SigmaDSP® 28-/56-Bit Audio Processor with Two ADCs and Four DACs

ADAU1702

FEATURES

28-/56-bit, 25 MIPS digital audio processor
Two ADCs: SNR of 100 dB, THD + N of −83 dB
Four DACs: SNR of 104 dB, THD + N of −90 dB
Complete standalone operation
Self-boot from serial EEPROM
Auxiliary ADC with 4-input mux for analog control
GPIOs for digital controls and outputs
Fully programmable with SigmaStudio™ graphical tool
28-bit × 28-bit multiplier with 56-bit accumulator for full double-precision processing
Clock oscillator for generating master clock from crystal
PLL for generating master clock from 64 × f_S, 256 × f_S, 384 × f_S, or 512 × f_S clocks
Flexible serial data input/output ports with I²S-compatible, left-justified, right-justified, and TDM modes
Sampling rates up to 192 kHz supported
On-chip voltage regulator for compatibility with 3.3 V systems
48-lead, plastic LQFP

APPLICATIONS

Multimedia speaker systems
MP3 player speaker docks
Automotive head units
Minicomponent stereos
Digital televisions
Studio monitors
Speaker crossovers
Musical instrument effects processors
In-seat sound systems (aircraft/motor coaches)

GENERAL DESCRIPTION

The ADAU1702 is a complete single-chip audio system with a 28-/56-bit audio DSP, ADCs, DACs, and microcontroller-like control interfaces. Signal processing includes equalization, crossover, bass enhancement, multiband dynamics processing, delay compensation, speaker compensation, and stereo image widening and can be used to compensate for real-world limitations of speakers, amplifiers, and listening environments, providing a dramatic improvements of perceived audio quality.

Its signal processing is comparable to that found in high end studio equipment. Most processing is done in full 56-bit, double-precision mode, resulting in very good low level signal performance. The ADAU1702 is a fully programmable DSP. The easy to use SigmaStudio software allows the user to graphically configure a custom signal processing flow using blocks such as biquad filters, dynamics processors, level controls, and GPIO interface controls.

ADAU1702 programs can be loaded on power-up either from a serial EEPROM through its own self-boot mechanism or from an external microcontroller. On power-down, the current state of the parameters can be written back to the EEPROM from the ADAU1702 to be recalled the next time the program is run.

Two Σ-Δ ADCs and four Σ-Δ DACs provide a 98.5 dB analog input to analog output dynamic. Each ADC has a THD + N of −83 dB, and each DAC has a THD + N of −90 dB. Digital input and output ports allow a glueless connection to additional ADCs and DACs. The ADAU1702 communicates through an I²C® bus or a 4-wire SPI® port.

FUNCTIONAL BLOCK DIAGRAM

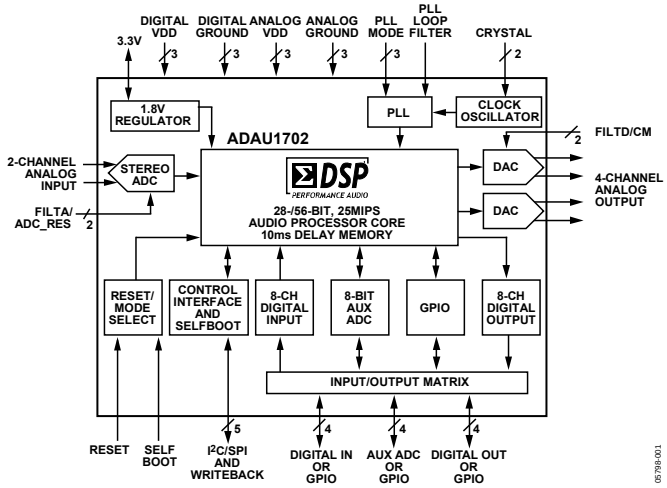

Figure 1.

Rev. 0

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 **www.analog.com**
Fax: 781.461.3113

TABLE OF CONTENTS

REVISION HISTORY

10/06—Revision 0: Initial Version

SPECIFICATIONS

AVDD = 3.3 V, DVDD = 1.8 V, PVDD = 3.3 V, IOVDD = 3.3 V, ambient temperature 25° C, master clock input 12.288 MHz, unless otherwise noted .

ANALOG PERFORMANCE

Table 1.

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ADC INPUTS					
Number of Channels		2			Stereo input
Resolution		24		Bits	
Full-Scale Input		100 (283)		μA_{rms} (μA_{p-p})	2 V_{rms} input with 20 kΩ (18 kΩ external + 2 kΩ internal) series resistor
Signal-to-Noise Ratio					
A-Weighted		100		dB	
Dynamic Range					−60 dB with respect to full-scale analog input
A-Weighted	95	100		dB	
Total Harmonic Distortion + Noise		−83		dB	−3 dB with respect to full-scale analog input
Interchannel Gain Mismatch		25	250	mdB	
Crosstalk		−82		dB	Analog channel-to-channel crosstalk
DC Bias		1.5		V	
Gain Error	−11		+11	%	
DAC OUTPUTS					
Number of Channels		4			Two stereo output channels
Resolution		24		Bits	
Full-Scale Analog Output		0.9 (2.5)		V_{rms} (V_{P-P})	
Signal-to-Noise Ratio					
A-Weighted		104		dB	
Dynamic Range					−60 dB with respect to full-scale analog output
A-Weighted	99	104		dB	
Total Harmonic Distortion + Noise		−90		dB	−1 dB with respect to full-scale analog output
Crosstalk		−100		dB	Analog channel-to-channel crosstalk
Interchannel Gain Mismatch		25	250	mdB	
Gain Error	−10		+10	%	
DC Bias		1.5		V	
VOLTAGE REFERENCE					
Absolute Voltage (CM, FILTA, FILTD)		1.5		V	
AUXILIARY ADC					
Full-Scale Analog Input		3.0		V	
INL		0.5		LSB	
DNL		1.0		LSB	
Offset		15		mV	
Input Impedance		30		kΩ	

DIGITAL INPUT/OUTPUT

Table 2.

Parameter	Min	Typ	Max	Unit	Comments
Input Voltage, High (V_{IH})	2.0		IOVDD	V	
Input Voltage, Low (V_{IL})			0.8	V	
Input Leakage, High (I_{IH})			1	µA	Excluding MCLKI
Input Leakage, Low (I_{IL})			1	µA	Excluding MCLKI and bidirectional pins
Bidirectional Pin Pull-Up Current, Low		150		µA	
MCLKI Input Leakage, High (I_{IH})			3	µA	
MCLKI Input Leakage, Low (I_{IL})			3	µA	
High Level Output Voltage (V_{OH}), I_{OH} = 2 mA	2.0			V	
Low Level Output Voltage (V_{OL}), I_{OL} = 2 mA			0.8	V	
Input Capacitance			5	pF	
GPIO Output Drive		2		mA	

POWER

Table 3.

Parameter	Min	Typ	Max[1]	Unit
SUPPLY VOLTAGE				
Analog Voltage		3.3		V
Digital Voltage		1.8		V
PLL Voltage		3.3		V
IOVDD Voltage		3.3		V
SUPPLY CURRENT				
Analog Current (AVDD and PVDD)		50	85	mA
Digital Current (DVDD)		40	60	mA
Analog Current, Reset		35	55	mA
Digital Current, Reset		1.5	4.5	mA
DISSIPATION				
Operation (AVDD, DVDD, PVDD)[2]		286.5		mW
Reset, All Supplies		118		mW
POWER SUPPLY REJECTION RATIO (PSRR)				
1 kHz, 200 mV_{P-P} Signal at AVDD		50		dB

[1] Maximum specifications are measured across a temperature range of −40°C to +130°C (case) and across a DVDD range of 1.62 V to 1.98 V and an AVDD range of 2.97 V to 3.63 V.
[2] Power dissipation does not include IOVDD power because the current drawn from this supply is dependent on the loads at the digital output pins.

TEMPERATURE RANGE

Table 4.

Parameter	Min	Typ	Max	Unit
Functionality Guaranteed	0°C		70°C	°C ambient

PLL AND OSCILLATOR

Table 5.

Parameter	Min	Typ	Max	Unit	Comments
PLL Operating Range	MCLK_Nom − 20%		MCLK_Nom + 20%	MHz	MCLK_Nom is the nominal input in a given mode (for example, 12.288 MHz in 256 × f_S mode with a f_S of 48 kHz)
PLL Lock Time			20	ms	
Crystal Oscillator g_m (Transconductance)		78		mmho	

REGULATOR

Table 6. Regulator[1]

Parameter	Min	Typ	Max	Unit
DVDD Voltage	1.7	1.8	1.84	V

[1] Regulator specifications are calculated using a Zetex Semiconductors FZT953 transistor in the circuit.

DIGITAL TIMING SPECIFICATIONS

Table 7. Digital Timing[1]

	Limit			
Parameter	**T_{MIN}**	**T_{MAX}**	**Unit**	**Description**
MASTER CLOCK				
t_{MP}	36	244	ns	MCLK period, 512 f_S mode.
t_{MP}	48	366	ns	MCLK period, 384 f_S mode.
t_{MP}	73	488	ns	MCLK period, 256 f_S mode.
t_{MP}	291	1953	ns	MCLK period, 64 f_S mode.
SERIAL PORT				
t_{BIL}	40		ns	INPUT_BCLK low pulse width.
t_{BIH}	40		ns	INPUT_BCLK high pulse width.
t_{LIS}	10		ns	INPUT_LRCLK setup. Time to INPUT_BCLK rising.
t_{LIH}	10		ns	INPUT_LRCLK hold. Time from INPUT_BCLK rising.
t_{SIS}	10		ns	SDATA_INx setup. Time to BCLK_IN rising.
t_{SIH}	10		ns	SDATA_INx hold. Time from BCLK_IN rising.
t_{LOS}	10		ns	OUTPUT_LRCLK setup in slave mode.
t_{LOH}	10		ns	OUTPUT_LRCLK hold in slave mode.
t_{TS}		5	ns	OUTPUT_BCLK falling to OUTPUT_LRCLK timing skew.
t_{SODS}		40	ns	SDATA_OUTx delay. Time from OUTPUT_BCLK falling in slave mode.
t_{SODM}		40	ns	SDATA_OUTx delay. Time from OUTPUT_BCLK falling in master mode.
SPI PORT				
f_{CCLK}		6.25	MHz	CCLK frequency.
t_{CCPL}	80		ns	CCLK pulse width low.
t_{CCPH}	80		ns	CCLK pulse width high.
t_{CLS}	0		ns	CLATCH setup. Time to CCLK rising.
t_{CLH}	100		ns	CLATCH hold. Time from CCLK rising.
t_{CLPH}	80		ns	CLATCH pulse width high.
t_{CDS}	0		ns	CDATA setup. Time to CCLK rising.
t_{CDH}	80		ns	CDATA hold. Time from CCLK rising.
t_{COD}		101	ns	COUT delay. Time from CCLK falling.
I^2C PORT				
f_{SCL}		400	kHz	SCL frequency.
t_{SCLH}	0.6		µs	SCL high.
t_{SCLL}	1.3		µs	SCL low.
t_{SCS}	0.6		µs	Setup time, relevant for repeated start condition.
t_{SCH}	0.6		µs	Hold time. After this period, the first clock is generated.
t_{DS}	100		ns	Data setup time.
t_{SCR}		300	ns	SCL rise time.
t_{SCF}		300	ns	SCL fall time.
t_{SDR}		300	ns	SDA rise time.
t_{SDF}		300	ns	SDA fall time.
t_{BFT}	0.6			Bus-free time. Time between stop and start.

Parameter	Limit		Unit	Description
	T_{MIN}	T_{MAX}		
MULTIPURPOSE PINS AND RESET				
t_{GRT}		50	ns	GPIO rise time.
t_{GFT}		50	ns	GPIO fall time.
t_{GIL}		1.5 × $1/f_S$	µs	GPIO input latency. Time until high/low value is read by core.
t_{RLPW}	20		ns	$\overline{\text{RESET}}$ low pulse width.

[1] All timing specifications are given for the default (I^2S) states of the serial input port and the serial output port (see Table 66).

DIGITAL TIMING DIAGRAMS

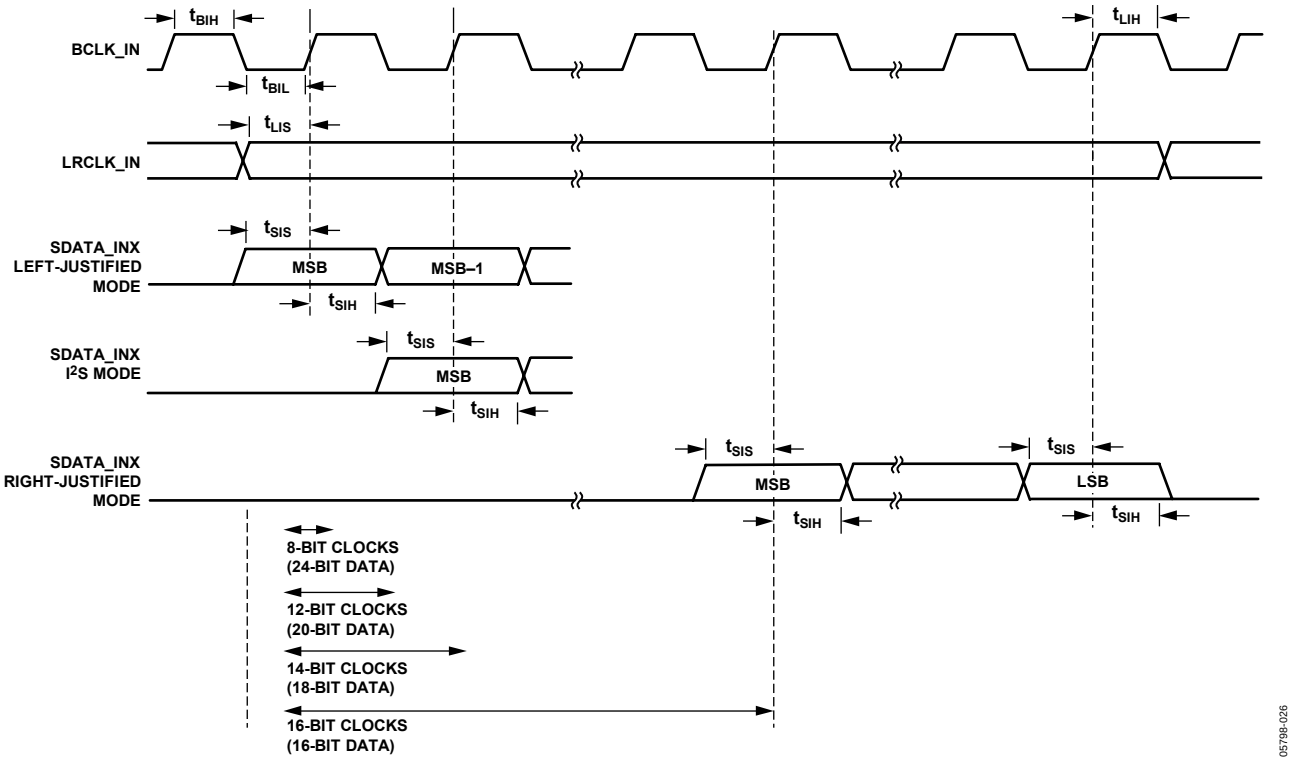

Figure 2. Serial Input Port Timing

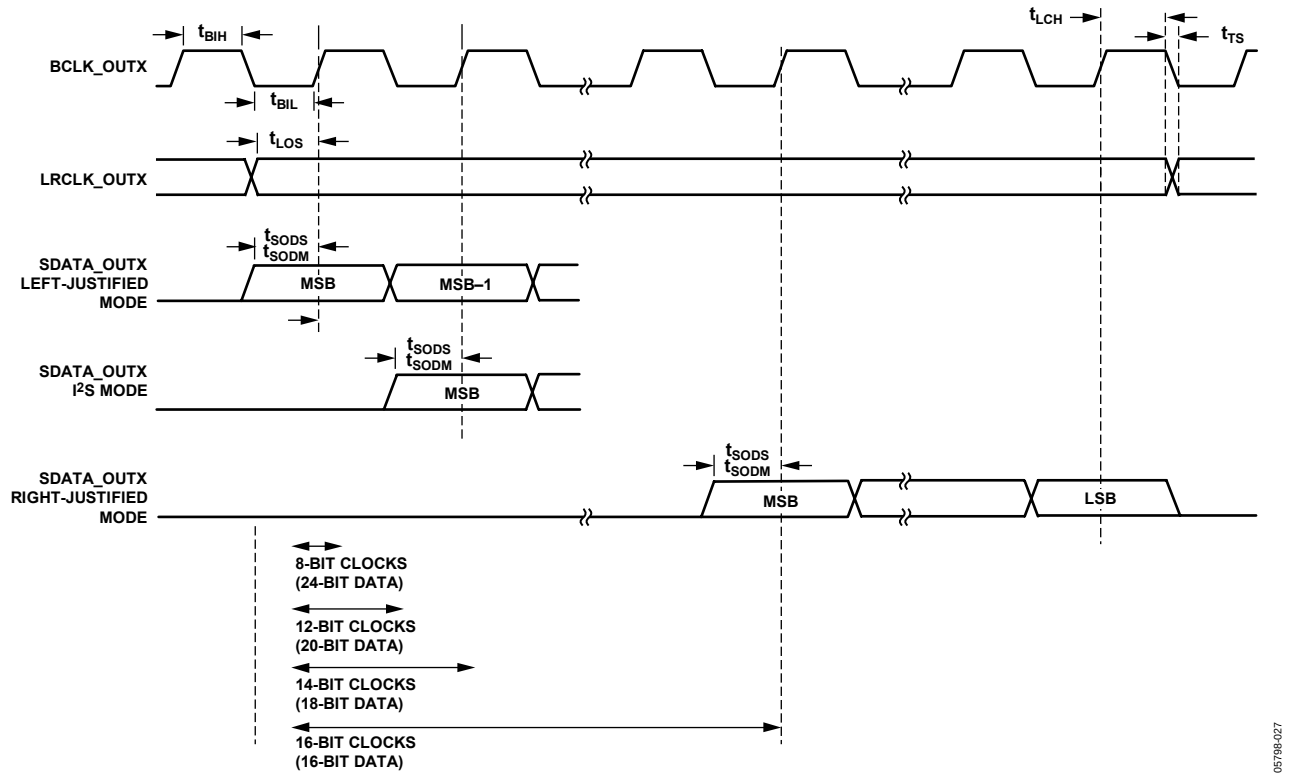

Figure 3. Serial Output Port Timing

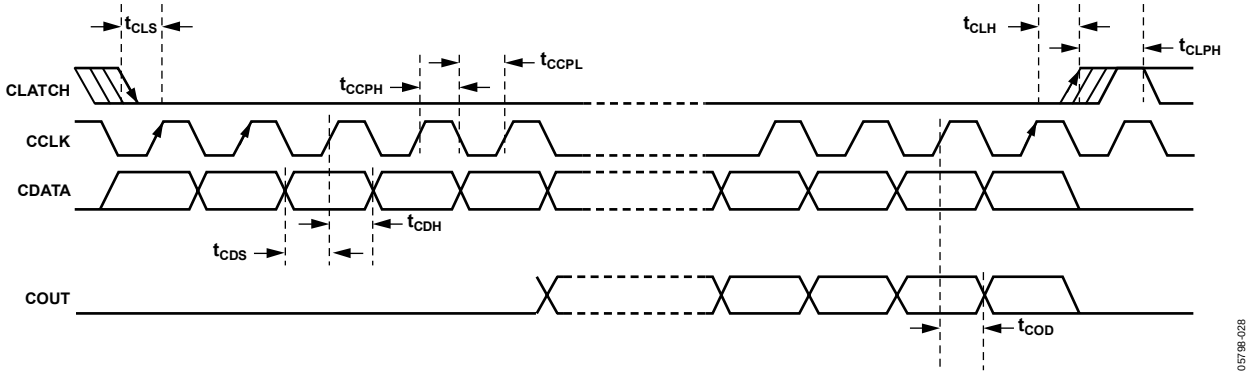

Figure 4. SPI Port Timing

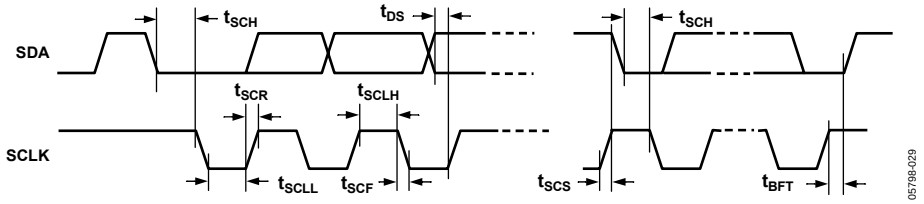

Figure 5. I²C Port Timing

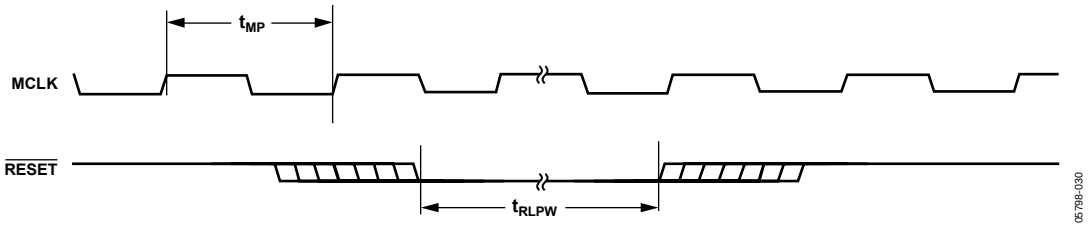

Figure 6. Master Clock and $\overline{RESET}$ Timing

ABSOLUTE MAXIMUM RATINGS

Table 8.

Parameter	Rating
DVDD to GND	0 V to 2.2 V
AVDD to GND	0 V to 4.0 V
IOVDD to GND	0 V to 4.0 V
Digital Inputs	DGND − 0.3 V, IOVDD + 0.3 V
Maximum Junction Temperature	135°C
Storage Temperature Range	−65°C to +150°C
Soldering (10 sec)	300°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL RESISTANCE

θ_{JA} is specified for the worst-case conditions, that is, a device soldered in a circuit board for surface-mount packages.

Table 9. Thermal Resistance

Package Type	θ_{JA}	θ_{JC}	Unit
48-Lead LQFP	72	19.5	°C/W

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

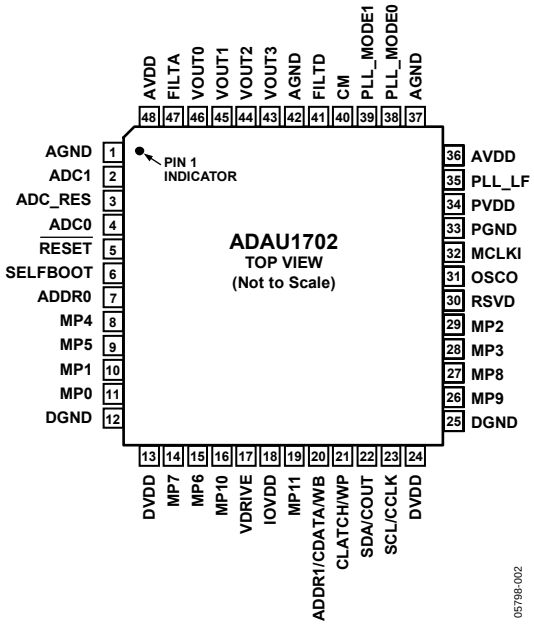

Figure 7. 48-Lead LQFP Pin Configuration

Table 10. Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Page No.	Description
1, 37, 42	AGND	PWR		Analog Ground Pin. The AGND, DGND, and PGND pins can be tied directly together in a common ground plane. AGND should be decoupled to an AVDD pin with a 100 nF capacitor.
2	ADC1	A_IN	19	Analog Audio Input 1. Full-scale 100 μA_{rms} input. Current input allows input voltage level to be scaled with an external resistor. An 18 kΩ resistor gives a 2 V_{rms} full-scale input.
3	ADC_RES	A_IN	19	ADC Reference Current. The full-scale current of the ADCs can be set with an external 18 kΩ resistor connected between this pin and ground.
4	ADC0	A_IN	19	Analog Audio Input 0. Full-scale 100 μA_{rms} input. Current input allows input voltage level to be scaled with an external resistor. An 18 kΩ resistor gives a 2 V_{rms} full-scale input.
5	$\overline{\text{RESET}}$	D_IN		Active Low Reset Input. Reset is triggered on a high-to-low edge and the ADAU1702 exits reset on a low-to-high edge. For more information about initialization, see the Power-Up Sequence section.
6	SELFBOOT	D_IN	26	Enable/Disable Self-Boot. SELFBOOT selects control port (low) or self-boot (high). Setting this pin high initiates a self-boot operation when the ADAU1702 is brought out of reset. This pin can be tied directly to the control voltage or pulled up/down with a resistor.
7	ADDR0	D_IN	22	I^2C and SPI Address 0. In combination with ADDR1, this pin allows up to four ADAU1702s to be used on the same I^2C bus and up to two ICs to be used with a common SPI CLATCH signal.
8	MP4	D_IO	44	Multipurpose GPIO or Serial Input Port LRCLK (INPUT_LRCLK).
9	MP5	D_IO	44	Multipurpose GPIO or Serial Input Port BCLK (INPUT_BCLK).
10	MP1	D_IO	44	Multipurpose GPIO or Serial Input Port Data 1 (SDATA_IN0).
11	MP0	D_IO	44	Multipurpose GPIO or Serial Input Port Data 0 (SDATA_IN1).
12, 25	DGND	PWR		Digital Ground Pin. The AGND, DGND, and PGND pins can be tied directly together in a common ground plane. DGND should be decoupled to a DVDD pin with a 100 nF capacitor.
13, 24	DVDD	PWR		1.8 V Digital Supply. This can be supplied either externally or generated from a 3.3 V supply with the on-board 1.8 V regulator. DVDD should be decoupled to DGND with a 100 nF capacitor.

Pin No.	Mnemonic	Type[1]	Page No.	Description
14	MP7	D_IO	44	Multipurpose GPIO or Serial Output Port Data 1 (SDATA_OUT1).
15	MP6	D_IO	44	Multipurpose GPIO, Serial Output Port Data 0, or TDM Data Output (SDATA_OUT0).
16	MP10	D_IO	44	Multipurpose GPIO or Serial Output Port LRCLK (OUTPUT_LRCLK).
17	VDRIVE	A_OUT	5	Drive for 1.8 V Regulator. The base of the voltage regulator external PNP transistor is driven from VDRIVE.
18	IOVDD	PWR		Supply for Input and Output Pins. The voltage on this pin sets the highest input voltage that should be seen on the digital input pins. This pin is also the supply for the digital output signals on the control port and MP pins. IOVDD should always be set to 3.3 V. The current draw of this pin is variable because it is dependent on the loads of the digital outputs.
19	MP11	D_IO	44	Multipurpose GPIO or Serial Output Port BCLK (OUTPUT_BCLK).
20	ADDR1/CDATA/WB	D_IN	22, 24, 26	ADDR1: I²C Address 1. In combination with ADDR0, this sets the I²C address of the IC so that four ADAU1702s can be used on the same I²C bus. CDATA: SPI Data Input. WB: EEPROM Writeback Trigger. A rising (default) or falling (if set in the EEPROM messages) edge on this pin triggers a writeback of the interface registers to the external EEPROM. This function can be used to save parameter data on power-down.
21	CLATCH/WP	D_IO	24, 26	CLATCH: SPI Latch Signal. Must go low at the beginning of an SPI transaction and high at the end of a transaction. Each SPI transaction can take a different number of CCLKs to complete, depending on the address and read/write bit that are sent at the beginning of the SPI transaction. WP: Self-Boot EEPROM Write Protect. This pin is an open-collector output when in self-boot mode. The ADAU1702 pulls this low to prohibit writes to an external EEPROM. This pin should be pulled high to 3.3 V.
22	SDA/COUT	D_IO	21, 24	SDA: I²C Data. This pin is a bidirectional open-collector. The line connected to this pin should have a 2.2 kΩ pull-up resistor. COUT: This SPI data output is used for reading back registers and memory locations. It is three-stated when an SPI read is not active.
23	SCL/CCLK	D_IO	21, 24	SCL: I²C Clock. This pin is always an open-collector input when in I²C control mode. In self-boot mode, this pin is an open-collector output (I²C master). The line connected to this pin should have a 2.2 kΩ pull-up resistor. CCLK: SPI Clock. This pin can either run continuously or be gated off in between SPI transactions.
26	MP9	D_IO/A_IO	44	Multipurpose GPIO, Serial Output Port Data 3 (SDATA_OUT3), or Auxiliary ADC Input 0.
27	MP8	D_IO/A_IO	44	Multipurpose GPIO, Serial Output Port Data 2 (SDATA_OUT2), or Auxiliary ADC Input 3.
28	MP3	D_IO/A_IO	44	Multipurpose GPIO, Serial Input Port Data 3 (SDATA_IN3), or Auxiliary ADC Input 2.
29	MP2	D_IO/A_IO	44	Multipurpose GPIO, Serial Input Port Data 2 (SDATA_IN2), or Auxiliary ADC Input 1.
30	RSVD	X		Reserved. Tie to ground, either directly or through a pull-down resistor.
31	OSCO	D_OUT	17	Crystal Oscillator Circuit Output. A 100 Ω damping resistor should be connected between this pin and the crystal. This output should not be used to directly drive a clock to another IC. If the crystal oscillator is not used, this pin can be left disconnected.
32	MCLKI	D_IN	17	Master Clock Input. MCLKI can either be connected to a 3.3 V clock signal or can be the input from the crystal oscillator circuit.
33	PGND	PWR		PLL Ground Pin. The AGND, DGND, and PGND pins can be tied directly together in a common ground plane. PGND should be decoupled to PVDD with a 100 nF capacitor.
34	PVDD	PWR		3.3 V Power Supply for the PLL and the Auxiliary ADC Analog Section. This should be decoupled to PGND with a 100 nF capacitor.
35	PLL_LF	A_OUT	17	PLL Loop Filter Connection. Two capacitors and a resistor need to be connected to this pin, as shown in the Setting Master Clock/PLL Mode section.
36, 48	AVDD	PWR		3.3 V Analog Supply. This should be decoupled to AGND with a 100 nF capacitor.
38	PLL_MODE0	D_IN	17	PLL Mode Setting. PLL_MODE0 and PLL_MODE1 set the output frequency

Pin No.	Mnemonic	Type[1]	Page No.	Description
39	PLL_MODE1	D_IN	17	of the master clock PLL. See the Setting Master Clock/PLL Mode section for more details.
40	CM	A_OUT		1.5 V Common-Mode Reference. A 47 µF decoupling capacitor should be connected between this pin and ground to reduce crosstalk between the ADCs and DACs. The material of the capacitors is not critical. This pin can be used to bias external analog circuits, as long as they are not drawing current from CM (for example, the noninverting input of an op amp).
41	FILTD	A_OUT		DAC Filter Decoupling Pin. Should be connected to a 10 µF capacitor to ground. The capacitor material is not critical. The voltage on the FILTD is 1.5 V.
43	VOUT3	A_OUT	20	VOUT0 to VOUT3 are the DAC Outputs. Full-scale output voltage is 0.9 V_{rms}. These outputs can be used with either active or passive output reconstruction filters.
44	VOUT2	A_OUT	20	
45	VOUT1	A_OUT	20	
46	VOUT0	A_OUT	20	
47	FILTA	A_OUT		ADC Filter Decoupling Pin. Should be connected to a 10 µF capacitor to ground. The capacitor material is not critical. The voltage on the FILTA pin is 1.5 V.

[1] PWR = power/ground, A_IN = analog input, D_IN = digital input, A_OUT = analog output, D_IO = digital input/output, D_IO/A_IO = digital input/output or analog input/output.

TYPICAL PERFORMANCE CHARACTERISTICS

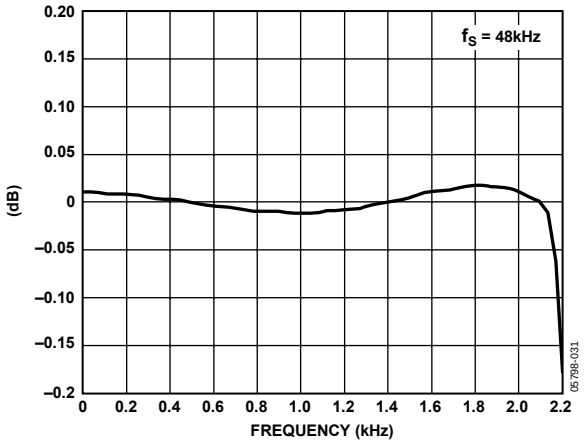

Figure 8. ADC Pass-Band Filter Response, f_S = 48 kHz

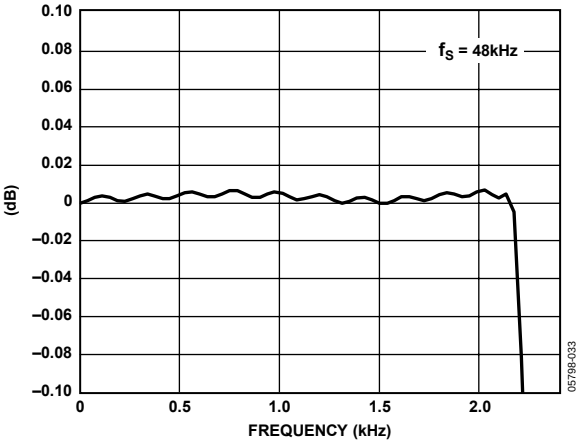

Figure 10. DAC Pass-Band Filter Response, f_S = 48 kHz

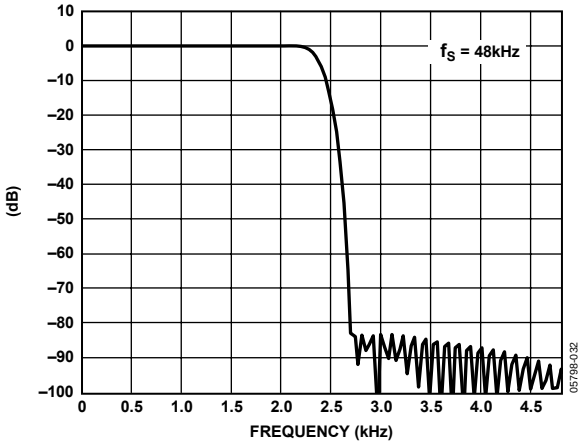

Figure 9. ADC Stop-Band Filter Response, f_S = 48 kHz

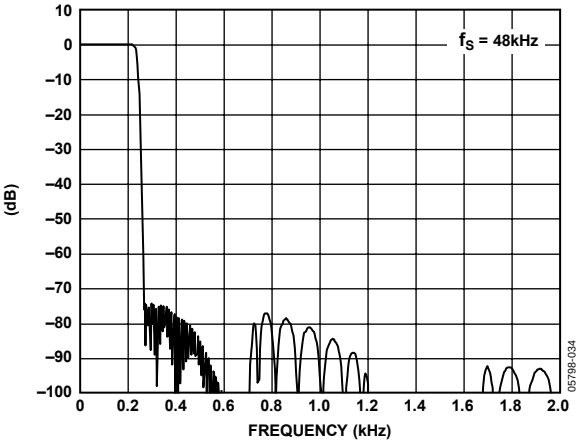

Figure 11. DAC Stop-Band Filter Response, f_S = 48 kHz

SYSTEM BLOCK DIAGRAM

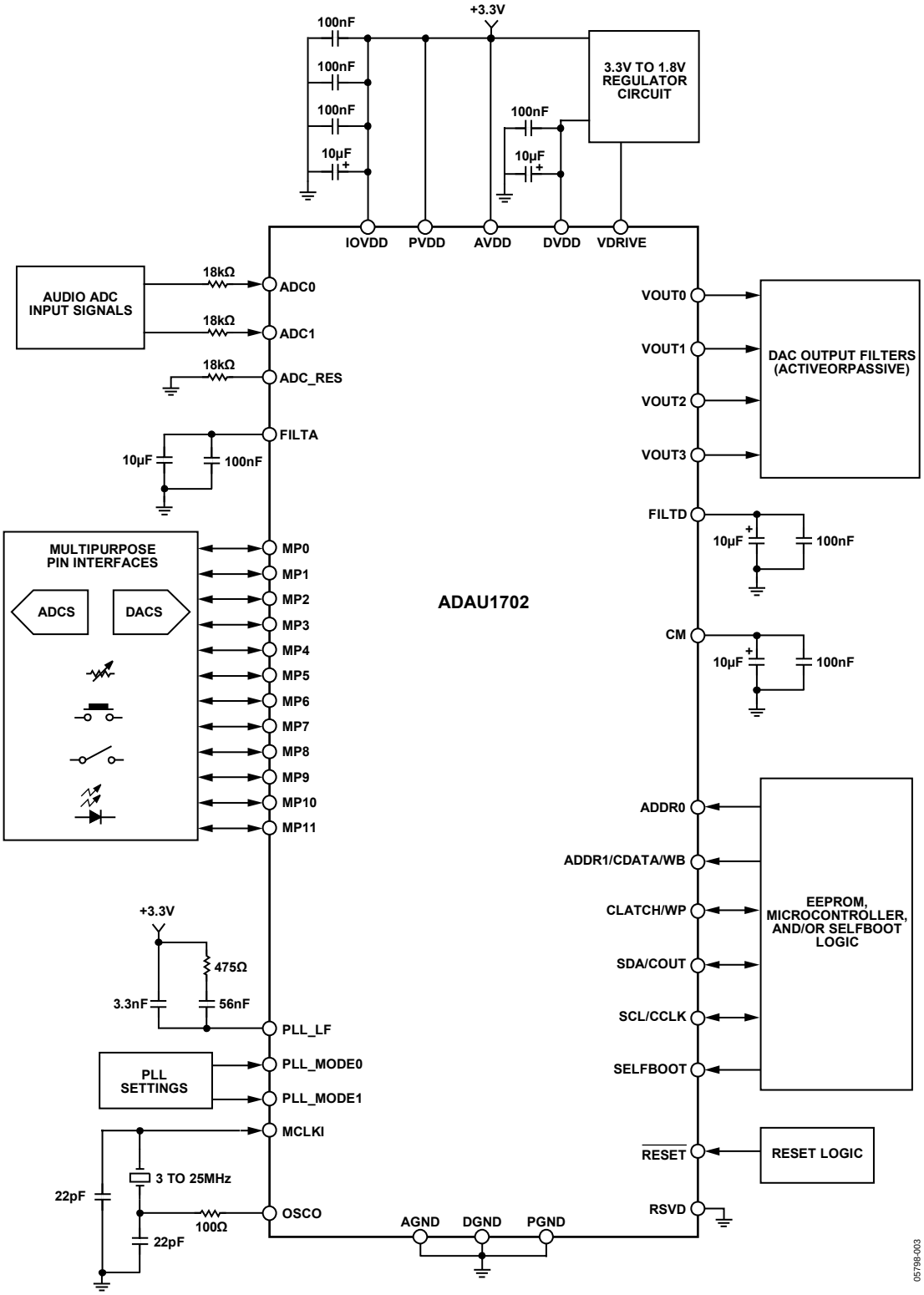

Figure 12. System Block Diagram

OVERVIEW

The core of the ADAU1702 is a 28-bit DSP (56-bit with double-precision processing) optimized for audio processing. The program and parameter RAMs can be loaded with a custom audio processing signal flow built by using SigmaStudio graphical programming software from Analog Devices, Inc. The values stored in the parameter RAM control individual signal processing blocks, such as equalization filters, dynamics processors, audio delays, and mixer levels. A safeload feature allows for transparent parameter updates and prevents clicks in the output signals.

The program RAM, parameter RAM, and register contents can be saved in an external EEPROM, from which the ADAU1702 can self-boot on start-up. In this standalone mode, parameters can be controlled through the on-board multipurpose pins. The ADAU1702 can accept controls from switches, potentiometers, rotary encoders, and IR receivers. Parameters such as volume and tone settings can be saved to the EEPROM on power-down and recalled again on power-up.

The ADAU1702 can operate with digital or analog inputs and outputs, or a mix of both. The stereo ADC and four DACs each have an SNR of at least +100 dB and a THD + N of at least −83 dB. The 8-channel, flexible serial data input/output ports allow glueless interconnection to a variety of ADCs, DACs, general-purpose DSPs, S/PDIF receivers and transmitters, and sample rate converters. The serial ports of the ADAU1702 can be configured in I^2S, left-justified, right-justified, or TDM serial port compatible modes.

Twelve multipurpose (MP) pins allow the ADAU1702 to input external control signals and output flags or controls to other devices in the system. The MP pins can be configured as digital I/Os, inputs to the 4-channel auxiliary ADC, or set up as the serial data I/O ports. As inputs, they can be connected to buttons, switches, rotary encoders, potentiometers, IR receivers, or other external circuitry to control the internal signal processing program. When configured as outputs, these pins can be used to drive LEDs, control other ICs, or connect to other external circuitry in an application.

The ADAU1702 has a sophisticated control port that supports complete read/write capability of all memory locations. Control registers are provided to offer complete control of the chip's configuration and serial modes. The ADAU1702 can be configured for either SPI or I^2C control, or can self-boot from an external EEPROM.

An on-board oscillator can be connected to an external crystal to generate the master clock. In addition, a master clock phase-locked loop (PLL) allows the ADAU1702 to be clocked from a variety of different clock speeds. The PLL can accept inputs of $64 \times f_S$, $256 \times f_S$, $384 \times f_S$, or $512 \times f_S$ to generate the internal master clock of the core.

The SigmaStudio software is used to program and control the SigmaDSP through the control port. Along with designing and tuning a signal flow, the tools can be used to configure all of the DSP registers and burn a new program into the external EEPROM. SigmaStudio's graphical interface allows anyone with digital or analog audio processing knowledge to easily design a DSP signal flow and port it to a target application. At the same time, it provides enough flexibility and programmability for an experienced DSP programmer to have in-depth control of the design. In SigmaStudio, the user can connect graphical blocks (such as biquad filters, dynamics processors, mixers, and delays), compile the design, and load the program and parameter files into the ADAU1702 memory through the control port. Signal processing blocks available in the provided libraries include

- Single- and double-precision biquad filters
- Processors with peak or rms detection for monochannel and multichannel dynamics
- Mixers and splitters
- Tone and noise generators
- Fixed and variable gain
- Loudness
- Delay
- Stereo enhancement
- Dynamic bass boost
- Noise and tone sources
- FIR filters
- Level detectors
- GPIO control and conditioning

Additional processing blocks are always being developed. Analog Devices also provides proprietary and third-party algorithms for applications such as matrix decoding, bass enhancement, and surround virtualizers. Contact Analog Devices for information about licensing these algorithms.

The ADAU1702 operates from a 1.8 V digital power supply and a 3.3 V analog supply. An on-board voltage regulator can be used to operate the chip from a single 3.3 V supply. It is fabricated on a single monolithic, integrated circuit and is packaged in a 48-lead LQFP for operation over the 0°C to +70°C temperature range.

INITIALIZATION

This section details the procedure for properly setting up the ADAU1702. The following five-step sequence provides an overview of how to initialize the IC:

1. Apply power to ADAU1702.
2. Wait for PLL to lock.
3. Load SigmaDSP program and parameters.
4. Set up registers (including multipurpose pins and digital interfaces).
5. Turn off the default muting of the converters, clear the data registers, and initialize the DAC setup register (see the Control Registers Setup section for specific settings).

To only test analog audio pass-through (ADCs to DACs), Steps 3 and 4 can be skipped and the default internal program can be used.

POWER-UP SEQUENCE

The ADAU1702 has a built-in power-up sequence that initializes the contents of all internal RAMs on power-up or when the device is brought out of a reset. On the positive edge of $\overline{\text{RESET}}$, the contents of the internal program boot ROM are copied to the internal program RAM memory, the parameter RAM is filled with values (all 0s) from its associated boot ROM, and all registers are initialized to 0s. The default boot ROM program copies audio from the inputs to outputs without processing it (see Figure 13). In this program, serial digital Input 0 and Input 1 are output on DAC0 and DAC1 and serial digital Output 0 and Output 1. ADC0 and ADC1 are output on DAC2 and DAC3. The data memories are also zeroed at power-up. New values should not be written to the control port until the initialization is complete.

Table 11. Power-Up Time

MCLKI Input	Init. Time	Max Program/ Parameter/Register Boot Time (I^2C)	Total
3.072 MHz (64 × f_S)	85 ms	133 ms	218 ms
11.289 MHz (256 × f_S)	23 ms	133 ms	156 ms
12.288 MHz (256 × f_S)	21 ms	133 ms	154 ms
18.432 MHz (384 × f_S)	16 ms	133 ms	149 ms
24.576 MHz (512 × f_S)	11 ms	133 ms	144 ms

The PLL start-up time lasts for 2^{18} cycles of the clock on the MCLKI pin. This time ranges from 10.7 ms for a 24.576 MHz (512 × f_S) input clock to 85.3 ms for a 3.072 MHz (64 × f_S) input clock. This start-up time is measured from the rising edge of $\overline{\text{RESET}}$. Following the PLL startup, the duration of the ADAU1702 boot cycle is about 42 µs for a f_S of 48 kHz. The user should avoid writing to or reading from the ADAU1702 during this start-up time. For an MCLK input of 12.288 MHz, the full initialization sequence (PLL startup plus boot cycle) is approximately 21 ms. As the device comes out of a reset, the clock mode is immediately set by the PLL_MODE0 and PLL_MODE1 pins. The reset is synchronized to the falling edge of the internal clock.

Table 11 lists typical times to boot the ADAU1702 into an application's operational state, assuming a 400 kHz I^2C clock loading a full program, parameter set, and all registers (about 6.5 kB). In reality, most applications will not fill the RAMs and therefore boot time (Column 3 of Table 11) will be less.

CONTROL REGISTERS SETUP

The following registers must be set as described in this section to initialize the ADAU1702. These settings are the basic minimum settings needed to operate the IC with an analog input/output of 48 kHz. More registers may need to be set, depending on the application. See the RAMs and Registers section for additional settings.

DSP Core Control Register (Address 2076)

Set Bits [4:2] (ADM, DAM, and CR) each to 1.

DAC Setup Register (Address 2087)

Set Bits [0:1] (DS [1:0]) to 01.

RECOMMENDED PROGRAM/PARAMETER LOADING PROCEDURE

When writing large amounts of data to the program or parameter RAM in direct write mode, the processor core should be disabled to prevent unpleasant noises from appearing at the audio output.

1. Set Bit 3 and Bit 4 (active low) of the core control register to 1 to mute the ADCs and DACs. This begins a volume ramp-down.
2. Set Bit 2 (active low) of the core control register to 1. This zeroes the SigmaDSP accumulators, the data output registers, and the data input registers.
3. Fill the program RAM using burst mode writes.
4. Fill the parameter RAM using burst mode writes.
5. Deassert Bit 2 to Bit 4 of the core control register.

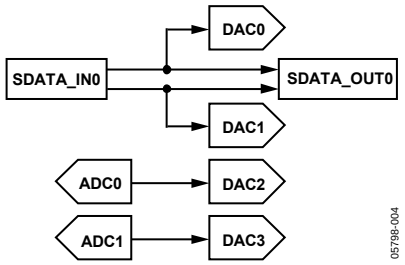

Figure 13. Default Program Signal Flow

POWER-REDUCTION MODES

Sections of the ADAU1702 chip can be turned on and off as needed to reduce power consumption. These include the ADCs, DACs, and voltage reference.

The individual analog sections can be turned off by writing to the auxiliary ADC and power control register. By default, the ADCs, DACs, and reference are enabled (all bits set to 0). Each

of these can be turned off by writing a 1 to the appropriate bits in this register. The ADC power-down mode powers down both ADCs, and each DAC can be powered down individually. The current savings is about 15 mA when the ADCs are powered down and about 4 mA for each DAC that is powered down. The voltage reference, which is supplied to both the ADCs and DACs, should only be powered down if all ADCs and DACs are powered down. The reference is powered down by setting both Bit 6 and Bit 7 of the control register.

USING THE OSCILLATOR

The ADAU1702 can use an on-board oscillator to generate its master clock. The oscillator is designed to work with a 256 × f_S master clock, which is 12.288 MHz for a f_S of 48 kHz and 11.2896 MHz for a f_S of 44.1 kHz. The crystal in the oscillator circuit should be an AT-cut, parallel resonator operating at its fundamental frequency. Figure 14 shows the external circuit recommended for proper operation.

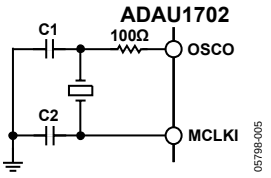

Figure 14. Crystal Oscillator Circuit

The 100 Ω damping resistor on OSCO gives the oscillator a voltage swing of approximately 2.2 V. The crystal shunt capacitance should be 7 pF. Its load capacitance should be about 18 pF, although the circuit supports values of up to 25 pF. The necessary values of the C1 and C2 load capacitors can be calculated from the crystal load capacitance as follows:

$$C_L = \frac{C1 \times C2}{C1 + C2} + C_{stray}$$

where C_{stray} is the stray capacitance in the circuit and is usually assumed to be approximately 2 pF to 5 pF.

OSCO should not be used to directly drive the crystal signal to another IC. This signal is an analog sine wave and is not appropriate to drive a digital input. There are two options for using the ADAU1702 to provide a master clock to other ICs in the system. The first, and less recommended method, is to use a high impedance input digital buffer on the OSCO signal. If this is done, minimize the trace length to the buffer input. The second method is to use a clock from the serial output port. Pin MP11 can be set as an output (master) clock divided down from the internal core clock. If this pin is set to serial output port (OUTPUT_BCLK) mode in the multipurpose pin configuration register (2081) and the port is set to master in the serial output control register (2078), the desired output frequency can also be set in the serial output control register with Bits OBF [1:0] (see Table 49).

If the oscillator is not utilized in the design, it can be powered down to save power. This can be done if a system master clock is already available in the system. By default, the oscillator is powered on. The oscillator powers down when a 1 is written to the OPD bit of the oscillator power-down register (see Table 60).

SETTING MASTER CLOCK/PLL MODE

The MCLK input of the ADAU1702 feeds a PLL, which generates the 25 MIPS SigmaDSP core clock. In normal operation, the input to MCLK must be one of the following: 64 × f_S, 256 × f_S, 384 × f_S, or 512 × f_S, where f_S is the input sampling rate. The mode is set on PLL_MODE0 and PLL_MODE1 as described in Table 12. If the ADAU1702 is set to receive double-rate signals (by reducing the number of program steps per sample by a factor of 2 using the core control register), the master clock frequencies must be 32 × f_S, 128 × f_S, 192 × f_S, or 256 × f_S. If the ADAU1702 is set to receive quad-rate signals (by reducing the number of program steps per sample by a factor of 4 using the core control register), the master clock frequencies must be 16 × f_S, 64 × f_S, 96 × f_S, or 128 × f_S. On power-up, a clock signal must be present on MCLK so that the ADAU1702 can complete its initialization routine.

Table 12. PLL Modes

MCLKI Input	PLL_MODE0	PLL_MODE1
64 × f_S	0	0
256 × f_S	0	1
384 × f_S	1	0
512 × f_S	1	1

The clock mode should not be changed without also resetting the ADAU1702. If the mode is changed during operation, a click or pop can result in the output signals. The state of the PLL_MODEx pins should be changed while $\overline{\text{RESET}}$ is held low.

The PLL loop filter should be connected to the PLL_LF pin. This filter, shown in Figure 15, includes three passive components—two capacitors and a resistor. The values of these components do not need to be exact; the tolerance can be up to 10% for the resistor and up to 20% for the capacitors. The 3.3 V signal shown in Figure 15 can be connected to the AVDD supply of the chip.

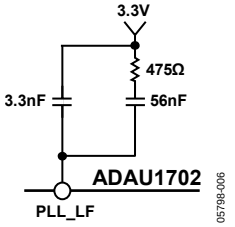

Figure 15. PLL Loop Filter

VOLTAGE REGULATOR

The digital voltage of the ADAU1702 must be set to 1.8 V. The chip includes an on-board voltage regulator that allows the device to be used in systems without an available 1.8 V supply but with an available 3.3 V supply. The only external components needed in such instances are a PNP transistor, a resistor, and a few bypass capacitors. Only one pin, VDRIVE, is necessary to support the regulator.

The recommended design for the voltage regulator is shown in Figure 16. The 10 µF and 100 nF capacitors shown in this configuration are recommended for bypassing, but are not necessary for operation. Each DVDD pin should have its own 100 nF bypass capacitor, but only one bulk capacitor (10 µF to 47 µF) is needed for both DVDD pins. With this configuration, 3.3 V is the main system voltage; 1.8 V is generated at the transistor's collector, which is connected to the DVDD pins. VDRIVE is connected to the base of the PNP transistor. If the regulator is not used in the design, VDRIVE can be tied to ground.

Two specifications must be considered when choosing a regulator transistor: The transistor's current amplification factor (h_{FE} or beta) should be at least 100, and the transistor's collector must be able to dissipate the heat generated when regulating from 3.3 V to 1.8 V. The maximum digital current drawn from the ADAU1702 is 60 mA. The equation to determine the minimum power dissipation of the transistor is as follows:

$$(3.3\ V - 1.8\ V) \times 60\ mA = 90\ mW$$

There are many transistors, such as the FZT953 from Zetex Semiconductors, with these specifications available in small SOT-23 or SOT-223 packages.

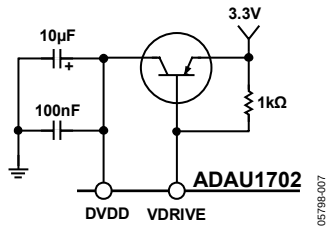

Figure 16. Voltage Regulator Configuration

AUDIO ADCS

The ADAU1702 has two Σ-Δ ADCs. The signal-to-noise ratio (SNR) of the ADCs is 100 dB and the THD + N is −83 dB.

The stereo audio ADCs are current input; therefore, a voltage-to-current resistor is required on the inputs. This means that the voltage level of the input signals to the system can be set to any level; only the input resistors need to be scaled to provide the proper full-scale current input. The ADC0 and ADC1 input pins, as well as ADC_RES, have an internal 2 kΩ resistor for ESD protection. The voltage seen directly on the ADC input pins is the 1.5 V common mode.

The external resistor connected to ADC_RES sets the full-scale current input of the ADCs. The full range of the ADC inputs is 100 μA_{rms} with an external 18 kΩ resistor on ADC_RES (20 kΩ total, because it is in series with the internal 2 kΩ). The only reason to change the ADC_RES resistor is if a sampling rate other than 48 kHz is used.

The voltage-to-current resistors connected to ADC0/ADC1 set the full-scale voltage input of the ADCs. With a full-scale current input of 100 μA_{rms}, a 2.0 V_{rms} signal with an external 18 kΩ resistor (in series with the 2 kΩ internal resistor) results in an input using the full range of the ADC. The matching of these resistors to the ADC_RES resistor is important to the operation of the ADCs. For these three resistors, a 1% tolerance is recommended.

Either the ADC0 and/or ADC1 input pins can be left unconnected if that channel of the ADC is unused.

These calculations of resistor values assume a 48 kHz sample rate. The recommended input and current setting resistors scale linearly with the sample rate because the ADCs have a switched-capacitor input. The total value (2 kΩ internal plus external resistor) of the ADC_RES resistor with sample rate f_{S_NEW} can be calculated as follows:

$$R_{total} = 20\ \text{k}\Omega \times \frac{48{,}000}{f_{S\_NEW}}$$

The values of the resistors (internal plus external) in series with the ADC0 and ADC1 pins can be calculated as follows:

$$R_{Input\ Total} = (RMS\ Input\ Voltage) \times 10\ \text{k}\Omega \times \frac{48{,}000}{f_{S\_NEW}}$$

Table 13 lists the external and total resistor values for common signal input levels at a 48 kHz sampling rate. A full-scale rms input voltage of 0.9 V is shown in the table because a full-scale signal at this input level is equal to a full-scale output on the DACs.

Table 13. ADC Input Resistor Values

Full-Scale RMS Input Voltage (V)	ADC_RES Value (kΩ)	ADC0/ADC1 Resistor Value (kΩ)	Total ADC0/ADC1 Input Resistance (External + Internal) (kΩ)
0.9	18	7	9
1.0	18	8	10
2.0	18	18	20

Figure 17 shows a typical configuration of the ADC inputs for a 2.0 V_{rms} input signal for a f_S of 48 kHz. The 47 µF capacitors are used to ac-couple the signals so that the inputs are biased at 1.5 V.

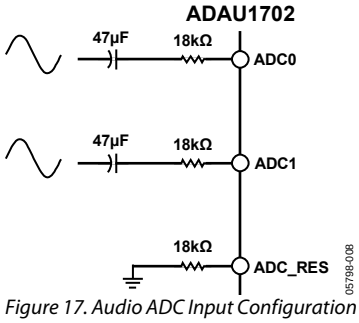

Figure 17. Audio ADC Input Configuration

AUDIO DACS

The ADAU1702 includes four Σ-Δ DACs. The SNR of the DAC is 104 dB and the THD + N is −90 dB. A full-scale output on the DACs is 0.9 V_{rms} (2.5 V_{p-p}).

The DACs are in an inverting configuration. If a signal inversion from input to output is undesirable, it can be reversed by using either an inverting configuration for the output filter or by simply inverting the signal in the SigmaDSP program flow.

The DAC outputs can be filtered with either an active or a passive reconstruction filter. A single-pole, passive low-pass filter with a 50 kHz corner frequency, as shown in Figure 18, is sufficient to filter the DAC out-of-band noise, although an active filter may provide better audio performance. Figure 19 shows a triple-pole, active, low-pass filter that provides a steeper roll-off and better stop-band attenuation than the passive filter. In this configuration, the V+ and V− pins of the AD8606 op amp are set to VDD and ground, respectively.

To properly initialize the DACs, Bits DS [1:0] in the DAC setup register (Address 2087) should be set to 01.

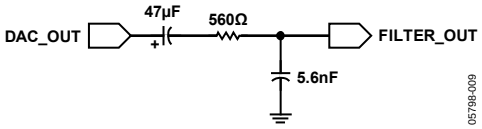

Figure 18. Passive DAC Output Filter

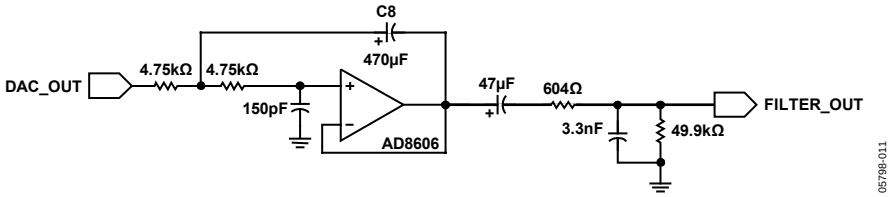

Figure 19. Active DAC Output Filter

CONTROL PORTS

The ADAU1702 can operate in one of three control modes:

- I²C control
- SPI control
- Self-boot (no external controller)

The ADAU1702 has both a 4-wire SPI control port and a 2-wire I²C bus control port. Each can be used to set the RAMs and registers. When the SELFBOOT pin is low at power-up, the part defaults to I²C mode but can be put into SPI control mode by pulling the CLATCH/WP pin low three times. When the SELFBOOT pin is set high at power-up, the ADAU1702 loads its program, parameters, and register settings from an external EEPROM on startup.

The control port is capable of full read/write operation for all addressable memory and registers. Most signal processing parameters are controlled by writing new values to the parameter RAM using the control port. Other functions, such as mute and input/output mode control, are programmed by writing to the registers.

All addresses may be accessed in both a single-address mode or a burst mode. The first byte (Byte 0) of a control port write contains the 7-bit chip address plus the R/$\overline{W}$ bit. The next two bytes (Byte 1 and Byte 2) together form the subaddress of the memory or register location within the ADAU1702. This subaddress must be two bytes because the memory locations within the ADAU1702 are directly addressable and their sizes exceed the range of single-byte addressing. All subsequent bytes (starting with Byte 3) contain the data, such as control port data, program data, or parameter data. The number of bytes per word depends on the type of data that is being written. The exact formats for specific types of writes are shown in Table 22 to Table 31.

The ADAU1702 has several mechanisms for updating signal processing parameters in real time without causing pops or clicks. If large blocks of data need to be downloaded, the output of the DSP core can be halted (using the CR bit in the DSP core control register (Address 2076)), new data can be loaded, and then the device can be restarted. This is typically done during the booting sequence at start-up or when loading a new program into RAM. In cases where only a few parameters need to be changed, they can be loaded without halting the program. To avoid unwanted side effects while loading parameters on the fly, the SigmaDSP provides the safeload registers. The safeload registers can be used to buffer a full set of parameters (for example, the five coefficients of a biquad) and then transfer these parameters into the active program within one audio frame. The safeload mode uses internal logic to prevent contention between the DSP core and the control port.

The control port pins are multifunctional, depending on the mode in which the part is operating. Table 14 details these multiple functions.

Table 14. Control Port Pins and SELFBOOT Pin Functions

Pin	I²C Mode	SPI Mode	Self-Boot
SCL/CCLK	SCL—input	CCLK—input	SCL—output
SDA/COUT	SDA—open-collector output	COUT—output	SDA—open-collector output
ADDR1/CDATA/WB	ADDR1—input	CDATA—input	WB—writeback trigger
CLATCH/WP	Unused input—tie to ground or VDD	CLATCH—input	WP—EEPROM write protect, open-collector output
ADDR0	ADDR0—input	ADDR0—input	Unused input—tie to ground or VDD

I²C PORT

The ADAU1702 supports a 2-wire serial (I²C-compatible) microprocessor bus driving multiple peripherals. Two pins, serial data (SDA) and serial clock (SCL), carry information between the ADAU1702 and the system I²C master controller. In I²C mode, the ADAU1702 is always a slave on the bus, meaning it cannot initiate a data transfer. Each slave device is recognized by a unique address. The address byte format is shown in Table 15. The ADAU1702 slave addresses are set with the ADDR0 and ADDR1 pins. The address resides in the first seven bits of the I²C write. The LSB of this byte sets either a read or write operation. Logic Level 1 corresponds to a read operation, and Logic Level 0 corresponds to a write operation. Bit 5 and Bit 6 of the address are set by tying the ADDRx pins of the ADAU1702 to Logic Level 0 or Logic Level 1. The full byte addresses, including the pin settings and read/$\overline{\text{write}}$ bit, are shown in Table 16.

Burst mode addressing, where the subaddresses are automatically incremented at word boundaries, can be used for writing large amounts of data to contiguous memory locations. This increment happens automatically after a single-word write unless a stop condition is encountered. The registers and RAMs in the ADAU1702 range in width from one to five bytes, so the auto-increment feature knows the mapping between subaddresses and the word length of the destination register (or memory location). A data transfer is always terminated by a stop condition.

Both SDA and SCL should have 2.2 kΩ pull-up resistors on the lines connected to them. The voltage on these signal lines should not be more than IOVDD (3.3 V).

Table 15. ADAU1702 I²C Address Byte Format

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0	1	1	0	1	ADDR1	ADDR0	R/$\overline{\text{W}}$

Table 16. ADAU1702 I²C Addresses

ADDR1	ADDR0	Read/$\overline{\text{Write}}$	Slave Address
0	0	0	0x68
0	0	1	0x69
0	1	0	0x6A
0	1	1	0x6B
1	0	0	0x6C
1	0	1	0x6D
1	1	0	0x6E
1	1	1	0x6F

Addressing

Initially, each device on the I²C bus is in an idle state and monitoring the SDA and SCL lines for a start condition and the proper address. The I²C master initiates a data transfer by establishing a start condition, defined by a high-to-low transition on SDA while SCL remains high. This indicates that an address/data stream follows. All devices on the bus respond to the start condition and shift the next eight bits (the 7-bit address plus the R/$\overline{\text{W}}$ bit) MSB first. The device that recognizes the transmitted address responds by pulling the data line low during the ninth clock pulse. This ninth bit is known as an acknowledge bit. All other devices withdraw from the bus at this point and return to the idle condition. The R/$\overline{\text{W}}$ bit determines the direction of the data. A Logic 0 on the LSB of the first byte means the master will write information to the peripheral, whereas a Logic 1 means the master will read information from the peripheral after writing the subaddress and repeating the start address. A data transfer takes place until a stop condition is encountered. A stop condition occurs when SDA transitions from low to high while SCL is held high. Figure 20 shows the timing of an I²C write, and Figure 21 shows an I²C read.

Stop and start conditions can be detected at any stage during the data transfer. If these conditions are asserted out of sequence with normal read and write operations, the ADAU1702 immediately jumps to the idle condition. During a given SCL high period, the user should only issue one start condition, one stop condition, or a single stop condition followed by a single start condition. If an invalid subaddress is issued by the user, the ADAU1702 does not issue an acknowledge and returns to the idle condition. If the user exceeds the highest subaddress while in auto-increment mode, one of two actions is taken. In read mode, the ADAU1702 outputs the highest subaddress register contents until the master device issues a no acknowledge, indicating the end of a read. A no-acknowledge condition is where the SDA line is not pulled low on the ninth clock pulse on SCL. If the highest subaddress location is reached while in write mode, the data for the invalid byte is not loaded into any subaddress register, a no acknowledge is issued by the ADAU1702, and the part returns to the idle condition.

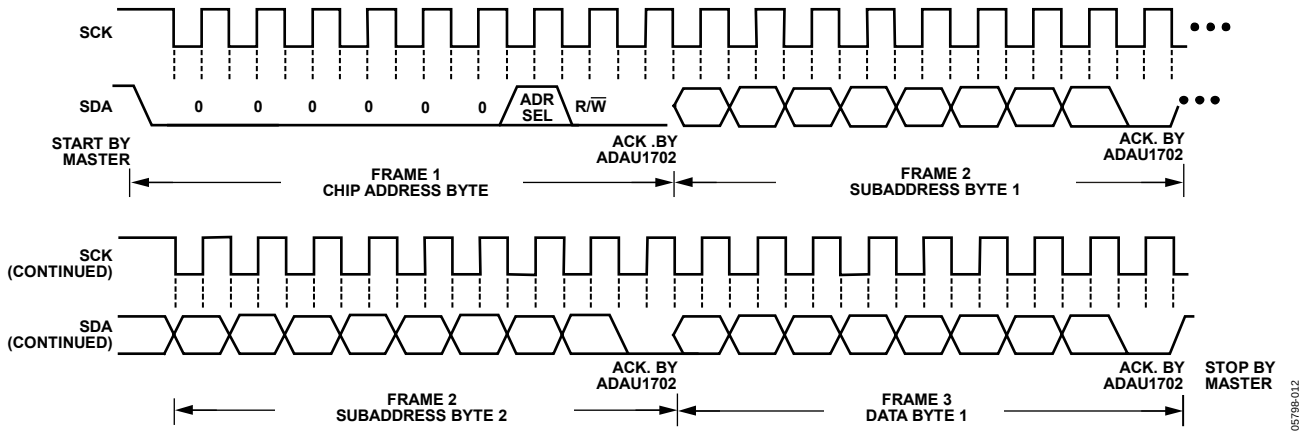

Figure 20. I²C Write to ADAU1702 Clocking

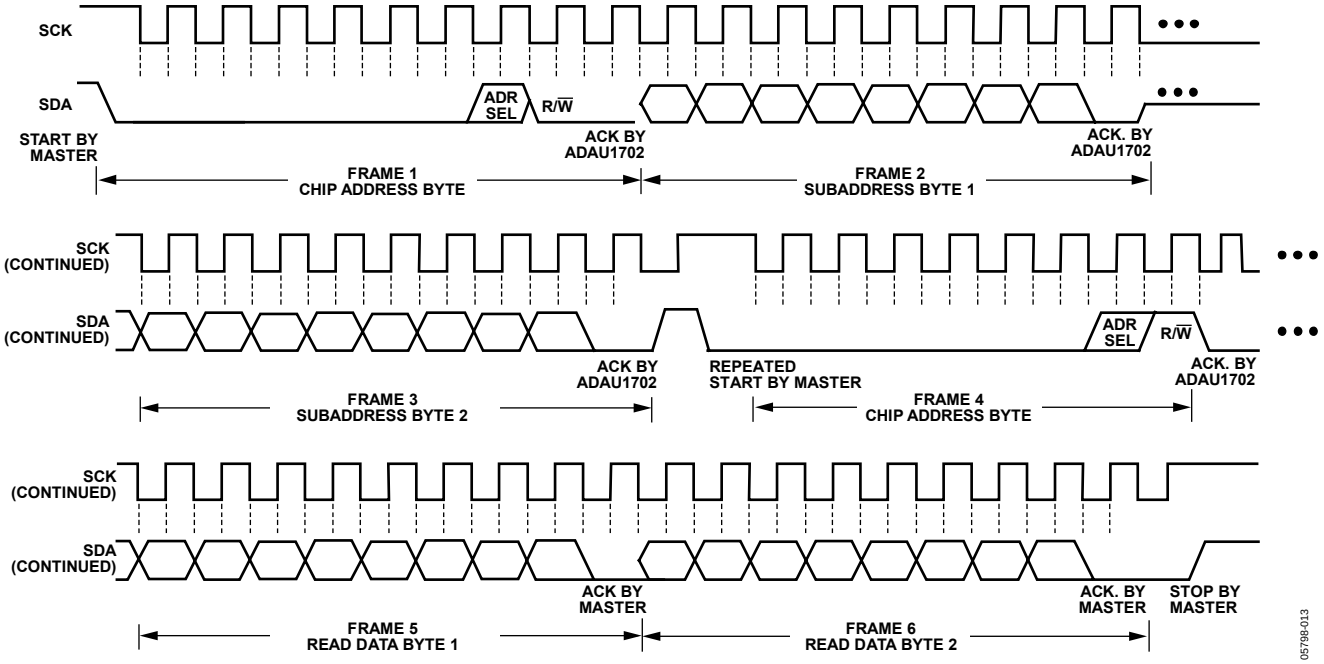

Figure 21. I²C Read from ADAU1702 Clocking

I²C Read and Write Operations

Figure 22 shows the timing of a single-word write operation. Every ninth clock, the ADAU1702 issues an acknowledge by pulling SDA low.

Figure 23 shows the timing of a burst mode write sequence. This figure shows an example where the target destination registers are two bytes. The ADAU1702 knows to increment its subaddress register every two bytes because the requested subaddress corresponds to a register or memory area with a 2-byte word length.

The timing of a single-word read operation is shown in Figure 24. Note that the first R/$\overline{W}$ bit is 0, indicating a write operation. This is because the subaddress still needs to be written to set up the internal address. After the ADAU1702 acknowledges the receipt of the subaddress, the master must issue a repeated start command followed by the chip address byte with the R/$\overline{W}$ set to 1 (read). This causes the ADAU1702 SDA to reverse and begin driving data back to the master. The master then responds every ninth pulse with an acknowledge pulse to the ADAU1702.

Figure 25 shows the timing of a burst mode read sequence. This figure shows an example where the target read registers are two bytes. The ADAU1702 increments its subaddress every two bytes because the requested subaddress corresponds to a register or memory area with word lengths of two bytes. Other address ranges may have a variety of word lengths ranging from one to five bytes. The ADAU1702 always decodes the subaddress and sets the auto-increment circuit so that the address increments after the appropriate number of bytes.

Figure 22 to Figure 25 use the following abbreviations:
S = start bit
P = stop bit
AM = acknowledge by master
AS = acknowledge by slave

S	Chip address, R/$\overline{W}$ = 0	AS	Subaddress high	AS	Subaddress low	AS	Data Byte 1	AS	Data Byte 2	…	AS	Data Byte N	P

Figure 22. Single-Word I²C Write Format

S	Chip address, R/$\overline{W}$ = 0	AS	Subaddress high	AS	Subaddress low	AS	Data-Word 1, Byte 1	AS	Data-Word 1, Byte 2	AS	Data-Word 2, Byte 1	AS	Data-Word 2, Byte 2	AS	…	P

Figure 23. Burst Mode I²C Write Format

S	Chip address, R/$\overline{W}$ = 0	AS	Subaddress high	AS	Subaddress low	AS	S	Chip address, R/$\overline{W}$ = 1	AS	Data Byte 1	AM	Data Byte 2	…	AM	Data Byte N	P

Figure 24. Single-Word I²C Read Format

S	Chip address, R/$\overline{W}$ = 0	AS	Subaddress high	AS	Subaddress low	AS	S	Chip address, R/$\overline{W}$ = 1	AS	Data-Word 1, Byte 1	AM	Data-Word 1, Byte 2	AM	…	P

Figure 25. Burst Mode I²C Read Format

SPI PORT

By default, the ADAU1702 is in I²C mode, but can be put into SPI control mode by pulling CLATCH/WP low three times. The SPI port uses a 4-wire interface, consisting of CLATCH, CCLK, CDATA, and COUT signals and is always a slave port. The CLATCH signal should go low at the beginning of a transaction and high at the end of a transaction. The CCLK signal latches CDATA on a low-to-high transition. COUT data is shifted out of the ADAU1702 on the falling edge of CCLK and should be clocked into a receiving device, such as a microcontroller, on the CCLK rising edge. The CDATA signal carries the serial input data, and the COUT signal is the serial output data. The COUT signal remains three-stated until a read operation is requested. This allows other SPI-compatible peripherals to share the same readback line. All SPI transactions have the same basic format shown in Table 18. A timing diagram is shown in Figure 4. All data should be written MSB first. The ADAU1702 cannot be taken out of SPI mode without a full reset.

Chip Address R/$\overline{W}$

The first byte of an SPI transaction includes the 7-bit chip address and a R/$\overline{W}$ bit. The chip address is set by the ADDR0 pin. This allows two ADAU1702s to share a CLATCH signal, yet still operate independently. When ADDR0 is low, the chip address is 0000000; when it is high, the address is 0000001 (see Table 17). The LSB of this first byte determines whether the SPI transaction is a read (Logic Level 1) or a write (Logic Level 0).

Table 17. ADAU1702 SPI Address Byte Format

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0	0	0	0	0	0	ADDR0	R/$\overline{W}$

Subaddress

The 12-bit subaddress word is decoded into a location in one of the memories or registers. This subaddress is the location of the appropriate RAM location or register. The MSBs of the subaddress are zero-padded to bring the word to a full 2-byte length.

Data Bytes

The number of data bytes varies according to the register or memory being accessed. During a burst mode write, an initial subaddress is written followed by a continuous sequence of data for consecutive memory/register locations. The detailed data format for continuous mode operation is shown in Table 23 and Table 25 in the Read/Write Data Formats section.

A sample timing diagram for a single-write SPI operation to the parameter RAM is shown in Figure 26. A sample timing diagram of a single-read SPI operation is shown in Figure 27. The COUT pin goes from three-state to being driven at the beginning of Byte 3. In this example, Byte 0 to Byte 2 contain the addresses and R/$\overline{W}$ bit and subsequent bytes carry the data.

Table 18. Generic Control Word Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4[1]
chip_adr [6:0], R/$\overline{W}$	0000, subadr [11:8]	subadr [7:0]	data	data

[1] Continues to end of data.

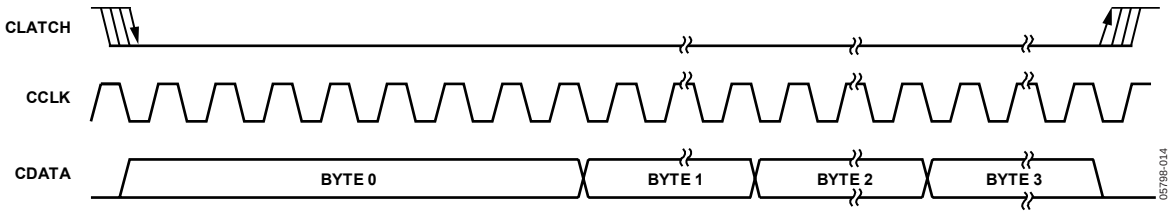

Figure 26. SPI Write to ADAU1702 Clocking (Single-Write Mode)

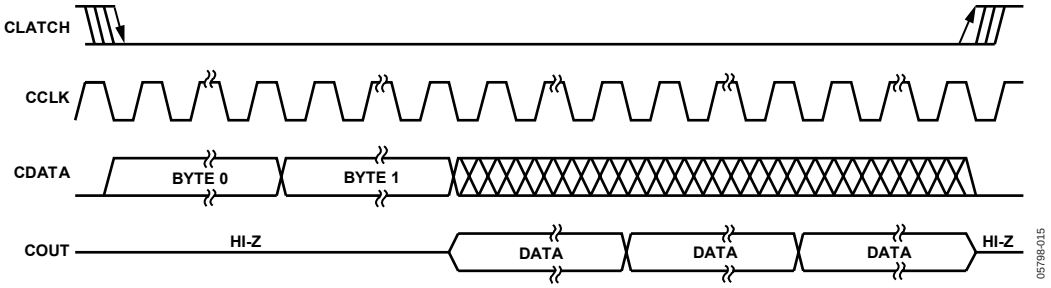

Figure 27. SPI Read from ADAU1702 Clocking (Single-Read Mode)

SELF-BOOT

On power-up, the ADAU1702 can load a program and a set of parameters that have been saved in an external EEPROM. Combined with the auxiliary ADC and the multipurpose pins, this eliminates the need for a microcontroller in the system. The self-booting is accomplished by the ADAU1702 acting as a master on the I²C bus on start-up, which occurs when the SELFBOOT pin is set high. The ADAU1702 cannot self-boot in SPI mode.

The maximum necessary EEPROM size for program and parameters is 6688 bytes, or just over 6.5 kB. This does not include register settings or overhead bytes, but such factors do not add a significant number of bytes. This much memory is only needed if the program RAM (512 × five bytes), parameter RAM (1024 × four bytes), and interface registers (8 × four bytes) are completely full. An 8 kB EEPROM has sufficient memory for this application.

A self-boot operation is triggered on the rising edge of $\overline{\text{RESET}}$ when the SELFBOOT and WP pins are set high. The ADAU1702 reads the program, parameters, and register settings from the EEPROM. After the ADAU1702 finishes self-booting, additional messages can be sent to the ADAU1702 on the I²C bus, although this typically is not necessary in a self-booting application. The I²C device address is 0x68 for a write and 0x69 for a read in this mode. The ADDRx pins have different functions when the chip is in this mode, so the settings on them are ignored.

The ADAU1702 does not self-boot if WP is set low. Holding this pin low allows the EEPROM to be programmed in-circuit. The WP pin is pulled low (it typically has a resistor pull-up) to enable writes to the EEPROM, but this in turn disables the self-boot function until the WP pin is returned high.

The ADAU1702 is a master on the I²C bus during self-boot and writeback. Although it is uncommon for an application using self-boot to also have a microcontroller connected to the control lines, care should be taken that no other device tries to write to the I²C bus during self-boot or writeback. The ADAU1702 generates SCL at 8 × f_S; therefore, for a f_S of 48 kHz, SCL runs at 384 kHz. SCL has a duty cycle of 3/8 in accordance with the I²C specification.

The ADAU1702 reads from EEPROM Chip Address 0xA1. The LSBs of the addresses of some EEPROMs are pin configurable; in most cases, these pins should be tied low to set this address.

EEPROM Format

The EEPROM data contains a sequence of messages. Each discrete message is one of the seven types defined in Table 19. Each message consists of a sequence of one or more bytes. The first byte identifies the message type. Bytes are written MSB first. Most messages are block write (0x01) types, which are used for writing to the ADAU1702 program RAM, parameter RAM, and control registers.

The body of the message following the message type should start with a 0x00 byte—this is the chip address. As with all other control port transactions, following the chip address is a 2-byte register/memory address field.

Table 20 shows an example of what should be stored in the EEPROM, starting with EEPROM Address 0. In this example, the interface registers are first set to control port write mode (Line 1), which is followed by 18 no-operation (no-op) bytes (Line 2 to Line 4) so that the interface register data appears on Page 2 of the EEPROM. Next, follows the write header (Line 4) and then 32 bytes of interface register data (Line 5 to Line 8). Finally, the program RAM data, starting at ADAU1702 Address 0x04, 0x00 is written (Line 9 to Line 11). In this example, the program length is 70 words, or 350 bytes, so 332 more bytes are included in the EEPROM but are not shown in Table 20.

Writeback

A writeback occurs when the WB pin is triggered and data is written to the EEPROM from the ADAU1702. This function is typically used to save the volume setting and other parameter settings to the EEPROM just before power is removed from the system. A rising edge on the WB pin triggers a writeback when the device is in self-boot mode, unless a message to set the WB to the falling edge sensitive (0x05) is contained in the self-boot message sequence. Only one writeback takes place unless a message to set multiple writebacks (0x04) is contained in the self-boot message sequence. The WP pin is pulled low when a writeback is triggered to allow writing to the EEPROM.

The ADAU1702 is only capable of writing back the contents of the interface registers to the EEPROM. These registers are usually set by the DSP program, but can also be written to directly after setting Bit 6 of the core control register. The parameter settings that should be saved are configured in SigmaStudio.

The writeback function writes data from the ADAU1702 interface registers to the second page of the self-boot EEPROM, Address 32 to Address 63. Starting at EEPROM Address 26 (so that the interface register data begins at Address 32), the EEPROM should be programmed with six bytes—the message byte (0x01), two length bytes, the chip address (0x00), and the 2-byte subaddress for the interface registers (0x08, 0x00). There must be a message to the DSP core control register to enable writing to the interface registers prior to the interface register data in the EEPROM. This should be stored in EEPROM Address 0. No-op messages (0x03) can be used in between messages to ensure that these conditions are met.

The ADAU1702 writes to EEPROM Chip Address 0xA0. The LSBs of the addresses of some EEPROMs are pin configurable; in most cases, these pins should be tied low to set the address to 0xA0.

The maximum number of bytes that is written back from the ADAU1702 is 35 (eight 4-byte interface registers plus three bytes of EEPROM-addressing overhead). With SCL running at 384 kHz, the writeback operation takes approximately 73 µs to complete after being triggered. Ensure that sufficient power is available to the system to allow enough time for a writeback to complete, especially if the WB signal is triggered from a falling power supply voltage.

Table 19. EEPROM Message Types

Message ID	Message Type	Following Bytes
0x00	End	None
0x01	Write	Two bytes indicating message length followed by appropriate number of data bytes
0x02	Delay	Two bytes for delay
0x03	No operation executed	None
0x04	Set multiple writeback	None
0x05	Set WB to falling edge sensitive	None
0x06	End and wait for writeback	None

Table 20. EEPROM Data Example

1	**0x01**	**0x00**	**0x05**	**0x00**	**0x08**	**0x1C**	**0x00**	**0x40**
	Write	Length		Device addr.	Core control register address		Core control register data	
2	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**
	No-op bytes							
3	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**	**0x03**
	No-op bytes							
4	**0x03**	**0x03**	**0x01**	**0x00**	**0x23**	**0x00**	**0x08**	**0x00**
	No-op bytes		Write	Length		Device addr.	Interface register address	
5	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**
	Interface register data							
6	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**
	Interface register data							
7	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**
	Interface register data							
8	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**	**0x00**
	Interface register data							
9	**0x01**	**0x01**	**0x61**	**0x00**	**0x04**	**0x00**	**0x00**	**0x00**
	Write	Length		Device addr.	Program RAM address		Program RAM data	
10	**0x00**	**0x00**	**0x01**	**0x00**	**0x00**	**0x00**	**0xE8**	**0x01**
	Program RAM data							
11	**0x00**	**0x00**	**0x00**	**0x00**	**0x01**	**0x00**	**0x08**	**0x00**
	Program RAM data (continues for 332 more bytes)							

SIGNAL PROCESSING

The ADAU1702 is designed to provide all audio signal processing functions commonly used in stereo or multichannel playback systems. The signal processing flow is designed using the SigmaStudio software, which allows graphical entry and real-time control of all signal processing functions.

Many of the signal processing functions are coded using full, 56-bit, double-precision arithmetic data. The input and output word lengths of the DSP core are 24 bits. Four extra headroom bits are used in the processor to allow internal gains of up to 24 dB without clipping. Additional gains can be achieved by initially scaling down the input signal in the DSP signal flow.

NUMERIC FORMATS

DSP systems commonly use a standard numeric format. Fractional number systems are specified by an A.B format, where A is the number of bits to the left of the decimal point and B is the number of bits to the right of the decimal point.

The ADAU1702 uses the same numeric format for both the parameter and data values. The format is as follows.

Numerical Format: 5.23

Linear range: −16.0 to (+16.0 − 1 LSB)

Examples:
1000 0000 0000 0000 0000 0000 0000 = −16.0
1110 0000 0000 0000 0000 0000 0000 = −4.0
1111 1000 0000 0000 0000 0000 0000 = −1.0
1111 1110 0000 0000 0000 0000 0000 = −0.25
1111 1111 0011 0011 0011 0011 0011 = −0.1
1111 1111 1111 1111 1111 1111 1111 = (1 LSB below 0.0)
0000 0000 0000 0000 0000 0000 0000 = 0.0
0000 0000 1100 1100 1100 1100 1101 = 0.1
0000 0010 0000 0000 0000 0000 0000 = 0.25
0000 1000 0000 0000 0000 0000 0000 = 1.0
0010 0000 0000 0000 0000 0000 0000 = 4.0
0111 1111 1111 1111 1111 1111 1111 = (16.0 − 1 LSB).

The serial port accepts up to 24 bits on the input and is sign-extended to the full 28 bits of the DSP core. This allows internal gains of up to 24 dB without internal clipping.

A digital clipper circuit is used between the output of the DSP core and the DACs or serial port outputs (see Figure 28). This clips the top four bits of the signal to produce a 24-bit output with a range of 1.0 (minus 1 LSB) to −1.0. Figure 28 shows the maximum signal levels at each point in the data flow in both binary and decibel levels.

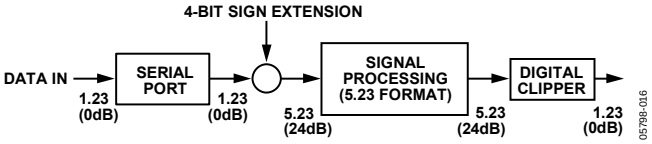

Figure 28. Numeric Precision and Clipping Structure

PROGRAMMING

On power-up, the ADAU1702 default program passes the unprocessed input signals to the outputs (shown in Figure 13), but the outputs are muted by default (see the Power-Up Sequence section). There are 512 instruction cycles per audio sample, resulting in about 25 MIPS available. The SigmaDSP runs in a stream-oriented manner, meaning that all 512 instructions are executed each sample period. The ADAU1702 can also be set up to accept double- or quad-speed inputs by reducing the number of instructions per sample that are set in the core control register.

The part can be programmed easily using SigmaStudio (Figure 29), a graphical tool provided by Analog Devices. No knowledge of writing line-level DSP code is required. More information about SigmaStudio can be found at www.analog.com.

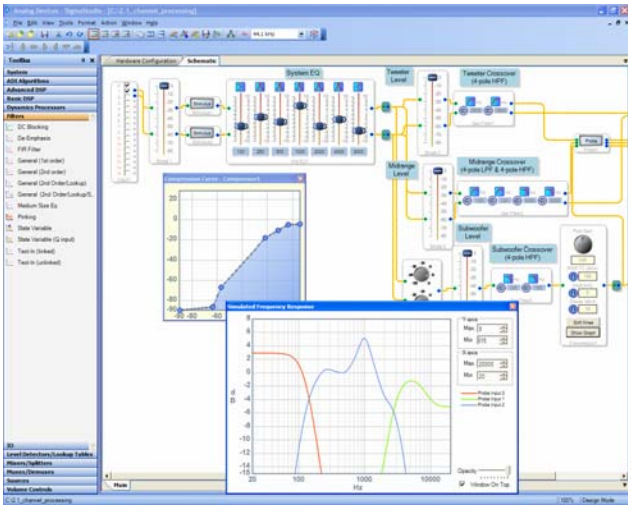

Figure 29. SigmaStudio Screen Shot

RAMS AND REGISTERS

Table 21. RAM Map and Read/Write Modes

Memory	Size	Address Range	Read	Write	Write Modes
Parameter RAM	1024 × 32	0 to 1023 (0x0000 to 0x03FF)	Yes	Yes	Direct write[1] safeload write
Program RAM	512 × 40	1024 to 1535 (0x0400 to 0x05FF)	Yes	Yes	Direct write[1]
Reserved	N/A	1536 to 2047 (0x0600 to 0x07FF) [2]	No	No	N/A

[1] Internal registers should be cleared first to avoid clicks/pops.
[2] Addresses 1536 to 2047 (0x0600 to 0x07FF) are reserved RAM locations and data can not be written to them.

ADDRESS MAPS

Table 21 shows the RAM map and Table 32 shows the ADAU1702 register map. The address space encompasses a set of registers and two RAMs: one holds signal processing parameters and one holds the program instructions. The program RAM and parameter RAM are initialized on power-up from on-board boot ROMs (see the Power-Up Sequence section).

All RAMs and registers have a default value of all 0s, except for the program RAM, which is loaded with the default program (see the Initialization section).

PARAMETER RAM

The parameter RAM is 32 bits wide and occupies Address 0 to Address 1023. Each parameter is padded with four 0s before the MSB to extend the 28-bit word to a full 4-byte width. The parameter RAM is initialized to all 0s on power-up. The data format of the parameter RAM is twos complement, 5.23. This means that the coefficients can range from +16.0 (minus 1 LSB) to −16.0, with 1.0 represented by the binary word 0000 1000 0000 0000 0000 0000 0000 or by the hexadecimal word 0x00 0x80 0x00 0x00.

The parameter RAM can be written using one of the two following methods: a direct read/write or a safeload write.

Direct Read/Write

This method allows direct access to the program RAM and parameter RAM. This mode of operation is typically used during a complete new loading of the RAM using burst mode addressing. The clear registers bit in the core control register should be set to 0 using this mode to avoid any clicks or pops in the outputs. Note that this mode can be used during live program execution, but because there is no handshaking between the core and the control port, the parameter RAM is unavailable to the DSP core during control writes, resulting in clicks and pops in the audio stream.

Safeload Write

Up to five safeload registers can be loaded with the parameter RAM address/data. The data is then transferred to the requested address when the RAM is not busy. This method can be used for dynamic updates while live program material is playing through the ADAU1702. For example, a complete update of one biquad section can occur in one audio frame while the RAM is not busy. This method is not available for writing to the program RAM or control registers.

DATA RAM

The ADAU1702 data RAM is used to store audio data words for processing. For the most part, this process is transparent to the user. The user cannot directly address this RAM space, which has a size of 0.5k words, from the control port.

Data RAM utilization should be considered when implementing blocks that require large amounts of data RAM space, such as delays. The SigmaDSP core processes delay times in one-sample increments; therefore, the total pool of delay available to the user equals 512 multiplied by the sample period. For a f_S of 48 kHz, the pool of available delay is a maximum of about 11 ms. In practice, this much data memory is not available to the user because every block in a design uses a few data memory locations for its processing. In most DSP programs, this does not significantly impact the total delay time. The SigmaStudio compiler manages the data RAM and indicates if the number of addresses needed in the design exceeds the maximum available.

READ/WRITE DATA FORMATS

The read/write formats of the control port are designed to be byte oriented. This allows easy programming of common microcontroller chips. To fit into a byte-oriented format, 0s are appended to the data fields before the MSB to extend the data-word to eight bits. For example, 28-bit words written to the parameter RAM are appended with four leading 0s to equal 32 bits (4 bytes); 40-bit words written to the program RAM are not appended with 0s because they are already a full five bytes. These zero-padded data fields are appended to a 3-byte field consisting of a 7-bit chip address, a read/write bit, and an 11-bit RAM/register address. The control port knows how many data bytes to expect based on the address given in the first three bytes.

The total number of bytes for a single-location write command can vary from four bytes (for a control register write) to eight bytes (for a program RAM write). Burst mode can be used to fill contiguous register or RAM locations. A burst mode write begins by writing the address and data of the first RAM or register location to be written. Rather than ending the control port transaction (by issuing a stop command in I^2C mode or by bringing the CLATCH signal high in SPI mode after the data-word), as would be done in a single-address write, the next data-word can be written immediately without specifying its address. The

ADAU1702 control port auto-increments the address of each write even across the boundaries of the different RAMs and registers. Table 23 and Table 25 show examples of burst mode writes.

Table 22. Parameter RAM Read/Write Format (Single Address)

Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4:6
chip_adr [6:0], $\overline{W}$/R	000000, param_adr [9:8]	param_adr [7:0]	0000, param [27:24]	param [23:0]

Table 23. Parameter RAM Block Read/Write Format (Burst Mode)

Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4:6	Bytes 7:10	Bytes 11:14
chip_adr [6:0], $\overline{W}$/R	000000, param_adr [9:8]	param_adr [7:0]	0000, param [27:24]	param [23:0]		
			<—param_adr—>		param_adr + 1	param_adr + 2

Table 24. Program RAM Read/Write Format (Single Address)

Byte 0	Byte 1	Byte 2	Bytes 3:7
chip_adr [6:0], $\overline{W}$/R	00000, prog_adr [10:8]	prog_adr [7:0]	prog [39:0]

Table 25. Program RAM Block Read/Write Format (Burst Mode)

Byte 0	Byte 1	Byte 2	Bytes 3:7	Bytes 8:12	Bytes 13:17
chip_adr [6:0], $\overline{W}$/R	00000, prog_adr [10:8]	prog_adr [7:0]	prog [39:0]		
			<—prog_adr—>	prog_adr + 1	prog_adr + 2

Table 26. Control Register Read/Write Format (Core, Serial Out 0, Serial Out 1)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
chip_adr [6:0], $\overline{W}$/R	0000, reg_adr [11:8]	reg_adr [7:0]	data [15:8]	data [7:0]

Table 27. Control Register Read/Write Format (RAM Configuration, Serial Input)

Byte 0	Byte 1	Byte 2	Byte 3
chip_adr [6:0], $\overline{W}$/R	0000, reg_adr [11:8]	reg_adr [7:0]	data [7:0]

Table 28. Data Capture Register Write Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
chip_adr [6:0], $\overline{W}$/R	0000, data_capture_adr [11:8]	data_capture_adr [7:0]	000, progCount [10:6][1]	progCount [5:0][1], regSel [1:0][2]

[1] ProgCount [10:0] is the value of the program counter where the data capture occurs (the table of values is generated by the SigmaStudio compiler).
[2] RegSel [1:0] selects one of four registers (see the 2074 to 2075 (0x081A to 0x081B)—Data Capture Registers section).

Table 29. Data Capture (Control Port Readback) Register Read Format

Byte 0	Byte 1	Byte 2	Bytes 3:5
chip_adr [6:0], $\overline{W}$/R	0000, data_capture_adr [11:8]	data_capture_adr [7:0]	data [23:0]

Table 30. Safeload Address Register Write Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
chip_adr [6:0], $\overline{W}$/R	0000, safeload_adr [11:8]	safeload_adr [7:0]	000000, param_adr [9:8]	param_adr [7:0]

Table 31. Safeload Data Register Write Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Bytes 5:7
chip_adr [6:0], $\overline{W}$/R	0000, safeload_adr [11:8]	safeload_adr [7:0]	00000000	0000, data [27:24]	data [23:0]

CONTROL REGISTER MAP

Table 32. Register Map[1]

Reg (Hex)	Reg (Dec)	No. of Bytes	Name	MSB D31 D15	D30 D14	D29 D13	D28 D12	D27 D11	D26 D10	D25 D9	D24 D8	D39 D23 D7	D38 D22 D6	D37 D21 D5	D36 D20 D4	D35 D19 D3	D34 D18 D2	D33 D17 D1	LSB D32 D16 D0	Default
0x0800	2048	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0801	2049	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0802	2050	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0803	2051	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0804	2052	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0805	2053	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0806	2054	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0807	2055	4	Interface 0 [31:16]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
			Interface 0 [15:0]	IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000
0x0808	2056	2	GPIO pin setting	0	0	0	0	MP11	MP10	MP09	MP08	MP07	MP06	MP05	MP04	MP03	MP02	MP01	MP00	0x0000
0x0809	2057	2	Auxiliary ADC Data 0	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000
0x080A	2058	2	Auxiliary ADC Data 1	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000
0x080B	2059	2	Auxiliary ADC Data 2	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000
0x080C	2060	2	Auxiliary ADC Data 3	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000
0x080D	2061	5	Reserved [39:32]									RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x00
			Reserved [31:16]	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
			Reserved [15:0]	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
0x080E	2062	5	Reserved [39:32]									RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x00
			Reserved [31:16]	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
			Reserved [15:0]	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
0x080F	2063	5	Reserved [39:32]									RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x00
			Reserved [31:16]	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
			Reserved [15:0]	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
0x0810	2064	5	Safeload Data 0 [39:32]									SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	0x00
			Safeload Data 0 [31:16]	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	0x0000
			Safeload Data 0 [15:0]	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	0x0000
0x0811	2065	5	Safeload Data 1 [39:32]									SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	0x00
			Safeload Data 1 [31:16]	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	0x0000
			Safeload Data 1 [15:0]	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	0x0000
0x0812	2066	5	Safeload Data 2 [39:32]									SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	0x00
			Safeload Data 2 [31:16]	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	0x0000
			Safeload Data 2 [15:0]	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	0x0000
0x0813	2067	5	Safeload Data 3 [39:32]									SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	0x00
			Safeload Data 3 [31:16]	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	0x0000
			Safeload Data 3 [15:0]	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	0x0000
0x0814	2068	5	Safeload Data 4 [39:32]									SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	0x00
			Safeload Data 4 [31:16]	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	0x0000
			Safeload Data 4 [15:0]	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	0x0000
0x0815	2069	2	Safeload Address 0	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	0x0000
0x0816	2070	2	Safeload Address 1	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	0x0000
0x0817	2071	2	Safeload Address 2	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	0x0000
0x0818	2072	2	Safeload Address 3	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	0x0000
0x0819	2073	2	Safeload Address 4	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	0x0000

Reg (Hex)	Reg (Dec)	No. of Bytes	Name	MSB D31 D15	D30 D14	D29 D13	D28 D12	D27 D11	D26 D10	D25 D9	D24 D8	D39 D23 D7	D38 D22 D6	D37 D21 D5	D36 D20 D4	D35 D19 D3	D34 D18 D2	D33 D17 D1	LSB D32 D16 D0	Default
0x081A	2074	2	Data Capture 0	0	0	0	0	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	RS01	RS00	0x0000
0x081B	2075	2	Data Capture 1	0	0	0	0	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	RS01	RS00	0x0000
0x081C	2076	2	DSP core control	RSVD	RSVD	GD1	GD0	RSVD	RSVD	RSVD	AACW	GPCW	IFCW	IST	ADM	DAM	CR	SR1	SR0	0x0000
0x081D	2077	1	Reserved									RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x00
0x081E	2078	2	Serial output control	0	0	OLRP	OBP	M/S	OBF1	OBF0	OLF1	OLF0	FST	TDM	MSB2	MSB1	MSB0	OWL1	OWL0	0x0000
0x081F	2079	1	Serial input control									0	0	0	ILP	IBP	M2	M1	M0	0x00
0x0820	2080	3	MP Pin Config. 0 [23:16]									MP53	MP52	MP51	MP50	MP43	MP42	MP41	MP40	0x00
			MP Pin Config. 0 [15:0]	MP33	MP32	MP31	MP30	MP23	MP22	MP21	MP20	MP13	MP12	MP11	MP10	MP03	MP02	MP01	MP00	0x0000
0x0821	2081	3	MP Pin Config. 1 [23:16]									MP113	MP112	MP111	MP110	MP103	MP102	MP101	MP100	0x00
			MP Pin Config. 1 [15:0]	MP93	MP92	MP91	MP90	MP83	MP82	MP81	MP80	MP73	MP72	MP71	MP70	MP63	MP62	MP61	MP60	0x0000
0x0822	2082	2	Auxiliary ADC and Power Control	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	FIL1	FIL0	AAPD	VBPD	VRPD	RSVD	D0PD	D1PD	D2PD	D3PD	0x0000
0x0823	2083	2	Reserved	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
0x0824	2084	2	Auxiliary ADC Enable	AAEN	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
0x0825	2085	2	Reserved	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000
0x0826	2086	2	Oscillator Power-Down	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	OPD	RSVD	RSVD	0x0000
0x0827	2087	2	DAC Setup	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	DS1	DS0	0x0000

[1] Shading indicates that registers do not fill these locations, so control bits do not exist in these locations.

CONTROL REGISTER DETAILS

2048 TO 2055 (0X0800 TO 0X0807)—INTERFACE REGISTERS

The interface registers are used in self-boot mode to save parameters that need to be written to the external EEPROM. The ADAU1702 then recalls these parameters from the EEPROM after the next reset or power-up. Therefore, system parameters such as volume and EQ settings can be saved during power-down and recalled the next time the system is turned on.

There are eight 32-bit interface registers, which allow eight 28-bit (plus zero-padding) parameters to be saved. The parameters to be saved in these registers are selected in the graphical programming tools. These registers are updated with their corresponding parameter RAM data once per sample period. An edge, which can be set to be either rising or falling, triggers the ADAU1702 to write the current contents of the interface registers to the EEPROM. See the Self-Boot section for details.

The user can write directly to the interface registers after the interface registers control port write mode (IFCW) in the DSP core control register has been set. In this mode, the data in the registers is written from the control port, not from the DSP core.

Table 33.

D31 D15	D30 D14	D29 D13	D28 D12	D27 D11	D26 D10	D25 D9	D24 D8	D23 D7	D22 D6	D21 D5	D20 D4	D19 D3	D18 D2	D17 D1	D16 D0	Default
0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	0x0000
IF15	IF14	IF13	IF12	IF11	IF10	IF09	IF08	IF07	IF06	IF05	IF04	IF03	IF02	IF01	IF00	0x0000

Table 34.

Bit Name	Description
IF [27:0]	Interface register 28-bit parameter

2056 (0x808)—GPIO PIN SETTING REGISTER

This register allows the user to set the GPIO pins through the control port. High or low settings can be directly written to or read from this register after setting the GPIO pin setting register control port write mode (GPCW) in the core control register. This register is updated once every LRCLK frame ($1/f_S$).

Table 35.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
0	0	0	0	MP11	MP10	MP09	MP08	MP07	MP06	MP05	MP04	MP03	MP02	MP01	MP00	0x0000

Table 36.

Bit Name	Description
MP [11:0]	Setting of multipurpose pin when controlled through SPI or I^2C

2057 TO 2060 (0x809 TO 0x80C)—AUXILIARY ADC DATA REGISTERS

These registers hold the data generated by the 4-channel auxiliary ADC. The ADCs have eight bits of precision and can be extended to 12 bits if filtering is selected in Bits FIL [1:0] of the auxiliary ADC and power control register. The SigmaDSP program reads this data as a 1.11 format data-word with a range of 0 to 1.0. This data-word is mapped to the 5.23 format parameter word with the four MSBs and 12 LSBs set to 0. A full-scale code of 255 results in a value of 1.0. These registers can be written to directly if the auxiliary ADC data registers control port write mode (AACW) bit is set in the DSP core control register.

Table 37.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000

Table 38.

Bit Name	Description
AA [11:0]	Auxiliary ADC output data, MSB first

2064 TO 2068 (0x0810 TO 0x814)—SAFELOAD DATA REGISTERS

Many applications require real-time microcontroller control of signal processing parameters, such as filter coefficients, mixer gains, multichannel virtualizing parameters, or dynamics processing curves. When controlling a biquad filter, for example, all of the parameters must be updated at the same time. Doing so prevents the filter from executing with a mix of old and new coefficients for one or two audio frames, thus avoiding temporary instability and transients that may take a long time to decay. To accomplish this, the ADAU1702 uses safeload data registers to simultaneously load a set of five 28-bit values to the desired parameter RAM address. Five registers are used because a biquad filter uses five coefficients and, as previously mentioned, it is desirable to do a complete update in one transaction.

The first step in performing a safeload operation is writing the parameter address to one of the safeload address registers (2069 to 2073). The 10-bit data-word to be written is the address in parameter RAM to which the safeload is being performed. After this address is written, the 28-bit data-word can be written to the corresponding safeload data register (2064 to 2068).

The data formats for these writes are detailed in Table 30 and Table 31. Table 39 shows how each of the five address registers maps to its corresponding data register.

After the address and data registers are loaded, the initiate safeload transfer bit in the core control register should be set to initiate the loading into RAM. Each of the five safeload registers takes one of the 512 core instructions to load into the parameter RAM. The total program lengths should therefore be limited to 507 cycles (512 minus 5) to ensure that the SigmaDSP core always has at least five cycles available. The safeload is guaranteed to occur within one LRCLK period (21 µs for a f_S of 48 kHz) of the initiate safeload transfer bit being set.

The safeload logic automatically sends data to be loaded into RAM from only those safeload registers that have been written to since the last safeload operation. For example, if two parameters are to be updated in the RAM, only two of the five safeload registers must be written. When the initiate safeload transfer bit is asserted, only data from those two registers are sent to the RAM; the other three registers are not sent to the RAM and may hold old or invalid data.

Table 39. Safeload Address and Data Register Mapping

Safeload Register	Safeload Address Register	Safeload Data Register
0	2069	2064
1	2070	2065
2	2071	2066
3	2072	2067
4	2073	2068

Table 40.

D31 D15	D30 D14	D29 D13	D28 D12	D27 D11	D26 D10	D25 D9	D24 D8	D39 D23 D7	D38 D22 D6	D37 D21 D5	D36 D20 D4	D35 D19 D3	D34 D18 D2	D33 D17 D1	D32 D16 D0	Default
								SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	0x00
SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	0x0000
SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	0x0000

Table 41.

Bit Name	Description
SD [39:0] Safeload Data	Data (program, parameters, register contents) to be loaded into the RAMs or registers

2069 TO 2073 (0x0815 TO 0x819) SAFELOAD ADDRESS REGISTERS

Table 42.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	0x0000

Table 43.

Bit Name	Description
SA [11:0] Safeload Address	Address of data that is to be loaded into the RAMs or registers

2074 TO 2075 (0X081A TO 0X081B)—DATA CAPTURE REGISTERS

The ADAU1702 data capture feature allows the data at any node in the signal processing flow to be sent to one of two readable registers. This feature is useful for monitoring and displaying information about internal signal levels or compressor/limiter activity.

For each of the data capture registers, a capture count and a register select must be set. The capture count is a number between 0 and 1023 that corresponds to the program step number where the capture is to occur. The register select field programs one of four registers in the DSP core that transfers this information to the data capture register when the program counter reaches this step.

The captured data is in 5.19, twos complement data format, which comes from the internal 5.23 data-word with the four LSBs truncated.

The data that must be written to set up the data capture is a concatenation of the 10-bit program count index with the 2-bit register select field. The capture count and register select values that correspond to the desired point to be monitored in the signal processing flow can be found in a file output from the program compiler. The capture registers can be accessed by reading from Location 2074 and Location 2075. The format for reading and writing to the data capture registers is shown in Table 28 and Table 29.

Table 44.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
0	0	0	0	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	RS01	RS00	0x0000

Table 45.

Bit Name	Description	
PC [9:0]	10-bit program counter address	
RS [1:0]	Select the register to be transferred to the data capture output	
	RS [1:0]	**Register**
	00	Multiplier X input (Mult_X_input)
	01	Multiplier Y input (Mult_Y_input)
	10	Multiplier-accumulator output (MAC_out)
	11	Accumulator feedback (Accum_fback)

2076 (0x081C)—DSP CORE CONTROL REGISTER

Table 46.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
RSVD	RSVD	GD1	GD0	RSVD	RSVD	RSVD	AACW	GPCW	IFCW	IST	ADM	DAM	CR	SR1	SR0	0x0000

Table 47. DSP Core Control Register

Bit Name	Description	
GD [1:0] GPIO Debounce Control	Sets debounce time of multipurpose pins that are set as GPIO inputs.	
	GD [1:0]	**Time (ms)**
	00	20
	01	40
	10	10
	11	5
AACW Auxiliary ADC Data Registers Control Port Write Mode	Setting this bit allows data to be written directly to the auxiliary ADC data registers (2057 to 2060) from the control port. When this bit is set, the auxiliary ADC data registers ignores the settings on the multipurpose pins.	
GPCW GPIO Pin Setting Register Control Port Write Mode	When this bit is set, the GPIO pin setting register (2056) can be written to directly from the control port and this register ignores the input settings on the multipurpose pins.	
IFCW Interface Registers Control Port Write Mode	When this bit is set, data can be written directly to the interface registers (2048 to 2055) from the control port. In that state, the interface registers are not written from the SigmaDSP program.	
IST Initiate Safeload Transfer	Setting this bit to 1 initiates a safeload transfer to the parameter RAM. This bit is automatically cleared when the operation is complete. There are five safeload register pairs (address/data); only those registers that have been written since the last safeload event are transferred to the parameter RAM.	
ADM Mute ADCs	This bit mutes the output of the ADCs. The bit defaults to 0 and is active low; therefore, it must be set to 1 to transmit audio signals from the ADCs.	
DAM Mute DACs	This bit mutes the output of the DACs. The bit defaults to 0 and is active low; therefore, it must be set to 1 to transmit audio signals from the DACs.	
CR Clear Internal Registers to 0	This bit defaults to 0 and is active low. It must be set to 1 for a signal to pass through the SigmaDSP core.	
SR [1:0] Sample Rate	These bits set the number of DSP instructions for every sample and the sample rate at which the ADAU1702 operates. At the default setting of 1×, there are 512 instructions per audio sample. This setting should be used with sample rates such as 48 kHz and 44.1 kHz. At the 2× setting, the number of instructions per frame is halved to 256 and the ADCs and DACs nominally run at a 96 kHz sample rate. At the 4× setting, there are 128 instructions per cycle and the converters run at a 192 kHz sample rate.	
	SR [1:0]	**Setting**
	00	1× (512 instructions)
	01	2× (256 instructions)
	10	4× (128 instructions)
	11	Reserved

2078 (0x081E)—SERIAL OUTPUT CONTROL REGISTER

Table 48.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
0	0	OLRP	OBP	M/S	OBF1	OBF0	OLF1	OLF0	FST	TDM	MSB2	MSB1	MSB0	OWL1	OWL0	0x0000

Table 49.

Bit Name	Description	
OLRP OUTPUT_LRCLK Polarity	When this bit is set to 0, the left-channel data is clocked when OUTPUT_LRCLK is low and the right-channel data is clocked when OUTPUT_LRCLK is high. When this bit is set to 1, the right-channel data is clocked when OUTPUT_LRCLK is low and the left-channel data is clocked when OUTPUT_LRCLK is high.	
OBP OUTPUT_BCLK Polarity	This bit controls on which edge of the bit clock the output data is clocked. Data changes on the falling edge of OUTPUT_BCLK when this bit is set to 0 and on the rising edge when this bit is set to 1.	
M/S Master/Slave	This bit sets whether the output port is a clock master or slave. The default setting is slave; on power-up, the OUTPUT_BCLK and OUTPUT_LRCLK pins are set as inputs until this bit is set to 1, at which time they become clock outputs.	
OBF [1:0] OUTPUT_BCLK Freq (Master Mode Only)	When the output port is being used as a clock master, these bits set the frequency of the output bit clock, which is divided down from an internal 1024 × f_S clock (49.152 MHz for a f_S of 48 kHz).	
	OBF [1:0]	**Setting**
	00	Internal clock/16
	01	Internal clock/8
	10	Internal clock/4
	11	Internal clock/2
OLF [1:0] OUTPUT_LRCLK Freq (Master Mode Only)	When the output port is used as a clock master, these bits set the frequency of the output word clock on the OUTPUT_LRCLK pins, which is divided down from an internal 1024 × f_S clock (49.152 MHz for a f_S of 48 kHz).	
	OLF [1:0]	**Setting**
	00	Internal clock/1024
	01	Internal clock/512
	10	Internal clock/256
	11	Reserved
FST Frame Sync Type	This bit sets the type of signal on the OUTPUT_LRCLK pins. When this bit is set to 0, the signal is a word clock with a 50% duty cycle; when this bit is set to 1, the signal is a pulse with a duration of one bit clock at the beginning of the data frame.	
TDM TDM Enable	Setting this bit to 1 changes the output port from four serial stereo outputs to a single 8-channel TDM output stream on the SDATA_OUT0 pin (MP6).	
MSB [2:0] MSB Position	These three bits set the position of the MSB of data with respect to the LRCLK edge. The data output of the ADAU1702 is always MSB first.	
	MSB [2:0]	**Setting**
	000	Delay by 1
	001	Delay by 0
	010	Delay by 8
	011	Delay by 12
	100	Delay by 16
	101	Reserved
	111	Reserved
OWL [1:0] Output Word Length	These bits set the word length of the output data-word. All bits following the LSB are set to 0.	
	OWL [1:0]	**Setting**
	00	24 bits
	01	20 bits
	10	16 bits
	11	Reserved

2079 (0x081F)—SERIAL INPUT CONTROL REGISTER

Table 50.

D7	D6	D5	D4	D3	D2	D1	D0	Default
0	0	0	ILP	IBP	M2	M1	M0	0x00

Table 51.

Bit Name	Description	
ILP INPUT_LRCLK Polarity	When this bit is set to 0, the left-channel data on the SDATA_INx pins is clocked when INPUT_LRCLK is low and the right-channel data is clocked when INPUT_LRCLK is high. When this bit is set to 1, the clocking of these channels is reversed. In TDM mode when this bit is set to 0, data is clocked in, starting with the next appropriate BCLK edge (set in Bit 3 of this register) after a falling edge on the INPUT_LRCLK pin. When this bit is set to 1 and the device is running in TDM mode, the input data is valid on the BCLK edge after a rising edge on the word clock (INPUT_LRCLK). INPUT_LRCLK can also operate with a pulse input, rather than a clock. In this case, the first edge of the pulse is used by the ADAU1702 to start the data frame. When this polarity bit is set to 0, a low pulse should be used; when the bit it set to 1, a high pulse should be used.	
IBP INPUT_BCLK Polarity	This bit controls on which edge of the bit clock the input data changes and on which edge it is clocked. Data changes on the falling edge of INPUT_BCLK when this bit is set to 0 and on the rising edge when this bit is set at 1.	
M [2:0] Serial Input Mode	These two bits control the data format that the input port expects to receive. Bit 3 and Bit 4 of this control register override the settings of Bits 2:0; therefore, all four bits must be changed together for proper operation in some modes. The clock diagrams for these modes are shown in Figure 31, Figure 32, and Figure 33. Note that for left-justified and right-justified modes the LRCLK polarity is high and then low, which is opposite from the default setting of ILP. When these bits are set to accept a TDM input, the ADAU1702 data starts after the edge defined by ILP. The ADAU1702 TDM data stream should be input on Pin SDATA_IN0. Figure 34 shows a TDM stream with a high-to-low triggered LRCLK and data changing on the falling edge of the BCLK. The ADAU1702 expects the MSB of each data slot to be delayed by one BCLK from the beginning of the slot, as it would in stereo I²S format. In TDM mode, Channel 0 to Channel 3 are in the first half of the frame, and Channel 4 to Channel 7 are in the second half. Figure 35 shows an example of a TDM stream running with a pulse word clock, which is used to interface to ADI codecs in auxiliary mode. To work in this mode with either the input or output serial ports, set the ADAU1702 to begin the frame on the rising edge of LRCLK, to change data on the falling edge of BCLK, and to delay the MSB position from the start of the word clock by one BCLK.	
	M [2:0]	**Setting**
	000	I²S
	001	Left justified
	010	TDM
	011	Right justified, 24 bits
	100	Right justified, 20 bits
	101	Right justified, 18 bits
	110	Right justified, 16 bits
	111	Reserved

2080 TO 2081 (0x0820 TO 0x0821)—MULTIPURPOSE PIN CONFIGURATION REGISTERS

Each multipurpose pin can be set to different functions from these registers (2080 to 2081). The two 3-byte registers are broken up into 12 4-bit (nibble) sections that each control a different MP pin. Table 54 lists the function of each nibble setting within the MP pin configuration registers. The MSB of each pin's 4-bit configuration inverts the input to or output from the pin. The internal pull-up resistor (approximately 10 kΩ) of each MP pin is enabled when it is set as a digital input (either a GPIO input or a serial data port input).

Table 52. Register 2080

D15	D14	D13	D12	D11	D10	D9	D8	D23 D7	D22 D6	D21 D5	D20 D4	D19 D3	D18 D2	D17 D1	D16 D0	Default
								MP53	MP52	MP51	MP50	MP43	MP42	MP41	MP40	0x00
MP33	MP32	MP31	MP30	MP23	MP22	MP21	MP20	MP13	MP12	MP11	MP10	MP03	MP02	MP01	MP00	0x0000

Table 53. Register 2081

D15	D14	D13	D12	D11	D10	D9	D8	D23 D7	D22 D6	D21 D5	D20 D4	D19 D3	D18 D2	D17 D1	D16 D0	Default
								MP113	MP112	MP111	MP110	MP103	MP102	MP101	MP100	0x00
MP93	MP92	MP91	MP90	MP83	MP82	MP81	MP80	MP73	MP72	MP71	MP70	MP63	MP62	MP61	MP60	0x0000

Table 54.

Bit Name	Description	
MPx [3:0]	Set the function of each multipurpose pin	
	MPx [3:0]	**Setting**
	1111	Auxiliary ADC input (see Table 63)
	1110	Reserved
	1101	Reserved
	1100	Serial data port—inverted (see Table 65)
	1011	Open-collector output—inverted
	1010	GPIO output—inverted
	1001	GPIO input, no debounce—inverted
	1000	GPIO input, debounced—inverted
	0111	N/A
	0110	Reserved
	0101	Reserved
	0100	Serial data port (see Table 65)
	0011	Open-collector output
	0010	GPIO output
	0001	GPIO input, no debounce
	0000	GPIO input, debounced

2082 (0x0822)—AUXILIARY ADC AND POWER CONTROL

Table 55.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	FIL1	FIL0	AAPD	VBPD	VRPD	RSVD	D0PD	D1PD	D2PD	D3PD	0x0000

Table 56.

Bit Name	Description	
FIL [1:0]	Auxiliary ADC filtering	
	FIL [1:0]	**Setting**
	00	4-bit hysteresis (12-bit level)
	01	5-bit hysteresis (12-bit level)
	10	Filter and hysteresis bypassed
	11	Low-pass filter bypassed
AAPD	ADC power-down (both ADCs)	
VBPD	Voltage reference buffer power-down	
VRPD	Voltage reference power-down	
D0PD	DAC0 power-down	
D1PD	DAC1 power-down	
D2PD	DAC2 power-down	
D3PD	DAC3 power-down	

2084 (0x0824)—AUXILIARY ADC ENABLE

Table 57.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
AAEN	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000

Table 58.

Bit Name	Description
AAEN	Enable the auxiliary ADC

2086 (0x0826)—OSCILLATOR POWER-DOWN

Table 59.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	OPD	RSVD	RSVD	0x0000

Table 60.

Bit Name	Description
OPD	Power-down the oscillator

2087 (0x0827)—DAC SETUP

To properly initialize the DACs, Bits DS [1:0] in this register should be set to 01.

Table 61.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	DS1	DS0	0x0000

Table 62.

Bit Name	Description	
DS [1:0]	DAC setup	
	DS [1:0]	**Setting**
	00	Reserved
	01	Initialize DACs
	10	Reserved
	11	Reserved

MULTIPURPOSE PINS

The ADAU1702 has 12 multipurpose (MP) pins that can be individually programmed to be used as serial data inputs, serial data outputs, digital control inputs/outputs to and from the SigmaDSP core, or inputs to the 4-channel auxiliary ADC. These pins allow the ADAU1702 to be used with external ADCs and DACs. They also use analog or digital inputs to control settings such as volume control, or use output digital signals to drive LED indicators.

AUXILIARY ADC

The ADAU1702 has a 4-channel, auxiliary, 8-bit ADC that can be used in conjunction with a potentiometer to control volume, tone, or other parameter settings in the DSP program. Each of the four channels is sampled at the audio sampling frequency (f_S). Full-scale input on this ADC is 3.3 V, so the step size is approximately 13 mV (3.3 V/256 steps). The input resistance of the ADC is approximately 20 kΩ. Table 63 indicates which four MP pins are mapped to the four channels of the auxiliary ADC. The auxiliary ADC is enabled for those pins by writing 1111 to the appropriate portion of the multipurpose pin configuration registers.

The auxiliary ADC is turned on by setting the AAEN bit of the auxiliary ADC enable register (see Table 58).

Noise on the ADC input can cause the digital output to constantly change by a few LSBs. If the auxiliary ADC is used to control volume, this constant change causes small gain fluctuations. To avoid this, add a low-pass filter or hysteresis to the auxiliary ADC signal path by enabling either function in the auxiliary ADC and power control register (2082), as described in Table 56. The filter is enabled by default when the auxiliary ADC is enabled. When data is read from the auxiliary ADC registers, two bytes (12 bits of data, plus zero-padded LSBs) are available because of this filtering.

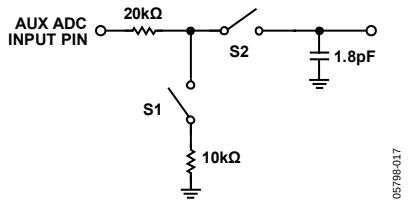

Figure 30. Auxiliary ADC Input Circuit

Figure 30 shows the input circuit for the auxiliary ADC. Switch S1 enables the auxiliary ADC and is set by Bit 15 of the auxiliary ADC enable register. The sampling switch, S2, operates at the audio sampling frequency (f_S).

The auxiliary ADC data registers can be written to directly after AACW in the DSP core control register has been set. In this mode, the voltages on the analog inputs are not written into the registers, but rather the data in the registers is written from the control port.

PVDD supplies the 3.3 V power for the auxiliary ADC analog input. The digital core of the auxiliary ADC is powered with the 1.8 V DVDD signal.

Table 63. Multipurpose Pin Auxiliary ADC Mapping

Multipurpose Pin	Function
MP0	N/A
MP1	N/A
MP2	ADC1
MP3	ADC2
MP4	N/A
MP5	N/A
MP6	N/A
MP7	N/A
MP8	ADC3
MP9	ADC0
MP10	N/A
MP11	N/A

GENERAL-PURPOSE INPUT/OUTPUT PINS

The general-purpose input/output (GPIO) pins can be used as either inputs or outputs. These pins are readable and can be set either through the control interface or directly by the SigmaDSP core. When set as inputs, these pins can be used with push-button switches or rotary encoders to control DSP program settings. Digital outputs can be used to drive LEDs or external logic to indicate the status of internal signals and control other devices. Examples of this use include indicating signal overload, signal present, and button press confirmation.

When set as an output, each pin can typically drive 2 mA. This is enough current to directly drive some high efficiency LEDs. Standard LEDs require about 20 mA of current and can be driven from a GPIO output with an external transistor or buffer. Because of issues that could arise from simultaneously driving or sinking a large current on many pins, care should be taken in the application design to avoid connecting high efficiency LEDs directly to many or all of the MPx pins. If many LEDs are required, use an external driver.

When the GPIO pins are set as open-collector outputs, they should be pulled up to a maximum voltage of 3.3 V (the voltage on IOVDD).

SERIAL DATA INPUT/OUTPUT PORTS

The flexible serial data input and output ports of the ADAU1702 can be set to accept or transmit data in 2-channel format or in an 8-channel TDM stream. Data is processed in twos complement, MSB-first format. The left-channel data field always precedes the right-channel data field in the 2-channel streams. In TDM mode, Slot 0 to Slot 3 are in the first half of the audio frame, and Slot 4 to Slot 7 are in the second half of the frame. TDM mode allows fewer multipurpose pins to be used, freeing more pins for other functions. The serial modes are set in the serial output and serial input control registers.

The serial data clocks need to be synchronous with the ADAU1702 master clock input.

The input control register allows control of clock polarity and data input modes. The valid data formats are I²S, left-justified, right-justified (24-/20-/18-/16-bit), and 8-channel TDM. In all modes except for the right-justified modes, the serial port accepts an arbitrary number of bits up to a limit of 24. Extra bits do not cause an error, but they are truncated internally. Proper operation of the right-justified modes requires that there be exactly 64 BCLKs per audio frame. The TDM data is input on SDATA_IN0. The LRCLK in TDM mode can be input to the ADAU1702 either as a 50/50 duty cycle clock or as a bit-wide pulse.

In TDM mode, the ADAU1702 can be a master for 48 kHz and 96 kHz data, but not for 192 kHz data. Table 64 lists the modes in which the serial output port can function.

Table 64. Serial Output Port Master/Slave Mode Capabilities

f_S	2-Channel Modes (I²S, Left-Justified, Right-Justified)	8-Channel TDM
48 kHz	Master and slave	Master and slave
96 kHz	Master and slave	Master and slave
192 kHz	Master and slave	Slave only

The output control registers allow the user to control clock polarities, clock frequencies, clock types, and data format. In all modes except for the right-justified modes (MSB delayed by 8, 12, or 16 bits), the serial port accepts an arbitrary number of bits up to a limit of 24. Extra bits do not cause an error, but are truncated internally. Proper operation of the right-justified modes requires the LSB to align with the edge of the LRCLK. The default settings of all serial port control registers correspond to 2-channel I²S mode. All register settings apply to both master and slave modes unless otherwise noted.

The function of each multipurpose pin in serial data port mode is shown in Table 65. Pin MP0 to Pin MP5 support digital data input to the ADAU1702, and Pin MP6 to Pin MP11 handle digital data output from the DSP. The configuration of the serial data input port is set in the serial input control register (Table 51), and the configuration of the corresponding output port is controlled with the serial output control register (Table 49). The clocks of the input port function only as slaves, whereas the output port clocks can be set to function as either masters or slaves. The INPUT_LRCLK (MP4) and INPUT_BCLK (MP5) pins are used to clock the SDATA_INx (MP0 to MP3) signals, and the OUTPUT_LRCLK (MP10) and OUTPUT_BCLK (MP11) pins are used to clock the SDATA_OUTx (MP6 to MP9) signals.

If an external ADC is connected as a slave to the ADAU1702, use both the input and output port clocks. The OUTPUT_LRCLK (MP10) and OUTPUT_BCLK (MP11) pins must be set into master mode and connected externally to the INPUT_LRCLK (MP4) and INPUT_BCLK (MP5) pins as well as to the external ADC clock input pins. The data is output from the external ADC into the SigmaDSP on one of the four SDATA_INx pins (MP0 to MP3).

Connections to an external DAC are handled exclusively with the output port pins. The OUTPUT_LRCLK and OUTPUT_BCLK pins can be set to function as either masters or slaves, and the SDATA_OUTx pins are used to output data from the SigmaDSP to the external DAC.

Table 66 describes the proper configurations for standard audio data formats.

Table 65. Multipurpose Pin Serial Data Port Functions

Multipurpose Pin	Function
MP0	SDATA_IN0/TDM_IN
MP1	SDATA_IN1
MP2	SDATA_IN2
MP3	SDATA_IN3
MP4	INPUT_LRCLK (slave only)
MP5	INPUT_BCLK (slave only)
MP6	SDATA_OUT0/TDM_OUT
MP7	SDATA_OUT1
MP8	SDATA_OUT2
MP9	SDATA_OUT3
MP10	OUTPUT_LRCLK (master or slave)
MP11	OUTPUT_BCLK (master or slave)

Table 66. Data Format Configurations

Format	LRCLK Polarity	LRCLK Type	BCLK Polarity	MSB Position
I²S (Figure 31)	Frame begins on falling edge	Clock	Data changes on falling edge	Delayed from LRCLK edge by 1 BCLK
Left-Justified (Figure 32)	Frame begins on rising edge	Clock	Data changes on falling edge	Aligned with LRCLK edge
Right-Justified (Figure 33)	Frame begins on rising edge	Clock	Data changes on falling edge	Delayed from LRCLK edge by 8, 12, or 16 BCLKs
TDM with Clock (Figure 34)	Frame begins on falling edge	Clock	Data changes on falling edge	Delayed from start of word clock by 1 BCLK
TDM with Pulse (Figure 35)	Frame begins on rising edge	Pulse	Data changes on falling edge	Delayed from start of word clock by 1 BCLK

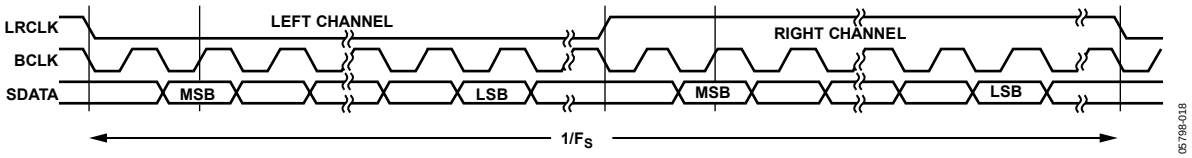

Figure 31. I²S Mode—16 Bits to 24 Bits per Channel

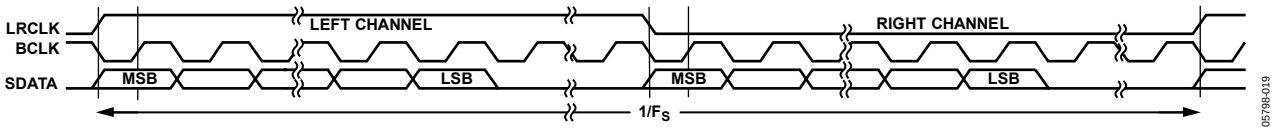

Figure 32. Left-Justified Mode—16 Bits to 24 Bits per Channel

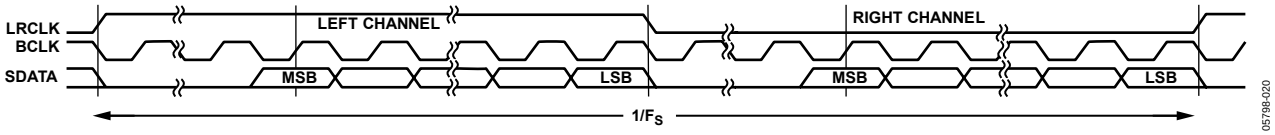

Figure 33. Right-Justified Mode—16 Bits to 24 Bits per Channel

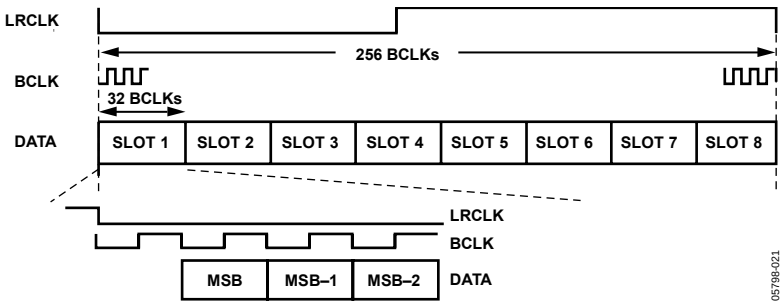

Figure 34. TDM Mode

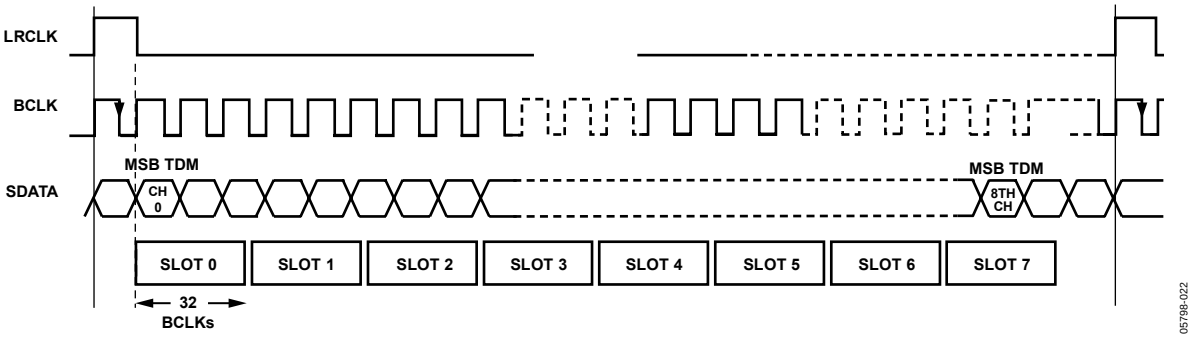

Figure 35. TDM Mode with Pulse Word Clock

LAYOUT RECOMMENDATIONS

PARTS PLACEMENT

The ADC input voltage-to-current resistors and the ADC current set resistor should be placed as close as possible to the 2, 3, and 4 input pins.

All 100 nF bypass capacitors, which are recommended for every analog, digital, and PLL power/ground pair, should be placed as close as possible to the ADAU1702. The 3.3 V and 1.8 V signals on the board should also each be bypassed with a single bulk capacitor (10 μF to 47 μF).

All traces in the crystal oscillator circuit (Figure 14) should be kept as short as possible to minimize stray capacitance. In addition, avoid long board traces connected to any of these components because such traces may affect crystal start-up and operation.

GROUNDING

A single ground plane should be used in the application layout. Components in an analog signal path should be placed away from digital signals.

TYPICAL APPLICATION SCHEMATICS

SELF-BOOT MODE

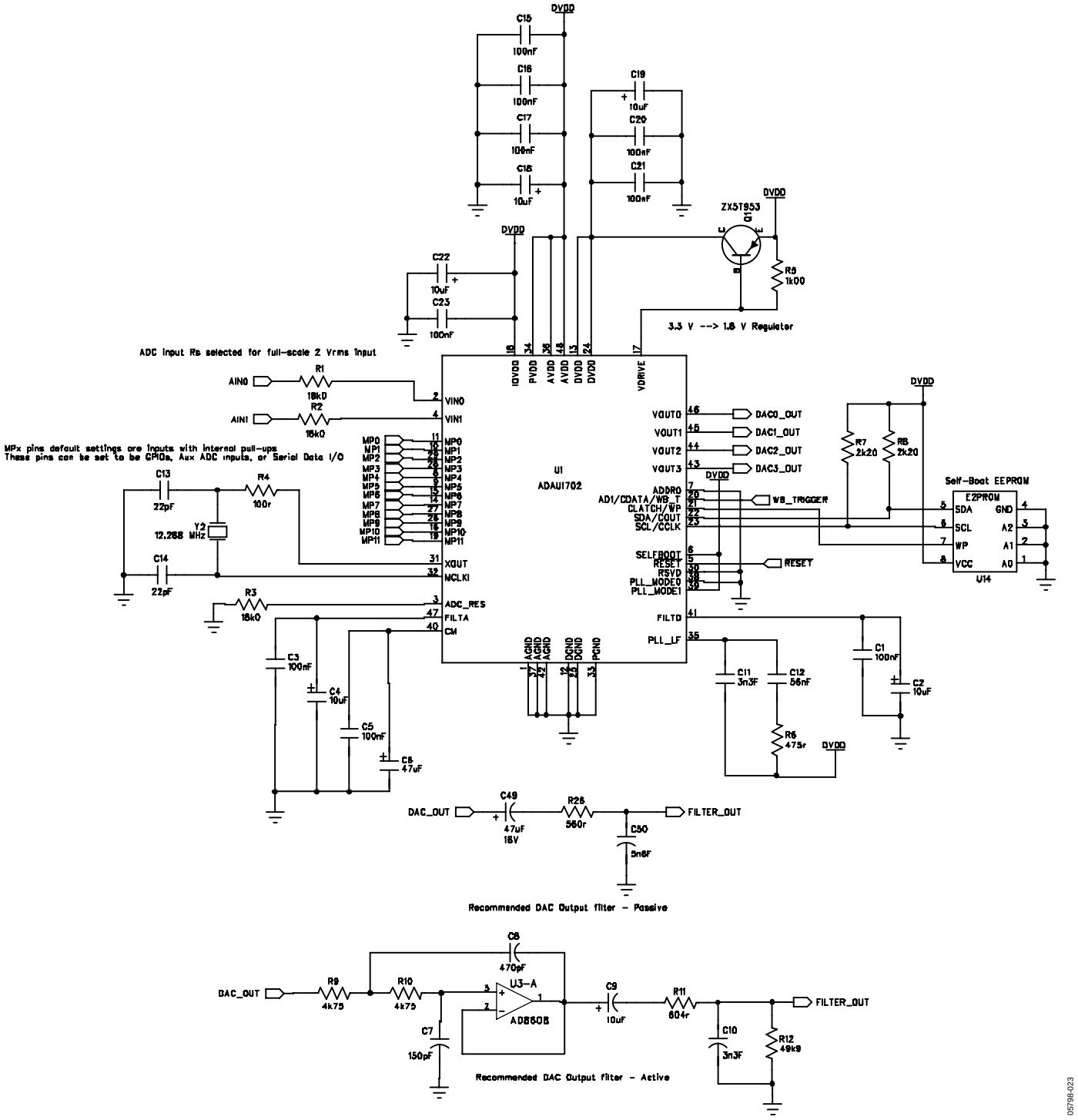

Figure 36. Self-Boot Mode Schematic

I²C CONTROL

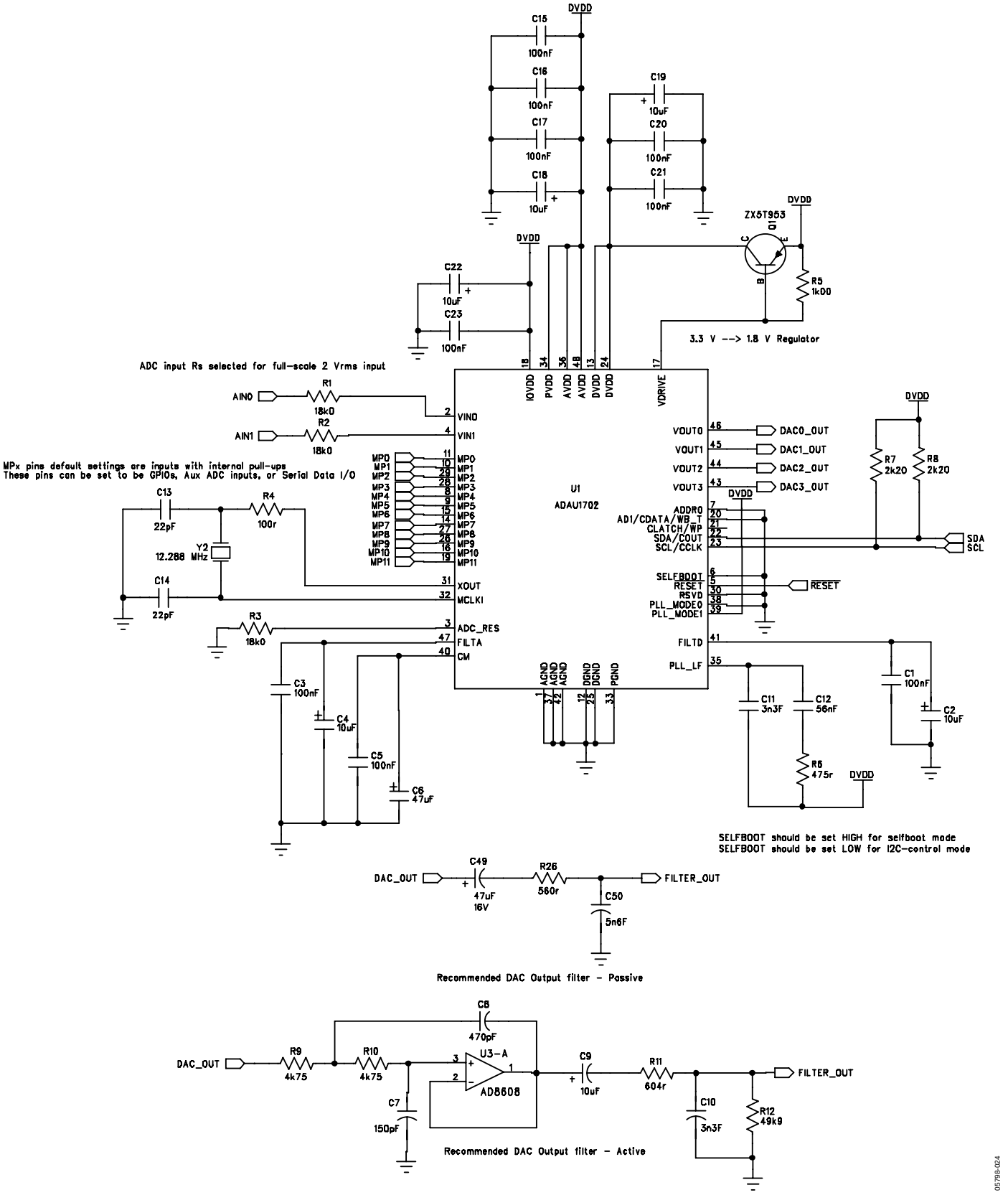

Figure 37. I²C Control Schematic

SPI CONTROL

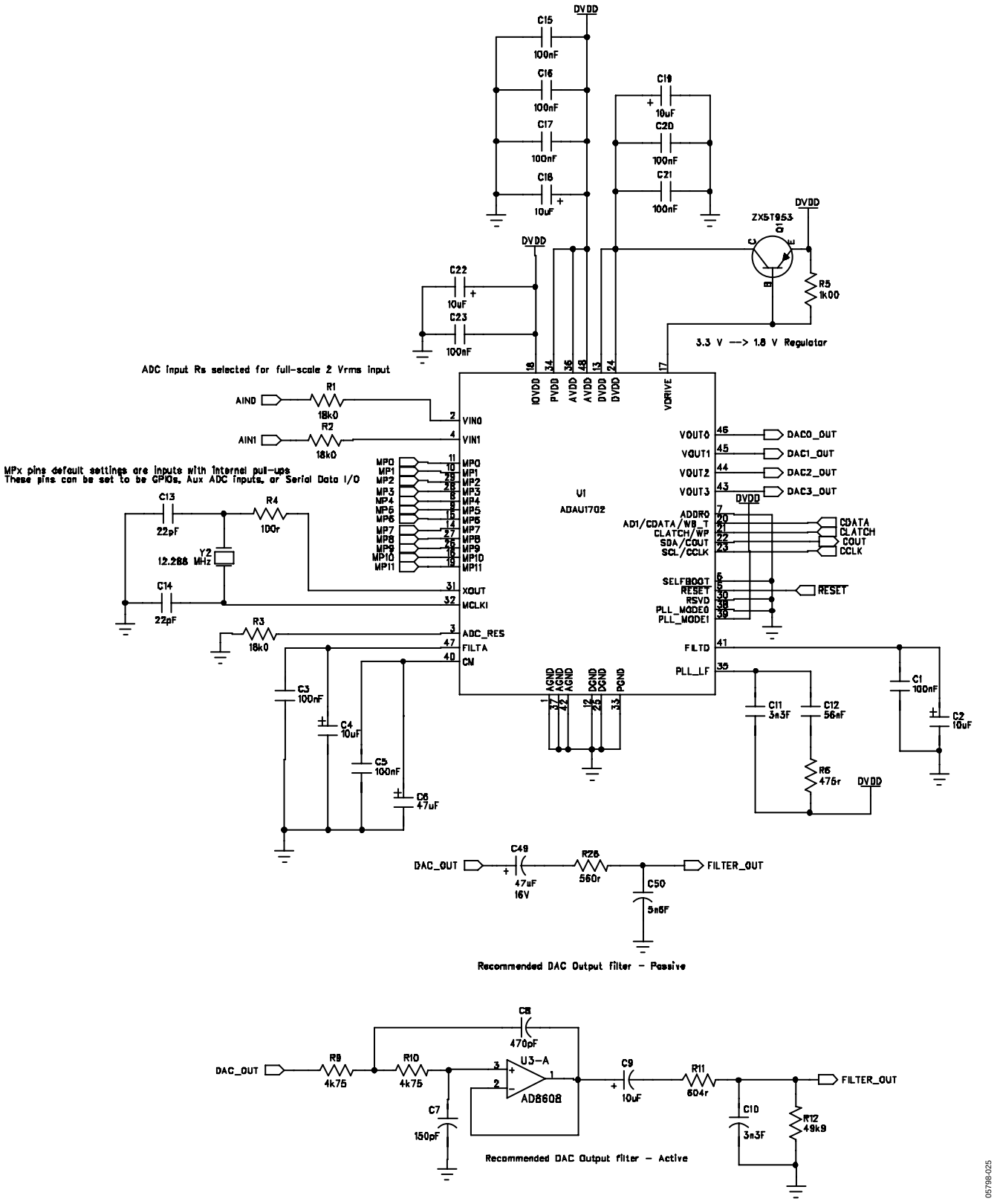

Figure 38. SPI Control Schematic

OUTLINE DIMENSIONS

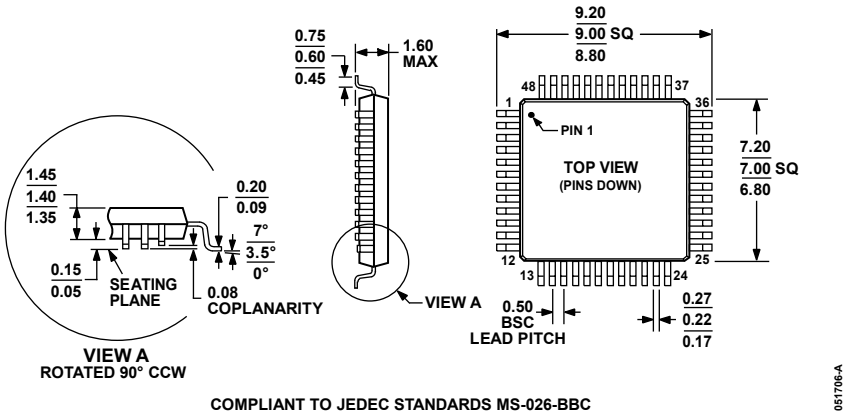

Figure 39. 48-Lead Low-Profile Quad Flat Package [LQFP]
(ST-48)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
ADAU1702JSTZ[1]	0°C to 70°C	48-Lead LQFP	ST-48
ADAU1702JSTZ-RL[1]	0°C to 70°C	48-Lead LQFP, 13 in Reel	ST-48
EVAL-ADAU1702EB		Evaluation Board	

[1] Z = Pb-free part.

NOTES

D05798-0-10/06(0)