Second, switch buffers and write again to Buffer A's registers:
w 30 00 2c 05
#
# BIQUAD A, Left Channel (Page 44, Register 12, C1-C5)
w 30 00 2c
w 30 0C 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
# BIQUAD A, Right Channel (Page 45, Register 20, C33-C37)
w 30 00 2d
w 30 14 7c 73 e4 00 c1 c6 0f 00 00 00 00 00 3c 73 e6 00 00 00 00 00
#
###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

Appendix C Power Scripts

C.1 Configure Power Using External Supplies for AVDD and DVDD

The following script fragment programs the power registers for use with external AVDD and DVDD supplies. The script assumes that the LDO_SELECT pin is tied low. The commands highlighted in blue are necessary for proper operation of the device. The first two commands highlighted in blue should be executed only if AVDD is present (internally or externally). The highest performance PowerTune™ mode for both ADC and DAC channels is used for this script.

```
# Configure Power Supplies
################################################
#
# Select Page 1
w 30 00 01
#
# Disable weak AVDD in presence of external
# AVDD supply
w 30 01 08
#
# Enable Master Analog Power Control
w 30 02 00
#
# Set full chip common mode to 0.9V
# HP output CM = full chip CM
# HP driver supply = AVDD
# Line output CM = full chip CM
# Line output supply = AVDD
w 30 0A 00
#
# Select ADC PTM_R4
w 30 3d 00
#
# Select DAC PTM_P3/4
w 30 03 00 00
#
# Set the input power-up time to 3.1ms (for ADC)
w 30 47 32
#
# Set the REF charging time to 40ms
w 30 7b 01
#
################################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

C.2 Configure Power Using Internal LDOs and 1.65-V Output Common-Mode

The following script fragment programs the power registers for use with the internal LDOs. This script assumes that the LDO_SELECT pin is pulled high and that the LDOin voltage is between 1.9 V and 3.6 V. The commands highlighted in blue are necessary for proper operation of the device.

```
# Configure Power Supplies
###############################################
#
# Select Page 1
w 30 00 01
#
# Power up AVDD LDO
w 30 02 09
#
# Disable weak AVDD in presence of external
# AVDD supply
w 30 01 08
#
# Enable Master Analog Power Control
# Power up AVDD LDO
w 30 02 01
#
# Set full chip common mode to 0.9V
# HP output CM = 1.65V
# HP driver supply = LDOin voltage
# Line output CM = 1.65V
# Line output supply = LDOin voltage
w 30 0A 3B
#
# Select ADC PTM_R4
w 30 3d 00
#
# Select DAC PTM_P3/4
w 30 03 00 00
#
# Set the input power-up time to 3.1ms (for ADC)
w 30 47 32
#
# Set the REF charging time to 40ms
w 30 7b 01
#
###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

Appendix D ADC Channel Scripts

D.1 Configure the ADC Channel for Single-ended Stereo Operation

The following script fragment programs IN1_L and IN1_R pins as single-ended stereo inputs to the left and right ADCs, respectively.

```
# Configure ADC Channel
###############################################
#
# Select Page 1
w 30 00 01
#
# Route IN1L to LEFT_P with 20K input impedance
w 30 34 80
#
# Route CM1L to LEFT_M with 20K input impedance
w 30 36 80
#
# Route IN1R to RIGHT_P with 20K input impedance
w 30 37 80
#
# Route CM1R to RIGHT_M with 20K input impedance
w 30 39 80
#
# Unmute Left MICPGA, Gain selection of 6dB to
# make channel gain 0dB, since 20K input
# impedance is used single ended
w 30 3b 0c
#
# Unmute Right MICPGA, Gain selection of 6dB to
# make channel gain 0dB, since 20K input
# impedance is used single ended
w 30 3c 0c
#
# Select Page 0
w 30 00 00
#
# Power up LADC/RADC
w 30 51 c0
#
# Unmute LADC/RADC
w 30 52 00
#
###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

D.2 Configure the ADC Channel for a Differential Electret Microphone

For systems in which an electret microphone is used, a differential configuration is often desired to for better noise rejection. The following script fragment programs IN3_L and IN3_R pins as a differential pair to the left ADC. The actual input gain is 6 dB because the input resistors are set to 10 kΩ.

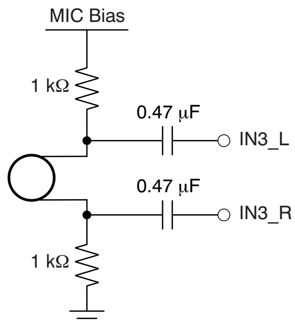


Figure 9. Differential Electret Microphone Configuration

```
# Configure ADC Channel
################################################
#
# Select Page 1
w 30 00 01
#
# Power-up MIC BIAS
w 30 33 40
#
# Route IN3L to LEFT_P with 10K input impedance
w 30 34 04
#
# Route IN3R to LEFT_M with 10K input impedance
w 30 36 04
#
# Unmute Left MICPGA
w 30 3b 00
#
# Select Page 0
w 30 00 00
#
# Power up LADC
w 30 51 80
#
# Unmute LADC
w 30 52 08
#
################################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

Appendix E DAC Channel Scripts

E.1 Configure the DAC Channel for Single-ended Stereo Outputs

The following script fragment programs the headphone and line outputs. The left and right digital channels are routed to the left and right DACs, respectively.

```
# Configure DAC Channel
###############################################
#
# Select Page 1
w 30 00 01
#
# De-pop: 5 time constants, 6k resistance
w 30 14 25
#
# Route LDAC/RDAC to HPL/HPR
w 30 0c 08 08
#
# Route LDAC/RDAC to LOL/LOR
w 30 0e 08 08
#
# Power up HPL/HPR and LOL/LOR drivers
w 30 09 3C
#
# Unmute HPL/HPR driver, 0dB Gain
w 30 10 00 00
#
# Unmute LOL/LOR driver, 0dB Gain
w 30 12 00 00
#
# Select Page 0
w 30 00 00
#
# DAC => 0dB
w 30 41 00 00
#
# Power up LDAC/RDAC
w 30 3f d6
#
# Unmute LDAC/RDAC
w 30 40 00
#
###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

E.2 Configure the DAC Channel for Differential Headphone Output

The following script fragment programs the headphone outputs for differential drive. The left channel digital data are routed to the left DAC and into the HP outputs. For this case, AV_{DD} must be used as the amplifier supply.

```
# Configure DAC Channel
###############################################
#
# Select Page 1
w 30 00 01
#
# De-pop: 5 time constants, 6k resistance
w 30 14 25
#
# Set HP outputs in BTL mode, LDAC is used
w 30 0c 08 01
#
# Power up HPL/HPR
w 30 09 30
#
# Unmute HPL/HPR driver, 0dB Gain
w 30 10 00 00
#
# Select Page 0
w 30 00 00
#
# DAC => 0dB
w 30 41 00 00
#
# Power up LDAC/RDAC
w 30 3f b2
#
# Unmute LDAC/RDAC
w 30 40 04
#
###############################################
```


Copyright © 2009–2010, Texas Instruments Incorporated

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2010, Texas Instruments Incorporated