TEXAS INSTRUMENTS

TAS3103A

Digital Audio Processor With 3D Effects

Data Manual

January 2006 *Digital Audio Solutions*

SLES166

IMPORTANT NOTICE

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

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

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

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

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

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

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303, Dallas, Texas 75265

Contents

List of Illustrations

List of Tables

1 Introduction

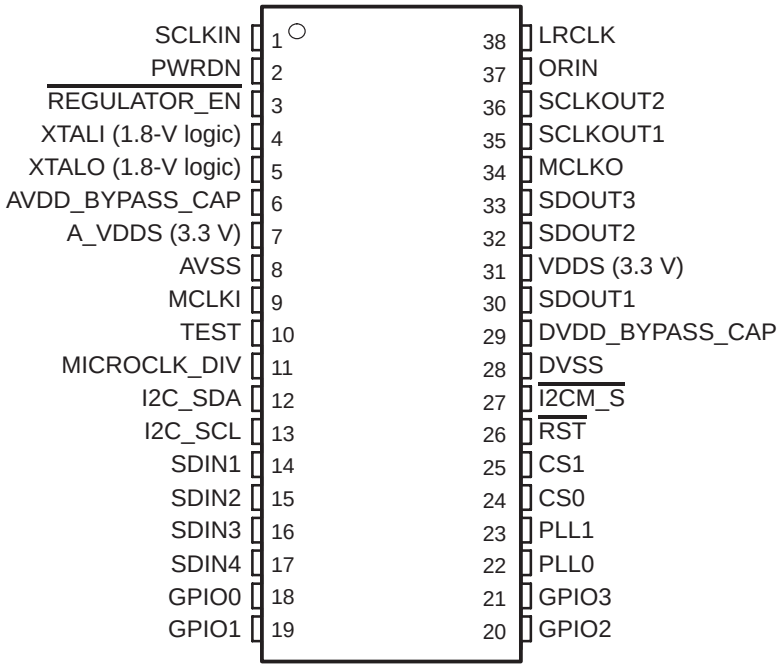
The TAS3103A is a fully configurable digital audio processor that preserves high-quality audio by using a 48-bit data path, 28-bit filter coefficients, a single-cycle 28 × 48-bit multiplier and a 76-bit accumulator. Because of the coefficient-configurable fixed-program architecture of the TAS3103A, a complete set of user-specific audio processing functions can be realized, with short development times, in a small, low-power, low-cost device. A personal computer (PC) GUI-based software development package and a comprehensive evaluation board provide additional facilities to further reduce development times. The TAS3103A uses 1.8-V core logic with 3.3-V I/O buffers, and requires only 3.3-V power. The TAS3103A is available in a 38-pin TSSOP package.

1.1 Features

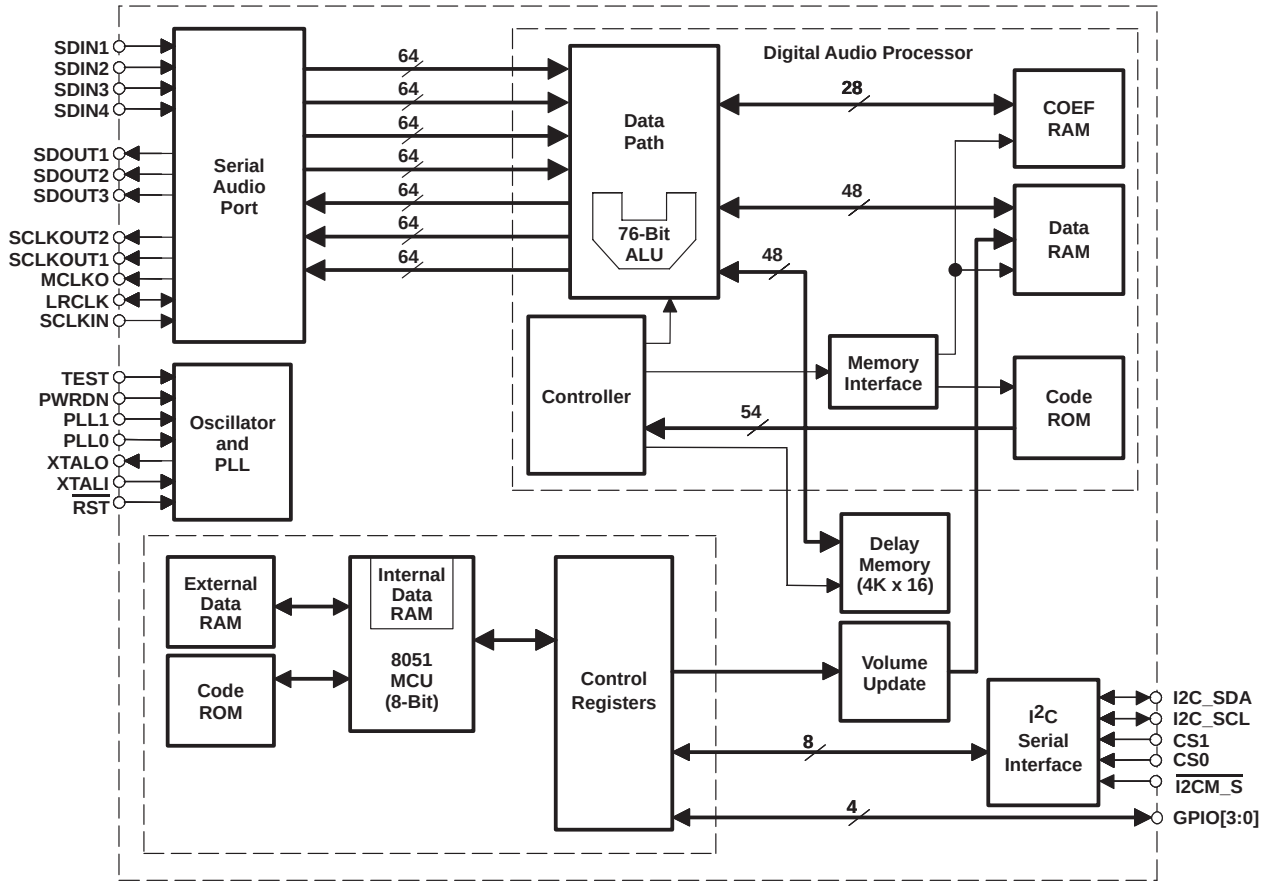
- Audio Input/Output
  - Four Serial Audio Input Channels
  - Three Serial Audio Output Channels
  - 8-kHz to 96-kHz Sample Rates Supported
  - 15 Stereo/TDM Data Formats Supported
  - Input/Output Data Format Selections Independent
  - 16-, 18-, 20-, 24-, and 32-Bit Word Sizes Supported
- Serial Master/Slave I^2C Control Channel
- Three Independent Monaural Processing Channels
  - Programmable Four-Stereo-Input Digital Mixer
  - 3D Effect and Reverberation (Reverb) Structure and Filters
  - Programmable 12-Band Digital Parametric Equalization
  - Programmable Digital Bass and Treble Controls
  - Programmable Digital Soft Volume Control (24 dB to −∞ dB)
  - Soft Mute/Unmute
  - Programmable Dither
  - Programmable Loudness Compensation
  - VU Meter and Spectral Analysis I^2C Output
  - Programmable Channel Delay (Up to 42 ms at 48 kHz)
  - 192-dB Dynamic Range (Supports Up to 32-Bit Audio Data)
  - Dual Threshold Dynamic Range Compression/Expansion
- Electrical and Physical
  - Single 3.3-V Power Supply
  - 38-Pin TSSOP Package
  - Low-Power Standby

1.2 Terminal Assignments

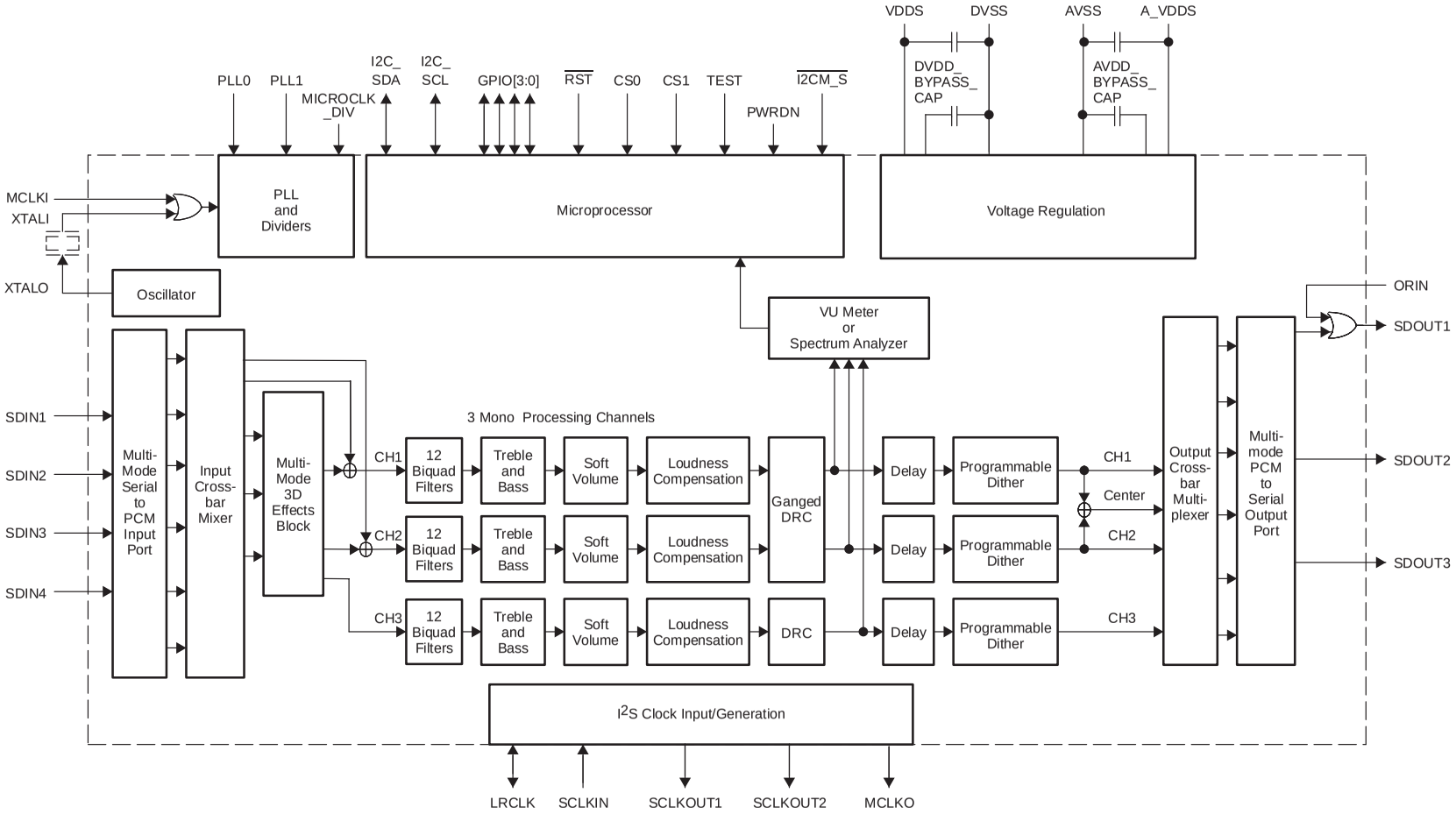
DBT PACKAGE
(TOP VIEW)



1.3 Hardware Block Diagram



1.4 Functional Block Diagram



1.5 Ordering Information

T_A	PLASTIC 38-PIN TSSOP (DBT)
0°C to 70°C	TAS3103ADBT

1.6 Terminal Functions

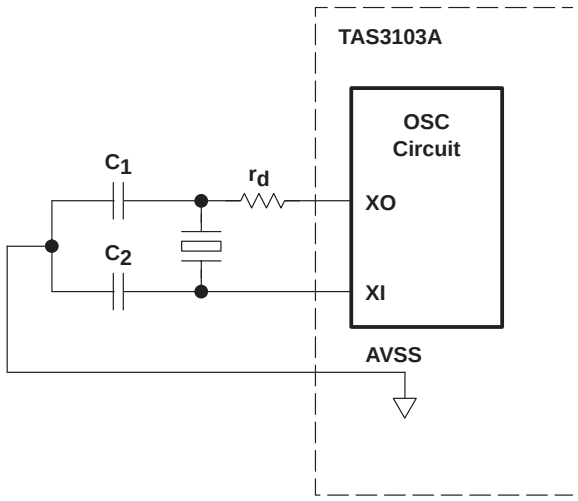
TERMINAL				DESCRIPTION	PULLUP/ DOWN(2)
NAME	NO.	I/O	TYPE(1)		
A_VDDS (3.3 V)	7		PWR	The PWR pin is used to input 3.3-V power to the DPLL and clock oscillator. This pin can be connected to the same power source used to drive the DVSS power pin. To achieve low DPLL jitter, this pin should be bypassed to AVSS with a 0.47-μF capacitor (low-ESR preferable).	None
AVDD_BYPASS_CAP	6		PWR	AVDD_BYPASS_CAP is a pinout of the internally regulated 1.8-VDC power used by the DPLL and crystal oscillator. This pin should be connected to pin 8 with a 0.47-μF capacitor (low-ESR preferable). This pin must not be used to power external devices.	None
AVSS	8		PWR	AVSS is the ground reference for the internal DPLL and oscillator circuitry. This pin needs to reference the same ground as DVSS power pin. To achieve low DPLL jitter, ground noise at this pin must be minimized. The availability of the AVSS pin allows a designer to use optimizing techniques such as star ground connections, separate ground planes, or other quiet ground distribution techniques to achieve a quiet ground reference at this pin.	None
CS0	24	I	D	CS0 is the LSB of a 2-bit code used to generate part of an I^2C device address that makes it possible to address four TAS3103A ICs on the same bus without additional chip select logic. The pulldowns on the inputs select 00 as a default when neither CS0 nor CS1 is connected.	Pulldown
CS1	25	I	D	CS1 is the MSB of a 2-bit code used to generate part of an I^2C device address that makes it possible to address four TAS3103A ICs on the same bus without additional chip select logic.	Pulldown
DVDD_BYPASS_CAP	29		PWR	DVDD_BYPASS_CAP is a pinout of the internally regulated 1.8-V power used by all internal digital logic. This pin must not be used to power external devices. A ceramic capacitor of at least 4.7 μF should be placed as close to the device as possible between this pin and pin 28. A 0.01 μF should be connected in parallel for high-frequency decoupling.	None
DVSS	28		PWR	DVSS is the digital ground pin.	None
GPIO0	18	I/O	D	GPIO0 is a general-purpose I/O, controlled by the internal microprocessor through I^2C commands. When in the I^2C master mode, GPIO0 serves as a volume-up command for CH1/CH2.	Pullup
GPIO1	19	I/O	D	GPIO1 is a general-purpose I/O, controlled by the internal microprocessor through I^2C commands. When in the I^2C master mode, GPIO1 serves as a volume-down command for CH1/CH2.	Pullup
GPIO2	20	I/O	D	GPIO2 is a general-purpose I/O, controlled by the internal microprocessor through I^2C commands. When in the I^2C master mode, GPIO2 serves as a volume-up command for CH3.	Pullup
GPIO3	21	I/O	D	GPIO3 is a general-purpose I/O, controlled by the internal microprocessor through I^2C commands. When in the I^2C master mode, GPIO3 serves as a volume-down command for CH3.	Pullup
$\overline{\text{I2CM\_S}}$	27	I	D	$\overline{\text{I2CM\_S}}$ is a non-latched input that determines whether the TAS3103A acts as an I^2C master or slave. Logic high, or no connection, sets the TAS3103A as an I^2C master device. A logic low sets the TAS3103A as an I^2C slave device. As a master I^2C device, the TAS3103A I^2C port must have access to an external EEPROM for input.	Pullup

TERMINAL				DESCRIPTION	PULLUP/ DOWN(2)
NAME	NO.	I/O	TYPE(1)		
I2C_SCL	13	I/O	D	I2C_SCL is the I^2C clock pin. When the TAS3103A I^2C port is a master, I2C_SCL is $(1/2^N) \times (1/(M+1)) \times 1/10$ times the microprocessor clock, where N and M are set to 2 and 8, respectively. When the TAS3103A I^2C port is a slave, input clock rates up to 400 kHz can be supported. This pin must be provided an external pullup (2 kΩ is recommended for most applications).	External pullup required
I2C_SDA	12	I/O	D	I2C_SDA is the I^2C bidirectional data pin. The TAS3103A I^2C port can support data rates up to 400 kbps. This pin must be provided an external pullup (2 kΩ is recommended for most applications).	External pullup required
LRCLK	38	I/O	D	LRCLK is either an input or an output, depending on whether the TAS3103A is in a master or slave serial audio port mode, which is determined by bit 22 of subaddress 0xF9.	Pulldown
MCLKI	9	I	D	MCLKI is a master clock input that provides an alternative to using a fixed crystal frequency. In DPLL modes, the input frequency of this clock can range from 2.8 MHz to 24.576 MHz. In PLL bypass mode, frequencies up to 136 MHz can be used. Whenever MCLKI is not used and XTALI/XTALO provide the master clock input, MCLKI must be grounded.	None
MCLKO	34	O	D	MCLKO is the master output clock pin. It is produced by dividing MCLKI/XTALI by 1, 2, or 4 (depending on the setting of a subaddress control field). MCLKO is provided to interconnect, without the need for additional glue logic, the TAS3103A interfaces with chips that require different multiples of the audio sample rate (Fs) as a master clock.	None
MICROCLK_DIV	11	I	D	MICROCLK_DIV sets the division ratio between the digital audio processing clock and the internal microprocessor clock. The audio-processing clock is the DPLL output clock if PLL_bypass is not enabled. The audio-processing clock is MCLKI/XTALI master clock if PLL_bypass is enabled. Logic high on this pin sets the microprocessor clock equal to the audio-processing clock. A logic low sets the microprocessor clock to 1/4 the digital audio-processing clock. MICROCLK_DIV must be set low if the audio processing clock is > 36 MHz. MICROCLK_DIV must be set high if the audio processing clock is ≤ 36 MHz.	Pulldown
ORIN	37	I	D	ORIN allows the processing of a multichannel signal set through two TAS3103As without any additional components. One use of ORIN would be to fully emulate a 6-channel audio processor at speeds up to a 96-kHz sample rate with only two TAS3103As and no glue logic. The two-chip configuration is accomplished by wiring the SDOUT1 port of one of the two TAS3103A chips to the ORIN port of the second TAS3103A. Internal to the chip, the ORIN input is ORed with internal SDOUT1 data to generate the resulting output data on channel SDOUT1. For TDM output formats, the SDOUT1 outputs of the two chips differ in phasing in both the left and right channels to arrive at the proper composite output. For discrete outputs, one chip contributes the left channel of the composite SDOUT1, and the other chip contributes the right channel of the composite SDOUT1. If not used, ORIN must be connected to ground.	Pulldown
PLL0	22	I	D	PLL0 is the LSB of a 2-bit code used to select four different modes of DPLL multiplexer/input divider operation. PLL[1:0] values of 00, 01, and 10 select the DPLL input clock to be MCLKI/XTALI divided by 1, 2, and 4, respectively. A value of 11 results in MCLKI/XTALI being substituted for the DPLL output. The pullup/pulldown combination provides a default of 01 when neither PLL0 nor PLL1 is connected.	Pullup
PLL1	23	I	D	PLL1 is the MSB of a 2-bit code used to select four different modes of DPLL multiplexer/input divider operation. PLL[1:0] values of 00, 01, and 10 select the DPLL input clock to be MCLKI/XTALI divided by 1, 2, and 4, respectively. A value of 11 results in MCLKI/XTALI being substituted for the DPLL output. The pullup/pulldown combination provides a default of 01 when neither PLL0 nor PLL1 is connected.	Pulldown

TERMINAL				DESCRIPTION	PULLUP/ DOWN(2)
NAME	NO.	I/O	TYPE(1)		
PWRDN	2	I	D	PWRDN powers down all logic and stops all clocks whenever logic high is applied. However, the coefficient memory remains stable through a power-down cycle, as long as a reset is not sent after a power-down cycle.	Pulldown
$\overline{\text{REGULATOR\_EN}}$	3	I	D	$\overline{\text{REGULATOR\_EN}}$ is only used in factory tests. This pin should always be tied to ground.	None
$\overline{\text{RST}}$	26	I	D	$\overline{\text{RST}}$ is the master reset input. Applying a logic low to this pin generates a master reset. The master reset results in all coefficients being set to their power-up default state, all data memories being cleared, and all logic signals being returned to their default values.	Pullup
SCLKIN	1	I	D	SCLKIN is the serial audio port (SAP) input data clock. This clock is only used when the SAP is a slave. In master mode, SCLKOUT1 internally provides the serial input clock (SCLKOUT1 from a given TAS3103A must not be connected to SCLKIN on the same TAS3103A chip).	Pulldown
SCLKOUT1	35	O	D	SCLKOUT1 is one of two serial output bit clocks. It is divided from MCLKI/XTALI in master mode, and SCLKIN in slave mode. Subaddress control fields determine the divide ratio in both cases. When the serial audio port is in master mode, SCLKOUT1 is used to receive incoming serial data and should be wired to the data source(s) providing data to the SDIN inputs.	None
SCLKOUT2	36	O	D	SCLKOUT2 is one of two serial output bit clocks. It is divided from MCLKI/XTALI in master mode, and SCLKIN in slave mode. Subaddress control fields determine the divide ratio in both cases. SCLKOUT2 is always used to clock out serial data from the three serial SDOUT output data channels. SCLKOUT2 is provided separately from SCLKOUT1 to allow discrete-in to TDM-out and TDM-in to discrete-out data format conversions without the use of external glue logic.	Output
SDIN1	14	I	D	SDIN1, SDIN2, SDIN3, and SDIN4 are the four TAS3103A serial data input ports. All four input ports support four discrete (stereo) data formats. SDIN1 is the only data input port that also supports 11 time division multiplexed data formats. All four ports are capable of receiving data with bit rates up to 24.576 MHz.	Pulldown
SDIN2	15	I	D	SDIN2 is one of the four TAS3103A serial data input ports. SDIN2 supports four discrete (stereo) data formats, and is capable of receiving data with bit rates up to 24.576 MHz.	Pulldown
SDIN3	16	I	D	SDIN3 is one of the four TAS3103A serial data input ports. SDIN4 supports four discrete (stereo) data formats, and is capable of receiving data with bit rates up to 24.576 MHz.	Pulldown
SDIN4	17	I	D	SDIN4 is one of the four TAS3103A serial data input ports. SDIN4 supports four discrete (stereo) data formats, and is capable of receiving data with bit rates up to 24.576 MHz.	Pulldown
SDOUT1	30	O	D	SDOUT1, SDOUT2, and SDOUT3 are the three TAS3103A serial data output ports. All three output ports support four discrete (stereo) data formats. SDOUT1 is the only data output port that also supports 11 time division multiplexed data formats. All three ports are capable of outputting data at bit rates up to 24.576 MHz.	None
SDOUT2	32	O	D	SDOUT2 is one of the three serial data output ports. SDOUT2 supports four discrete (stereo) data formats, and is capable of outputting data at bit rates up to 24.576 MHz.	None
SDOUT3	33	O	D	SDOUT3 is one of the three serial data output ports. SDOUT3 supports four discrete (stereo) data formats, and is capable of outputting data at bit rates up to 24.576 MHz.	None
TEST	10	I	D	TEST is only used in factory tests. This pin must be left unconnected or grounded.	Pulldown

TERMINAL				DESCRIPTION	PULLUP/ DOWN(2)
NAME	NO.	I/O	TYPE(1)		
VDDS (3.3 V)	31	-	PWR	VDDS is the 3.3-V pin that powers (1) the 1.8-V internal power regulator used to supply logic power to the chip and (2) the I/O ring. It is recommended that this pin be bypassed to DVSS (pin 28) with a low-ESR capacitor of 47 µF or greater. A 0.1-µf ceramic capacitor should be placed in parallel with the 47-µf capacitor to provide high-frequency decoupling.	None
XTALI (1.8-V logic)	4	I	A	XTALO and XTALI provide a master clock for the TAS3103A via use of an external fundamental-mode crystal. XTALI is the 1.8-V input port for the oscillator circuit. See Note 3 for recommended crystal type and accompanying circuitry. This pin should be grounded when the MCLKI pin is used as the source for the master clock.	None
XTALO (1.8-V logic)	5	O	A	XTALO and XTALI provide a master clock for the TAS3103A via use of an external fundamental-mode crystal. XTALO is the 1.8-V output drive to the crystal. XTALO can support crystal frequencies between 2.8 MHz and 20 MHz. See Note 3 for recommended crystal type and accompanying circuitry. This pin should be left unconnected in applications using an external clock input to MCLKI.	None

NOTES: 1. TYPE: A = analog; D = 3.3-V digital; PWR = power/ground/decoupling
2. All pullups are 20-µA weak pullups and all pulldowns are 20-µA weak pulldowns. The pullups and pulldowns are included to ensure proper input logic levels if the pins are left unconnected (pullups => logic 1 input; pulldowns => logic 0 input). Devices that drive inputs with pullups must be able to sink 20 µA while maintaining a logic 0 drive level. Devices that drive inputs with pulldowns must be able to source 20 µA while maintaining a logic 1 drive level.
3. Crystal type and recommended circuit:



- Crystal type = parallel-mode, fundamental-mode crystal
- r_d = drive level control resistor—vendor specified
- C_L = Crystal load capacitance (capacitance of circuitry between the two terminals of the crystal)
- $C_L = (C_1 \times C_2) / (C_1 + C_2) + C_S$ (where C_S = board stray capacitance ~2 pF)
  - Example: Vendor recommended C_L = 18 pF, C_S = 3 pF $\Rightarrow C_1 = C_2 = 2 \times (18 - 3) = 30$ pF

The TAS3103A crystal should comply with the following minimum specifications:

- Operation mode: fundamental
- Frequency tolerance: ±≤100 ppm at 25°C
- Frequency temperature characteristics: ±≤100 ppm, –20°C to 70°C
- Aging: ±≤10 ppm/year, maximum

1.7 Operational Modes

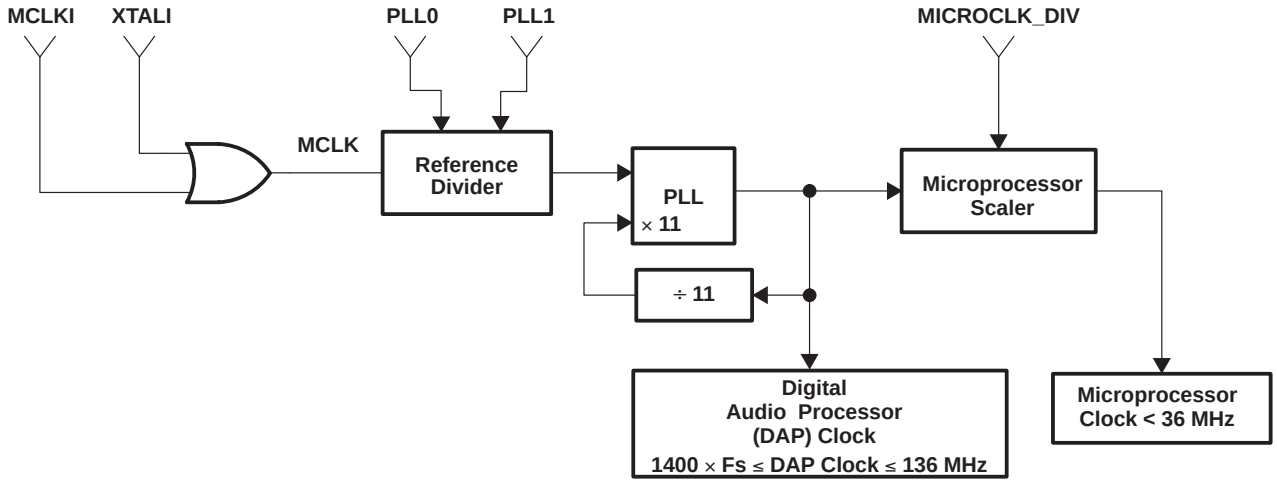
The TAS3103A operation is governed by I/O terminal voltage level settings and register/coefficient settings within the TAS3103A. The terminal settings are wholly sufficient to address all external environments, allowing the remaining configuration settings to be determined by either I^{2}C commands or by the content of an I^{2}C serial EEPROM (when the I^{2}C master mode is selected).

1.7.1 Terminal-Controlled Modes

1.7.1.1 Clock Control

PLL1	PLL0	DAP CLOCK
0	0	11 × MCLK
0	1	(11 × MCLK)/2 (default)
1	0	(11 × MCLK)/4
1	1	MCLK (PLL bypass)

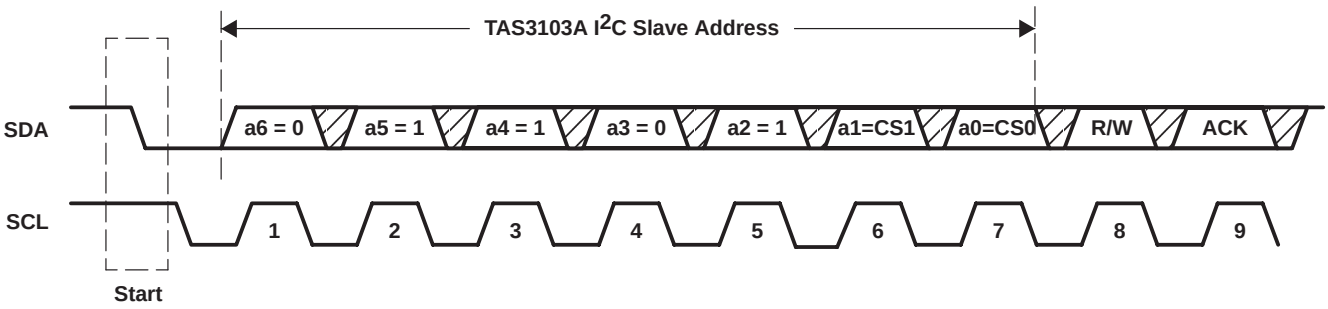
MICROCLK_DIV	MICROPROCESSOR CLOCK
0	DAP clock/4 (default)
1	DAP clock



1.7.1.2 I^{2}C Bus Setup

SLAVE ADDRESS	CS1	CS0
0x68/69 (default)	0	0
0x6A/6B	0	1
0x6C/6D	1	0
0x6E/6F	1	1

$\overline{\text{I2CM\_S}}$	I^{2}C BUS MODE
0	Slave
1	Master (default)—EEPROM device ID = 0xA0

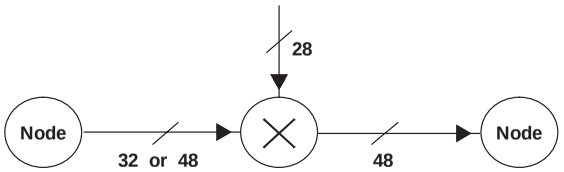


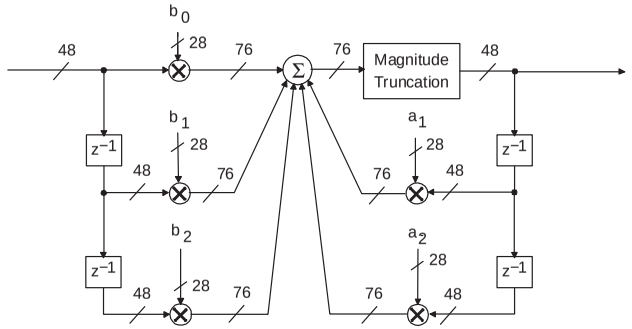
1.7.1.3 Power-Down/Sleep Selection

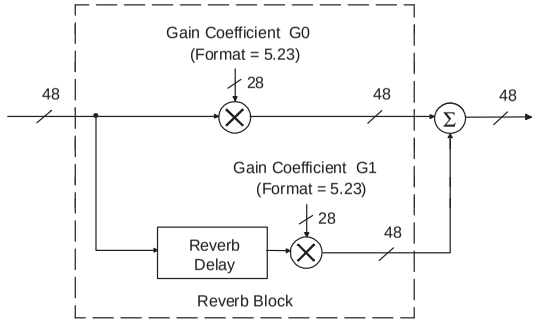
PWRDN	POWER STATUS
0	Active
1	Power down/sleep

1.7.2 I²C Bus-Controlled Modes

<table>
<tr><th>SUBADDRESS(es)</th><th>PARAMETER(s)</th></tr>
<tr><td>
0x00 – Starting I²C Check Word
0xFC – Ending I²C Check Word
<table>
<tr><td>S</td><td>Slave Addr</td><td>Ack</td><td>Subaddr</td><td>Ack</td><td>msb xxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td></tr>
</table>
</td><td>
• Check words apply to I²C master mode only
• In master I²C mode, the two check words are compared after EEPROM download. If comparison fails, a second attempt is made. If the second comparison fails, the parameters default to the slave default values.
• In slave I²C mode, the default value for both check words is:
0x81_42_24_18
</td></tr>
<tr><td>
Input Mixer 28-Bit Gain Coefficients
0x01 − 0x33

Output Mixer 28-Bit Gain Coefficients
0x84 – 0xA1
<table>
<tr><td>S</td><td>Slave Addr</td><td>Ack</td><td>Subaddr</td><td>Ack</td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td></tr>
</table>
</td><td>
Gain Coefficient
(Format = 5.23)


</td></tr>
<tr><td>
Effects Block Biquad Filter Coefficients
0x34−0x4B
<table>
<tr><td>S</td><td>Slave Addr</td><td>Ack</td><td>Subaddr</td><td>Ack</td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>a_1</td></tr>
<tr><td colspan="5"></td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>a_2</td></tr>
<tr><td colspan="5"></td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>b_0</td></tr>
<tr><td colspan="5"></td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>b_1</td></tr>
<tr><td colspan="5"></td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>b_2</td></tr>
</table>
</td><td>


NOTE: All gain coefficients 5.23 numbers.
</td></tr>
<tr><td>
Reverb Block Gains
<table>
<tr><th>Reverb Block</th><th>Subaddress</th></tr>
<tr><td>CH1</td><td>0x4C</td></tr>
<tr><td>CH2</td><td>0x4D</td></tr>
<tr><td>CH3</td><td>0x4E</td></tr>
</table>
<table>
<tr><td>S</td><td>Slave Addr</td><td>Ack</td><td>Subaddr</td><td>Ack</td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>G0</td></tr>
<tr><td colspan="5"></td><td>msb 0000 s xxx lsb</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxxxx</td><td>Ack</td><td>xxxxxx lsb</td><td>Ack</td><td>G1</td></tr>
</table>
</td><td>


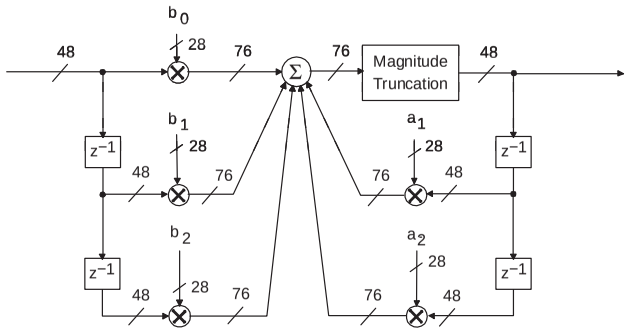
</td></tr>
</table>

SUBADDRESS(es)	PARAMETER(s)

Cascaded (Twelve/Channel) Main Filter Biquads

MAIN FILTER BLOCK	Subaddress
CH1	0x4F−0x5A
CH2	0x5B−0x66
CH3	0x67−0x72

S	Slave Addr	Ack	Subaddr	Ack	m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	a_1
					m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	a_2
					m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	b_0
					m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	b_1
					m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	b_2

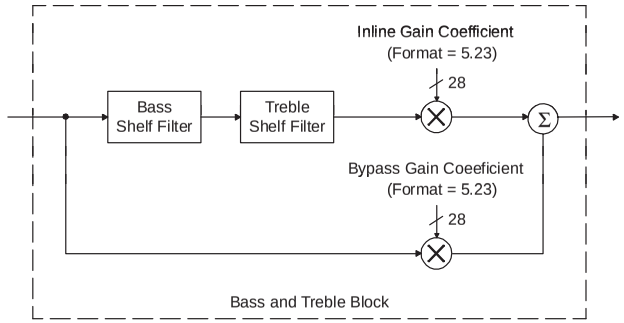


NOTE: All gain coefficients 5.23 numbers

Bass and Treble Gain Coefficients

CH1 = 0x73
CH2 = 0x74
CH3 = 0x75

S	Slave Addr	Ack	Subaddr	Ack	m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	Bypass Gain
					m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack	Inline Gain



Dynamic Range Control (DRC) Mixer Coefficients

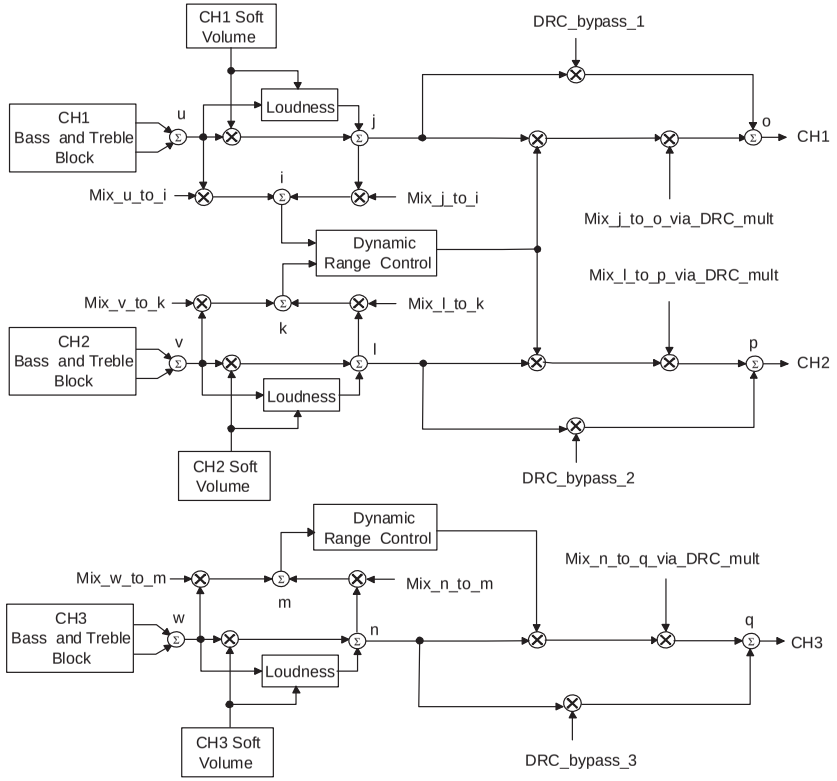
S	Slave Addr	Ack	Subaddr	Ack	m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	l xxxxxx s b	Ack

S	Slave Addr	Ack	Subaddr	Ack	m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	l xxxxxx s b	Ack	
					m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	l xxxxxx s b	Ack	Word2

Word1

- CH1
  - 0x76 = Mix u to i
  - 0x79 = Mix j to i
- CH2
  - 0x77 = Mix v to k
  - 0x7A = Mix l to k
- CH3
  - 0x78 = Mix w to m
  - 0x7B = Mix n to m

- CH1– 0x7C
  - Word 1 = Mix j to o – Inline
  - Word 2 = Mix j to o – Bypass
- CH2– 0x7D
  - Word 1 = Mix l to p – Inline
  - Word 2 = Mix l to p – Bypass
- CH1– 0x7E
  - Word 1 = Mix n to q – Inline
  - Word 2 = Mix n to q – Bypass

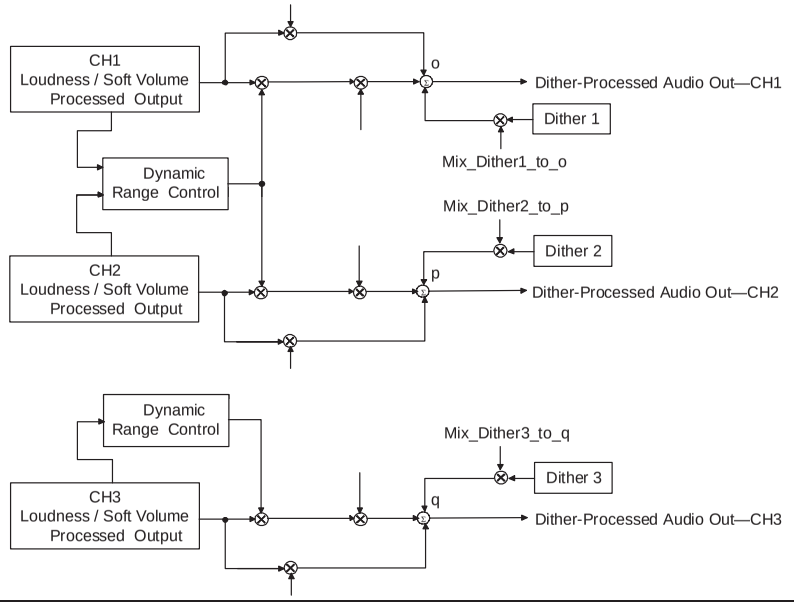


SUBADDRESS(es)	PARAMETER(s)

Dither Mix Gain Coefficients

S	Slave Addr	Ack	Subaddr	Ack	m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack

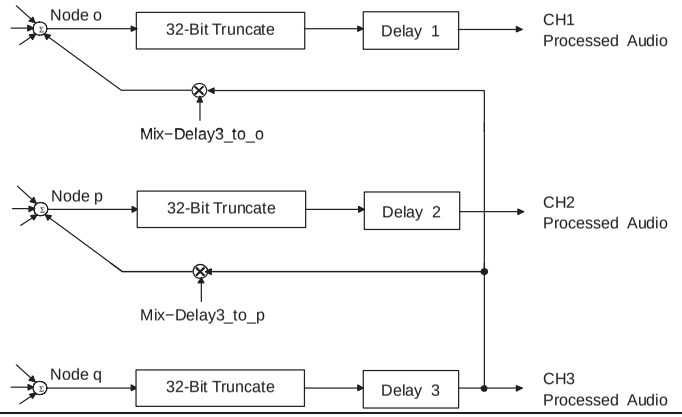
CH1 0x7F = Mix Dither 1 to o − 28-Bit Coefficient
CH2 0x80 = Mix Dither 2 to p − 28-Bit Coefficient
CH3 0x81 = Mix Dither 3 to q − 28-Bit Coefficient



CH3 to CH1 and CH2 Mix Gain Coefficients

S	Slave Addr	Ack	Subaddr	Ack	m 0000 s xxx b	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx s l b	Ack

CH1 0x82 = Mix CH3 Output to o − 28-Bit Coefficient
CH2 0x83 = Mix CH3 Output to p − 28-Bit Coefficient



Soft Volume and Loudness Subaddress

Soft Volume

- Mute/Unmute = 0xF0
- Volume Slew-Rate Command = 0xF1

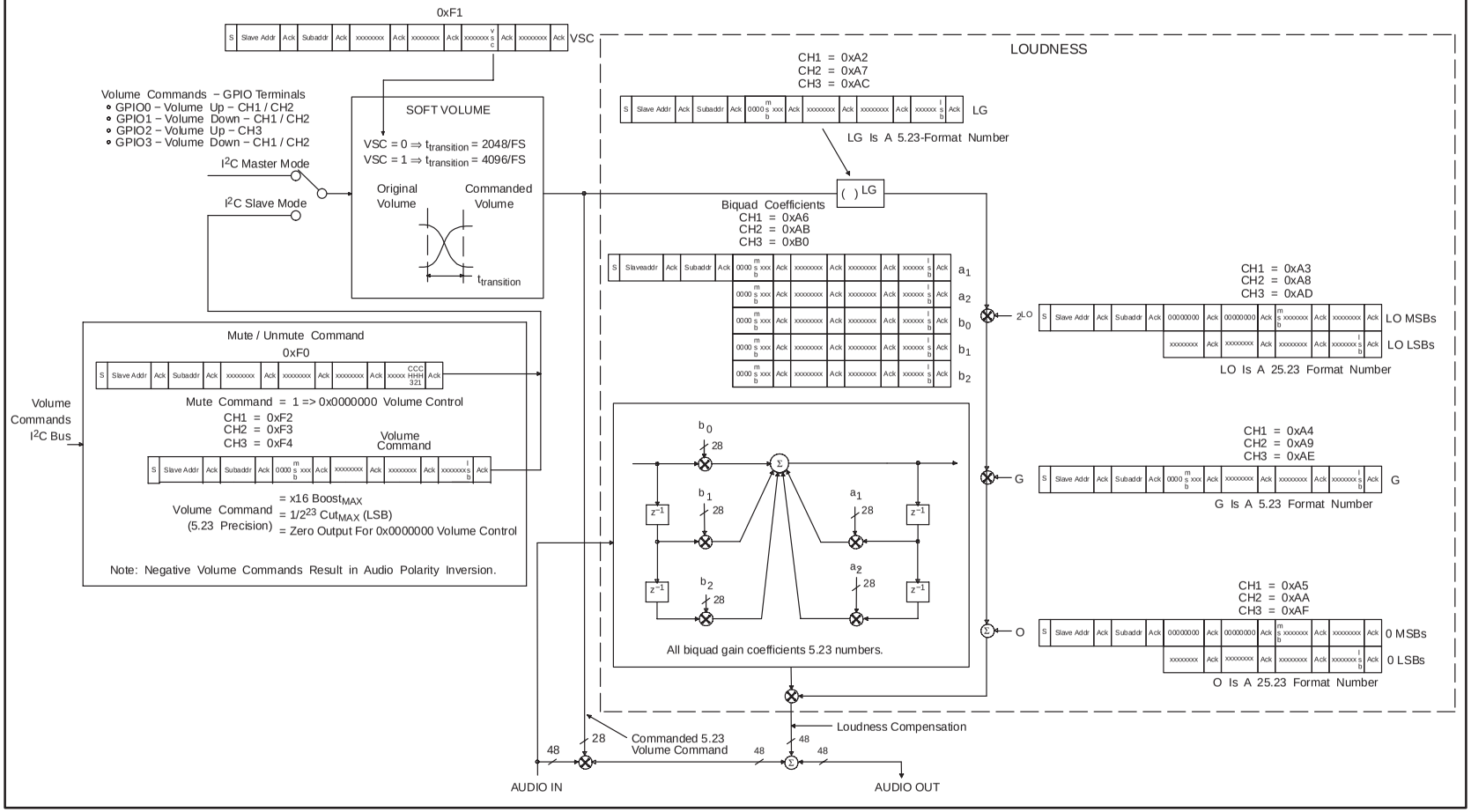
Volume Command

Parameter	Subaddress		
	CH1	CH2	CH3
Volume Command	0xF2	0xF3	0xF4

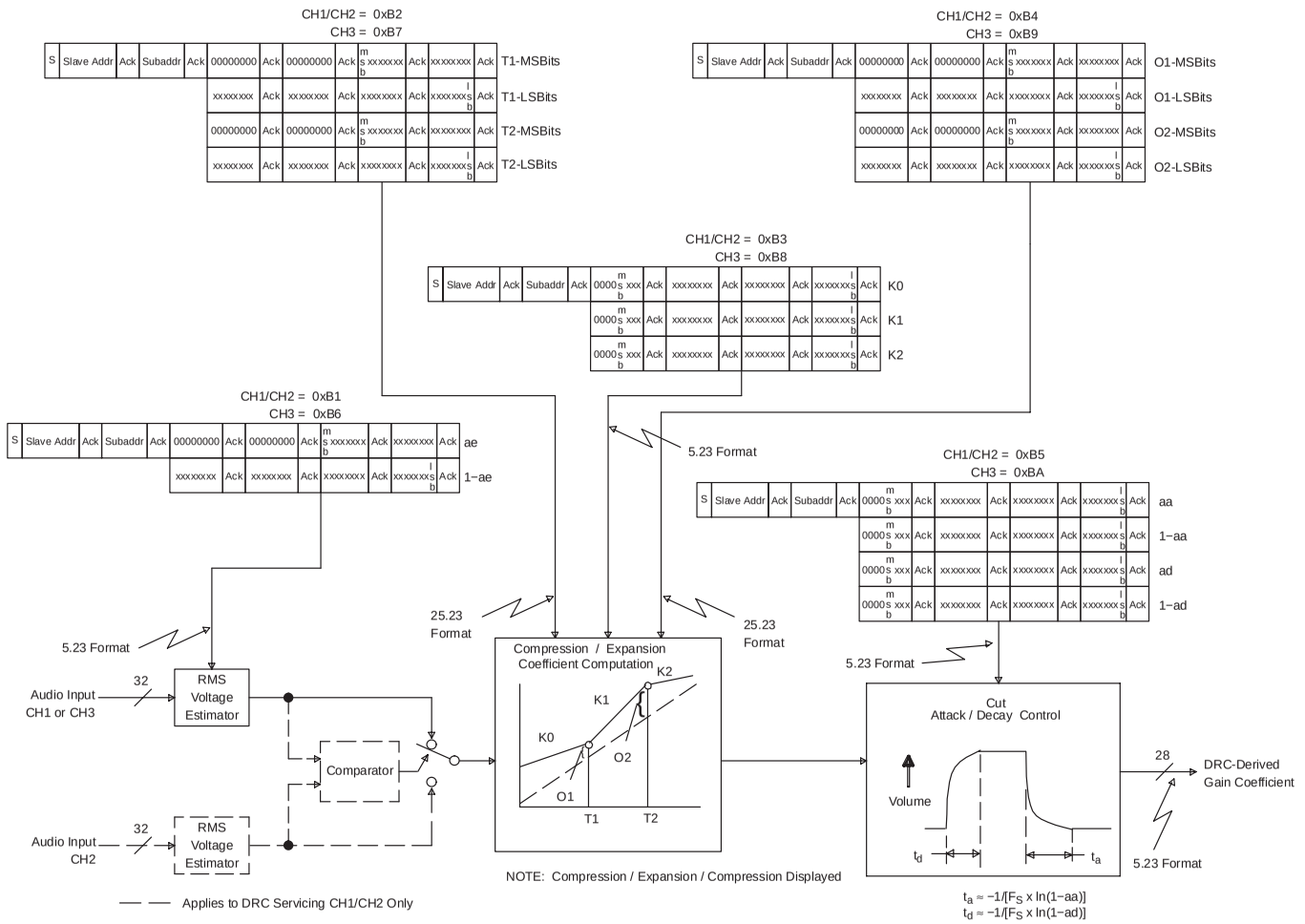
Loudness

Parameter	Subaddress		
	CH1	CH2	CH3
LG	0xA2	0xA7	0xAC
LO	0xA3	0xA8	0xAD
G	0xA4	0xA9	0xAE
O	0xA5	0xAA	0xAF
Biquad	0xA6	0xAB	0xB0

NOTE: The value of the VSC bit is not correctly reported when the register is read. VSC behaves like a write-only bit.



Subaddress — Dynamic Range Control (DRC) Block



Spectrum Analyzer/VU Meter

Biquad 1 to 10 Subaddresses = 0xBC to 0xC5

S	Slave Addr	Ack	Subaddr	Ack	0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	a_1
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	a_2
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	b_0
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	b_1
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	b_2

RMS Window Time Constant Subaddress = 0xBB

S	Slave Addr	Ack	Subaddr	Ack	0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	asa
					0000 msb xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxx lsb	Ack	1−asa

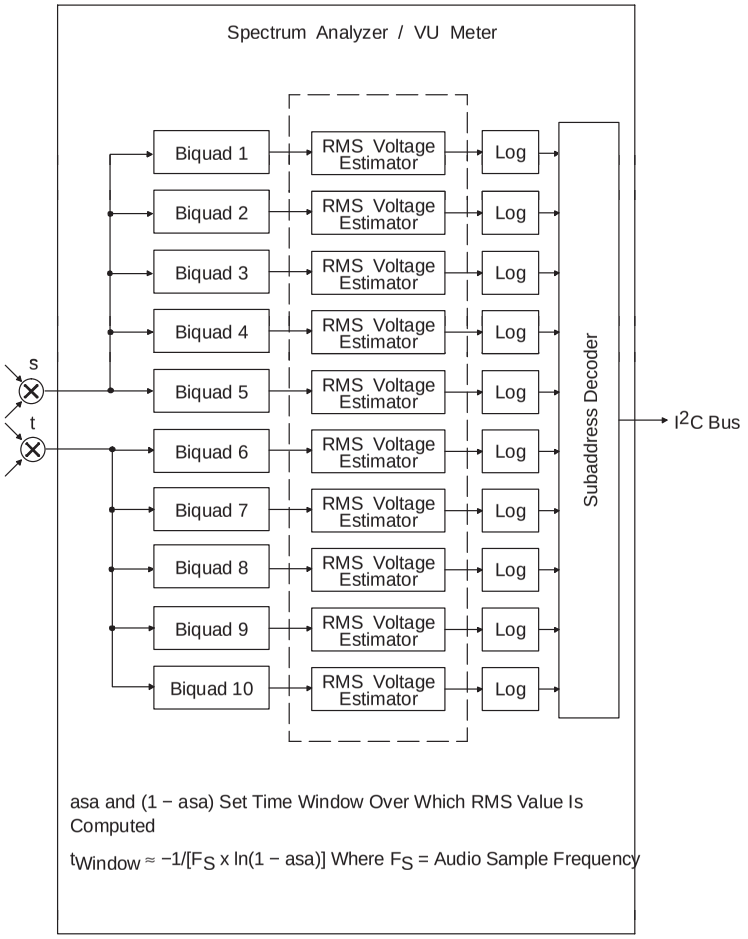
Spectrum Analyzer Output Subaddress = 0xFD

S	Slave Addr	Ack	Subaddr	Ack	xxxxx.xxx	Ack	
					xxxxx.xxx	Ack	Biquad 1
					xxxxx.xxx	Ack	Biquad 2
					xxxxx.xxx	Ack	Biquad 3
					xxxxx.xxx	Ack	Biquad 4
					xxxxx.xxx	Ack	Biquad 5
					xxxxx.xxx	Ack	Biquad 6
					xxxxx.xxx	Ack	Biquad 7
					xxxxx.xxx	Ack	Biquad 8
					xxxxx.xxx	Ack	Biquad 9
					xxxxx.xxx	Ack	Biquad 10

VU Meter Output = 0xFE

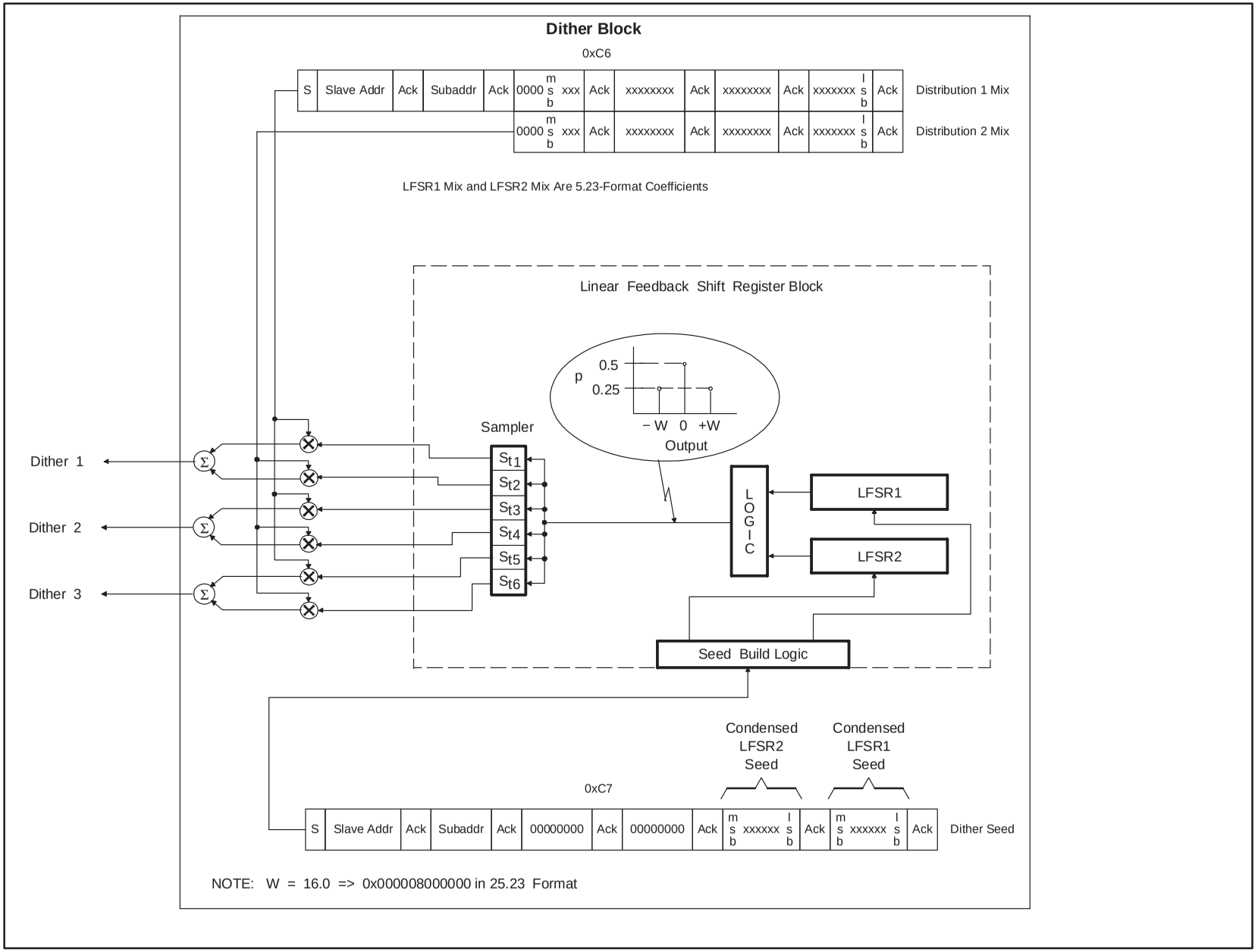
S	Slave Addr	Ack	Subaddr	Ack	xxxxx.xxx	Ack	
					xxxxx.xxx	Ack	VU Meter Output 1 (Biquad 5)
					xxxxx.xxx	Ack	VU Meter Output 1 (Biquad 6)

Spectrum Analyzer/VU Meter

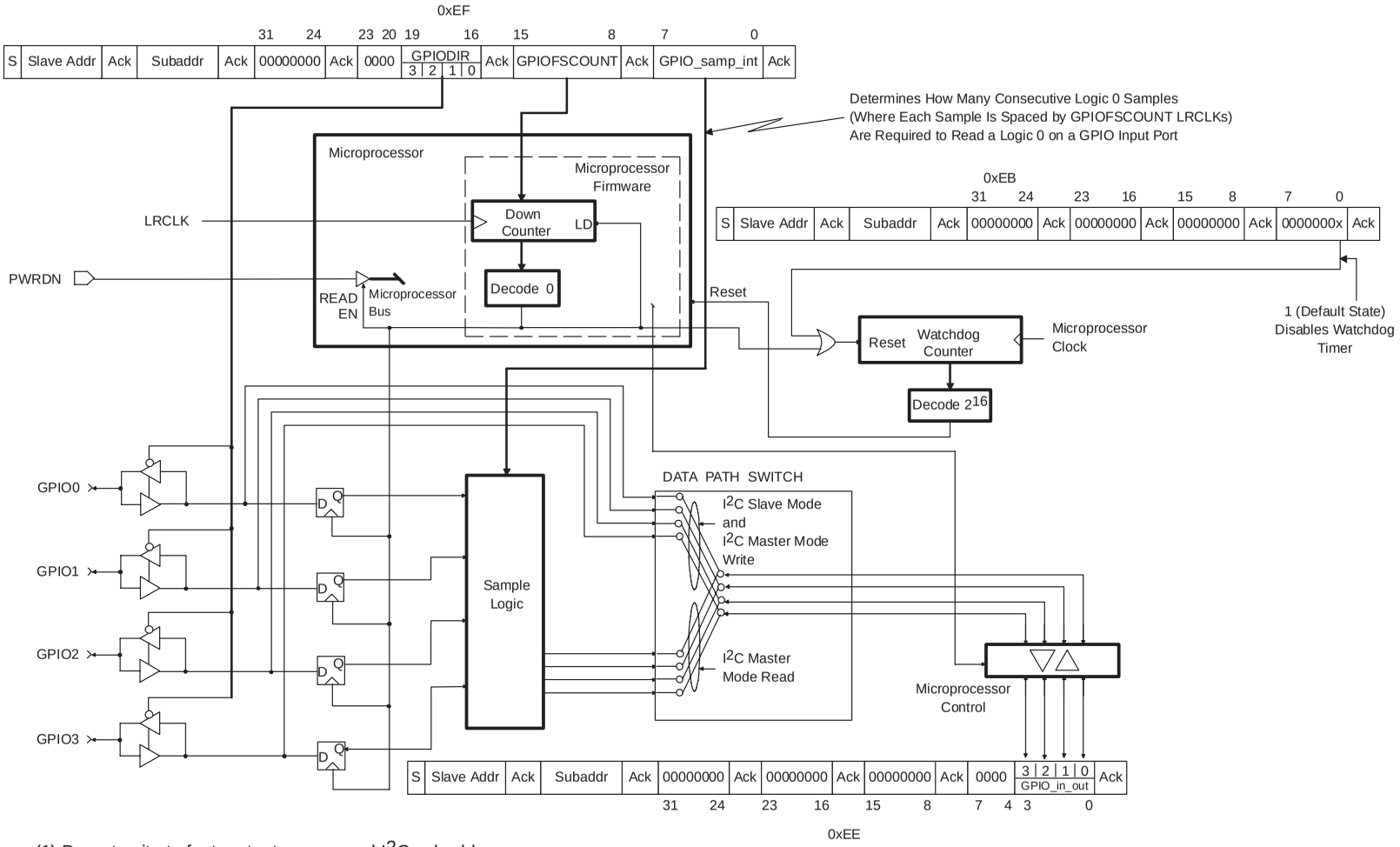


asa and (1 − asa) Set Time Window Over Which RMS Value Is Computed

$t_{Window} \approx -1/[F_S \times \ln(1 - asa)]$ Where F_S = Audio Sample Frequency

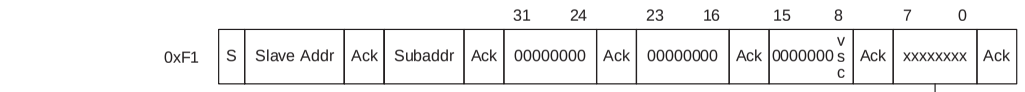

Dither Block
0xC6
S
Slave Addr
Ack
Subaddr
0000 msb xxx
xxxxxxxx
xxxxxxx lsb
Distribution 1 Mix
Distribution 2 Mix
LFSR1 Mix and LFSR2 Mix Are 5.23-Format Coefficients
Linear Feedback Shift Register Block
p
0.5
0.25
− W 0 +W
Output
Sampler
St1
St2
St3
St4
St5
St6
Dither 1
Dither 2
Dither 3
LOGIC
LFSR1
LFSR2
Seed Build Logic
Condensed LFSR2 Seed
Condensed LFSR1 Seed
0xC7
00000000
xxxxxx
Dither Seed
NOTE: W = 16.0 => 0x000008000000 in 25.23 Format

GPIO and Watchdog Timer Subaddresses	
SUBADDRESS(ES)	**PARAMETER(S)**
0xC8−0xC9	Factory Test Subaddresses (see Note 1)
0xCA−0xCF	SDIN4 Input Mixers
0xD0−0xD1	CH1/CH2 to CH3 *After Effects* Mixers
0xD2−0xEA	Reserved (see Note 1)



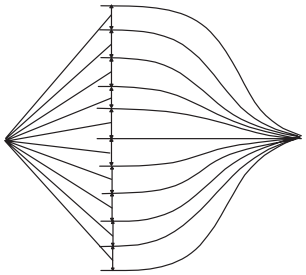
(1) Do not write to factory test or reserved I2C subaddresses.

SUBADDRESS(ES)	PARAMETER(S)
0xEC–0xED	Reserved/Factory Test Subaddresses
0xEE–0xEF—See Subaddress 0xEB	GPIO Port I/O Values and GPIO Parameters
0xF0—See Subaddress 0xA2	Master Mute/Unmute
0xF1—Also See Subaddress 0xA2 and Subaddress 0xF5	
0xF2−0xF4—See Subaddress 0xA2	CH1−CH3 Volume CMDS



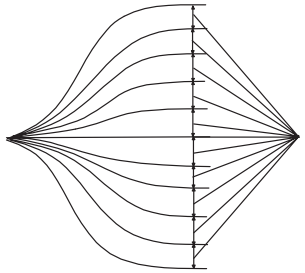
Treble and Bass
Slew Rate
TBLC[7:0]

Bass
Filter Set N



$t_{Transition}$ = TBLC[7:0] x 1/LRCLK

Treble
Filter Set N



$t_{Transition}$ = TBLC[7:0] x 1/LRCLK

Subaddress—Bass and Treble Shelf Filter Parameters

Treble/Bass Slew Rate Selection

													7		0
0xF1	S	Slave Addr	Ack	Subaddr	Ack	00000000	Ack	00000000	Ack	0000000	VSC	Ack	xxxxxxxx		Ack

Treble/Bass Slew Rate = TBLC
(Slew Rate = TBLC/Fs,
Where Fs = Audio Sample Rate)

Bass Filter Set Selection

								CH3		CH2		CH1			
0xF5	S	Slave Addr	Ack	Subaddr	Ack	00000000	Ack	00000xxx	Ack	00000xxx	Ack	00000xxx	Ack		

Treble Filter Set Selection

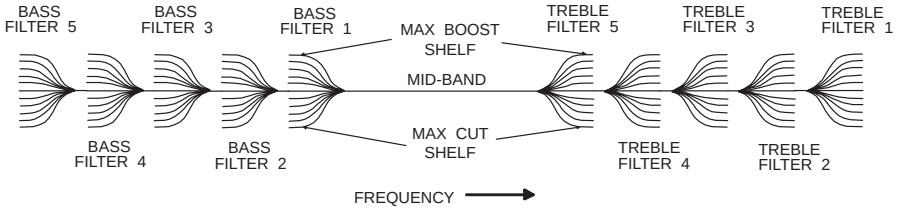
								CH3		CH2		CH1			
0xF7	S	Slave Addr	Ack	Subaddr	Ack	00000000	Ack	00000xxx	Ack	00000xxx	Ack	00000xxx	Ack		

Bass Shelf Selection (Filter Index)

								CH3		CH2		CH1			
0xF6	S	Slave Addr	Ack	Subaddr	Ack	00000000	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack		

Treble Shelf Selection (Filter Index)

								CH3		CH2		CH1			
0xF8	S	Slave Addr	Ack	Subaddr	Ack	00000000	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack		



Treble and Bass Filter Set Commands

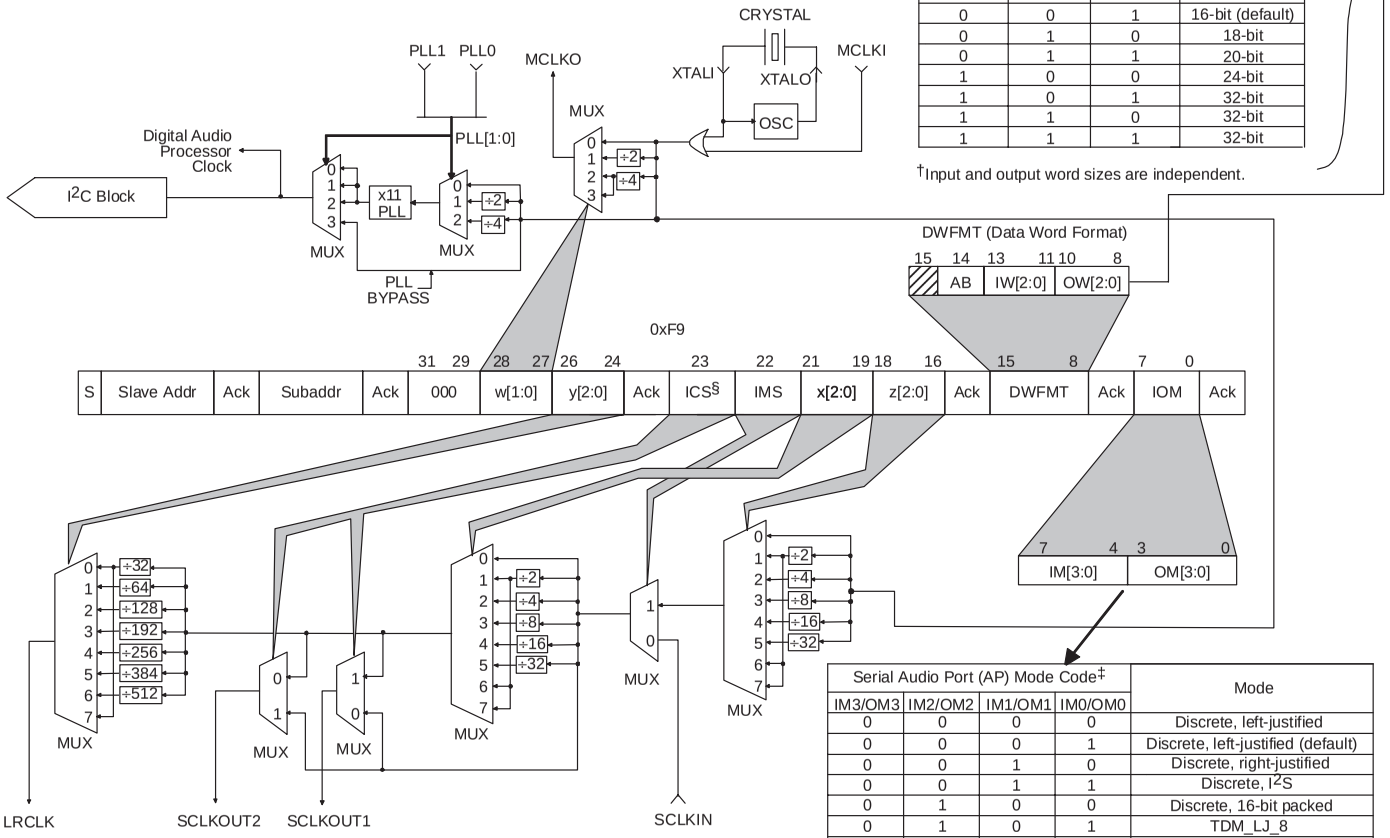
0 => No Change
1 – 5 => Filter Sets 1 – 5
6 – 7 => Illegal (Behavior Indeterminate)

Treble and Bass Filter Shelf Commands

0 => Illegal (Behavior Indeterminate)
1 – 150 => Filter Shelves 1 – 150
1 => +18-dB Boost
•
•
•
150 => −18-dB Cut
151 – 255 => Illegal (Behavior Indeterminate)

Fs (LRCLK)	3-dB CORNERS (kHz)									
	FILTER SET 5		FILTER SET 4		FILTER SET 3		FILTER SET 2		FILTER SET 1	
	BASS	TREBLE	BASS	TREBLE	BASS	TREBLE	BASS	TREBLE	BASS	TREBLE
96 kHz	0.25	6	0.5	12	0.75	18	1	24	1.5	36
88.4 kHz	0.23	5.525	0.46	11.05	0.691	16.575	0.921	22.1	1.381	33.15
64 kHz	0.167	4	0.333	8	0.5	12	0.667	16	1	24
48 kHz	0.125	3	0.25	6	0.375	9	0.5	12	0.75	18
44.1 kHz	0.115	2.756	0.23	5.513	0.345	8.269	0.459	11.025	0.689	16.538
32 kHz	0.083	2	0.167	4	0.25	6	0.333	8	0.5	12
24 kHz	0.063	1.5	0.125	3	0.188	4.5	0.25	6	0.375	9
22.05 kHz	0.057	1.378	0.115	2.756	0.172	4.134	0.23	5.513	0.345	8.269
16 kHz	0.042	1	0.083	2	0.125	3	0.167	4	0.25	6
12 kHz	0.031	0.75	0.063	1.5	0.094	2.25	0.125	3	0.188	4.5
11.025 kHz	0.029	0.689	0.057	1.378	0.086	2.067	0.115	2.756	0.172	4.134

DIGITAL AUDIO FORMAT AND CLOCK MANAGEMENT



AB assigns TDM time slots for those TDM outputs involving two TAS3103As. For these output formats, one of the TAS3103A chips must be defined as AB = 0. The other TAS3103A chip must be defined as AB = 1.

Word Size Code†			Word Size
IW2/OW2	IW1/OW1	IW0/OW0	
0	0	0	32-bit
0	0	1	16-bit (default)
0	1	0	18-bit
0	1	1	20-bit
1	0	0	24-bit
1	0	1	32-bit
1	1	0	32-bit
1	1	1	32-bit

†Input and output word sizes are independent.

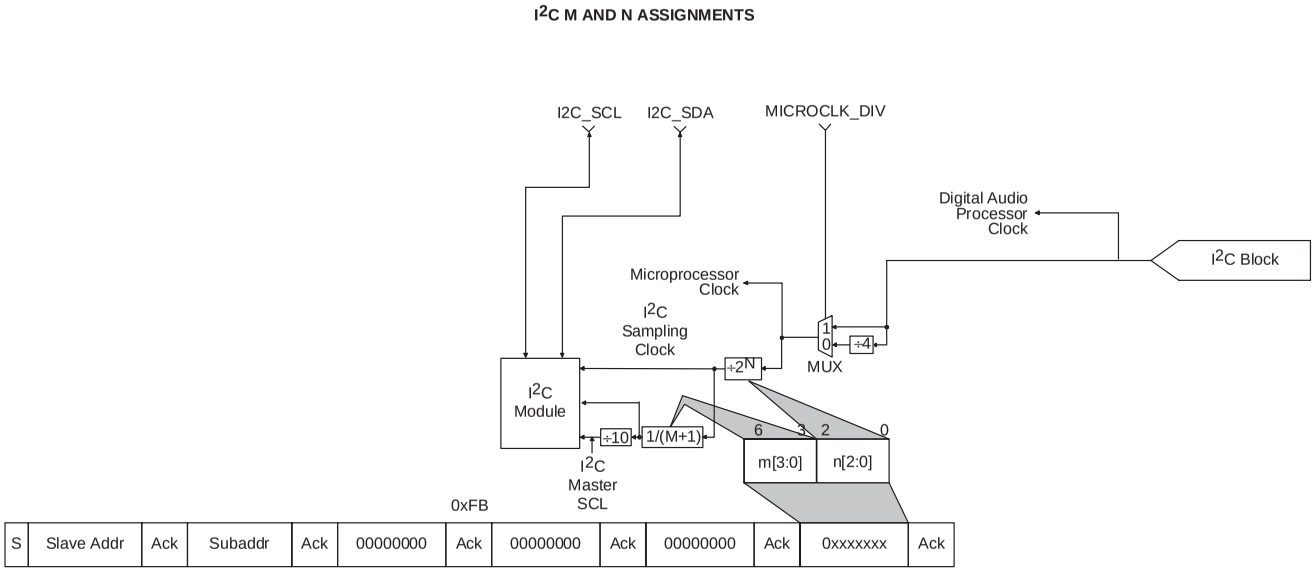
Serial Audio Port (AP) Mode Code‡				Mode
IM3/OM3	IM2/OM2	IM1/OM1	IM0/OM0	
0	0	0	0	Discrete, left-justified
0	0	0	1	Discrete, left-justified (default)
0	0	1	0	Discrete, right-justified
0	0	1	1	Discrete, I2S
0	1	0	0	Discrete, 16-bit packed
0	1	0	1	TDM_LJ_8
0	1	1	0	TDM_LJ_6
0	1	1	1	TDM_LJ_4
1	0	0	0	TDM_I2S_8
1	0	0	1	TDM_I2S_6
1	0	1	0	TDM_I2S_4
1	0	1	1	TDM_20Bit_6
1	1	0	0	6-CH, single-chip, crystal (LJ)
1	1	0	1	6-CH, single-chip (LJ)
1	1	1	0	6-CH, single-chip, crystal (I2S)
1	1	1	1	6-CH, single-chip, 20-bit

‡Input and output mode selections are independent.

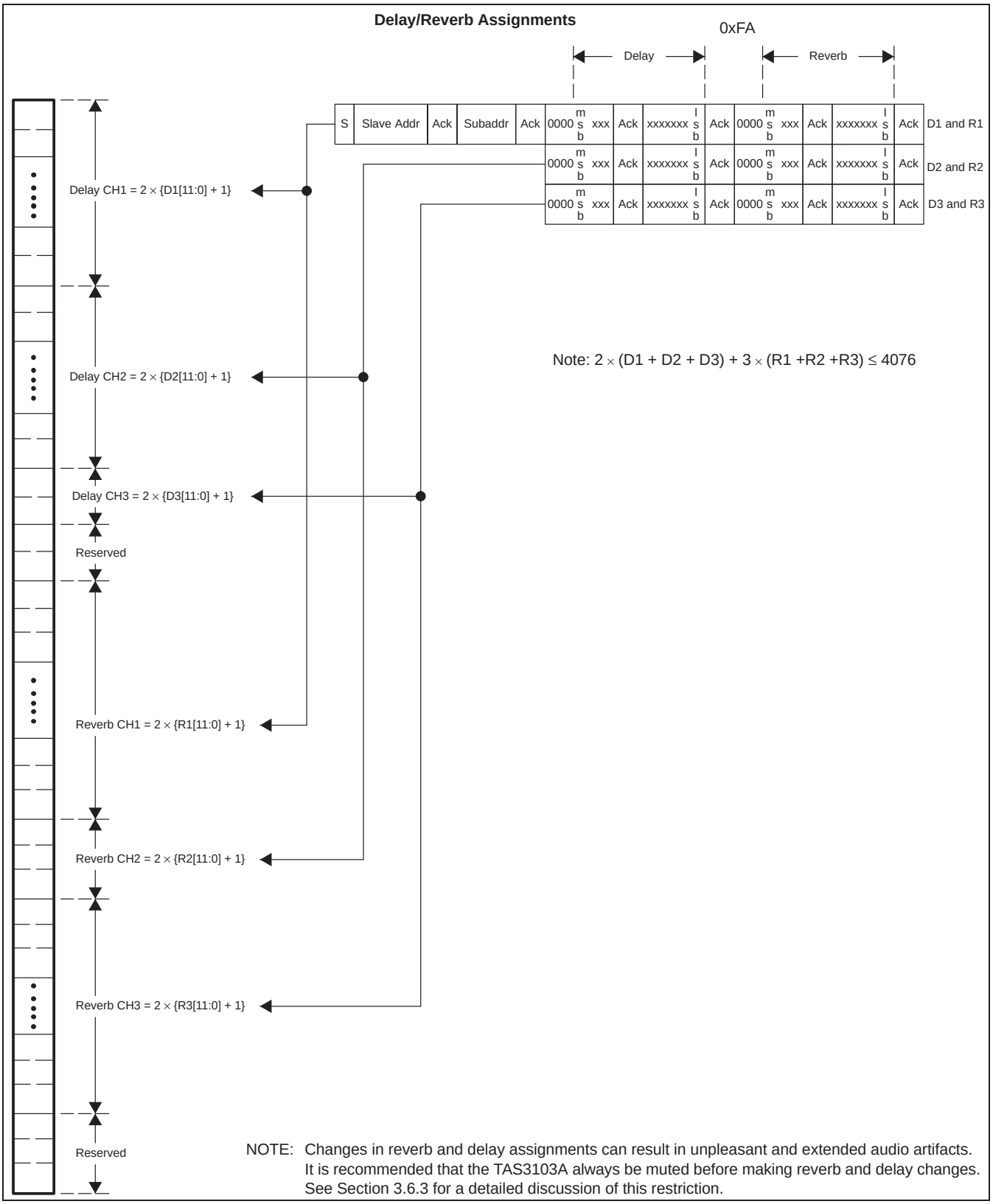
NOTE: See Section 2.1.1 for a detailed discussion of restrictions regarding updating of the 0xF9 register.

§ I2S MASTER/$\overline{\text{SLAVE}}$ bit

I²C M AND N ASSIGNMENTS

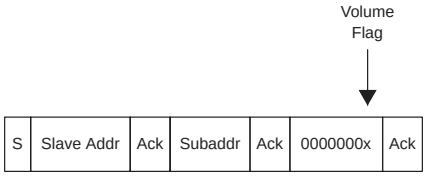


Delay/Reverb Assignments



Note: $2 \times (D1 + D2 + D3) + 3 \times (R1 + R2 + R3) \leq 4076$

NOTE: Changes in reverb and delay assignments can result in unpleasant and extended audio artifacts. It is recommended that the TAS3103A always be muted before making reverb and delay changes. See Section 3.6.3 for a detailed discussion of this restriction.

SUBADDRESS(ES)	PARAMETER(S)
0xFC—See Subaddress 0x00	Ending I^2C Check Word
0xFD−0xFE—See Subaddress 0xBB	Spectrum Analyzer/VU Meter Outputs
0xFF—Volume Busy Flag	
	• Volume Flag = 0 ⇒ No volume commands are active. • Volume Flag = 1 ⇒ One or more volume commands are active.

2 Hardware Architecture

Figure 2−1 depicts the hardware architecture of the chip. The architecture consists of five major blocks:

- Input serial audio port (SAP)
- Output serial audio port (SAP)
- DPLL and clock management
- Controller
- Digital audio processor (DAP)

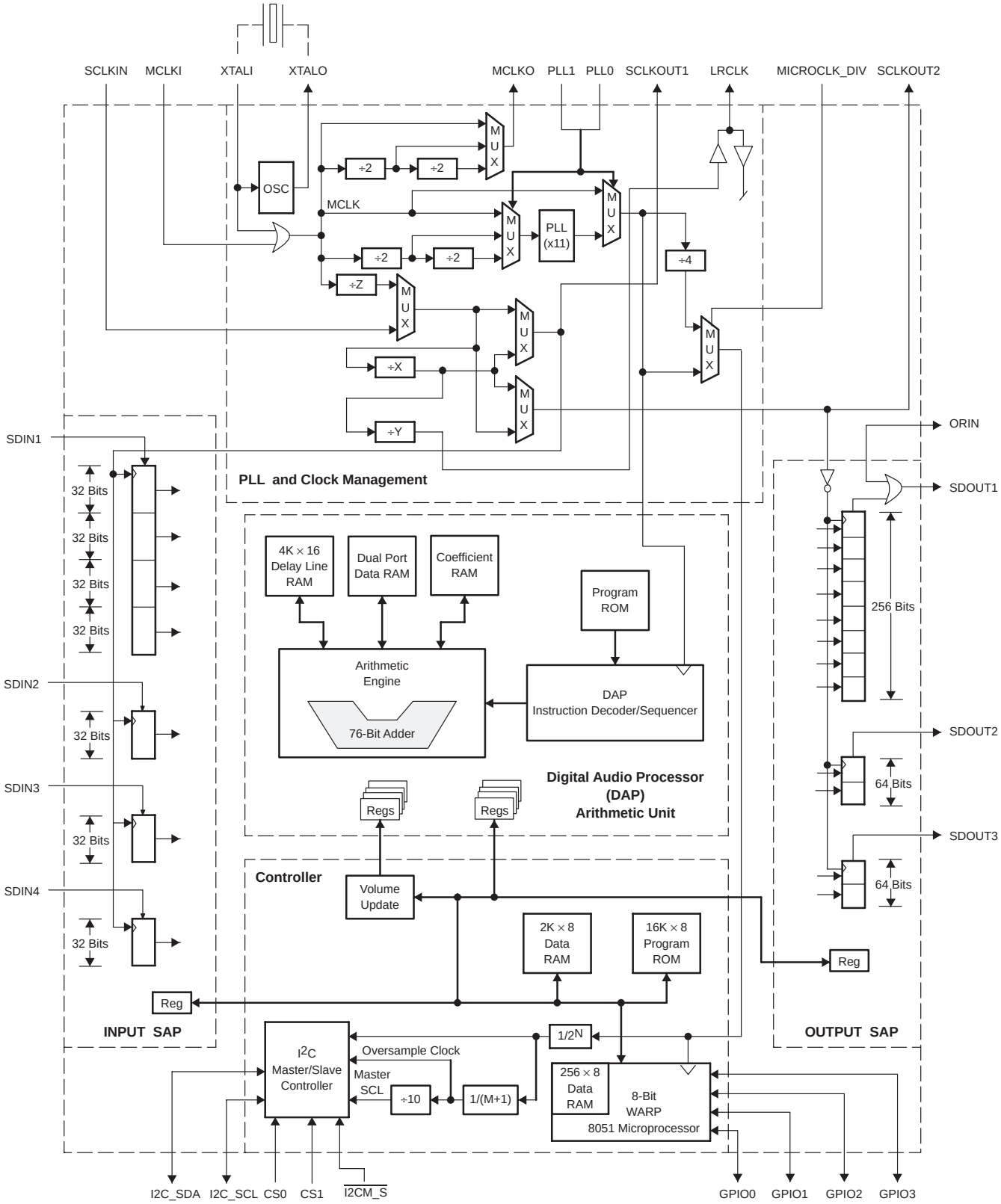


Figure 2−1. TAS3103A Detailed Hardware Block Diagram

2.1 Input and Output Serial Audio Ports (SAPs)

The TAS3103A accepts data in various serial data formats including left-/right-justified and I^{2}S, 16 through 32 bits, discrete, or time-division multiplex (TDM). Sample rates from 8 kHz through 96 kHz are supported. Each TAS3103A has four input serial ports and three output serial ports, labeled SDIN[4:1] and SDOUT[3:1], respectively. All ports accommodate stereo data formats, and SDIN1 and SDOUT1 also accommodate TDM data formats. The formats are selectable via I^{2}C commands. All input channels are assigned the same format and all output channels are assigned the same format; the two formats need not be the same. The TAS3103A can accommodate system architectures that require data format conversions without the need for additional glue logic. If a TDM format is selected for the input port, only SDIN1 is active; the other three input channels cannot be used. If a TDM format is selected for the output port, only SDOUT1 is active; the other two channels cannot be used.

2.1.1 SAP Configuration Options

The TAS3103A serial interface data format options for discrete (stereo) data are detailed in Figure 2−2.

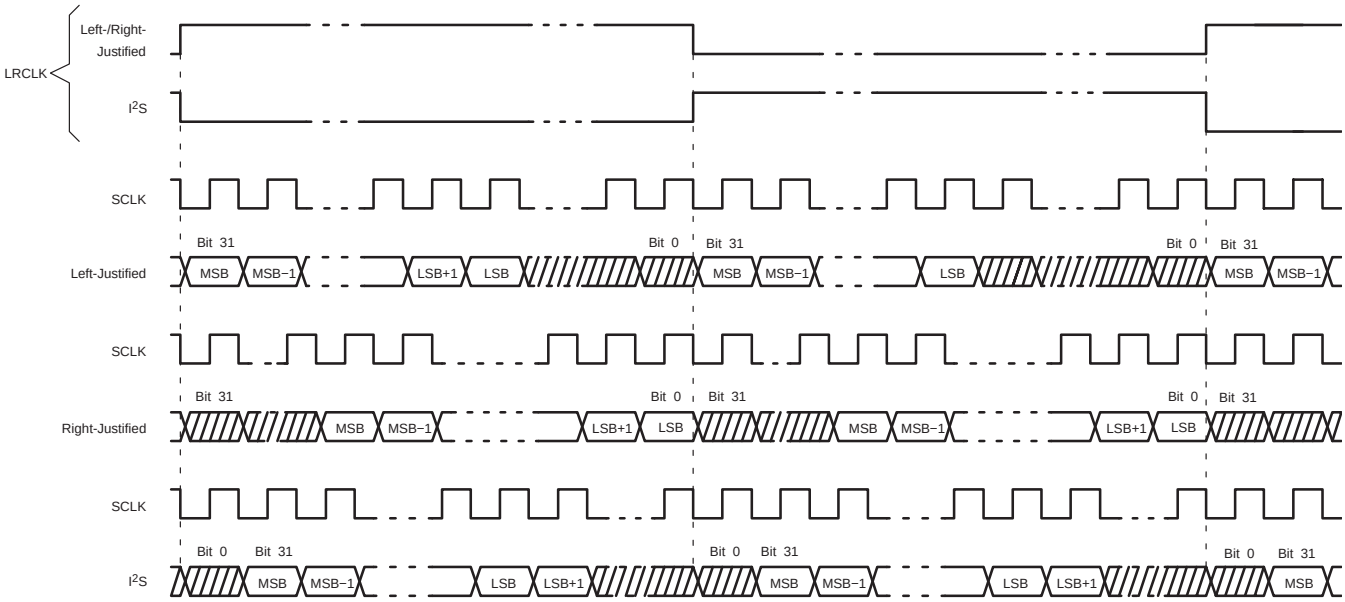


Figure 2−2. Discrete Serial Data Formats

When the TAS3103A is transmitting serial data, it uses the negative edge of SCLK to output a new data bit. The TAS3103A samples incoming serial data on the rising edge of SCLK.

The TDM modes on the TAS3103A only provide left-justified and I^{2}S formats, and each word in the TDM data stream adheres to the bit placement shown in Figure 2−2. Figure 2−3 illustrates the output data stream for a 4-channel TDM mode. Two cases are illustrated; an I^{2}S data format case (SAP output mode 1010) and a left-justified data format case (SAP output mode 0111).

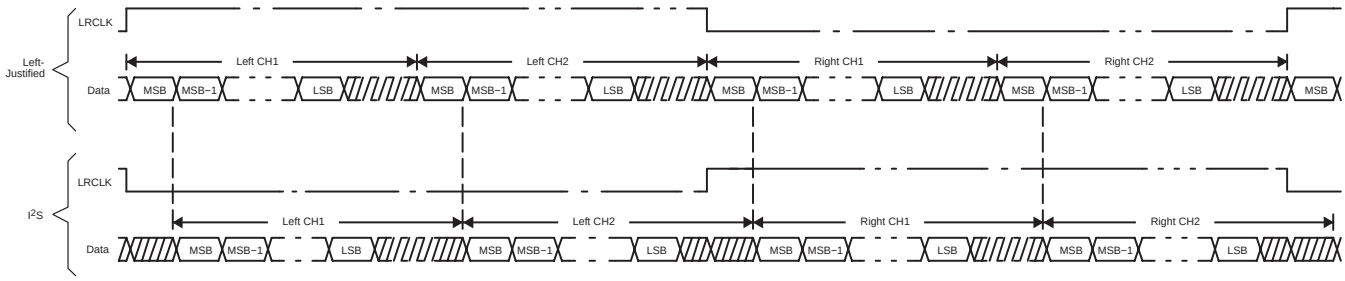


Figure 2−3. Four-Channel TDM Serial Data Formats

A 16-bit field contained in the 32-bit word located at I^{2}C subaddress 0xF9 configures both the input and output serial audio ports. Figure 2−4 illustrates the format of this 16-bit field. The data is shown in the transmitted I^{2}C protocol format, and thus, in addition to the data, the start bit S, the slave address, the subaddress, and the acknowledges required by every byte are also shown.

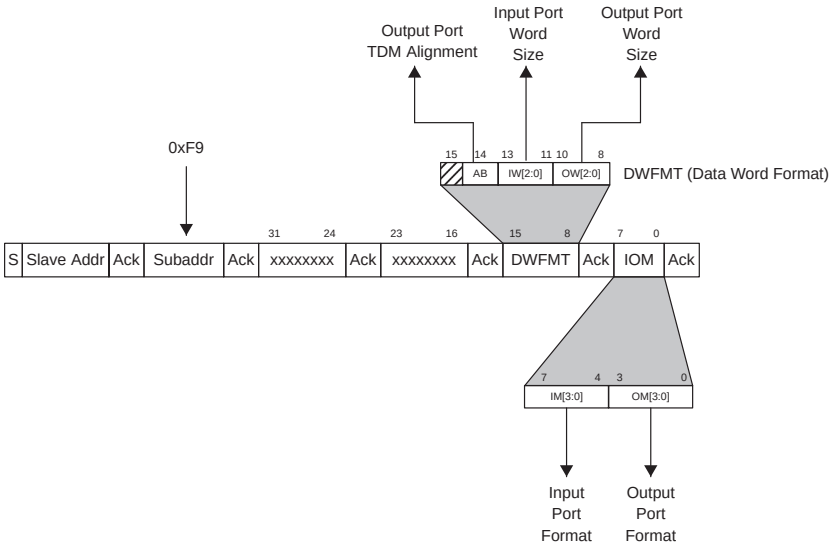
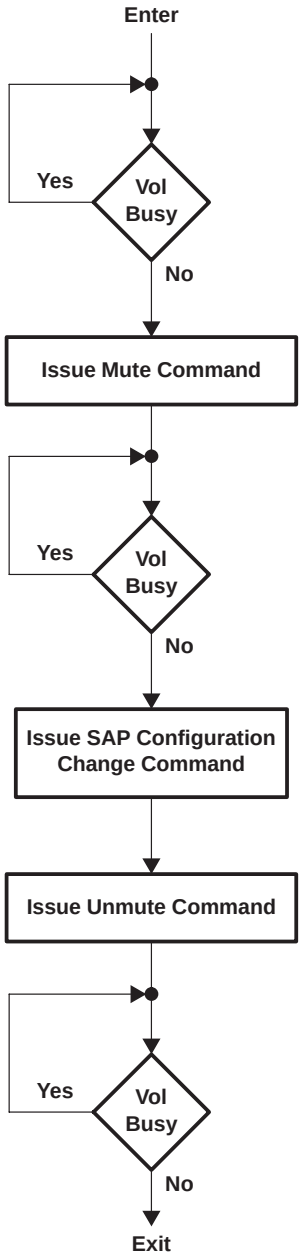


Figure 2−4. SAP Configuration Subaddress Fields

Commands to reconfigure the SAP cannot be issued as stand-alone commands, but must accompany mute and unmute commands. The reason for this is that an SAP configuration change while a volume, bass, or treble update is taking place can cause the update not to be completed properly. Figure 2−5 shows the recommended procedure for issuing SAP configuration update commands.

After a reset, ensure that 0xF9 is written before volume, treble, or bass.

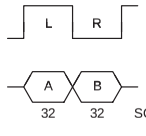
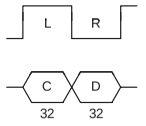
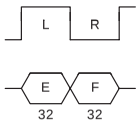
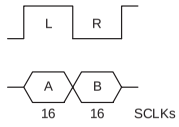
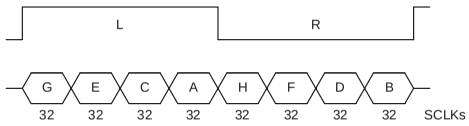
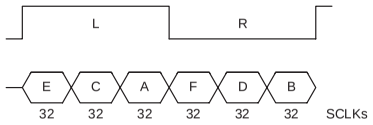
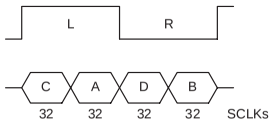
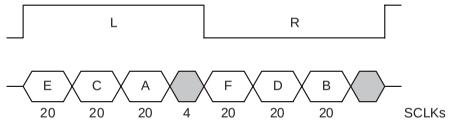
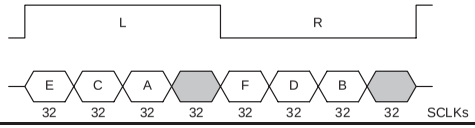


Mute Command = 0x00000007 at subaddress 0xF0
Unmute command = 0x00000000 at subaddress 0xF0
SAP configuration subaddress = 0xF9
Volume busy flag = LSB of subaddress 0xFF. Logic 1 = busy

Figure 2−5. Recommended Procedure for Issuing SAP Configuration Updates

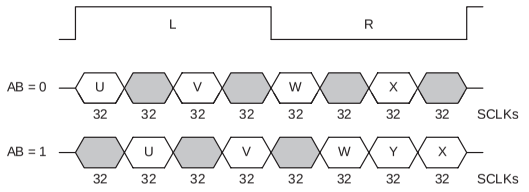
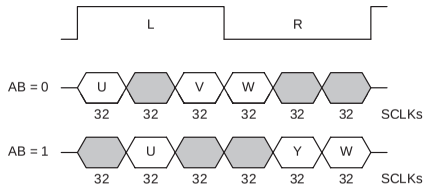
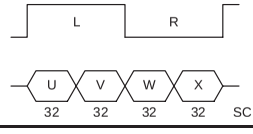
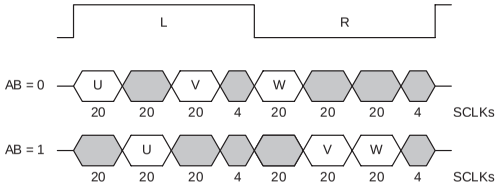
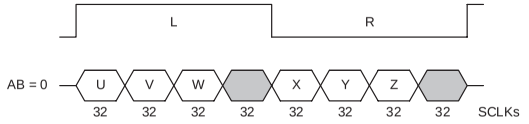
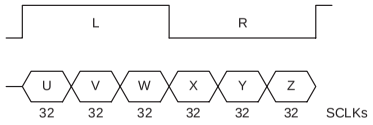
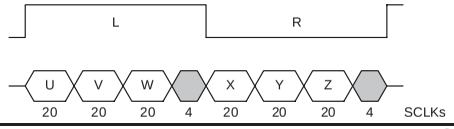
Figure 2−6, Figure 2−7, and Figure 2−8 tabularize the formatting and word size options available for the input SAP and output SAP. In these figures, data formats are paired when the only difference between the pair is whether the word placement within the LRCLK period is left-justified or I^2S. The TDM formats available include single-chip TDM output formats ($SDOUT1_{chipA}$ or $SDOUT1_{chipB}$) and two-chip TDM output formats ($SDOUT1_{chipA}$ ORed with $SDOUT1_{chipB}$). For two-chip TDM output formats, the ORing operation is accomplished by routing SDOUT1 from one of the two TAS3103A chips to ORIN of the other TAS3103A chip. The AB bit also comes into play for two-chip TDM formats; AB must be set to 1 on one of the two TAS3103A chips and set to 0 on the other TAS3103A chip. For a given connection of SDOUT1 to ORIN, it does not matter which TAS3103A is set up as chip AB = 1 and which chip is set up as chip AB = 0. However, the routing of the processed data to the output registers in the TAS3103A depends on which chip is chip AB = 0 and which chip is chip AB = 1. Figure 2−10 and Figure 2−11 illustrate this dependence.

In Figure 2−10 and Figure 2−11, the paired TDM output formats 0101 and 1000 are unique in that each format, in effect, services two distinct industry formats. For these two modes, if register Y in chip AB = 1 is set to zero (by appropriate output mixer coefficient settings), the resulting format is a standard 8-CH TDM format. This option is illustrated in Figure 2−7.

INPUT TYPE	IM[3:0]	FORMAT	WORD SIZE	DATA DISTRIBUTION: A, B, C, D, E, F, G, H = INPUT MIXER INPUTS		
				SDIN1	SDIN2	SDIN3
DISCRETE	000X(1)	Left-justified	All options valid			
	0010	Right-justified	All options valid			
	0011	I2S(5)	All options valid except 32-bit			
	0100	16-bit packed	IW[2:0] = 001		Not available	Not available
TIME-DIVISION MULTIPLEX (TDM)	0101	8-CH transfer, left-justified	All options valid		Not available	Not available
	1000	8-CH transfer, I2S(5)	All options valid except 32-bit		Not available	Not available
	0110/1101 (2)(3)	6-CH, left-justified	All options valid		Not available	Not available
	1001(3)	6-CH, I2S(5)	All options valid except 32-bit		Not available	Not available
	0111	4-CH, left-justified	All options valid		Not available	Not available
	1010	4-CH, I2S(5)	All options valid except 32-bit		Not available	Not available
	1011/1111(4)	6-CH, 20-bit	IW[2:0] = 011		Not available	Not available
	1100	6-CH data, 8 CH transfer, left-justified	All options valid		Not available	Not available
	1110	6-CH data, 8 CH transfer, I2S(5)	All options valid except 32-bit		Not available	Not available

NOTES: 1. Left-justified, stereo is the default input format.
2. IM[3:0] modes 0110 and 1101 are identical for the input SAP. OM[3:0] modes 0110 and 1101 do produce different results in the output SAP (see Figure 2−7).
3. If a 6 CH input format is selected, the output format must also be set to 6 CH. When in a 6 CH mode, data format selections—I2S and left-justified—for the 6 CH input SAP can be made independent of the data format selections—I2S and left-justified—made for the 6 CH output SAP.
4. IM[3:0] modes 1011 and 1111 are identical for the input SAP. OM[3:0] modes 1011 and 1111 do produce different results in the output SAP (see Figure 2−7).
5. LRCLK for the I2S format is ⌊L⌋R⌉.

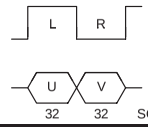
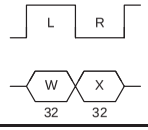
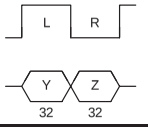
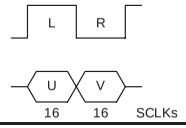
Figure 2−6. Format Options: Input Serial Audio Port

OUTPUT TYPE	OM[3:0]	FORMAT	WORD SIZE	DATA DISTRIBUTION: U, V, W, X, Y, Z = OUTPUT MIXER OUTPUTS		
				SDOUT1	SDOUT2	SDOUT3
TIME-DIVISION MULTIPLEX (TDM)	0101	8-CH, 2-chip, left-justified	All options valid		Not available	Not available
	1000	8-CH, 2-chip, I2S(2)	All options valid except 32-bit		Not available	Not available
	0110(1)	6-CH, 2-chip, left-justified	All options valid		Not available	Not available
	1001(1)	6-CH, 2-chip, I2S(2)	All options valid except 32-bit		Not available	Not available
	0111	4-CH, left-justified	All options valid		Not available	Not available
	1010	4-CH, I2S(2)	All options valid except 32-bit		Not available	Not available
	1011	6-CH, 2-chip, 20-bit	OW[2:0] = 011		Not available	Not available
	1100	6-CH data, 8-CH transfer, left-justified	All options valid		Not available	Not available
	1110	6-CH data, 8-CH transfer, I2S(2)	All options valid except 32-bit		Not available	Not available
	1101(1)	6-CH, left-justified	All options valid		Not available	Not available
	1111	6-CH, 20-bit	OW[2:0] = 011		Not available	Not available

NOTES: 1. If a 6-CH output format is selected, the input format must also be set to 6-CH. When in a 6-CH mode, data format selections—I2S and left-justified—for the output SAP can be made independent of the data format selections—I2S and left-justified—that are made for the input SAP

2. LRCLK for the I2S format is ⌊L⌈R⌉.

Figure 2−7. TDM Format Options: Output Serial Audio Port

OUTPUT TYPE	OM[3:0]	FORMAT	WORD SIZE	DATA DISTRIBUTION: U, V, W, X, Y, Z = OUTPUT MIXER OUTPUTS		
				SDOUT1	SDOUT2	SDOUT3
DISCRETE	000X(1)	Left-justified	All options valid			
	0010	Right-justified	All options valid			
	0011	I2S(2)	All options valid except 32-bit			
	0100	16-bit packed	OW[2:0] = 001		Not available	Not available

NOTES: 1. Left-justified, stereo is the default input format.
2. LRCLK for the I2S format is ⌊L⌈R⌋.

Figure 2–8. Discrete Format Options: Output Serial Audio Port

	SAMPLE SIZE	INPUT IW[2:0]			OUTPUT OW[2:0]		
		IW2	IW1	IW0	OW2	OW1	OW0
	(32)	0	0	0	0	0	0
DEFAULT →	16 Bit	0	0	1	0	0	1
	18 Bit	0	1	0	0	1	0
	20 Bit	0	1	1	0	1	1
	24 Bit	1	0	0	1	0	0
	32 Bit	1	0	1	1	0	1
	(32)	1	1	0	1	1	0
	(32)	1	1	1	1	1	1

(32) ⇒ Reserved for future family members. Selection of 000, 110, or 111 in the TAS3103A selects a 32-bit sample size.

Figure 2–9. Word-Size Settings

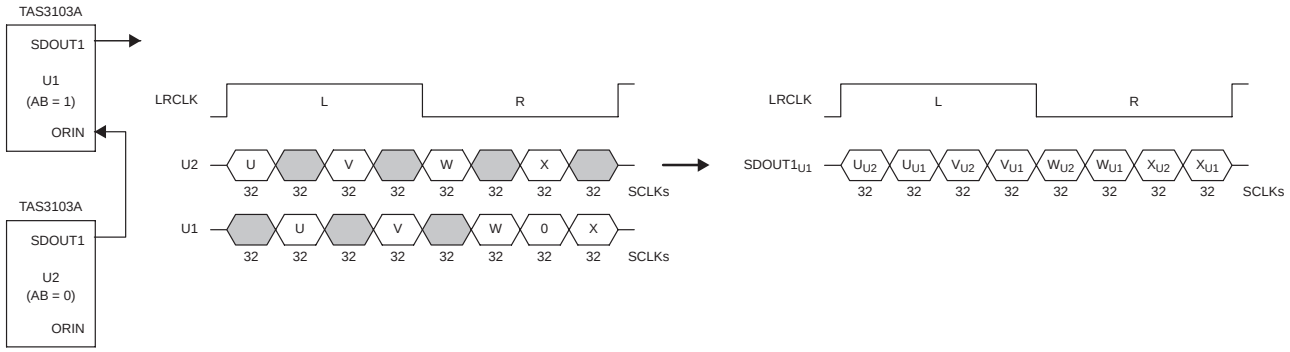


Figure 2−10. 8-CH TDM Format Using SAP Modes 0101 and 1000

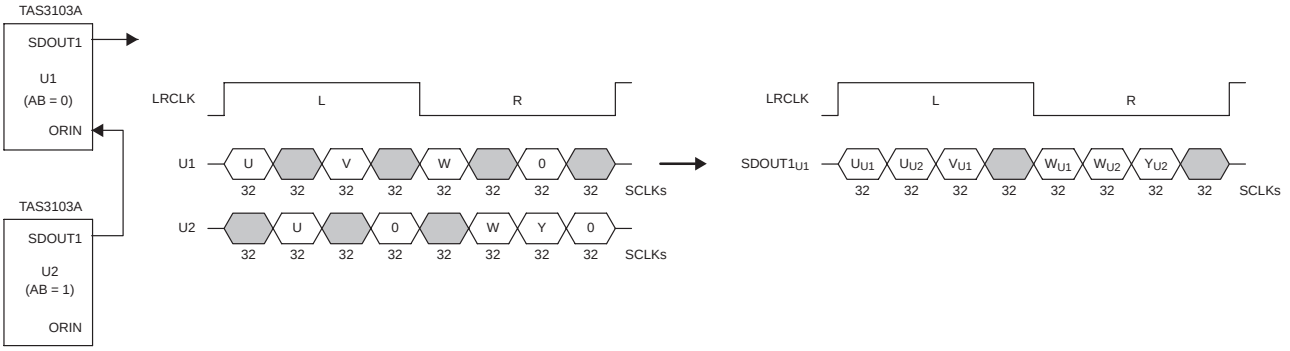


Figure 2−11. 6-CH Data, 8-CH Transfer TDM Format Using SAP Modes 0101 and 1000

For these same two modes, if register X in chip AB = 0 is set to zero, and registers V and X in chip AB = 1 are set to zero, the resulting format is a 6-CH data, 8-CH transfer format. This option is shown in Figure 2−11.

The data output format in Figure 2−11 is identical to that realized using data output formats 1100 and 1110 in Figure 2−7. The difference is that SAP modes 1010 and 1000 provide six independent monaural channels to process the data, whereas SAP modes 1100 and 1110 provide only three independent monaural channels to process the data.

2.1.2 Processing Flow—SAP Input to SAP Output

All SAP data format options other than I^2S result in a two-sample delay from input to output, as illustrated in Figure 2−12. Figure 2−12 is also relevant if I^2S formatting is used for both the input SAP and the output SAP (the polarity of LRCLK in Figure 2−12 must be inverted in this case). However, if I^2S format conversions are performed between input and output, the delay becomes either 1.5 samples or 2.5 samples, depending on the processing clock frequency selected for the digital audio processor (DAP) relative to the sample rate of the incoming data. The input-to-output delay for an I^2S input format and a non-I^2S output format is illustrated in Figure 2−13(a), and Figure 2−13(b) illustrates the delay for a non-I^2S input format and an I^2S output format. In each case, two distinct input-to-output delay times are shown: a 1.5-sample delay time if the processing time in the DAP is less than half the sample period, and a 2.5-sample delay time if the processing time in the DAP is greater than half the sample period.

The departure from the two-sample input-to-output processing delay when I^2S format conversions are performed is due to the use of a common LRCLK. The I^2S format uses the falling edge of LRCLK to begin a sample period, whereas all other formats use the rising edge of LRCLK to begin a sample period. This means that the input SAP and digital audio processor (DAP) operate on sample windows that are 180° out of phase with respect to the sample window used by the output SAP. This phase difference results in the output SAP outputting a new data sample at the midpoint of the sample period used by the DAP to process the data. If the processing cycle completes all processing tasks before the midpoint of the processing sample period, the output SAP outputs this processed data. However, if the processing time extends past the midpoint of the processing sample period, the output SAP outputs the data processed during the previous processing sample period. In the former case, the delay from input to output is 1.5 samples. In the latter case, the delay from input to output is 2.5 samples.

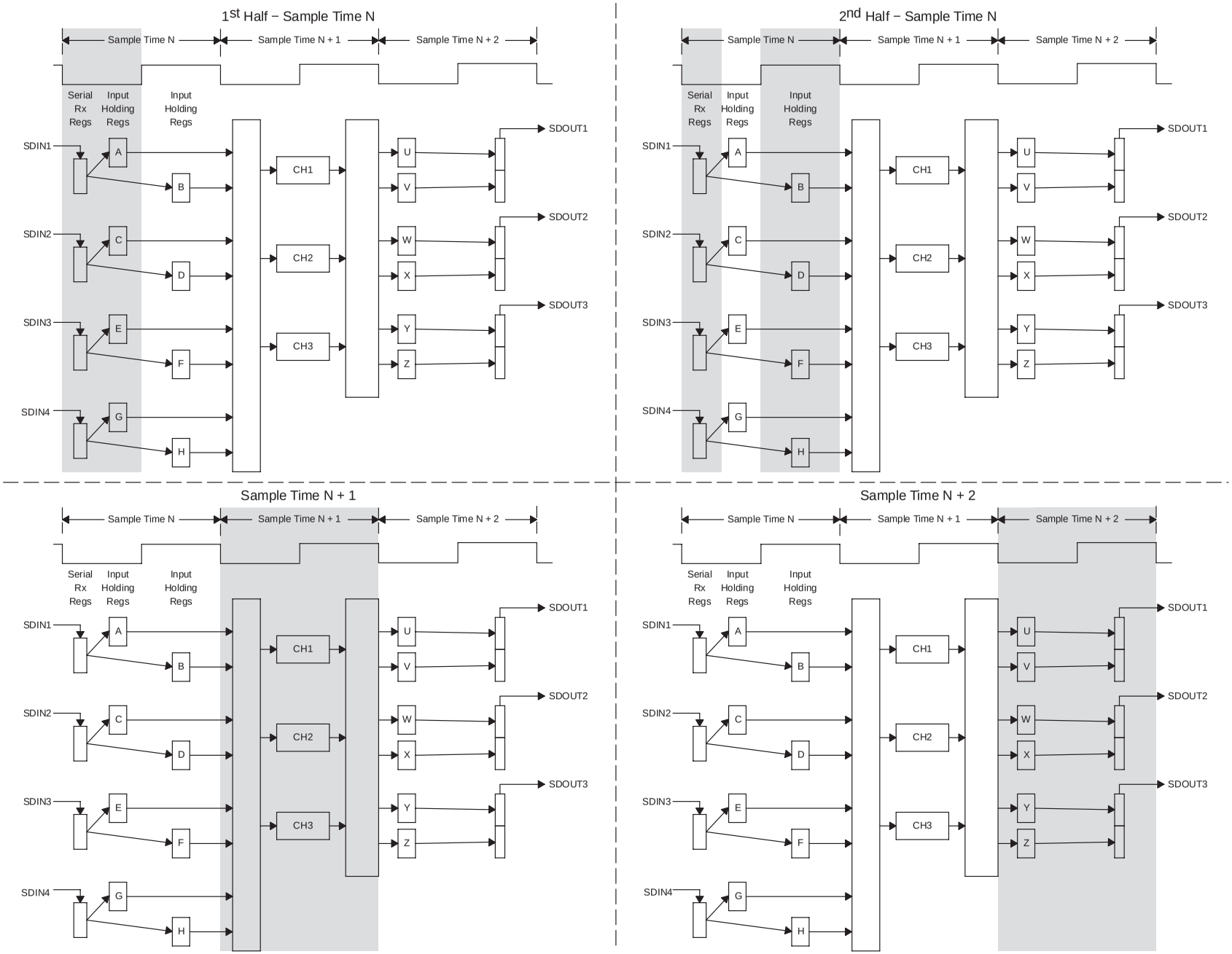
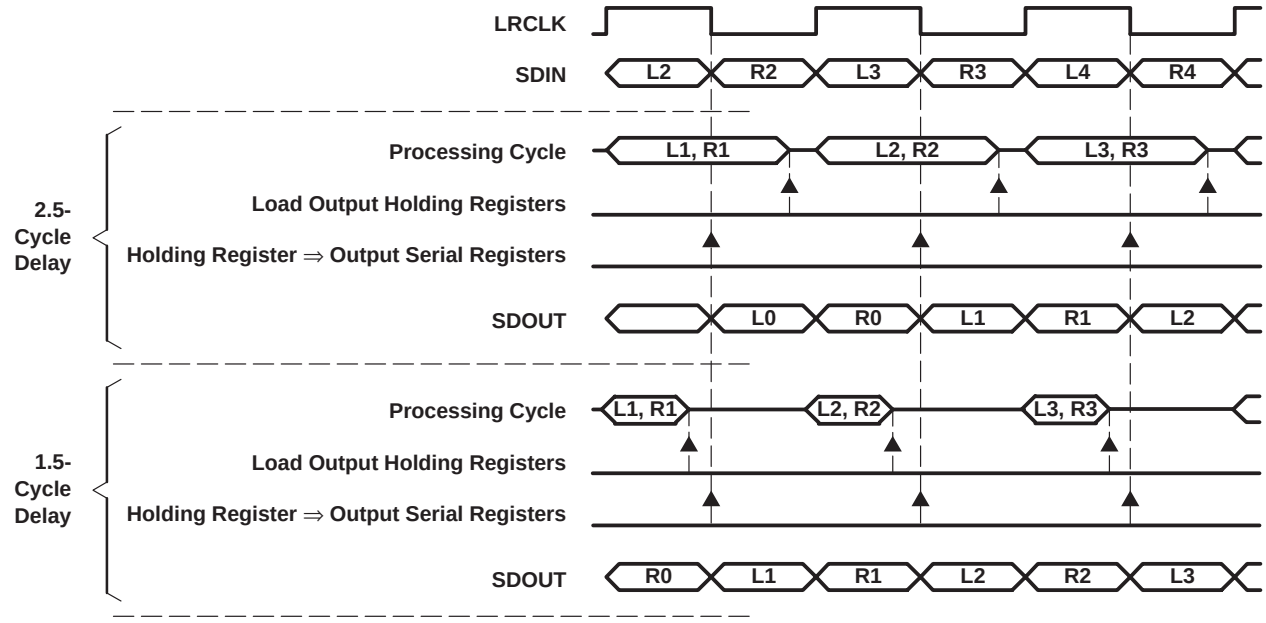
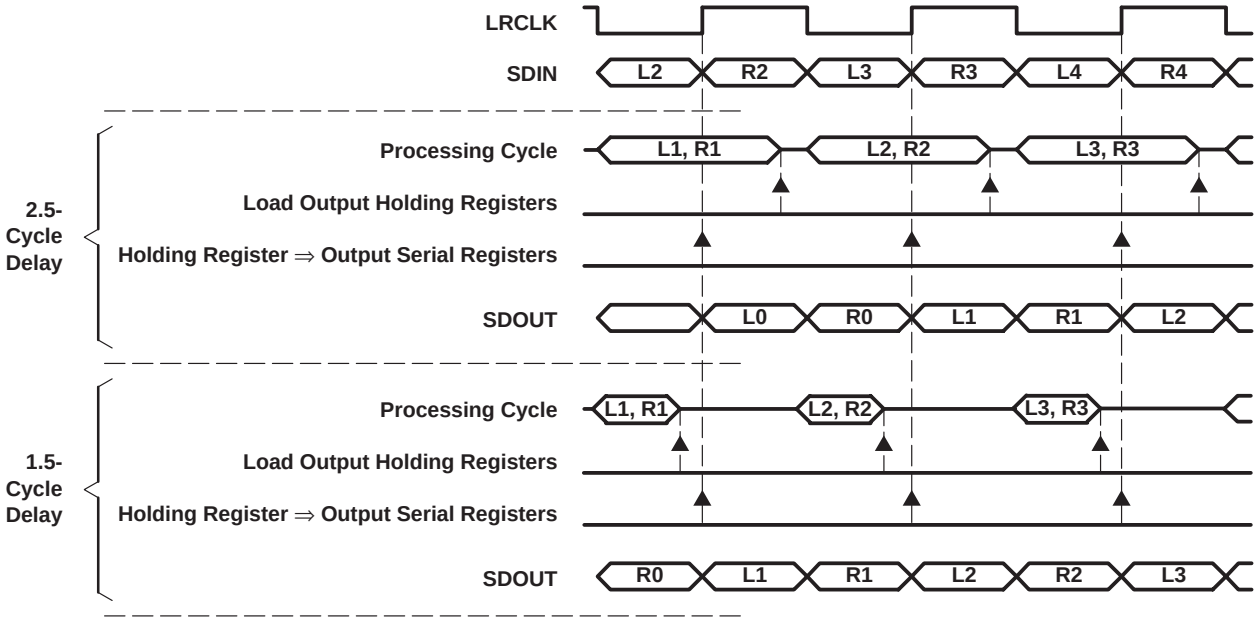


Figure 2–12. SAP Input-to-Output Latency



(a) Left-Justified Input / I²S Output



(b) I²S Input / Left-Justified Output

Figure 2−13. SAP Input-to-Output Latency for I²S Format Conversions

The delay from input to output can thus be either 1.5 or 2.5 sample times when data format conversions are performed that involve the I²S format. However, which delay time is obtained for a particular application is determinable and fixed for that application, providing care is taken in the selection of MCLKI/XTALI with respect to the incoming sample clock LRCLK.

Table 2−1 lists all viable clock selections for a given audio sample rate (LRCLK). The table only includes those clock choices that allow enough processing throughput to accomplish all tasks within a given sample time (T_S = 1/LRCLK). For each entry in the table, the DAP processing time is given in terms of whether the time is greater than 0.5 T_S (resulting in an input-to-output delay of 2.5 T_S), or less than 0.5 T_S (resulting in an input-to-output delay of 1.5 T_S).

Table 2−1 is valid for both master and slave I^2S modes (bit IMS at subaddress 0xF9 determines I^2S master/slave selection—see *DPLL and Clock Management*, Section 2.2). For all applications, MCLK must be ≥128 LRCLK (Fs). In the I^2S master mode, MCLK, SCLK (I^2S bit clock), and LRCLK are all harmonically related. Furthermore, in the I^2S master mode, if a master clock value given in Table 2−1 is used, the latency realized in performing I^2S format conversions, 1.5 samples or 2.5 samples, is stable and fixed over the duration of operation. However, greater care must be taken for the I^2S slave mode. In this mode, the device has the proper operational throughput to perform all required computations as long as MCLK is ≥128 LRCLK. But there is not a requirement that MCLK be harmonically related to SCLK and LRCLK. Values of MCLK could be chosen such that the output dithers between latencies of 1.5 and 2.5 sample times. There may be cases where part of the data stream output exhibits sample time latencies of 1.5 T_S and the other portion of the output data stream exhibits sample time latencies of 2.5 T_S. To ensure that such cases do not happen in the I^2S slave mode, the relationships between MCLK and LRCLK given in Table 2−1 should be followed for data format conversions involving the I^2S format. The MCLKI/XTALI frequencies given in Table 2−1 (if set to within ±5% of the nominal value shown) ensure that the DAP processing time falls above 0.5 T_S or below 0.5 T_S with enough margin to ensure that there is no race condition between the outputting of data and the completion of the processing tasks.

Table 2−1. TAS3103A Throughput Latencies vs MCLK and LRCLK

AUDIO SAMPLE RATE (LRCLK)	MASTER CLOCK(2) (MCLKI/XTALI)	DAP(1) CLOCK (PLL_OUTPUT)	DAP CLOCK CYCLES/LRCLK	DAP PROCESSING TIME	THROUGHPUT DELAY
96 kHz	24.576 MHz, 12.288 MHz	135.168 MHz	1408	> T_S/2	2.5 T_S
88.2 kHz	22.5792 MHz, 11.2896 MHz	124.1856 MHz	1408	> T_S/2	2.5 T_S
48 kHz	24.576 MHz, 12.288 MHz	135.168 MHz	2816	< T_S/2	1.5 T_S
	24.576 MHz, 12.288 MHz, 6.144 MHz	67.584 MHz	1408	> T_S/2	2.5 T_S
44.1 kHz	22.5792 MHz, 11.2896 MHz	124.1856 MHz	2816	< T_S/2	1.5 T_S
	22.5792 MHz, 11.2896 MHz, 5.6448 MHz	62.0928 MHz	1408	> T_S/2	2.5 T_S
32 kHz	16.384 MHz, 8.192 MHz	90.112 MHz	2816	< T_S/2	1.5 T_S
	16.384 MHz, 8.192 MHz, 4.096 MHz	45.056 MHz	1408	> T_S/2	2.5 T_S
24 kHz	24.576 MHz, 12.2858 MHz	135.168 MHz	5632	< T_S/2	1.5 T_S
	24.576 MHz, 12.2858 MHz, 6.144 MHz	67.584 MHz	2816	< T_S/2	1.5 T_S
	12.288 MHz, 6.144 MHz, 3.072 MHz	33.792 MHz	1408	> T_S/2	2.5 T_S
22.05 kHz	22.5792 MHz, 11.2896 MHz	124.1856 MHz	5632	< T_S/2	1.5 T_S
	22.5792 MHz, 11.2896 MHz, 5.6448 MHz	62.0928 MHz	2816	< T_S/2	1.5 T_S
	11.2896 MHz, 5.6448 MHz, 2.8224 MHz	31.0464 MHz	1408	> T_S/2	2.5 T_S
8 kHz	24.576 MHz, 12.288 MHz	135.168 MHz	16896	< T_S/2	1.5 T_S
	24.576 MHz, 12.288 MHz, 6.144 MHz	67.584 MHz	8448	< T_S/2	1.5 T_S
	12.288 MHz, 6.144 MHz, 3.072 MHz	33.792 MHz	4224	< T_S/2	1.5 T_S
	6.144 MHz, 3.072 MHz, 1.536 MHz	16.896 MHz	2112	> T_S/2	2.5 T_S

NOTES: 1. DAP clock is the internal digital audio processor clock. It is equal to 11 × MCLK1/XTALI, 11/2 × MCLKI/XTALI, or 11/4 × MCLKI/XTALI (as determined by a bit field in I^2C subaddress 0xF9). The DAP clock must always be greater than or equal to 1400 Fs (LRCLK).
2. Unless in PLL bypass, MCLKI must be ≤ 20 MHz.
3. XTALI must always be ≤ 20 MHz.

2.2 DPLL and Clock Management

Clock management for the TAS3103A consists of two control structures:

- Master clock management: oversees the selection of the clock frequencies for the microprocessor, the I^2C controller, and the digital audio processor (DAP). The master clock (MCLKI or XTALI) serves as the source for these clocks. In most applications, the master clock is input to an on-chip digital phase-locked loop (DPLL), and the DPLL output is used to drive the microprocessor and DAP clocks. A DPLL bypass mode can also be used, in which case the master clock is used to drive the microprocessor and DAP clocks.
- Serial audio port (SAP) clock management: oversees SAP master/slave mode, the settings of SCLKOUT1 and SCLKOUT2, and the setting of LRCLK in the SAP master mode.

Figure 2−15 illustrates the clock circuitry in the TAS3103A. The bold lines in Figure 2−15 highlight the default settings at power turnon, or after a reset. Inputs MCLKI and XTALI source the master clock for the TAS3103A. Within the TAS3103A, these two inputs are combined by an OR gate, and thus only one of these two sources can be active at any one time. The source that is not active must be set to logic 0. In normal operation, the master clock is divided by 1, 2, or 4 as determined by the logic levels set at input pins PLL0 and PLL1 (the state of PLL0, PLL1, and MICROCLK_DIV should only be changed while the TAS3103A $\overline{\text{RST}}$ is held LOW) and then multiplied by 11 in frequency by the on-chip DPLL. The DPLL output (or MCLKI/XTALI if the DPLL is bypassed) is the processing clock used by the digital audio processor (DAP).

The DAP processing clock can also serve as the clock for the on-chip microprocessor, or the DAP clock can be divided by four prior to sending it to the microprocessor. The input pin MICROCLK_DIV makes this clock choice. A logic 1 input level on this pin selects the DAP clock for the microprocessor clock; a logic 0 input level on this pin selects the DAP clock/4 for the microprocessor clock. Table 2−2 lists the primary clock modes of the TAS3103A.

Table 2−2. Sample Rate and CLK Ratios

Sample Rate	DAP Cycles Req'd	MCLK RATIO					
		768	512	384	256	192	128
PLL[1:0] = 00							
96	134,400						135.168
88.1	123,340						124.0448
48	67,200				135.168	101.376	67.584
44.1	61,740				124.1856	93.1392	62.0928
32	44,800			135.168	90.112	67.584	45.056
24	33,600		135.168	101.376	67.584	50.688	33.792
22.1	30,940		124.4672	93.3504	62.2336	46.6752	31.1168
8	11,200	67.584	45.056	33.792	22.528	16.896	
PLL[1:0] = 01							
96	134,400				135.168		
88.1	123,340				124.0448		
48	67,200		135.168	101.376	67.584		
44.1	61,740		124.4672	93.1392	62.0928		
32	44,800	135.168	90.112	67.584	45.056		
24	33,600	101.376	67.584	50.688	33.792		
22.1	30,940	93.3504	62.2336	46.6752	31.1168		
8	11,200	33.792	22.528	16.896			
PLL[1:0] = 10							
96	134,400						
88.1	123,340						
48	67,200		67.584				
44.1	61,740		62.0928				
32	44,800	67.584	45.056				
24	33,600	50.688	33.792				
22.1	30,940	46.6752	31.1168				
8	11,200	16.896	11.264				

2.2.1 TAS3103A Sample-Rate Changes

The TAS3103A supports dynamic sample rate changes when a fixed-frequency master clock is provided. During dynamic sample-rate changes, the TAS3103A remains in normal operation and the register contents are preserved. To avoid producing audio artifacts during sample-rate changes, the volume or mute control may be included in the application firmware that can mute the output signal during the sample-rate change. The fixed-frequency clock can be provided by a crystal attached to XTLI and XTLO, or an external fixed-frequency master clock attached to MCLKI. When the TAS3103A is used in a system in which the master clock frequency can change, then any time MCLK is stopped or the frequency is changed, reset should be applied. If the system master clock frequency changes or stops, reset should be applied. In these cases, the following procedures should be used (see Figure 2−14).

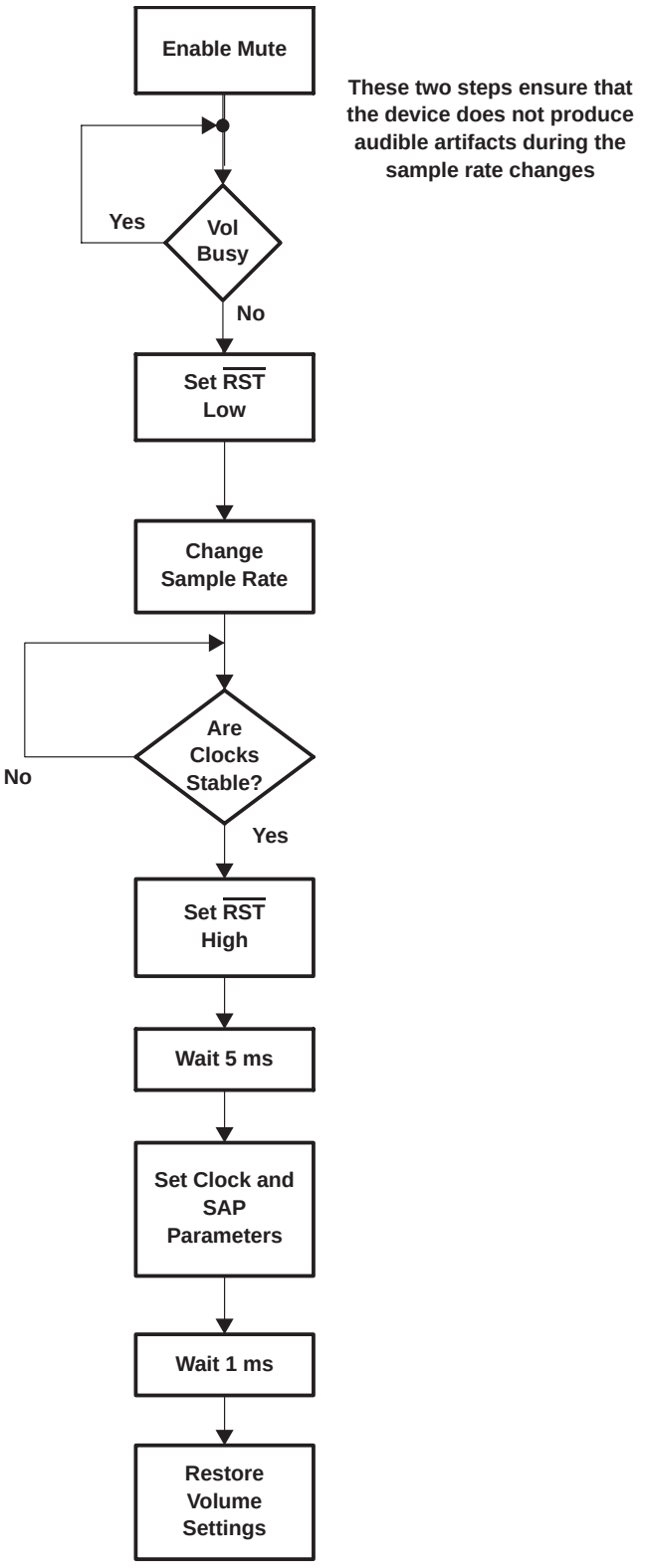


Figure 2−14. Load Coefficient and Restore Volume Settings

2.2.2 The Microprocessor Clock and I²C

The selected microprocessor clock is also used to drive the clocks used by the I²C control block. Two parameters, N and M, define the clocks used by the I²C control block. The I²C control block sampling frequency is set by $1/2^N$, where N can range in value from 0 to 7. A 1/(1 + M) divisor followed by a 1/10 divisor generates the data bit clock (SCL). This derived SCL clock is only used when the I²C control block is set to master mode (input pin $\overline{\text{I2CM\_S}}$ = 1). The default value for the I²C parameter N depends on whether the I²C controller is in slave mode ($\overline{\text{I2CM\_S}}$ = 0) or master mode ($\overline{\text{I2CM\_S}}$ = 1). In the I²C master mode, N = 2 (2^N = 4), which ensures that a 100-kHz I²C data clock (SCL) can be generated when the digital audio processor (DAP) is running at its maximum frequency of 135 MHz. In the I²C slave mode, N = 0 (2^N = 1), which ensures the I²C controller an adequate oversampling clock when the DAP is running at the minimum clock frequency required to process 8-kHz audio data (approximately 11.2 MHz). In I²C master mode, the values for M and N are fixed and cannot be changed.

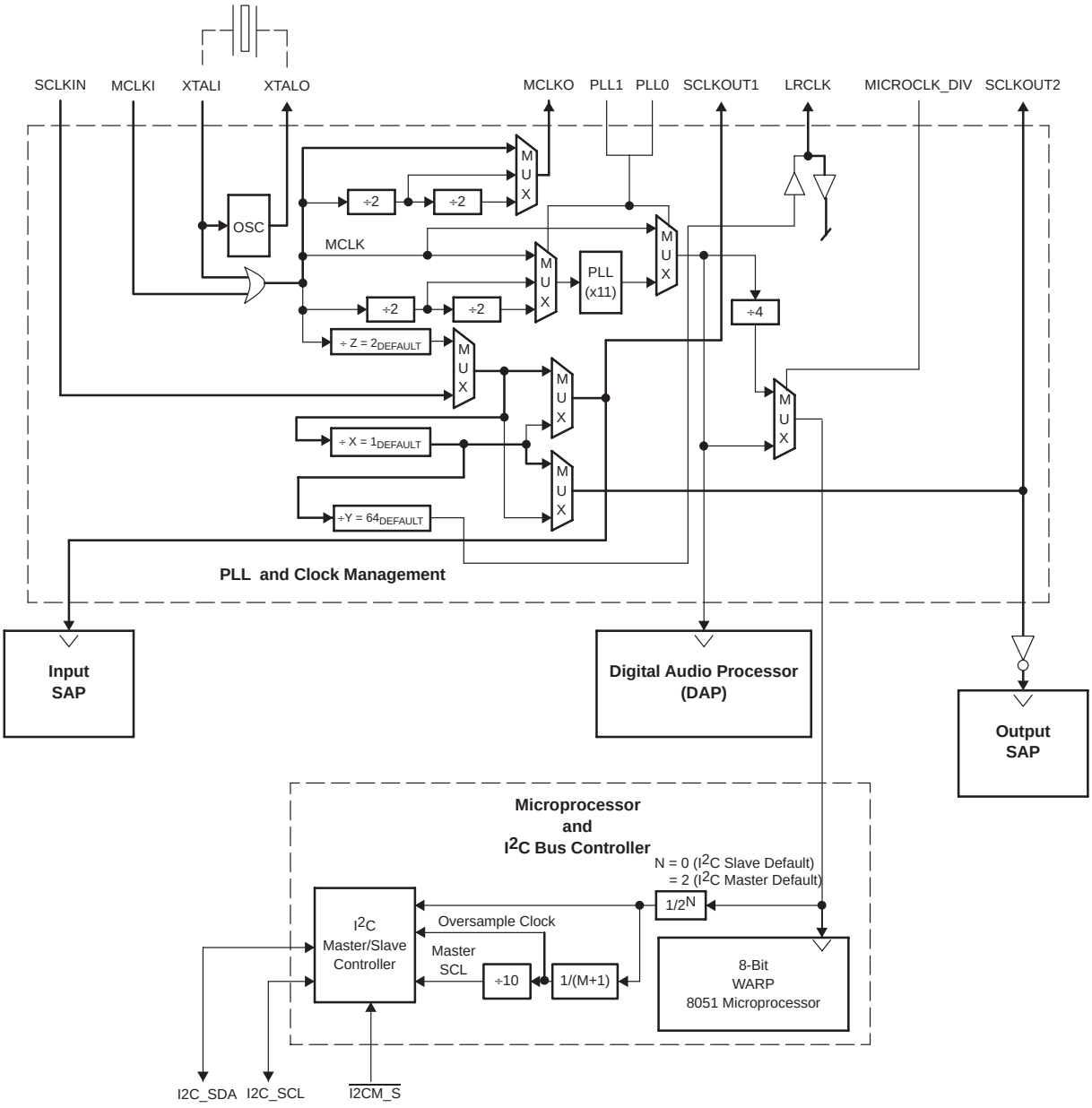


Figure 2−15. DPLL and Clock Management Block Diagram

When the SAP is in the master mode, it uses the MCLKI/XTALI master clock to drive the serial port clocks SCLKOUT1, SLCKOUT2, and LRCLK. When the SAP is in the slave mode, LRCLK is an input and SCLKOUT2 and SCLKOUT1 are derived from SCLKIN. As shown in Figure 2−15, SCLKOUT1 clocks data into the input SAP and SCLKOUT2 clocks data from the output SAP. Two distinct clocks are required to support TDM-to-discrete and discrete-to-TDM data format conversions. Such format conversions also require that SCLKIN be the higher valued bit clock frequency. For TDM-in/discrete-out format conversions, SCLKIN must be equal to the input bit clock. For discrete-in/TDM-out format conversions, SCLKIN must be equal to the output bit clock. The frequency settings for SCLKOUT1, SCLKOUT2, and LRCLK in the SAP master mode, as well as the SAP master/slave mode selection, are all controlled by I^2C commands.

Table 2−3 lists the default settings at power turnon or after a received reset.

Table 2−3. TAS3103A Clock Default Settings

CLOCK	DEFAULT SETTING
SCLKOUT1	SCLKIN
SCLKOUT2	SCLKIN
LRCLK	Input
MCLKO	MCLKI or XTALI
DAP processing clock	Set by pins PLL0 and PLL1
Microprocessor clock	Set by pin MICROCLK_DIV
I^2C sampling clock	**I^2C master mode** Microprocessor clock/4 **I^2C slave mode** Microprocessor clock
I^2C master SCL	I^2C sampling clock/90

The selections provided by the dedicated TAS3103A input pins and the programmable settings provided by I^2C subaddress commands give the TAS3103A a wealth of clocking options. Table 2−1, in the section describing the SAP, lists typical clocking selections for different audio sampling rates. However, the following clocking restrictions must be adhered to:

- MCLKI or XTALI ≥ 128 Fs (NOTE: For some TDM modes, MCLKI or XTALI must be ≥ 256 Fs)
- DAP clock ≥ 1400 × Fs
- DAP clock < 136 MHz
- Microprocessor clock/20 ≥ I^2C SCL clock
- Microprocessor clock ≤ 35 MHz
- XTALI ≤ 20 MHz
- MCLKI ≤ 25 MHz, unless PLL is bypassed

As long as these restrictions are met, all other clocking options are allowed.

2.3 Controller

The controller serves as the interface between the DAP, the asynchronous I^2C bus interface, and the four general-purpose I/O (GPIO) pins. Included in the controller block is an industry-standard 8051 microprocessor and an I^2C master/slave bus controller.

2.3.1 8051 Microprocessor

The 8051 microprocessor receives and distributes I^2C write data, retrieves and outputs to the I^2C bus controller the required I^2C read data, and participates in most processing tasks requiring multiframe processing cycles. The microprocessor also controls the flow of data into and out of the GPIO pins, which includes volume control when in the I^2C master mode. The microprocessor has its own data RAM for storing intermediate values and queuing I^2C commands, and a fixed program ROM. The microprocessor program cannot be altered.

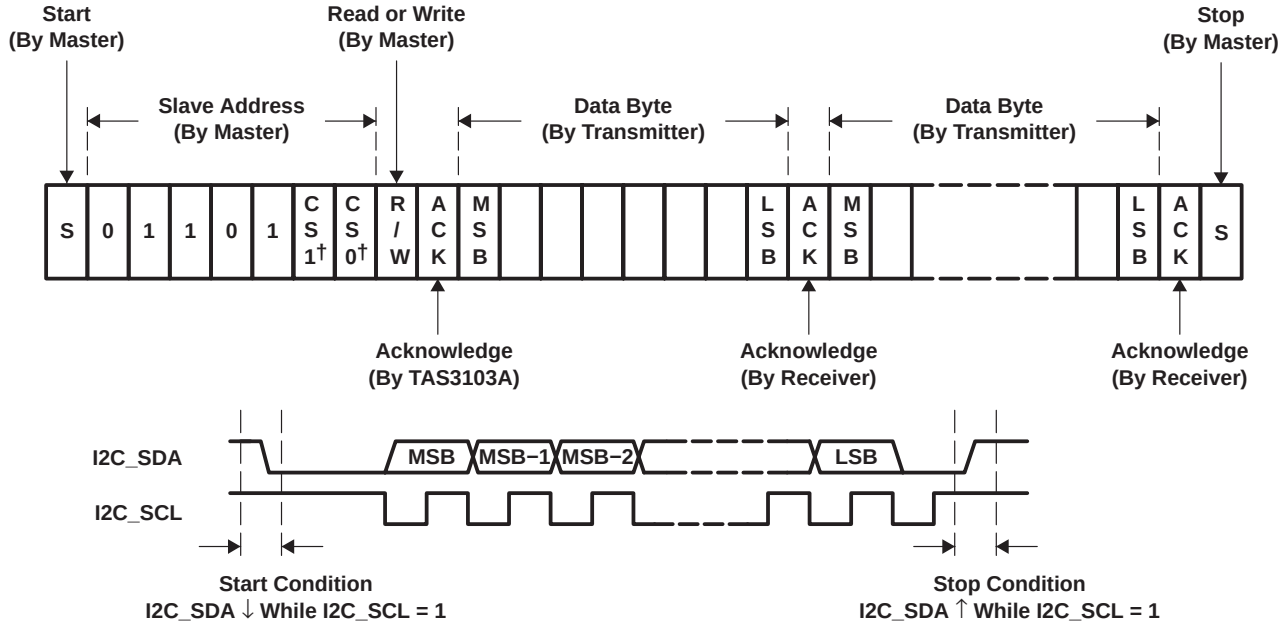
2.3.2 I²C Bus Controller

The TAS3103A has a bidirectional, two-wire, I²C-compatible interface. Both 100-kbps and 400-kbps data transfer rates are supported, and the TAS3103A controller can serve as either a master I²C device or a slave I²C device. Master/slave operation is defined by the logic level input into pin $\overline{\text{I2CM\_S}}$ (logic 1 = master mode, logic 0 = slave mode).

If the voltage input level to $\overline{\text{I2CM\_S}}$ is changed, the TAS3103A must be reset.

In the I²C master mode, data rate transfer is fixed at 100 kHz, assuming MCLKI or XTALI = 12.288 MHz, PLL0 = PLL1 = 0, and MICROCLK_DIV = 0. In the I²C slave mode, data rate transfer is determined by the master device. However, the setting of I²C parameter N at subaddress 0xFB (see *DPLL and Clock Management*, Section 2.2) does play a role in setting the data transfer rate. In the I²C slave mode, bit rates other than (and including) the I²C-specific 100-kbps and 400-kbps bit rates can be obtained, but N must always be set so that the oversample clock into the I²C master/slave controller is at least a factor of 20 higher in frequency than SCL.

The I²C communication protocol for the I²C slave mode is shown in Figure 2−16.



† Bits CS1 and CS0 in the TAS3103A slave address are compared to the logic levels on pins CS0 and CS1 for address verification. This provides the ability to address up to four TAS3103A chips on the same I²C bus.

Figure 2−16. I²C Slave-Mode Communication Protocol

In the slave mode, the I²C bus is used to:

- Update coefficient values and output data to those GPIO ports configured as output.
- Read status flags, input data from those GPIO ports configured as inputs and retrieve spectrum analyzer/VU meter data.

In the master mode, the I²C bus is used to download a user-specific configuration from an I²C compatible EEPROM.

In the slave mode only, specific registers and memory locations in the TAS3103A are accessible with the use of I²C subaddresses. There are 256 such I²C subaddresses. The protocol required to access a specific subaddress is presented in Figure 2−17.

As shown in Figure 2−17, a read transaction requires that the master device first issue a write transaction to give the TAS3103A the subaddress to be used in the read transaction that follows. This subaddress assignment write transaction is then followed by the read transaction. For write transactions, the subaddress is supplied in the first byte of data written, and this byte is followed by the data to be written. For write transactions, the subaddress must always

be included in the data written. There cannot be a separate write transaction to supply the subaddress, as was required for read transactions. If a subaddress assignment only write transaction is followed by a second write transaction supplying the data, erroneous behavior results. The first byte in the second write transaction is interpreted by the TAS3103A as another subaddress replacing the one previously written.

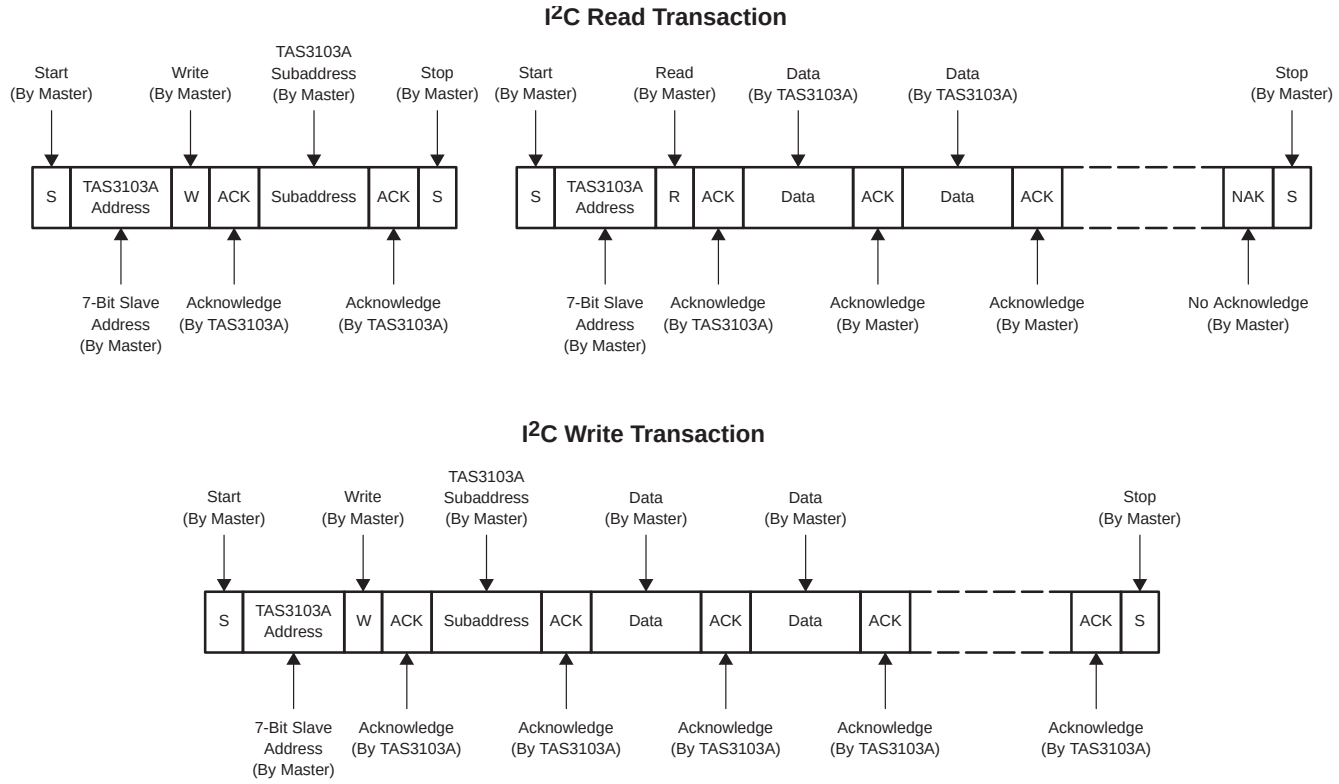


Figure 2−17. I2C Subaddress Access Protocol

2.3.2.1 I2C Master Mode Operation

The TAS3103A uses the master mode to download an operational configuration. The I2C in master mode is only used to download initialization parameters from EEPROM following reset or power down. The configuration downloaded must contain data for all 256 subaddresses, with spacer data supplied for those subaddresses that are GPIO subaddresses, read-only subaddresses, factory-test subaddresses, or unused (reserved) subaddresses. The spacer data must always be assigned the value zero. Table 2−4 organizes the 256 subaddresses (and their corresponding EEPROM addresses) into sequential blocks, with each block containing either valid data or spacer data.

Table 2−4 also illustrates that the subaddresses and their corresponding EEPROM memory addresses do not directly correlate. This is because many subaddresses are assigned more than one 32-bit word. For example, there is a unique subaddress for each biquad filter in the TAS3103A, but each subaddress is assigned five 32-bit coefficients—resulting in 20 bytes of memory being assigned to each biquad subaddress.

The TAS3103A, in the I2C master mode, can execute a complete download without requiring any wait states. After the TAS3103A has downloaded all 2367 bytes of coefficient and spacer data, the I2C bus is disabled and cannot be used to update coefficient values or retrieve status or spectrum/VU meter data. Volume control is available in the master mode via the four GPIO pins.

In I2C master mode, the watchdog timer must not be enabled.

When programming the EEPROM, make sure that the starting I2C check word (subaddress 0x00) and ending I2C check word (subaddress 0xFC) are identical.

Table 2−4. I²C EEPROM Data (1)(2)

DATA TYPE		EEPROM BYTE ADDRESSES	SUBADDRESS(ES)
Valid data	Starting I²C check word—must match ending check word	0x000−0x003	0x00
	Input mixers—set 1	0x004−0x0CF	0x01−0x33
	Effects block biquads	0x0D0−0x2AF	0x34−0x4B
	Reverberation (reverb) block mixers	0x2B0−0x2C7	0x4C−0x4E
	CH1 biquads	0x2C8−0x3B7	0x4F−0x5A
	CH2 biquads	0x3B8−0x4A7	0x5B−0x66
	CH3 biquads	0x4A8−0x597	0x67−0x72
	Bass and treble inline/bypass mixers	0x590−0x5AF	0x73−0x75
	DRC mixers	0x5B0−0x5DF	0x76−0x7E
	Dither input mixers	0x5E0−0x5EB	0x7F−0x81
	CH3 (subwoofer) to CH1/CH2 (L/R) mixers	0x5EC−0x5F3	0x82−0x83
	Spectrum analyzer/VU meter mixers	0x5F4−0x60B	0x84−0x89
	Output mixers	0x60C−0x66B	0x8A−0xA1
	CH1 loudness parameters	0x66C−0x697	0xA2−0xA6
	CH2 loudness parameters	0x698−0x6C3	0xA7−0xAB
	CH3 loudness parameters	0x6C4−0x6EF	0xAC−0xB0
	CH1/CH2 DRC parameters	0x6F0−0x733	0xB1−0xB5
	CH3 DRC parameters	0x734−0x777	0xB6−0xBA
	Spectrum analyzer parameters	0x778−0x847	0xBB−0xC5
	Dither output mixers	0x848−0x84F	0xC6
	Dither speed	0x850−0x853	0xC7
Factory Test Data (EEPROM Spacer Data) − Zeros		0x854−0x85F	0xC8−0xC9
Valid data	Input mixers—set 2	0x860−0x87F	0xCA−0xD1
Spacer Data		0x880−0x8E3	0xD2−0xEA
Valid data	Watchdog timer enable—must be disabled = 00 00 00 01	0x8E4−0x8E7	0xEB
Factory Test Data (EEPROM Spacer Data) − Zeros		0x8E8−0x8EF	0xEC−0xED (3)
Valid data	GPIO port parameters	0x8F0−0x8F7	0xEE−0xEF
	Volume parameters	0x8F8−0x90B	0xF0−0xF4
	Bass/treble filter selections	0x90C−0x91R	0xF5−0xF8
	I²S command word	0x91C−0x91F	0xF9
	Delay/reverb settings	0x920−0x92B	0xFA
	I²C M and N	0x92C−0x92F	0xFB
	Ending I²C check word—must match starting check word	0x930−0x933	0xFC
Read Only Data (EEPROM Spacer Data)		0x934−0x941	0xFD−0xFF

NOTES: 1. EEPROM organization must be big-endian—MS byte of data word allocated to the lowest address in memory.
2. EEPROM device ID = A0.
3. For subaddresses 0xEC and 0xED, the EEPROM byte space does not coincide.

The I²C master mode also uses the starting and ending I²C check words to verify a proper EEPROM download. The first 32-bit data word received from the EEPROM, the starting I²C check word at subaddress 0x00, is stored and compared against the 32-bit data word received for subaddress 0xFC, the ending I²C check word. These two data words must be equal as stored in the EEPROM. If the two words do not match when compared in the TAS3103A, the TAS3103A conducts another parameter download from the EEPROM. If the comparison check again fails, the TAS3103A discards all downloaded parameters and sets all parameters to the default values listed in the subaddress table presented in Appendix A. In the I²C slave mode, these default values are used to initialize the TAS3103A at power turnon or after a reset.

2.3.2.2 I²C Slave Mode Operation

The I²C slave mode permits configuration parameters (other than volume via the GPIO pins for the I²C master mode) to be changed. The TAS3103A can detect and reset the I²C interface when an invalid I²C command is received. This feature is enabled by setting the I²C configuration control value N to zero. This feature is also enabled by the default setting, 0x0000 0040, of the I²C configuration control register 0xFB. The I²C slave mode provides read access to the spectrum analyzer and VU meter outputs. Configuration downloads from a master device can be used to replace the I²C master mode EEPROM download.

For I²C read commands, the TAS3103A responds with data, a byte at a time, starting at the subaddress assigned, as long as the master device continues to respond with acknowledges. If a given subaddress does not use all 32 bits, the unused bits are read as logic 0. I²C write commands, however, are treated in accordance with the data assignment for that address space. If a write command is received for a biquad subaddress, the TAS3103A expects to see five 32-bit words. If fewer than five data words have been received when a stop command (or another start command) is received, the data received is discarded. If a write command is received for a mixer coefficient, the TAS3103A expects to see only one 32-bit word.

Supplying a subaddress for each subaddress transaction is referred to as random I²C addressing. The TAS3103A also supports sequential I²C addressing. For example, for write transactions, if a subaddress is issued followed by data for that subaddress and the fifteen subaddresses that follow, a sequential I²C write transaction has taken place, and the data for all 16 subaddresses is successfully received by the TAS3103A. For I²C sequential write transactions, the subaddress then serves as the start address, and the amount of data subsequently transmitted, before a stop or start is transmitted, determines how many subaddresses are written to. As was true for random addressing, sequential addressing requires that a block of data be transmitted. If only a partial set of data is written to the last subaddress, the data for the last subaddress is discarded. However, all other data written is accepted; just the incomplete data is discarded.

The GPIO subaddresses require the downloading of four bytes of zero-valued spacer data in order to proceed to the next subaddress. The TAS3103A can always receive sequential I²C addressing write data without issuing wait states. Do not write to reserved and factory-test subaddresses.

The TAS3103A also supports sequential read transactions. When an I²C subaddress assignment write transaction is followed by a read transaction, the TAS3103A outputs the data for that subaddress, and then continues to output data for the subaddresses that follow as long as the master continues to issue data received acknowledges. With only two exceptions, the TAS3103A outputs four bytes of zero-valued data for reserved and factory-test subaddresses. The subaddresses of the exceptions and the number of bytes supplied by the TAS3103A for each exception are given in Table 2−5. If a GPIO port is assigned as an output port, a logic 0 bit value is supplied by the TAS3103A for this GPIO port in response to a read transaction at subaddress 0xEE.

CAUTION: Sequential write transactions must be in ascending subaddress order. The TAS3103A does not wrap around from subaddress 0xFF to 0x00.

Sequential read transactions wrap around from subaddress 0xFF to 0x00.

Table 2−5. Four-Byte Read Exceptions—Reserved and Factory-Test I²C Subaddresses

SUBADDRESS	NUMBER BYTES SUPPLIED BY TAS3103A
0xC9	8
0xED	8

NOTE: Table 2−5 does not include read-only subaddresses and thus does not include subaddresses 0xFD, 0xFE, and 0xFF. When read, these read-only subaddresses output 10, 2, and 1 byte, respectively.

Thus, for all reserved and factory-test subaddresses, except subaddresses 0xC9 and 0xED, the master device must issue four data-received acknowledges for the four bytes of zero-valued data. For subaddresses 0xC9 and 0xED, the master device must issue eight data-received acknowledges for the eight bytes of zero-valued data.

Sequential read transactions do not have restrictions on outputting only complete subaddress data sets. If the master does not issue enough data-received acknowledges to receive all the data for a given subaddress, the master device

simply does not receive all the data. If the master device issues more data-received acknowledges than required to receive the data for a given subaddress, the master device simply receives complete or partial sets of data, depending on how many data-received acknowledges are issued from the subaddress(es) that follow.

I^2C read transactions, both sequential and random, can impose wait states. For the standard I^2C mode (SCL = 100 kHz), typical wait state time for an 8-MHz microprocessor clock is on the order of 1 µs. For the fast I^2C mode (SCL = 400 kHz) and the same 8-MHz microprocessor clock, typical wait state times are extended up to 4 µs in duration. Increasing the microprocessor clock frequency lowers the wait state times and for the standard I^2C mode, a higher microprocessor clock can totally eliminate the presence of wait states. For example, increasing the microprocessor clock to 33 MHz results in no wait states for the standard (100-kHz) I^2C mode. For the fast I^2C mode, higher microprocessor clocks shorten the wait state times encountered, but do not totally eliminate their presence.

2.4 Digital Audio Processor (DAP) Arithmetic Unit

The DAP arithmetic unit is a fixed-point computational engine consisting of an arithmetic unit and data and coefficient memory blocks. Figure 2−18 is a block diagram of the arithmetic unit.

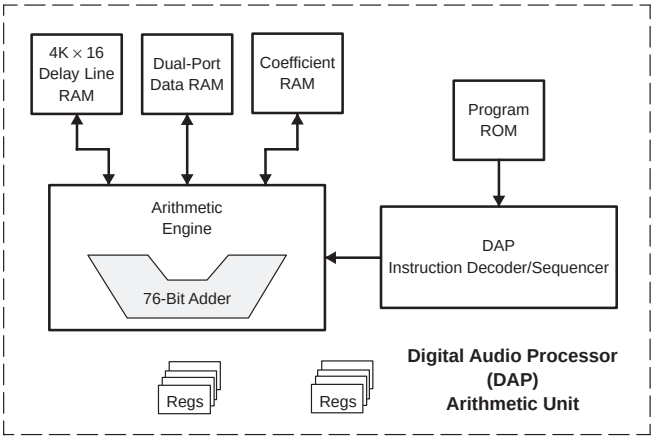


Figure 2−18. DAP Arithmetic Unit Block Diagram

The DAP arithmetic unit is used to implement all firmware functions—soft volume, loudness compensation, bass and treble processing, dynamic range control, channel filtering, 3D effects, input and output mixing, spectrum analyzer, VU meter, and dither.

Figure 2−19 shows the data word structure of the DAP arithmetic unit. Eight bits of overhead or guard bits are provided at the upper end of the 48-bit DAP word and 8 bits of computational precision or noise bits are provided at the lower end of the 48-bit word. The incoming digital audio words are all positioned with the most-significant bit abutting the 8-bit overhead/guard boundary. The sign bit in bit 39 indicates that all incoming audio samples are treated as signed data samples.

CAUTION: Audio data into the TAS3103A is always treated as sign-extended 2s-complement data.

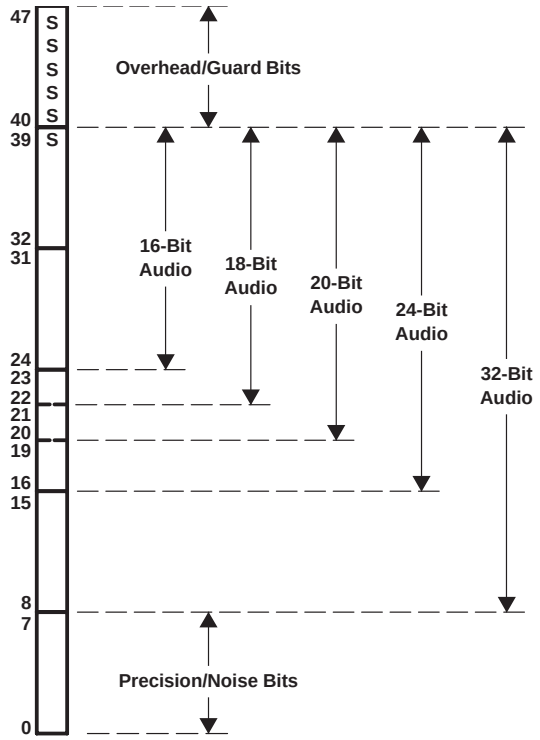


Figure 2−19. DAP Arithmetic Unit Data Word Structure

The arithmetic engine is a 48-bit (25.23-format) processor consisting of a general-purpose 76-bit arithmetic logic unit (ALU) and function-specific arithmetic blocks. Multiply operations (excluding the function-specific arithmetic blocks) always involve 48-bit DAP words and 28-bit coefficients (usually I^2C programmable coefficients). If a group of products is to be added together, the 76-bit product of each multiplication is applied to a 76-bit adder, where a DSP-like multiply-accumulate (MAC) operation takes place. Biquad filter computations use the MAC operation to maintain precision in the intermediate computational stages.

To maximize the linear range of the 76-bit ALU, saturation logic is not used. Intermediate overflows are then permitted in multiply-accumulate operations, but it is assumed that subsequent terms in the multiply-accumulate computation flow corrects the overflow condition. The biquad filter structure used in the TAS3103A is the direct form-I structure and has only one accumulation node. With this type of structure, intermediate overflow is allowed as long as the designer of the filters has ensured that the final output is bounded and does not overflow. Figure 2−20 shows a bounded computation that experiences an intermediate overflow condition. For ease of illustration, 8-bit arithmetic is used.

The DAP memory banks include a dual-port data RAM for storing intermediate results, a coefficient RAM, a 4K $\times$ 16 RAM for implementing the delay stages, and a fixed program ROM. Only the coefficient RAM, accessible via the I^2C bus, is available to the user.

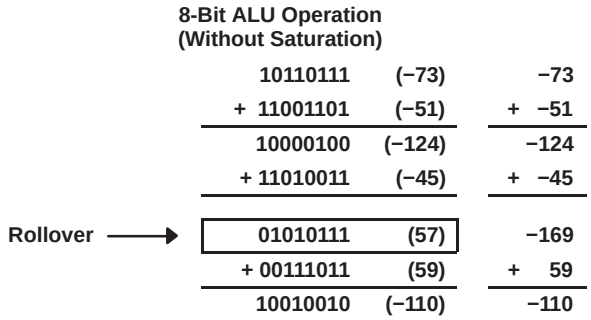


Figure 2−20. DAP ALU Operation With Intermediate Overflow

The DAP processing clock is set by pins PLL0 and PLL1, with the source clock XTALI or MCLKI. The DAP operates at speeds up to 136 MHz, which is sufficient to process 96-kHz audio.

2.5 Reset

The reset circuitry in the TAS3103A is shown in Figure 2−21. A reset is initiated by inputting logic 0 on the reset pin, $\overline{\text{RST}}$. A reset is also issued at power turnon by the internal 1.8-V regulator subsystem.

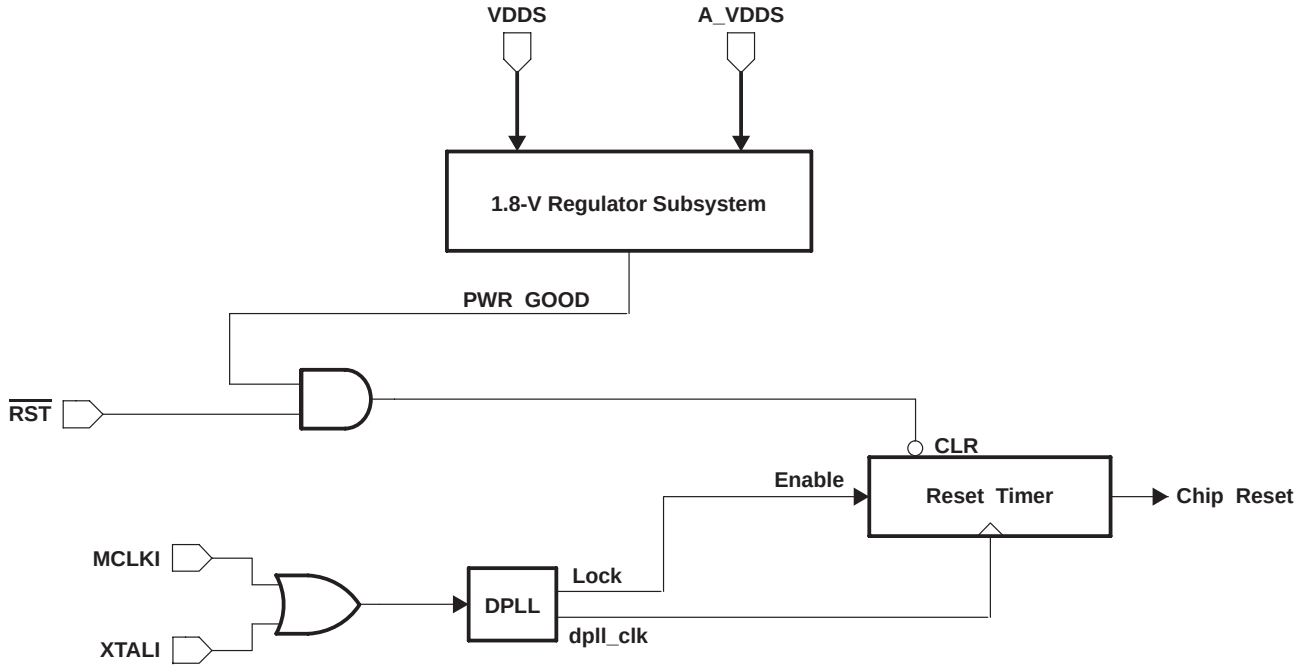


Figure 2−21. TAS3103A Reset Circuitry

When the VDDS and A_VDDS voltage rise times from 0.1 V to 3.0 V are 5 ms or less, the internal 1.8-V regulator subsystem holds the TAS3103A in reset at power on until regulation is reached. The duration of this signal permits the TAS3103A to complete all power-on initialization without external control or circuitry.

When the VDDS and A_VDDS voltage rise times from 0.1 V to 3.0 V are greater than 5 ms, the TAS3103A must be reset once VDDS and A_VDDS have reached a minimum of 3.0 V. This reset can be performed in one of two ways.

- A valid $\overline{\text{RST}}$ can be applied by an external controller once VDDS and A_VDDS are at 3.0 volts or higher.
- An external circuit can be employed to hold $\overline{\text{RST}}$ LOW until VDDS and A_VDDS are 3.0 volts or higher. An RC circuit implementation of this is shown in Figure 2−22. The values for R and C should be chosen to ensure that $\overline{\text{RST}}$ is held low until VDDS and A_VDDS have reached 3.0 volts or higher. The values of R and C in this figure provide a delay of approximately 20 ms.

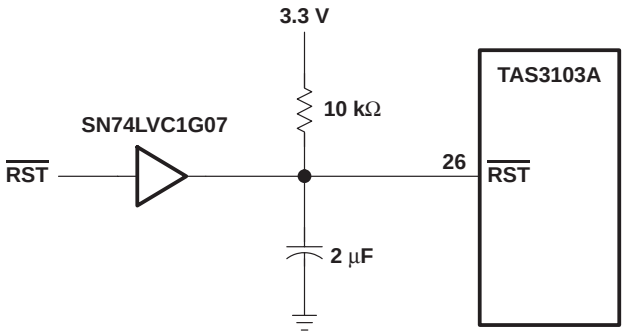


Figure 2−22. External Power-Good Reset-Control Circuit

Note that $\overline{\text{RST}}$ implements an asynchronous clear. This control can respond to narrow negative signal transitions. Some applications, therefore, might require a high-frequency capacitor on the $\overline{\text{RST}}$ pin to remove unwanted noise excursions.

2.6 Power Down

Power down requires stable and accurate clocks during the transition from the operational state to the power-down state, and during the transition for the power-down state to the operational state. If a clock error occurs during either of these transitions, then the operation of the TAS3103A can be compromised and a reset may be required to restore operation. Setting the PWRDN pin to logic 1 enables power down. Power down stops all clocks in the TAS3103A, but preserves the state of the TAS3103A. When PWRDN is deactivated (set to logic 0) after a period of activation, the TAS3103A resumes the processing of audio data on receiving the next LRCLK (indicating a new sample of audio data is available for processing). The configuration of the TAS3103A and all programmable parameters are retained during power down.

A time lag occurs between setting PWRDN to logic 1 and entering the power-down state. PWRDN is sampled every GPIOFSCOUNT LRCLK periods (see subaddress 0xEF in Appendix A; W*atchdog Timer*, Section 2.7; and *General-Purpose I/O (GPIO) Ports*; Section 2.8). This means that a time lag as great as GPIOFSCOUNT(1/LRCLK) could exist between the activation of PWRDN (setting to logic 1) and the time at which the microprocessor recognizes that the PWRDN pin has been activated. Normally, on recognizing that the PWRDN pin has been activated, the TAS3103A enters the power-down state approximately 80 microprocessor clock cycles later. However, if a soft volume update is in progress, the TAS3103A waits until the soft volume update is complete before entering the power-down state. For this case then, the worst-case time lag between recognizing the activation of pin PWRDN and entering the power-down state would be 4096 LRCLK periods, assuming a volume slew rate selection (bit VSC of I^2C subaddress 0xF1) of 4096 and the issuance of a volume update immediately preceding the reading of pin PWRDN. The worst-case time lag between setting PWRDN to logic 1 and entering the power-down state is then:

$$\text{power−down−time lag}_{\text{Worst}-\text{Case}} = \frac{4096 + \text{GPIOFSCOUNT}}{\text{LRCLK}} + \frac{80}{\text{Microprocessor} - \text{Clock}}$$

A time lag also between deactivating PWRDN (setting PWRDN to logic 0) and exiting the power-down state. This time lag is set by the time it takes the internal digital PLL to stabilize, and this time, in turn, is set by the master clock frequency (MCLKI or XTALI) and the PLL output clock frequency. For a 135-MHz PLL output clock and a 24.576 MCLKI, the time lag is approximately 25 μs. For an 11.264-MHz PLL output clock and a 1.024-MHz MCLKI, the time lag is approximately 360 μs.

Power consumption in the power-down state is approximately 12 mW.

2.7 Watchdog Timer

A watchdog timer in the TAS3103A monitors the microprocessor activity. If the microprocessor ever ceases to execute its stored program, the watchdog timer fires and resets the TAS3103A. This capability was included in the TAS3103A for factory test purposes and has little use in applications. The program structure used in the microprocessor ensures that the microprocessor always executes its stored program unless a hardware failure occurs.

The watchdog timer is governed by the parameter GPIOFSCOUNT in subaddress 0xEF and the LSB of the 32-bit word at subaddress 0xEB. The default value of the LSB of the 32-bit word at subaddress 0xEB is 1, and this value disables the watchdog timer. The GPIOFSCOUNT is also used in other functions and balancing the needs of these other functions regarding GPIOFSCOUNT with the requirements of the watchdog timer is an involved process. For this reason, it is strongly recommended that the LSB of the 32-bit word at subaddress 0xEB remain a 1. If an application does require use of the watchdog timer, it is requested that the user contact an application engineer in the Digital Audio Department of Texas Instruments for details in properly using this feature.

2.8 General-Purpose I/O (GPIO) Ports

The TAS3103A has four general-purpose I/O (GPIO) ports. Figure 2−23 is a block diagram of the GPIO circuitry in the TAS3103A.

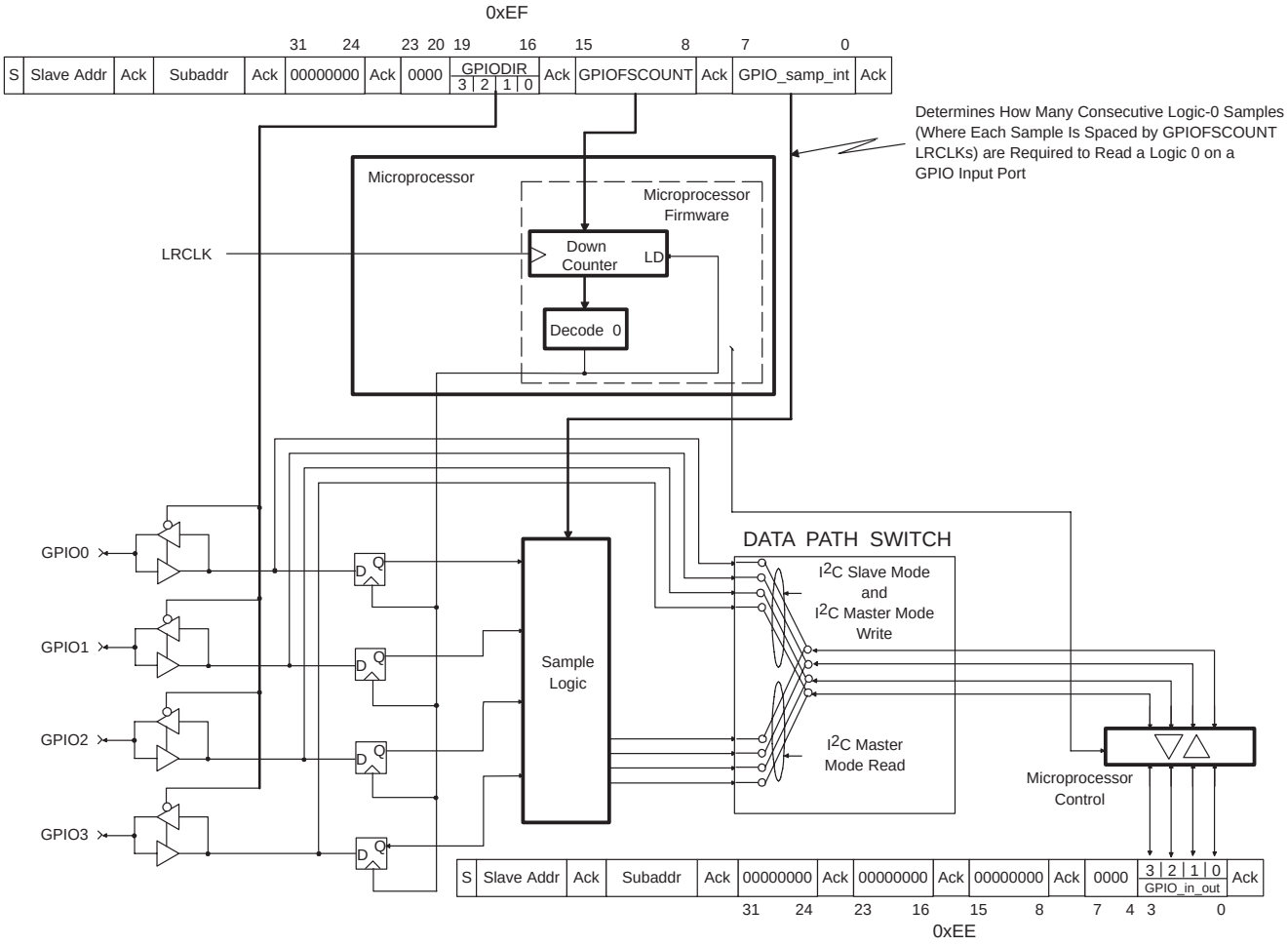


Figure 2−23. GPIO Port Circuitry

2.8.1 GPIO Functionality—I²C Master Mode

In the I²C master mode, the GPIO ports are strictly input ports and are used to control volume. Table 2−6 lists the functionality of each GPIO port in the I²C master mode. Bit field GPIOFSCOUNT (15:8) of I²C subaddress 0xEF governs the rate at which the GPIO pins are sampled for a volume update. The sample rate is:

$$f_{GPIO\_Port} = \frac{LRCLK}{GPIOFSCOUNT}$$

Table 2−6. GPIO Port Functionality—I2C Master Mode

GPIO PORT	FUNCTION
GPIO0 (pin 18)	Volume up—CH1 and CH2
GPIO1 (pin 19)	Volume down—CH1 and CH2
GPIO2 (pin 20)	Volume up—CH3
GPIO3 (pin 21)	Volume down—CH3

GPIOFSCOUNT also governs the rate at which the power-down pin PWRDN is sampled and the rate at which the watchdog counter is reset. GPIOFSCOUNT then cannot be independently used to tune the volume adjustment. For this reason, bit field GPIO_samp_int of the same I2C subaddress (0xEF) is included to provide the ability to adjust the responsiveness (or sluggishness) of the volume switches.

Each GPIO port has a weak pullup to VDDS. A volume control switch then typically switches the signal line to the GPIO port between ground and an open circuit. The parameter GPIO_samp_int sets how many consecutive GPIO port samples must be logic 0 before a logic 0 is read. A read logic 0 on a given GPIO port is interpreted as a command to increase or decrease volume. If a logic 0 is read, and the signal level into the GPIO port remains at logic 0 for another GPIO_samp_int consecutive samples, a second logic 0 value is read.

For each logic 0 read, the volume is increased or decreased 0.5 dB. After two consecutive logic 0 readings, each logic 0 reading that follows results in the volume level increasing or decreasing 5 dB instead of 0.5 dB. Figure 2−24 shows an example of activating a volume switch. For the example in Figure 2−24, GPIOFSCOUNT is set to 3 and GPIO_samp_int is set to 2. It is also noted in Figure 2−24 that the parameter GPIO_samp_int only comes into play on logic 0 valued samples. As soon as the GPIO sample goes to logic 1, the audio updating ceases.

2.8.2 GPIO Functionality—I2C Slave Mode

In the I2C slave mode, the GPIO ports can be used as true general-purpose ports. Each port can be individually programmed, via the I2C bus, to be either an input or an output port. The default assignment for all GPIO ports, in the I2C slave mode, is an input port.

When a given GPIO port is programmed as an output port, by setting the appropriate bit in the bit field GPIODIR (19:16) of subaddress 0xEF to logic 1, the logic level output is set by the logic level programmed into the appropriate bit in bit field GPIO_in_out (3:0) of subaddress 0xEE. The I2C bus then controls the logic output level for those GPIO ports assigned as output ports.

When a given GPIO port is programmed as an input port by setting the appropriate bit in bit field GPIODIR (19:16) of subaddress 0xEF to logic 0, the logic input level into the GPIO port is written to the appropriate bit in bit field GPIO_in_out (3:0) of subaddress 0xEE. The I2C bus can then be used to read bit field GPIO_in_out to determine the logic levels at the input GPIO ports. Whether a given bit in the bit field GPIO_in_out is a bit to be read via the I2C bus or a bit to be written to via the I2C bus is strictly determined by the corresponding bit setting in bit field GPIODIR.

In the I2C slave mode, the GPIO input ports are read every GPIOFSCOUNT LRCLKs, as was the case in the I2C master mode. However, parameter GPIO_samp_int does not have a role in the I2C slave mode. If a GPIO port is assigned as an output port, a logic 0 bit value is supplied by the TAS3103A for this GPIO port in response to a read transaction at subaddress 0xEE.

If the GPIO ports are left in their power turnon default state, they are input ports with a weak pullup on the input to VDSS.

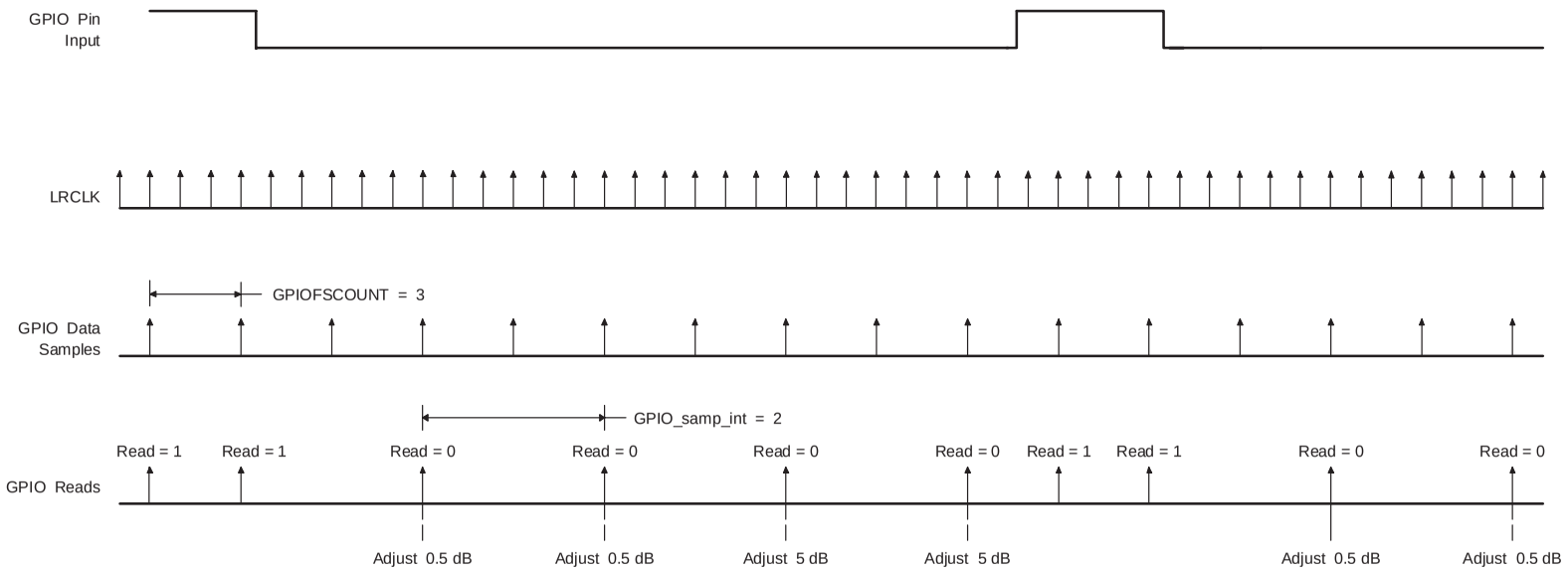


Figure 2−24. Volume Adjustment Timing—Master I^{2}C Mode

3 Firmware Architecture

3.1 I²C Coefficient Number Formats

The firmware for the TAS3103A is housed in ROM resources within the TAS3103A and cannot be altered. However, mixer gain, level offset, and filter tap coefficients, which can be entered via the I²C bus interface, provide a user with the flexibility to set the TAS3103A to a configuration that achieves the system-level goals.

The firmware is executed in a 48-bit signed fixed-point arithmetic machine. The most-significant bit of the 48-bit data path is a sign bit, and the 47 lower bits are data bits. Mixer gain operations are implemented by multiplying a 48-bit signed data value by a 28-bit signed gain coefficient. The 76-bit signed output product is then truncated to a signed 48-bit number. Level offset operations are implemented by adding a 48-bit signed offset coefficient to a 48-bit signed data value. In most cases, if the addition results in overflowing the 48-bit signed number format, saturation logic is used. This means that if the summation results in a positive number that is greater than 0x7FFF FFFF FFFF (the spaces are used to ease the reading of the hexadecimal number), the number is set to 0x7FFF FFFF FFFF. If the summation results in a negative number that is less than 0x8000 0000 0000, the number is set to 0x8000 0000 0000. There are exceptions to the use of saturation logic for summations that overflow—see *Digital Audio Processor (DAP) Arithmetic Unit*, Section 2.4.

3.1.1 28-Bit 5.23 Number Format

All mixer gain coefficients are 28-bit coefficients using a 5.23 number format. Numbers formatted as 5.23 numbers means that there are 5 bits to the left of the decimal point and 23 bits to the right of the decimal point. This is shown in the Figure 3−1.

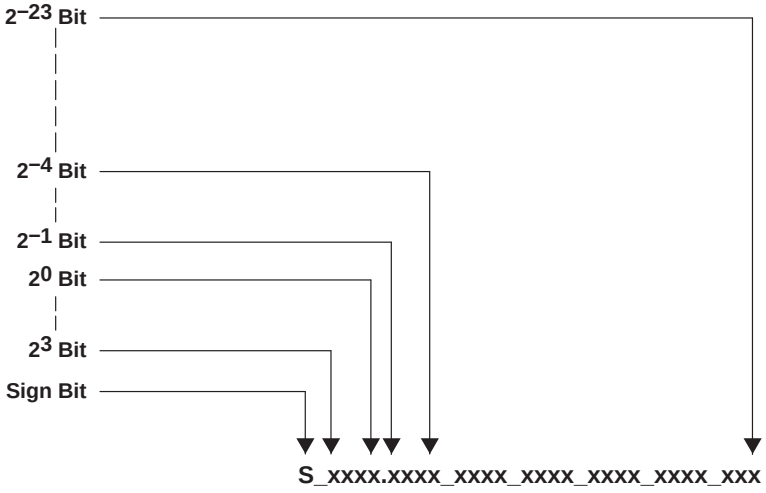


Figure 3−1. 5.23 Format

The decimal value of a 5.23-format number can be found by following the weighting shown in Figure 3−2. If the most-significant bit is logic 0, the number is a positive number, and the weighting shown yields the correct number. If the most-significant bit is a logic 1, then the number is a negative number. In this case every bit must be inverted, a 1 added to the result, and then the weighting shown in Figure 3−2 applied to obtain the magnitude of the negative number.

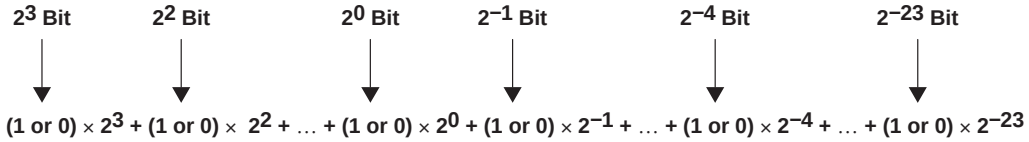


Figure 3−2. Conversion Weighting Factors—5.23 Format to Floating Point

Gain coefficients, entered via the I^{2}C bus, must be entered as 32-bit binary numbers. The format of the 32-bit number (4-byte or 8-digit hexadecimal number) is shown in Figure 3−3.

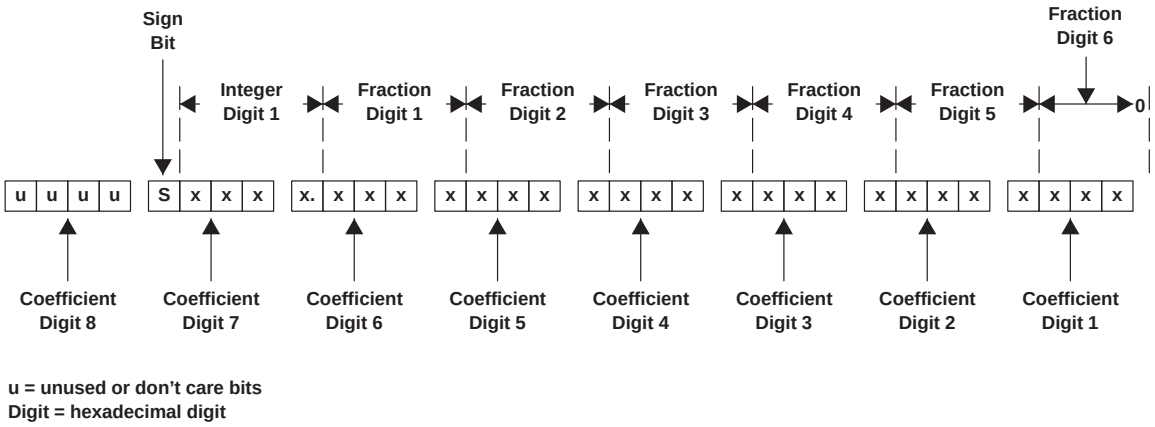


Figure 3−3. Alignment of 5.23 Coefficient in 32-Bit I^{2}C Word

As Figure 3−3 shows, the hexadecimal value of the integer part of the gain coefficient cannot be concatenated with the hexadecimal value of the fractional part of the gain coefficient to form the 32-bit I^{2}C coefficient. The reason is that the 28-bit coefficient contains 5 bits of integer, and thus the integer part of the coefficient occupies all of one hexadecimal digit and the most-significant bit of the second hexadecimal digit. In the same way, the fractional part occupies the lower 3 bits of the second hexadecimal digit, and then occupies the other five hexadecimal digits (with the eighth digit being the zero-valued most-significant hexadecimal digit).

3.1.2 48-Bit 25.23 Number Format

All level adjustment and threshold coefficients are 48-bit coefficients using a 25.23 number format. Numbers formatted as 25.23 numbers means that there are 25 bits to the left of the decimal point and 23 bits to the right of the decimal point. This is shown in Figure 3−4.

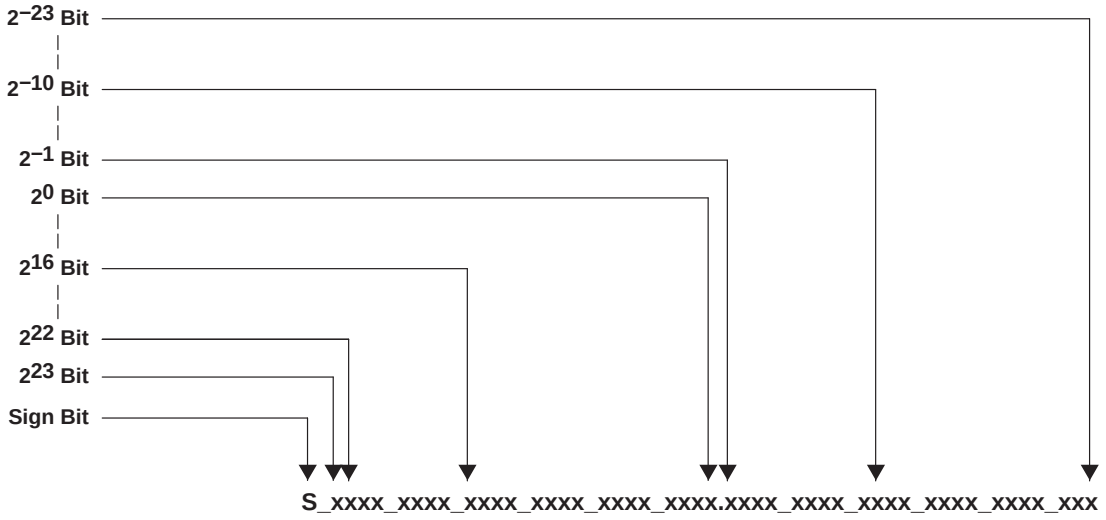


Figure 3−4. 25.23 Format

Figure 3−5 shows the derivation of the decimal value of a 48-bit 25.23-format number.

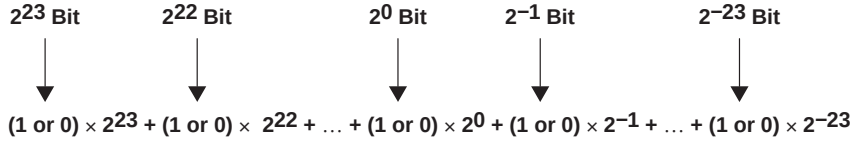
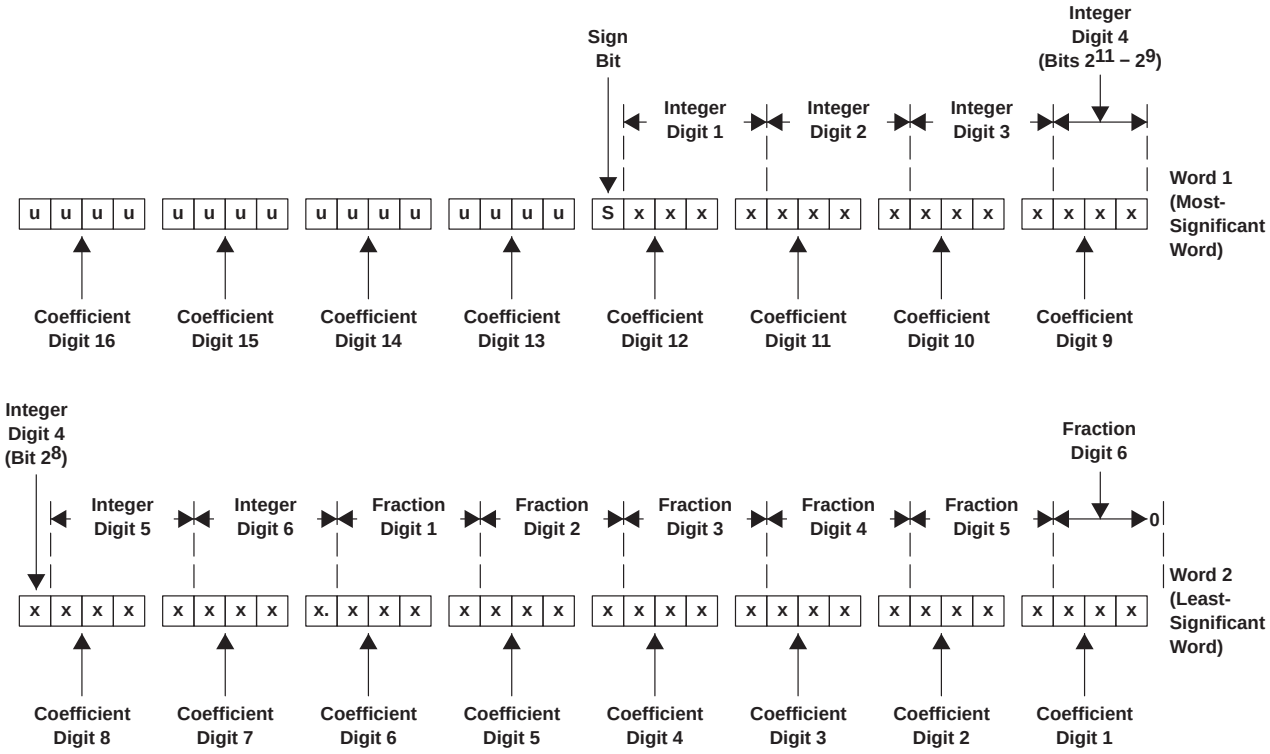


Figure 3−5. Alignment of 25.23 Coefficient in 32-Bit I²C Word

Two 32-bit words must be sent over the I²C bus to download a level or threshold coefficient into the TAS3103A. The alignment of the 48-bit, 25.23-formatted coefficient in the 8-byte (two 32-bit words) I²C word is shown in Figure 3−6.



u = unused or don't care bits
Digit = hexadecimal digit

Figure 3−6. Alignment of 25.23 Coefficient in Two 32-Bit I²C Words

3.2 Input Crossbar Mixers

The TAS3103A has four serial input ports—SDIN1, SDIN2, SDIN3, and SDIN4. SDIN1, SDIN2, and SDIN3 provide the input resources to process 5.1-channel audio in two TAS3103A chips. SDIN4 provides the capability to multiplex between a full 5.1-channel system and a stereo source or an *information/warning* audio message as might be found in an automotive application.

Each serial input port is assigned two internal processing nodes. The mixers following these internal processing nodes serve to distribute the input audio data to various processing nodes within the TAS3103A. Figure 3−7 shows the assignment of the internal processing nodes to the serial input ports. Two cases are shown in Figure 3−7—discrete mode and TDM mode.

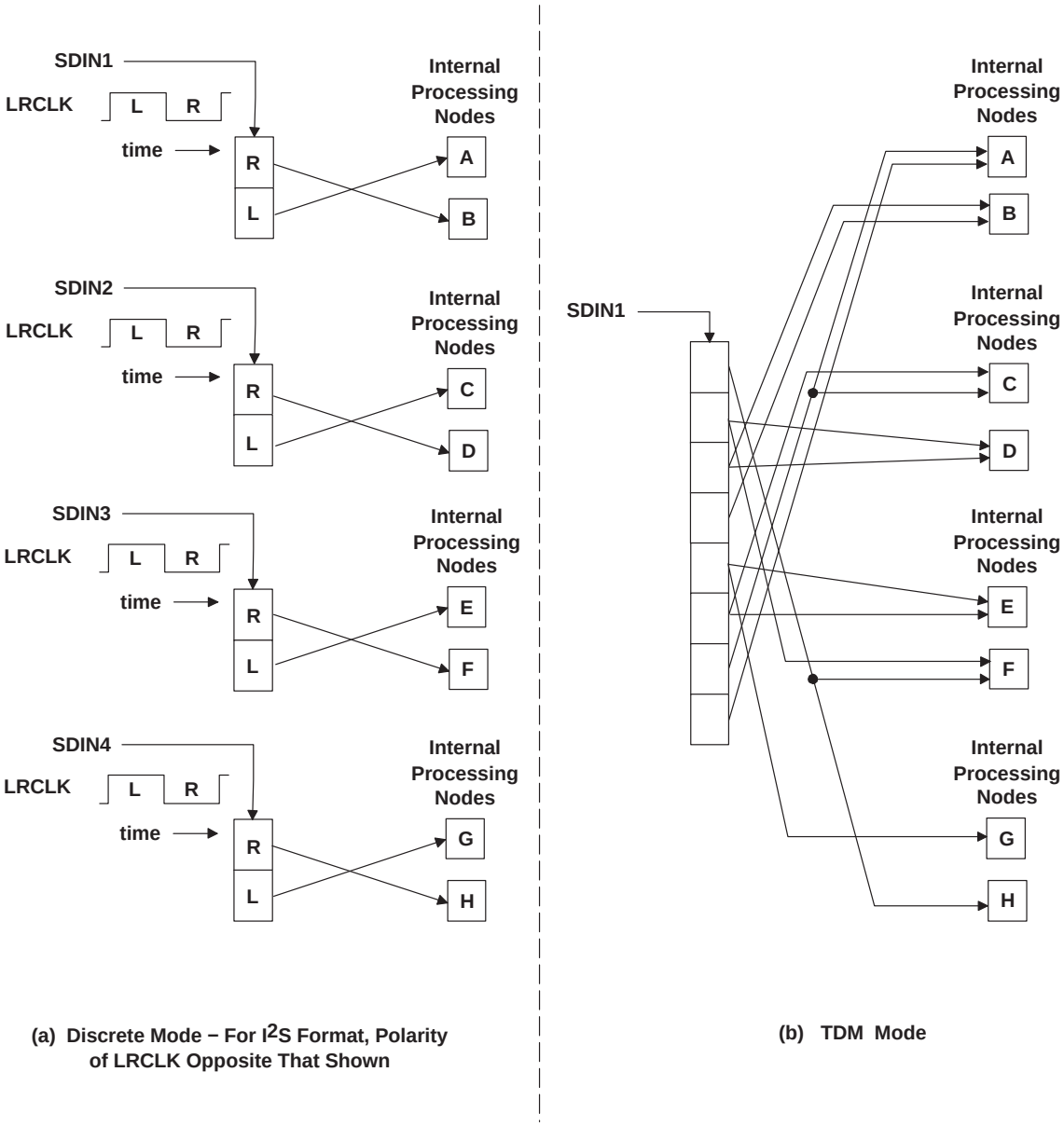
The input crossbar mixer topology for internal processing nodes A, B, C, D, E, and F is shown in Figure 3−8. Each of the six nodes is assigned six mixers. These six mixers provide the ability to route the incoming serial port data on SDIN1, SDIN2, and SDIN3 to:

- Processing node d—bypassing effects block and directly feeding monaural CH1
- Processing node e—bypassing effects block and directly feeding monaural CH2
- Processing node f—directly feeding the section of the effects block assigned to monaural CH3
- Processing nodes a and b—directly feeding paths that contain the reverberation (reverb) delay elements assigned to CH1 and CH2
- Processing node c—directly feeding an effects block path assigned to CH1 and CH2 that bypasses all reverb delay elements.

The ability to route all input nodes to the same set of processing nodes fully decouples the input order (what audio components are wired to which serial input ports) from the processing flow. As is seen in the discussion of the output crossbar mixers, the output serial ports are fully decoupled from the three monaural channels (any monaural channel output can be routed to either the left or the right side of any output port). The TAS3103A thus provides full flexibility in the routing of audio data into and out of the chip.

The mixer topology for internal processing nodes g and h is shown in Figure 3−9. Nodes g and h are each assigned three mixers. The mixers provide the ability to route the incoming data on serial port SDIN4 to:

- Output processing nodes to facilitate input-to-output pass-through
- CH1/CH2 effects block input nodes that bypass reverb delay
- CH3 effects block input node



(a) Discrete Mode − For I^2S Format, Polarity of LRCLK Opposite That Shown

(b) TDM Mode

Figure 3−7. Serial Input Port to Processing Node Topology

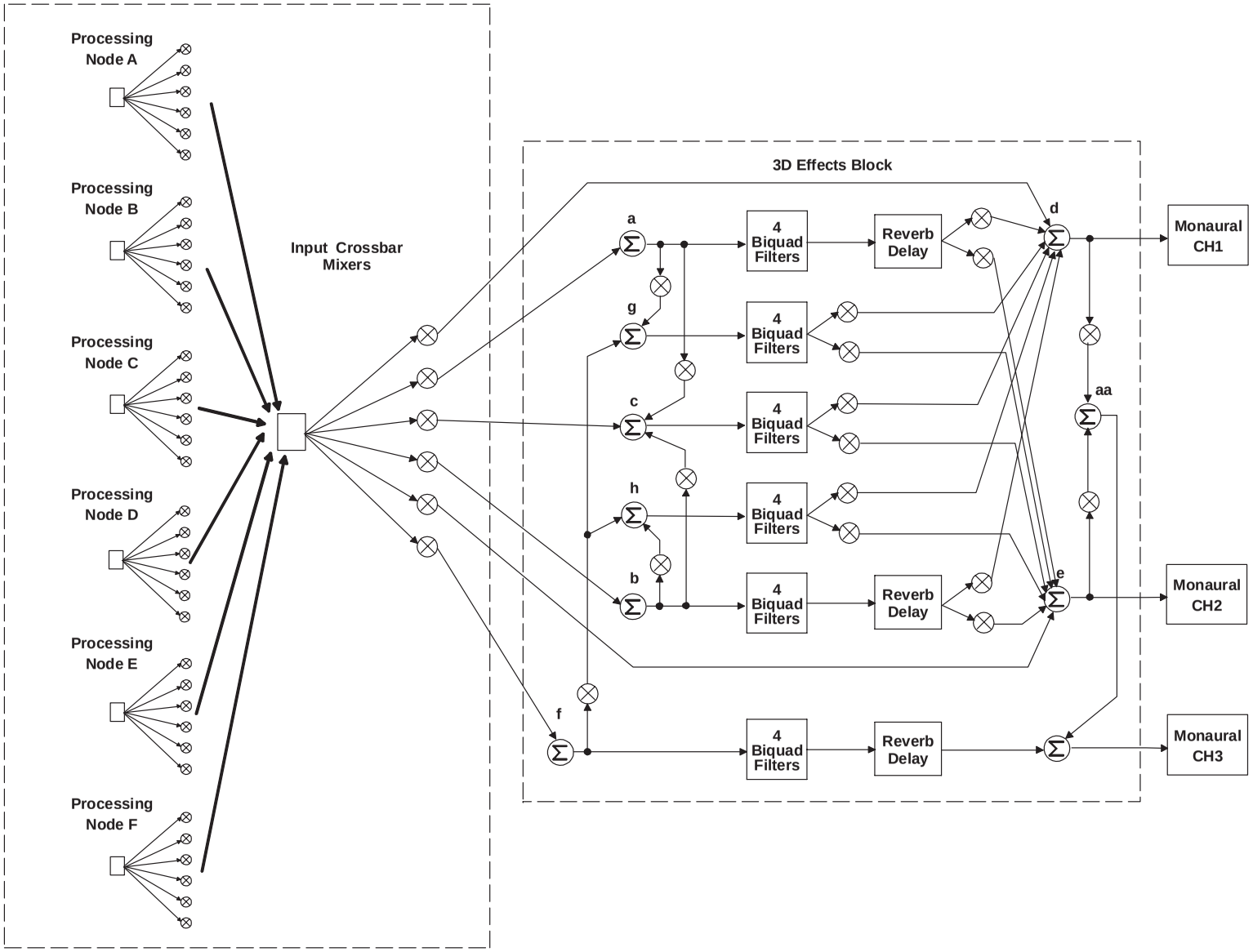
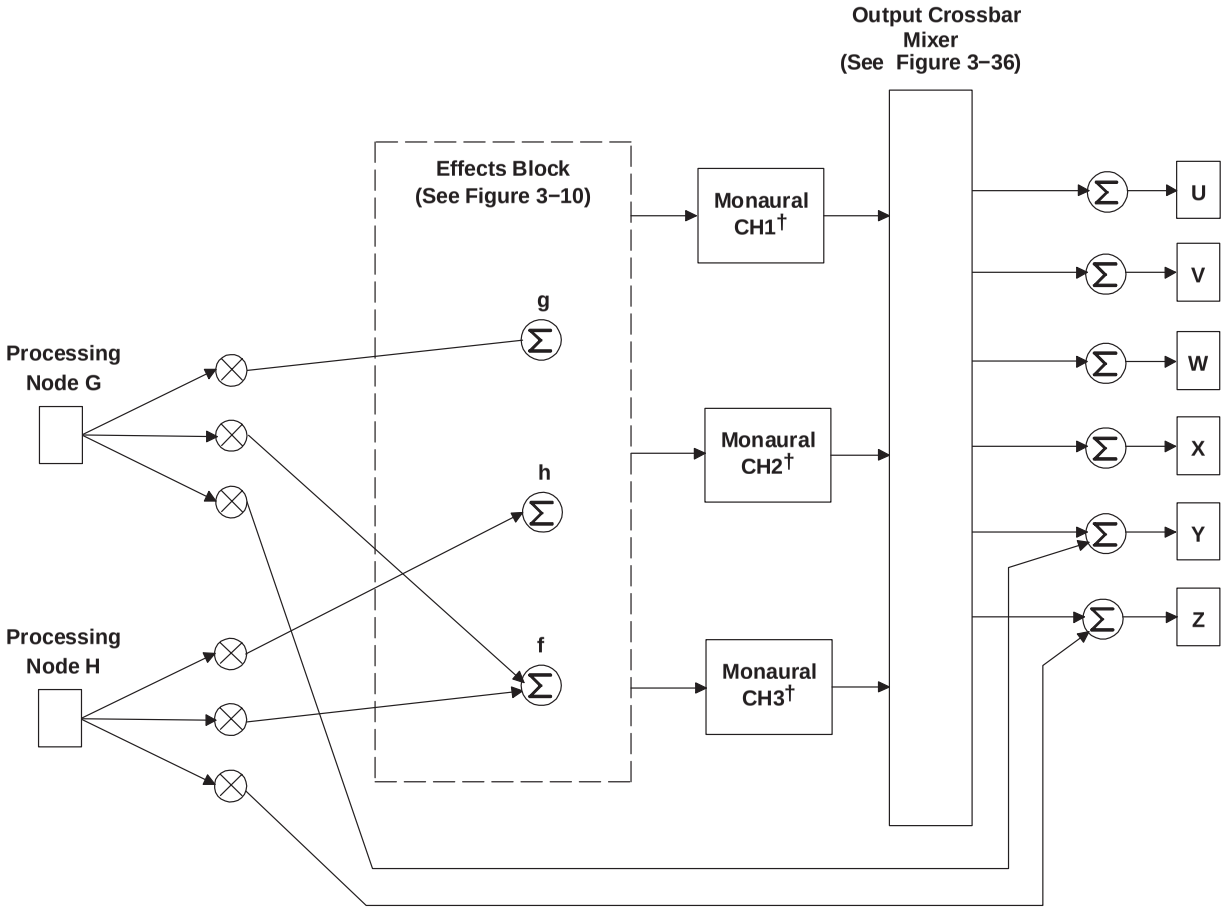


Figure 3–8. Input Mixer and Effects Block Topology—Internal Processing Nodes A, B, C, D, E, and F



† Monaural channels consist of 12 biquad filters, followed by bass and treble processing, followed by volume and loudness processing, followed by dynamic range control, followed by dither processing. See the TAS3103A Firmware Block Diagram in Appendix A.

Figure 3−9. Input Mixer Topology—Internal Processing Nodes G and H

3.3 3D Effects Block

The 3D effects block, shown in Figure 3−10, performs the first suite of processing tasks conducted on the incoming serial audio data streams. The TAS3103A has three monaural channels—CH1, CH2, and CH3. CH1 and CH2 share the same effects block, as well as the same dynamic range compression block. CH3 has its own effects block and its own dynamic range compression block. In typical two-TAS3103A-chip configurations for processing 5.1 audio, monaural channels CH1 and CH2 are used to process left/right front and left/right surround audio components, and CH3 is used to process the subwoofer and center audio components. To support such a processing structure, the effects block for CH1/CH2 offers more options for inserting audio effects into the audio data stream than does the effects block for CH3.

3.3.1 CH1/CH2 Effects Block

This block consists of five signal flow paths, starting at processing nodes a, b, c, g, and h. All five paths contain four programmable biquad filters, and paths a and b contain reverb delay lines as well. Nodes a, b, and c can be sourced by any of the input nodes A, B, C, D, E, and F (SDIN1, SDIN2, SDIN3). Nodes g and h can be sourced by input nodes G and H, respectively (SDIN4) and/or by weighted replicas of the data on nodes A and B, respectively. Nodes g and h can also be sourced by the same weighted replica of the data on node f. Node c can also be sourced by weighted replicas of the data on both node a and node b.

3D effects processing typically consists of installing sound direction effects. Sound direction effects are typically created by the use of three major components.

- Time differentiation
- Loudness differentiation
- Spectral differentiation

Time differentiation is achieved by using the paths containing the reverb delay elements. Loudness differentiation is achieved both by the mixers feeding the five paths and the two mixers located at the output of each of the five paths. Spectral differentiation is achieved using the four biquad filters located in each path.

3.3.2 CH3 Effects Block

This block, starting at node f, typically processes center and subwoofer audio components. Time and spectral differentiation with respect of CH1 and CH2 can be realized via the reverb delay line and the four biquad filters. Loudness differentiation can be achieved by adjusting the volume level of CH3. Weighted replicas of the effects block output for CH1 and CH2 can also be summed into the output of the CH3 effects block. This capability is typically used when processing the center channel audio component.

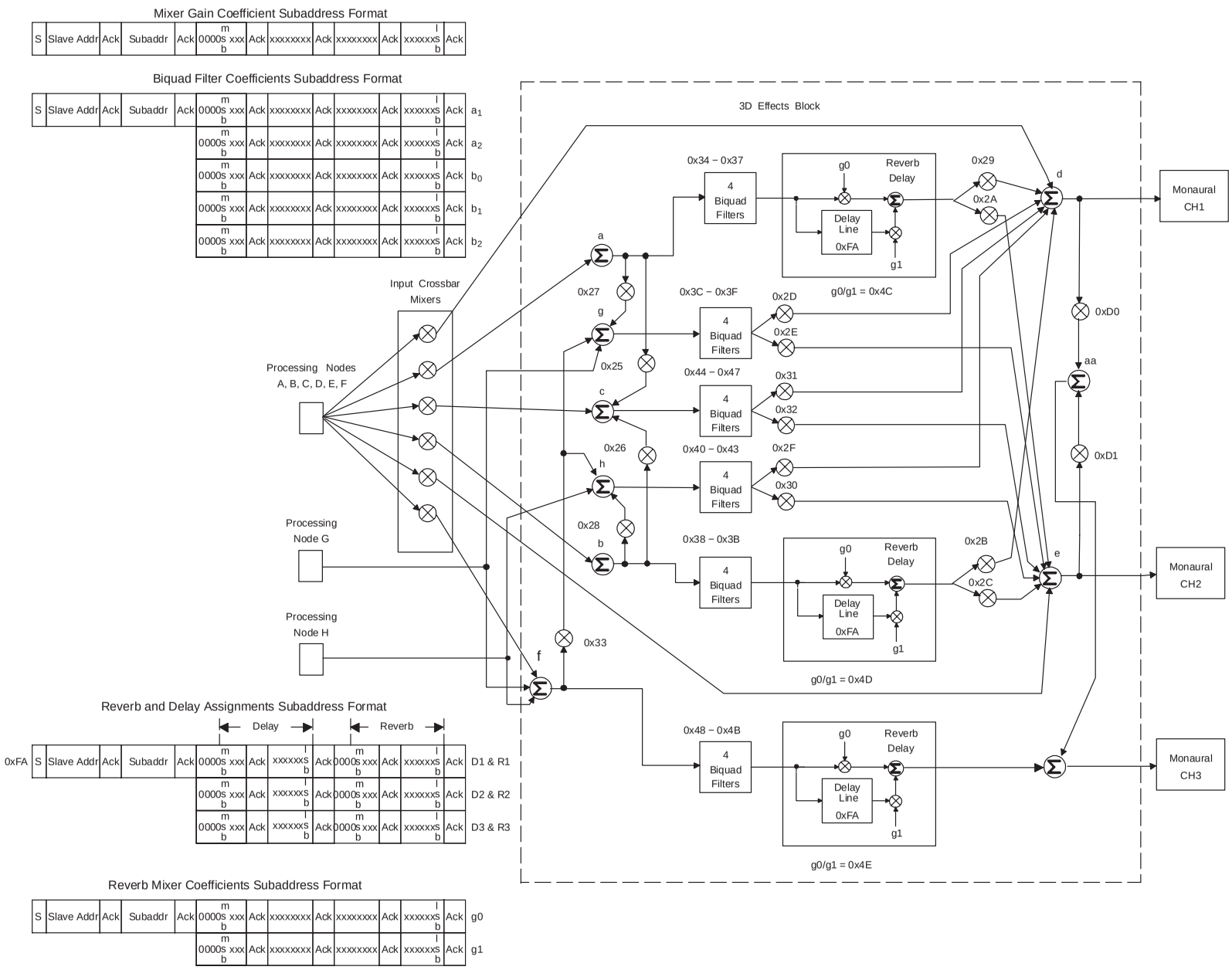


Figure 3−10. TAS3103A 3D Effects Processing Block

3.4 Biquad Filters

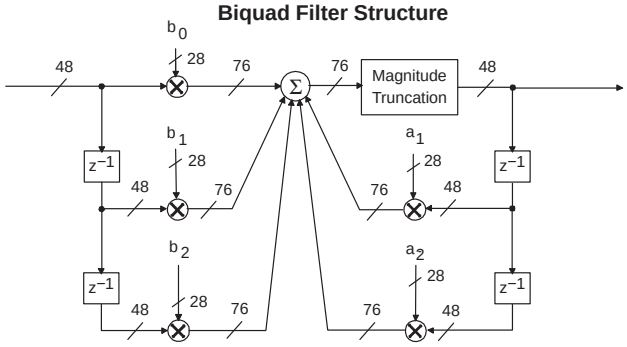
The TAS3103A has 73 biquad filters . The breakout of the biquad filters per functional element is given in Table 3−1.

Table 3−1. Biquad Filter Breakout

FUNCTION	BIQUAD FILTERS	SUBADDRESS
3D effects block	24	0x34−0x4B
Monaural channel CH1	12	0x4F−0x5A
Monaural channel CH2	12	0x5B−0x66
Monaural channel CH3	12	0x67−0x72
Loudness processing	3	0xA6−CH1
		0xAB−CH2
		0xB0−CH3
Spectrum analyzer/VU meter	10	0xBC−0xC5

All 73 biquad filters are second-order direct-form I structures. A block diagram of the structure of the biquad filter is shown in Figure 3−11.

Biquad Filter Structure



NOTE: All gain coefficients 5.23 numbers.

Biquad Filter Coefficient Subaddress Format

S	Slave Addr	Ack	Subaddr	Ack	msb 0000 xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	
					msb 0000 xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	a_1
					msb 0000 xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	a_2
					msb 0000 xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	b_0
					msb 0000 xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	b_1
					msb 0000 xxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	b_2

NOTE: Each biquad filter has one subaddress which contains the mixer gain coefficients a_1, a_2, b_0, b_1, b_2

Figure 3−11. Biquad Filter Structure and Coefficient Subaddress Format

The transfer function for the biquad filter is given by:

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

Note that the signs of a_n coefficients are opposite from Matlab-generated coefficients.

The direct-form I structure provides a separate delay element and mixer (gain coefficient) for each node in the biquad filter. Each mixer output is a signed 76-bit product of a signed 48-bit data sample (25.23-format number) and a signed 28-bit coefficient (5.23-format number). A 76-bit ALU in the TAS3103A allows the 76-bit resolution to be retained when summing the mixer outputs (filter products). This is an important factor, as it removes the need to carefully tailor the order of addition for each filter implementation to minimize the effects of finite-precision arithmetic. Intermediate overflows are allowed while summing the biquad terms to further minimize the effects of finite-precision arithmetic. See *Digital Audio Processor (DAP) Arithmetic Unit*, Section 2.4, for more discussion on intermediate overflow.

3.5 Bass and Treble Processing

The TAS3103A has three fully independent bass and treble adjustment blocks—one for each of the three monaural channels. Adjustments in bass and treble are accomplished by selecting a bass filter set and a treble filter set, and then selecting a shelf filter within each filter set. The filter set selected, of which there are five sets for treble and five sets for bass to select from, determines the frequency at which the bass and treble adjustments take effect. The shelf filters determine the gain to be applied to the bass and treble components of the incoming audio. All selections are independent of one another—any bass filter set can be combined with any treble filter set and any shelf filter can be selected in a given filter set.

Figure 3−13 shows the bass and treble selections available, their I^2C subaddresses, and the data fields in each subaddress used to make the selections. Each bass filter set has 150 low-pass shelf filters to choose from, with the shelves ranging from a cut (attenuation) of 18 dB to a boost (gain) of 18 dB. A shelf selection of 0 dB effectively removes bass processing. All 150 filters in a given filter set have the same 3-dB frequency, as measured from the shelf. The only difference in the 150 filters in a given bass filter set is the gain of the shelf.

Each treble filter set has 150 high-pass shelf filters to choose from, with the shelves again ranging from a cut (attenuation) of 18 dB to a boost (gain) of 18 dB. A shelf selection of 0 dB effectively removes treble processing. All 150 filters in a given filter set have the same 3-dB frequency, as measured from the shelf. The only difference in the 150 filters in a given treble filter set is the gain of the shelf. The three tone controls of the TAS3103A are enabled by registers 0x73, 0x74 and 0x75, shown in Figure 3−12.

Commands to adjust the bass and treble levels within a given filter set (by commanding the selection of different shelf filters) results in a soft adjustment to the newly commanded levels. The filters are labeled 1 through 150, with filter #1 implementing the maximum cut (18 dB) and filter #150 implementing the maximum boost (18 dB). If a command is received to change a shelf setting, the transition is made by stepping through each filter, one at a time, until the shelf filter commanded is reached. A soft transition is achieved by residing at each step (or shelf filter) for a period determined by the programmable parameter TBLC (bit field 7:0 of subaddress 0xF1). The time period set by TBLC is TBLC/Fs (where Fs is the audio sample rate). For an audio sample rate of 48 kHz and a TBLC setting of 64, the dwell time on each shelf filter is approximately 1.33 ms. The transition time then depends on the number of shelves separating the current and commanded shelf values. For 48-kHz audio and a TBLC setting of 64, a maximum transition time of approximately 198 ms is required to transition from shelf #1 to shelf #150 and a minimum time of approximately 1.33 ms is required to transition from shelf x to shelf x + 1.

Reverb Block Gains

Reverb Block	Subaddress
CH1	0x4C
CH2	0x4D
CH3	0x4E

S	Slave Addr	Ack	Subaddr	Ack	msb 0000 sxxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	G0
					msb 0000 sxxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	G1
S	Slave Addr	Ack	Subaddr	Ack	msb 0000 sxxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	G0
					msb 0000 sxxx	Ack	xxxxxxxx	Ack	xxxxxxxx	Ack	xxxxxx lsb	Ack	G1

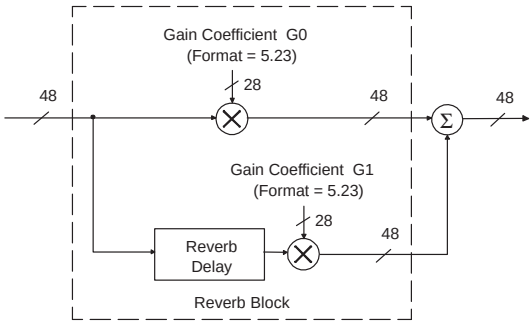
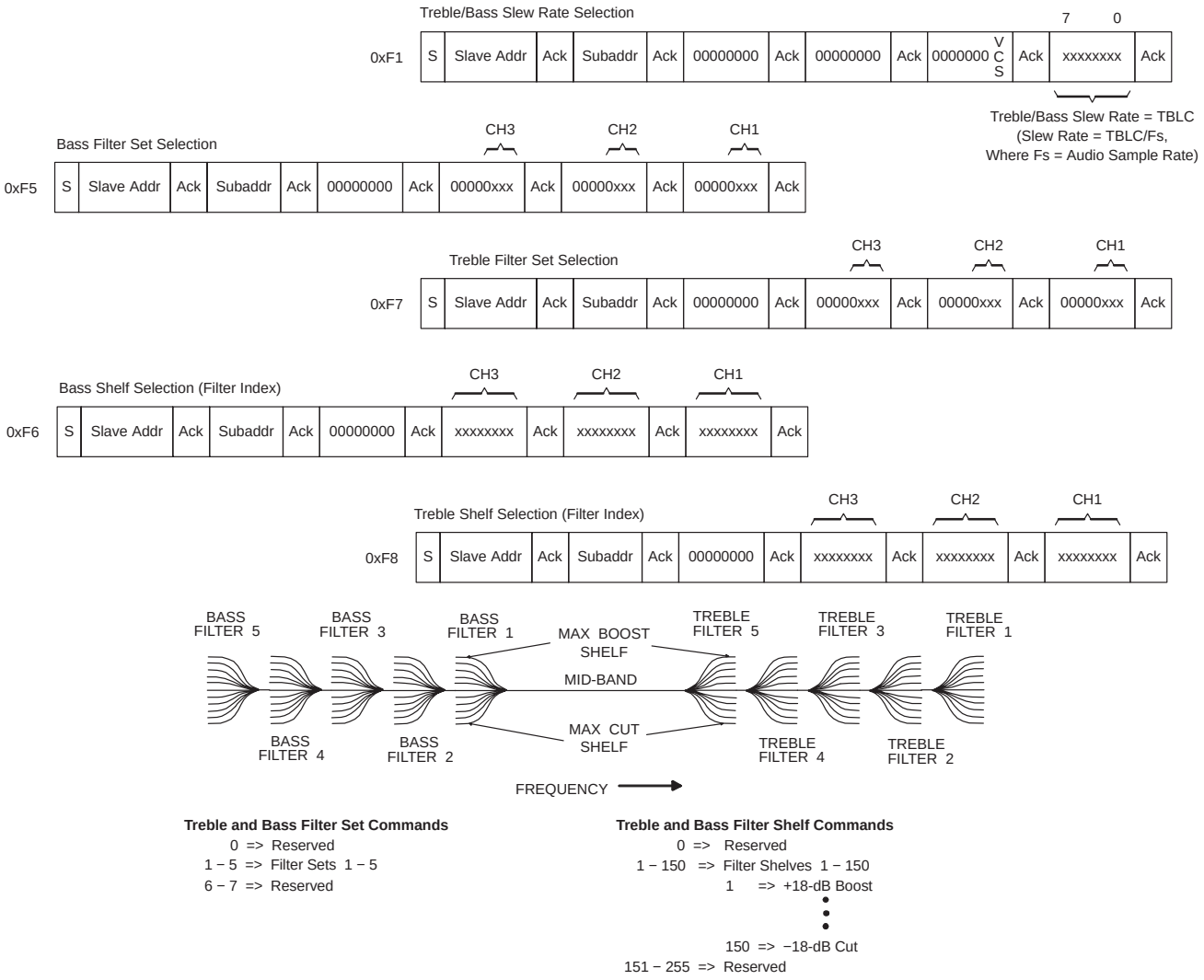


Figure 3−12. Tone Control Registers



Fs (LRCLK)	3-dB CORNERS (kHz)									
	FILTER SET 5		FILTER SET 4 (Default)		FILTER SET 3		FILTER SET 2		FILTER SET 1	
	BASS	TREBLE	BASS	TREBLE	BASS	TREBLE	BASS	TREBLE	BASS	TREBLE
96 kHz	0.25	6	0.5	12	0.75	18	1	24	1.5	36
88.4 kHz	0.23	5.525	0.46	11.05	0.691	16.575	0.921	22.1	1.381	33.15
64 kHz	0.167	4	0.333	8	0.5	12	0.667	16	1	24
48 kHz	0.125	3	0.25	6	0.375	9	0.5	12	0.75	18
44.1 kHz	0.115	2.756	0.23	5.513	0.345	8.269	0.459	11.025	0.689	16.538
38 kHz	0.099	2.375	0.198	4.75	0.297	7.125	0.396	9.5	0.594	14.25
32 kHz	0.083	2	0.167	4	0.25	6	0.333	8	0.5	12
24 kHz	0.063	1.5	0.125	3	0.188	4.5	0.25	6	0.375	9
22.05 kHz	0.057	1.378	0.115	2.756	0.172	4.134	0.23	5.513	0.345	8.269
16 kHz	0.042	1	0.083	2	0.125	3	0.167	4	0.25	6
12 kHz	0.031	0.75	0.063	1.5	0.094	2.25	0.125	3	0.188	4.5
11.025 kHz	0.029	0.689	0.057	1.378	0.086	2.067	0.115	2.756	0.172	4.134

Figure 3−13. Bass and Treble Filter Selections

CAUTION: No soft transition is implemented when changing bass and treble filter sets; soft transitions only apply when adjusting gains (shelves) within a given filter set. The variable TBLC should be set so that the dwell time at each shelf is never less than 32 audio sample periods; otherwise, audio artifacts could be introduced into the audio data stream. It is recommended that these registers be written before the bass/treble bypass values (subaddresses 0x73, 0x74, and 0x75) are set to pass audio through the bass/treble filters.

Figure 3−13 summarizes the bass and treble adjustments available within each monaural channel. As noted in Figure 3−13, the 3-dB frequency for the bass filter sets decreases in value as the filter set number is increased, whereas the 3-dB frequency for the treble filter set increases in value as the filter set number is increased. The valid selection for bass and treble sets ranges from 1 to 5.

Table 3−2 and Table 3−3 list, respectively, the bass and treble filter shelf selections for all 1/2-dB settings between 18-dB cut and 18-dB boost. The treble and bass selections are not the same, and the delta in the selection values between 1/2-dB points is not constant across the 36-dB range. Table 3−2 and Table 3−3 do not list all 150 filter shelf selections, but all 150 selection values for both bass and treble are valid, allowing the use of linear potentiometer or GUI-based sliders. Table 3−2 and Table 3−3 are provided for those applications requiring the adjustment of bass and treble in 1/2-dB steps.

CAUTION: Filter set selections 6 and 7 are reserved. Filter shelf selections 0 and 151 through 255 are reserved. Programming a reserved value could result in erratic and erroneous behavior.

As an example, consider the case of a 44.1-kHz audio sample rate. For this audio rate it is desired to have, for all three monaural channels

- A 3-dB bass shelf corner frequency of 100 Hz
- A bass shelf volume boost of 9 dB
- A 3-dB treble shelf corner frequency of 8.1 kHz
- A treble shelf volume cut of 4 dB

Bass and treble filter set selections can be made by referring to Figure 3−13. For a 44.1-kHz audio sample rate, filter set 5 provides a 3-dB bass corner frequency at 115 Hz, and filter set 3 provides a 3-dB treble corner frequency at 8.269 kHz. These corner frequencies are the closest realizable corner frequencies to the specified 100-Hz bass and 8.1-kHz treble corner frequencies.

Table 3−2 provides the indices for achieving specified bass volume levels. An index of 0x55 yields a bass shelf gain of 9 dB, which matches the specified shelf volume boost of 9 dB. Table 3−3 provides the indices for achieving specified treble volume levels. An index of 0x7A yields a treble shelf cut of 4 dB (−4-dB gain), which matches the specified shelf volume cut of 4 dB.

Figure 3−14 presents the resulting subaddress entries required to implement the parameters specified in the bass and treble example.

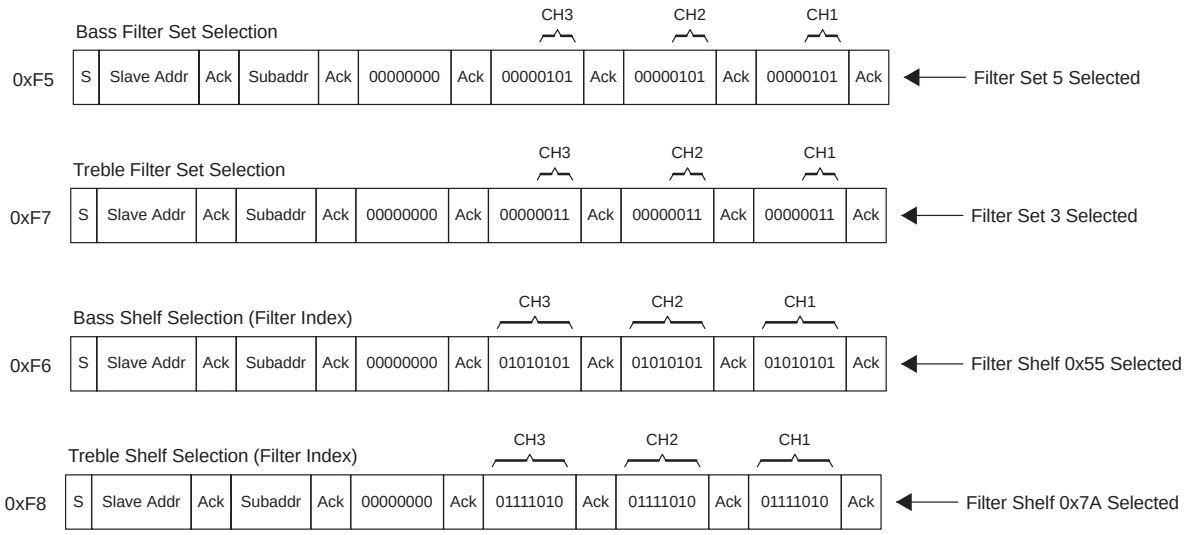


Figure 3−14. Bass and Treble Application Example—Subaddress Parameters

Table 3−2. Bass Shelf Filter Indices for 1/2-dB Adjustments

ADJUSTMENT (dB)	INDEX(1)	ADJUSTMENT (dB)	INDEX(1)	ADJUSTMENT (dB)	INDEX(1)
18	0x01	5.5	0x63	−7	0x80
17.5	0x08	5	0x64	−7.5	0x81
17	0x10	4.5	0x66	−8	0x82
16.5	0x16	4	0x67	−8.5	0x83
16	0x1D	3.5	0x69	−9	0x84
15.5	0x23	3	0x6A	−9.5	0x85
15	0x28	2.5	0x6B	−10	0x86
14.5	0x2D	2	0x6D	−10.5	0x87
14	0x32	1.5	0x6E	−11	0x88
13.5	0x37	1	0x6F	−11.5	0x89
13	0x3B	0.5	0x71	−12	0x8A
12.5	0x3F	0	0x72	−12.5	0x8B
12	0x42	−0.5	0x73	−13	0x8C
11.5	0x46	−1	0x74	−13.5	0x8D
11	0x49	−1.5	0x75	−14	0x8E
10.5	0x4C	−2	0x76	−14.5	0x8F
10	0x4F	−2.5	0x77	−15	0x90
9.5	0x52	−3	0x78	−15.5	0x91
9	0x55	−3.5	0x79	−16	0x92
8.5	0x58	−4	0x7A	−16.5	0x93
8	0x5A	−4.5	0x7B	−17	0x94
7.5	0x5C	−5	0x7C	−17.5	0x95
7	0x5E	−5.5	0x7D	−18	0x96
6.5	0x60	−6	0x7E		
6	0x62	−6.5	0x7F		

(1) CH1 index is subaddress 0xF6, bit field 7:0. CH2 index is subaddress 0xF6, bit field 15:8. CH3 index is subaddress 0xF6, bit field 23:16.

Table 3−3. Treble Shelf Filter Indices for 1/2-dB Adjustments

ADJUSTMENT (dB)	INDEX(1)	ADJUSTMENT (dB)	INDEX(1)	ADJUSTMENT (dB)	INDEX(1)
18	0x01	5.5	0x63	−7	0x80
17.5	0x09	5	0x65	−7.5	0x81
17	0x10	4.5	0x66	−8	0x82
16.5	0x16	40	0x68	−8.5	0x83
16	0x1C	3.5	0x69	−9	0x84
15.5	0x22	3	0x6B	−9.5	0x85
15	0x28	2.5	0x6C	−10	0x86
14.5	0x2D	2	0x6D	−10.5	0x87
14	0x31	1.5	0x6F	−11	0x88
13.5	0x35	1	0x70	−11.5	0x89
13	0x3A	0.5	0x71	−12	0x8A
12.5	0x3E	0	0x72	−12.5	0x8B
12	0x42	−0.5	0x73	−13	0x8C
11.5	0x45	−1	0x74	−13.5	0x8D
11	0x49	−1.5	0x75	−14	0x8E
10.5	0x4C	−2	0x76	−14.5	0x8F
10	0x4F	−2.5	0x77	−15	0x90
9.5	0x52	−3	0x78	−15.5	0x91
9	0x55	−3.5	0x79	−16	0x92
8.5	0x57	−4	0x7A	−16.5	0x93
8	0x5A	−4.5	0x7B	−17	0x94
7.5	0x5C	−5	0x7C	−17.5	0x95
7	0x5E	−5.5	0x7D	−18	0x96
6.5	0x60	−6	0x7E		
6	0x62	−6.5	0x7F		

(1) CH1 index is subaddress 0xF8, bit field 7:0. CH2 index is subaddress 0xF8, bit field 15:8. CH3 index is subaddress 0xF8, bit field 23:16.

3.5.1 Treble and Bass Processing and Concurrent I2C Read Transactions

I2C read transactions at subaddresses 0x01 through 0xD1 are not allowed during: (1) transitions between filter shelves within a given bass or treble filter set or (2) transitions between filter sets. This means that after issuing an I2C command to change treble or bass filter shelves or filter sets, time must be allowed to complete the bass/treble activity before proceeding to read subaddresses 0x01 through 0xD1. No subaddress is available to monitor bass/treble activity directly. However, the state of bass/treble activity can be probed via the *factory test* subaddresses 0xEC and 0xED.

If it is required to read subaddress data that falls in the subaddress range of 0x01 through 0xD1 after issuing an I2C command to change treble or bass filter shelves or filter sets, the procedure presented in Figure 3−15 can be used to monitor bass/treble activity. As Figure 3−15 shows, six readings must be taken to verify that none of the three monaural channels have ongoing bass or treble activity. If all six readings show the value zero in the 8th or least significant byte of the 8-byte data word output at subaddress 0xED, then all commanded bass/treble activity has completed and it is safe to resume I2C read transactions at subaddresses 0x01 through 0xD1.

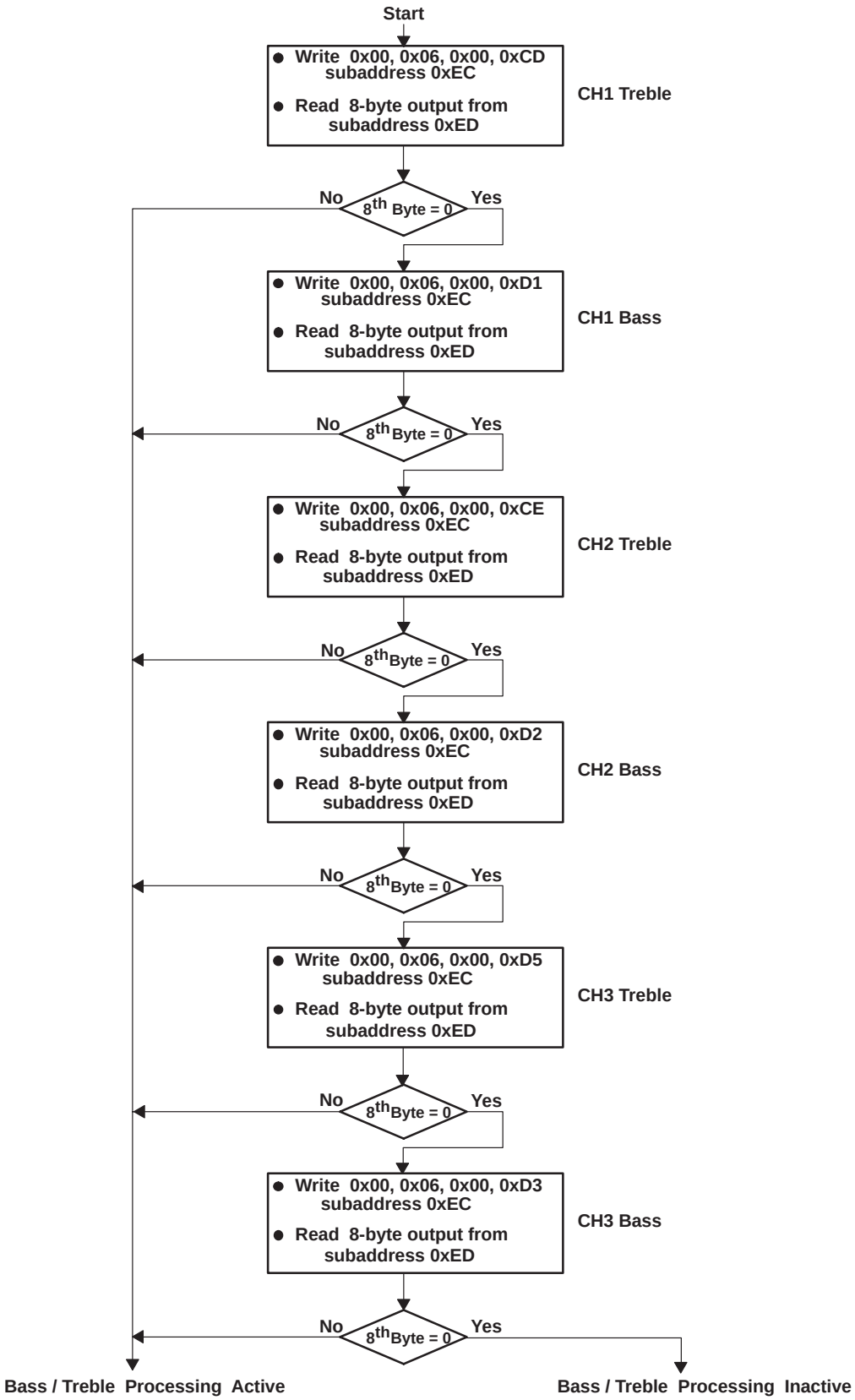


Figure 3−15. I²C Bass/Treble Activity Monitor Procedure

3.6 Soft Volume/Loudness Processing

Each of the three monaural channels in the TAS3103A has dedicated soft volume control and loudness compensation. Volume level changes are issued by I^2C bus commands in the I^2C slave mode and by setting the appropriate GPIO pin to logic 0 in the I^2C master mode. Commanded changes in volume are implemented softly, using a smooth S-curve trajectory to transition the volume to the newly commanded level.

Volume commands are formatted as signed 5.23 numbers. The maximum volume boost then is 24 dB (4 bits $\times$ 6 dB/bit); the maximum volume cut is ∞ with a volume command of zero. The maximum finite volume cut is 138 dB (23 bits $\times$ 6 dB/bit). The resolution of the volume adjustment is not fixed over all gain settings. For large gains, the resolution of the volume adjustment is fine, and the resolution of the volume adjustment decreases as the volume gain decreases. As an example, at maximum gain, the volume level can be adjusted to a resolution of 0.000001 dB [(138 + 24) dB adjustment range/2^{27} adjustment steps]. At the other end of the gain scale, if the volume setting is at the maximum finite cut (volume command = 0x0000001) and is increased by one count (volume command = 0x0000002), a 6-dB adjustment is realized. If an additional volume command is sent, the TAS3103A saves the last command entered.

Each monaural channel volume control is also assigned a separate mute command, which has the same effect as issuing a zero-valued volume command. If loudness is enabled, disabling it by setting the parameter G to zero is necessary to obtain a total cut ($-\infty$ dB). This requirement is further discussed in the paragraphs that follow.

Loudness compensation tracks the volume control setting to allow spectral compensation. An example of loudness compensation would be a boost in bass frequencies to compensate for weak perceived bass at low volume levels. Both linear and log control laws can be implemented for volume gain tracking, and a dedicated biquad filter can be used to achieve spectral discernment.

3.6.1 Soft Volume

Figure 3−16 is a simplified block diagram of the implementation of soft volume and loudness compensation. A volume level change (either via an I^2C-bus-issued command in the I^2C slave mode, or a GPIO-issued command in the I^2C master mode) initiates a transition process that ensures a smooth transition to the newly commanded volume level without producing artifacts such as pops, clicks, and zipper noise. The transition time, or volume slew rate, can be selected to occupy a time window of either 2048 or 4096 audio sample periods. The value of VSC (bit 8 of register 0xF1) is not correctly reported when the register is read. VSC behaves like a write-only bit. For 48-kHz audio, for example, this equates to a transition time of 42.67 ms or 85.34 ms. It is anticipated that 42.67 ms is the transition time of choice for most applications, and the 4096 sample transition option is primarily included to yield the same 42.67-ms transition time for 96-kHz audio. The slope of the S-curve (and its implementation) is proprietary and cannot be altered.

If additional volume commands for a given monaural channel are received, while a previously commanded volume change is still active, the last command received overwrites previous commands and is used. When the previously commanded transition completes, the volume command last received while the transition was taking place is acted on and a new transition to this volume level initiated. For example, assume three volume commands are sequentially issued, via the I^2C bus, to adjust the volume levels of the three monaural channels in the TAS3103A. The first command received immediately triggers the start of a soft volume transition to the newly commanded level. The other two volume commands are received and queued to await the completion of the currently active soft volume transaction. When the first soft volume transition completes, and assuming no further volume commands have been received to replace the other two volume commands received, the next two volume commands are activated, and soft volume transitions on both monaural channels takes place. The total time then, for a 48-kHz audio sample rate and a transition selection of 2048 Fs cycles is 2 $\times$ 42.67 ms or 85.34 ms. If, during the first soft volume transition, a second volume command is received for this same monaural channel, the second soft volume transition period would have soft volume transitions taking place on all three monaural channels. It is also noted that the soft volume transition time is independent of the magnitude of the adjustment. All volume commands take 2048 or 4096 Fs cycles, regardless of the magnitude of the change.

In the I^2C slave mode, the status of a commanded soft volume transition can be found by reading bit 0 of the 32-bit data word retrieved at subaddress 0xFF. If this bit is set to logic 1, one or more monaural channels are actively transitioning their volume setting. If a volume transition is taking place on one of the monaural channels in the

TAS3103A, volume commands received for the other two monaural channels are not acted on until the active volume transition completes. When the active volume transition does complete, the latest volume command received for the three monaural channels during the previous soft volume transition time are serviced. LRCLK should not be stopped during a volume transition.

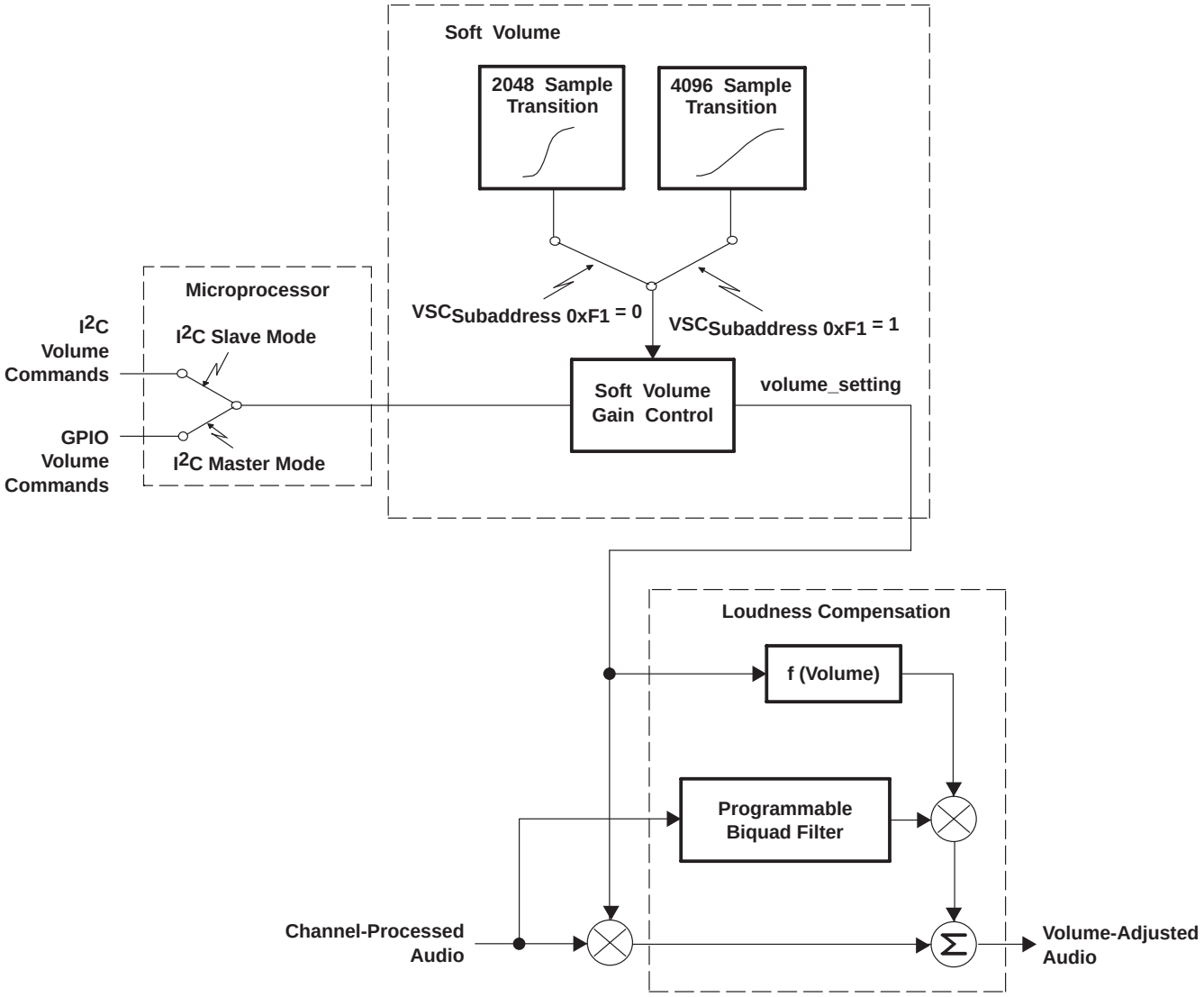


Figure 3−16. Soft Volume and Loudness Compensation Block Diagram

Figure 3−15 is a more detailed block diagram of soft volume and loudness compensation, and includes the I^2C subaddress commands that control volume and loudness compensation. Volume control is accomplished using three I^2C subaddresses—volume control, mute/unmute control, and volume slew rate control.

Volume control in the TAS3103A applies a linear gain. The volume commands issued via I^2C subaddresses 0xF2, 0xF3, and 0xF4 (monaural channels 1, 2, and 3, respectively) are signed 5.23-format numbers. These commands are applied to mixers, whose other input port is the audio data stream. The mixer output is the product of the audio data stream and the volume command. Examples of volume command settings follow.

Volume command = 0x3580 B07 = 0011 0.101 1000 0000 1011 0000 0111 = 6.6878365

Volume command = 0x01F0 000 = 0000 0.001 1111 0000 0000 0000 0000 = 0.2421875

Volume command = 0xCD6E FFE = 1100 1.101 0110 1110 1111 1111 1110 = −6.3208010

The volume control range is 0 = $-\infty$ to 2^{-23} = −138.47 dB to $2^{4} - 2^{-23}$ = 24.08 dB. Volume control is achieved by means of a 5.23-format gain coefficient that is applied to a linear mixer. The volume gain setting realized, for a given volume gain coefficient is:

Gain = 20 log (Volume_Gain_Coefficient)

Several techniques of volume management for a linear volume control process are:

- Precise calculations involving logarithms can be employed.
- A high-resolution gain table, with entries for every 0.5-dB step, can be employed.
- A more coarse gain table (entries in 3- to 6-dB steps with linear interpolation between entries) can be employed.
- Or approximations involving simple calculations can be employed.

As an example of using approximations, equations for increasing a linear 5.23 gain setting X by 0.5 dB that involve only simple binary shift and add operations and the accuracy of these equations follow.

$20 \log_{10}X + 0.5\ dB \approx X + 2^{-4}X$ gives 0.52657877-dB steps

$20 \log_{10}X + 0.5\ dB \approx X + 2^{-4}X - 2^{-8}X$ gives 0.49458651-dB steps

$20 \log_{10}X + 0.5\ dB \approx X + 2^{-4}X - 2^{-8}X + 2^{-11}X$ gives 0.49859199-dB steps

$20 \log_{10}X + 0.5\ dB \approx X[1 + 2^{-4} - 2^{-8} + 2^{-11} + 2^{-12} - 2^{-13} + 2^{-14} - 2^{-16} + 2^{-18}]$ gives 0.4999997332 dB−steps

Approximations can also be found for decreasing a linear 5.23 gain setting X by 0.5 dB that involve using only simple binary shift and add operations, but the equations differ slightly from those used to increase the gain in 0.5-dB steps. The approximations to decrease the volume by 0.5 dB and the accuracy of these approximations follow.

$20 \log_{10}X - 0.5\ dB \approx X - 2^{-4}X$ gives −0.56057447 dB-steps

$20 \log_{10}X - 0.5\ dB \approx X - 2^{-4}X + 2^{-7}X$ gives −0.48849199-dB steps

$20 \log_{10}X - 0.5\ dB \approx X - 2^{-4}X + 2^{-7}X - 2^{-10}X$ gives −0.49746965-dB steps

$20 \log_{10}X - 0.5\ dB \approx X[1 - 2^{-4} + 2^{-7} - 2^{-10} - 2^{-12} - 2^{-15} - 2^{-21}]$ gives −0.5000006792 dB-steps

Repeated use of a set of the preceding approximations results in an accumulation of the errors in the approximations. For example, if an application started at 0-dB volume, and repeatedly used the approximations to increase and decrease the volume, the exact reference point of 0 dB would be lost. If an application does require the maintenance of accurate reference points, it is necessary for the application to establish a set of exact gain reference settings and command these exact settings in place of a computed gain setting whenever the current gain setting and the next computed gain setting straddle an exact gain setting.

Table 3−4 lists the I^{2}C coefficient settings to adjust volume from 24 dB to −136 dB in 0.5-dB steps. For each volume setting, the gain in dB is presented in one column, the same gain in a floating point number $\left(\text{float} = 10^{\frac{\text{Gain}_{dB}}{20}}\right)$ is presented in the adjacent column, and the same gain formatted in the 32-bit hexadecimal gain coefficient format required to enter the value into the TAS3103A via the I^{2}C bus is presented in a third column.

Table 3−4. Volume Adjustment Gain Coefficients

GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)	GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)
24	15.84893192	07EC A9CD	2	1.25892541	00A1 2477
23.5	14.96235656	077B 2E7F	1.5	1.18850223	0098 20D7
23	14.12537545	0710 0C4D	1	1.12201845	008F 9E4C
22.5	13.33521432	06AA E84D	0.5	1.05925373	0087 95A0
22	12.58925412	064B 6CAD	0	1	0080 0000
21.5	11.88502227	05F1 4868	−0.5	0.94406088	0078 D6FC
21	11.22018454	059C 2F01	−1	0.89125094	0072 1482
20.5	10.59253725	054B D842	−1.5	0.84139514	006B B2D6
20	10	0500 0000	−2	0.79432823	0065 AC8C
19.5	9.44060876	04B8 65DE	−2.5	0.74989421	005F FC88
19	8.91250938	0474 CD1B	−3	0.70794578	005A 9DF7
18.5	8.41395142	0434 FC5C	−3.5	0.66834392	0055 8C4B
18	7.94328235	03F8 BD79	−4	0.63095734	0050 C335
17.5	7.49894209	03BF DD55	−4.5	0.59566214	004C 3EA8
17	7.07945784	038A 2BAC	−5	0.56234133	0047 FACC
16.5	6.68343918	0357 7AEF	−5.5	0.53088444	0043 F405
16	6.30957344	0327 A01A	−6	0.50118723	0040 26E7
15.5	5.95662144	02FA 7292	−6.5	0.47315126	003C 9038
15	5.62341325	02CF CC01	−7	0.44668359	0039 2CED
14.5	5.30884444	02A7 8836	−7.5	0.42169650	0035 FA26
14	5.01187234	0281 8508	−8	0.39810717	0032 F52C
13.5	4.73151259	025D A234	−8.5	0.37583740	0030 1B70
13	4.46683592	023B C147	−9	0.35481339	002D 6A86
12.5	4.21696503	021B C582	−9.5	0.33496544	002A E025
12	3.98107171	01FD 93C1	−10	0.31622777	0028 7A26
11.5	3.75837404	01E1 1266	−10.5	0.29853826	0026 3680
11	3.54813389	01C6 2940	−11	0.28183829	0024 1346
10.5	3.34965439	01AC C179	−11.5	0.26607251	0022 0EA9
10	3.16227766	0194 C583	−12	0.25118864	0020 26F3
9.5	2.98538262	017E 2104	−12.5	0.23713737	001E 5A84
9	2.81838293	0168 C0C5	−13	0.22387211	001C A7D7
8.5	2.66072506	0154 92A3	−13.5	0.21134890	001B 0D7B
8	2.51188643	0141 857E	−14	0.19952623	0019 8A13
7.5	2.37137371	012F 892C	−14.5	0.18836491	0018 1C57
7	2.23872114	011E 8E6A	−15	0.17782794	0016 C310
6.5	2.11348904	010E 86CF	−15.5	0.16788040	0015 7D1A
6	1.99526231	00FF 64C1	−16	0.15848932	0014 4960
5.5	1.88364909	00F1 1B69	−16.5	0.14962357	0013 26DD
5	1.77827941	00E3 9EA8	−17	0.14125375	0012 149A
4.5	1.67880402	00D6 E30C	−17.5	0.13335214	0011 11AE
4	1.58489319	00CA DDC7	−18	0.12589254	0010 1D3F
3.5	1.49623566	00BF 84A6	−18.5	0.11885022	000F 367B
3	1.41253754	00B4 CE07	−19	0.11220185	000E 5CA1
2.5	1.33352143	00AA B0D4	−19.5	0.10592537	000D 8EF6

Table 3−4. Volume Adjust Gain Coefficient (Continued)

GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)	GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)
−20	0.1	000C CCCC	−42	0.00794328	0001 0449
−20.5	0.09440609	000C 157F	−42.5	0.00749894	0000 F5B9
−21	0.08912509	000B 6873	−43	0.00707946	0000 E7FA
−21.5	0.08413951	000A C515	−43.5	0.00668344	0000 DB00
−22	0.07943282	000A 2ADA	−44	0.00630957	0000 CEC0
−22.5	0.07498942	0009 9940	−44.5	0.00595662	0000 C32F
−23	0.07079458	0009 0FCB	−45	0.00562341	0000 B844
−23.5	0.06683439	0008 8E07	−45.5	0.00530884	0000 ADF5
−24	0.06309573	0008 1385	−46	0.00501187	0000 A43A
−24.5	0.05956621	0007 9FDD	−46.5	0.00473151	0000 9B0A
−25	0.05623413	0007 32AE	−47	0.00446684	0000 925E
−25.5	0.05308844	0006 CB9A	−47.5	0.00421697	0000 8A2E
−26	0.05011872	0006 6A4A	−48	0.00398107	0000 8273
−26.5	0.04731513	0006 0E6C	−48.5	0.00375837	0000 7B27
−27	0.04466836	0005 B7B1	−49	0.00354813	0000 7443
−27.5	0.04216965	0005 65D0	−49.5	0.00334965	0000 6DC2
−28	0.03981072	0005 1884	−50	0.00316228	0000 679F
−28.5	0.03758374	0004 CF8B	−50.5	0.00298538	0000 61D3
−29	0.03548134	0004 8AA7	−51	0.00281838	0000 5C5A
−29.5	0.03349654	0004 499D	−51.5	0.00266073	0000 572F
−30	0.03162278	0004 0C37	−52	0.00251189	0000 524F
−30.5	0.02985383	0003 D240	−52.5	0.00237137	0000 4DB4
−31	0.02818383	0003 9B87	−53	0.00223872	0000 495B
−31.5	0.02660725	0003 67DD	−53.5	0.00211349	0000 4541
−32	0.02511886	0003 3718	−54	0.00199526	0000 4161
−32.5	0.02371374	0003 090D	−54.5	0.00188365	0000 3DB9
−33	0.02238721	0002 DD95	−55	0.00177828	0000 3A45
−33.5	0.02113489	0002 B48C	−55.5	0.00167880	0000 3702
−34	0.01995262	0002 8DCE	−56	0.00158489	0000 33EF
−34.5	0.01883649	0002 693B	−56.5	0.00149624	0000 3107
−35	0.01778279	0002 46B4	−57	0.00141254	0000 2E49
−35.5	0.01678804	0002 261C	−57.5	0.00133352	0000 2BB2
−36	0.01584893	0002 0756	−58	0.00125893	0000 2940
−36.5	0.01496236	0001 EA49	−58.5	0.00118850	0000 26F1
−37	0.01412538	0001 CEDC	−59	0.00112202	0000 24C4
−37.5	0.01333521	0001 B4F7	−59.5	0.00105925	0000 22B5
−38	0.01258925	0001 9C86	−60	0.001	0000 20C4
−38.5	0.01188502	0001 8572	−60.5	0.00094406	0000 1EEF
−39	0.01122018	0001 6FA9	−61	0.00089125	0000 1D34
−39.5	0.01059254	0001 5B18	−61.5	0.00084140	0000 1B92
−40	0.01	0001 47AE	−62	0.00079433	0000 1A07
−40.5	0.00944061	0001 3559	−62.5	0.00074989	0000 1892
−41	0.00891251	0001 240B	−63	0.00070795	0000 1732
−41.5	0.00841395	0001 13B5	−63.5	0.00066834	0000 15E6

Table 3−4. Volume Adjust Gain Coefficient (Continued)

GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)	GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)
−64	0.00063096	0000 14AC	−86	5.01188E−05	0000 01A4
−64.5	0.00059566	0000 1384	−86.5	4.73152E−05	0000 018C
−65	0.00056234	0000 126D	−87	4.46684E−05	0000 0176
−65.5	0.00053088	0000 1165	−87.5	4.21696E−05	0000 0161
−66	0.00050119	0000 106C	−88	3.98108E−05	0000 014D
−66.5	0.00047315	0000 0F81	−88.5	3.75838E−05	0000 013B
−67	0.00044668	0000 0EA3	−89	3.54814E−05	0000 0129
−67.5	0.00042170	0000 0DD1	−89.5	3.34966E−05	0000 0118
−68	0.00039811	0000 0D0B	−90	3.16228E−05	0000 0109
−68.5	0.00037584	0000 0C50	−90.5	2.98538E−05	0000 00FA
−69	0.00035481	0000 0BA0	−91	2.81838E−05	0000 00EC
−69.5	0.00033497	0000 0AF9	−91.5	2.66072E−05	0000 00DF
−70	0.00031623	0000 0A5C	−92	2.51188E−05	0000 00D2
−70.5	0.00029854	0000 09C8	−92.5	2.37138E−05	0000 00C6
−71	0.00028184	0000 093C	−93	2.23872E−05	0000 00BB
−71.5	0.00026607	0000 08B7	−93.5	2.11348E−05	0000 00B1
−72	0.00025119	0000 083B	−94	1.99526E−05	0000 00A7
−72.5	0.00023714	0000 07C5	−94.5	1.88365E−05	0000 009E
−73	0.00022387	0000 0755	−95	1.77828E−05	0000 0095
−73.5	0.00021135	0000 06EC	−95.5	1.67880E−05	0000 008C
−74	0.00019953	0000 0689	−96	1.58489E−05	0000 0084
−74.5	0.00018837	0000 062C	−96.5	1.49624E−06	0000 007D
−75	0.00017783	0000 05D3	−97	1.41254E−05	0000 0076
−75.5	0.00016788	0000 0580	−97.5	1.33352E−05	0000 006F
−76	0.00015849	0000 0531	−98	1.25893E−05	0000 0069
−76.5	0.00014962	0000 04E7	−98.5	1.18850E−05	0000 0063
−77	0.00014125	0000 04A0	−99	1.12202E−05	0000 005E
−77.5	0.00013335	0000 045E	−99.5	1.05925E−05	0000 0058
−78	0.00012589	0000 0420	−100	0.00001	0000 0053
−78.5	0.00011885	0000 03E4	−100.5	9.44060E−06	0000 004F
−79	0.00011220	0000 03AD	−101	8.91250E−06	0000 004A
−79.5	0.00010593	0000 0378	−101.5	8.41396E−06	0000 0046
−80	0.0001	0000 0346	−102	7.94328E−06	0000 0042
−80.5	9.44060E−05	0000 0317	−102.5	7.49894E−06	0000 003E
−81	8.91250E−05	0000 02EB	−103	7.07946E−06	0000 003B
−81.5	8.41396E−05	0000 02C1	−103.5	6.68344E−06	0000 0038
−82	7.94328E−05	0000 029A	−104	6.30958E−06	0000 0034
−82.5	7.49894E−05	0000 0275	−104.5	5.95662E−06	0000 0031
−83	7.07946E−05	0000 0251	−105	5.62342E−06	0000 002F
−83.5	6.68344E−05	0000 0230	−105.5	5.30884E−06	0000 002C
−84	6.30958E−05	0000 0211	−106	5.01188E−06	0000 002A
−84.5	5.95662E−05	0000 01F3	−106.5	4.73152E−06	0000 0027
−85	5.62342E−05	0000 01D7	−107	4.46684E−06	0000 0025
−85.5	5.30884E−05	0000 01BD	−107.5	4.21696E−06	0000 0023

Table 3−4. Volume Adjust Gain Coefficient (Continued)

GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)	GAIN (dB)	GAIN (FLOAT)	GAIN (COEFFICIENT)
−108	3.98108E−06	0000 0021	−122.5	7.49894E−07	0000 0006
−108.5	3.75838E−06	0000 001F	−123	7.07946E−07	0000 0005
−109	3.54814E−06	0000 001D	−123.5	6.68344E−07	0000 0005
−109.5	3.34966E−06	0000 001C	−124	6.30958E−07	0000 0005
−110	3.16228E−06	0000 001A	−124.5	5.95662E−07	0000 0004
−110.5	2.98538E−06	0000 0019	−125	5.62342E−07	0000 0004
−111	2.81838E−06	0000 0017	−125.5	5.30884E−07	0000 0004
−111.5	2.66072E−06	0000 0016	−126	5.01188E−07	0000 0004
−112	2.51188E−06	0000 0015	−126.5	4.73152E−07	0000 0003
−112.5	2.37138E−06	0000 0013	−127	4.46684E−07	0000 0003
−113	2.23872E−06	0000 0012	−127.5	4.21696E−07	0000 0003
−113.5	2.11348E−06	0000 0011	−128	3.98108E−07	0000 0003
−114	1.99526E−06	0000 0010	−128.5	3.75838E−07	0000 0003
−114.5	1.88365E−06	0000 000F	−129	3.54814E−07	0000 0002
−115	1.77828E−06	0000 000E	−129.5	3.34966E−07	0000 0002
−115.5	1.67880E−06	0000 000E	−130	3.16228E−07	0000 0002
−116	1.58489E−06	0000 000D	−130.5	2.98538E−07	0000 0002
−116.5	1.49624E−06	0000 000C	−131	2.81838E−07	0000 0002
−117	1.41254E−06	0000 000B	−131.5	2.66072E−07	0000 0002
−117.5	1.33352E−06	0000 000B	−132	2.51188E−07	0000 0002
−118	1.25893E−06	0000 000A	−132.5	2.37138E−07	0000 0001
−118.5	1.18850E−06	0000 0009	−133	2.23872E−07	0000 0001
−119	1.12202E−06	0000 0009	−133.5	2.11348E−07	0000 0001
−119.5	1.05925E−06	0000 0008	−134	1.99526E−07	0000 0001
−120	0.000001	0000 0008	−134.5	1.88365E−07	0000 0001
−120.5	9.44080E−07	0000 0007	−135	1.77828E−07	0000 0001
−121	8.91250E−07	0000 0007	−135.5	1.67880E−07	0000 0001
−121.5	8.41396E−07	0000 0007	−136	1.58489E−07	0000 0001
−122	7.94328E−07	0000 0006			

3.6.1.1 Soft Volume Adjustment Range Limitations

When the 2048 sample transition time is selected for the S-curve volume transition, the full −∞ to 24-dB adjustment range is available. All possible values but one in the signed 28-bit (5.23-format) volume-level command range are valid volume-level selections. The one exception is the maximum negative volume-level command 0x08000000. This value is reserved and if used, could result in erratic behavior that requires a reset to the part to correct.

When the 4096-sample transition time is selected, the upper two bits of the 28-bit volume command cannot be used. This means that the valid volume-level command range is reduced to a maximum positive value of 0x1FFFFFF (12 dB) and a maximum negative value of 0x0E000000 (also 12 dB, but with a 180° phase inversion of the audio signal). Values above these maximum levels could result in erratic behavior that requires a reset to the part to correct. Although the volume boost for the 4096-sample transition time is limited to 12 dB, volume boost can be realized elsewhere in the processing signal flow, such as in the input crossbar mixers, the biquad filters in the three monaural channels, or the biquad filters in the effects block.

3.6.1.2 Soft Volume Transitions and Concurrent I^2C Read Transactions

I^2C read transactions at subaddresses 0x01 through 0xD1 are not allowed during volume-level transitions, as read activity during volume-level transitions could result in erroneous data being output. If it is required to read subaddress

data that falls in the subaddress range 0x01 through 0xD1 after issuing an I^2C command to change the volume level on one of the three monaural channels, the *busy* bit at subaddress 0xFF must be monitored to determine when the volume activity has ceased and it is safe to resume I^2C read activity at subaddresses 0x01 through 0xD1. A value of 0 in the least-significant bit of the byte output on reading subaddress 0xFF signifies that all volume transition activity has completed.

3.6.2 Loudness Compensation

Loudness compensation employs a single, coefficient-programmable, biquad filter and a function block f(volume_setting), whose output is only a function of the volume control setting. The biquad filter input is the channel processed audio data stream that also feeds the volume gain-control mixer. The biquad output feeds a gain-control mixer whose other input is the volume-control setting after processing by the function block f(volume setting). Loudness compensation then allows a given spectral segment of the audio data stream (as determined by the biquad filter coefficients) to be given a delta adjustment in volume as determined by the programmable function block f(volume_setting).

The output of the function block f(volume_setting) can be expressed in terms of the programmable I^2C coefficients as:

$$f(volume\_setting) = [(volume\_setting)^{LG} \times 2^{LO} \times G] + O$$

where:

LG = logarithmic gain = 5.23-format number

LO = logarithmic offset = 25.23-format number

G = gain = 5.23-format number

O = offset = 25.23-format number

If LG = 0.5, LO = 0, G = 1.0, and O = 0, f(volume setting) becomes:

$$f(volume\_setting) = \sqrt{volume\_setting}$$

The output of the soft volume/loudness compensation block for the preceding case would be:

$$audio_{Out} = [audio_{In} \times volume\_setting] + [audio_{In} \times F(s)_{Biquad} \times \sqrt{volume\_setting}\,]$$

Figure 3−17 is a block diagram of the loudness logic. If a given monaural channel is set up as in the preceding example, a true mute is not obtained when a mute command is issued for the monaural channel. The function block f(volume_setting) approximates logarithmic arithmetic and a consequence of the mathematical approximations used is that the function block can output a nonzero value for an input volume setting of 0. For f(volume_setting) = $\sqrt{volume\_setting}$, a volume_setting value of 0.0 results in the function block outputting a gain coefficient that cuts the output of the biquad filter by approximately 90 dB. If true muting is required, it can be achieved by setting G = 0 after the volume control setting has softly transitioned to 0.

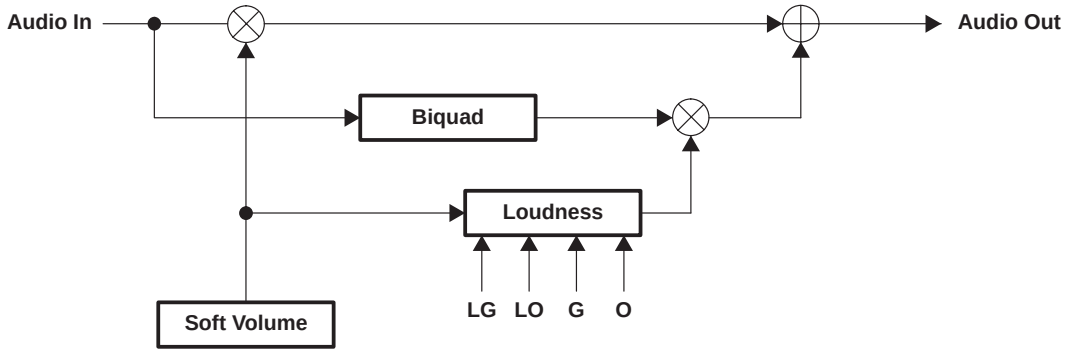


Figure 3−17. Loudness Compensation Block Diagram

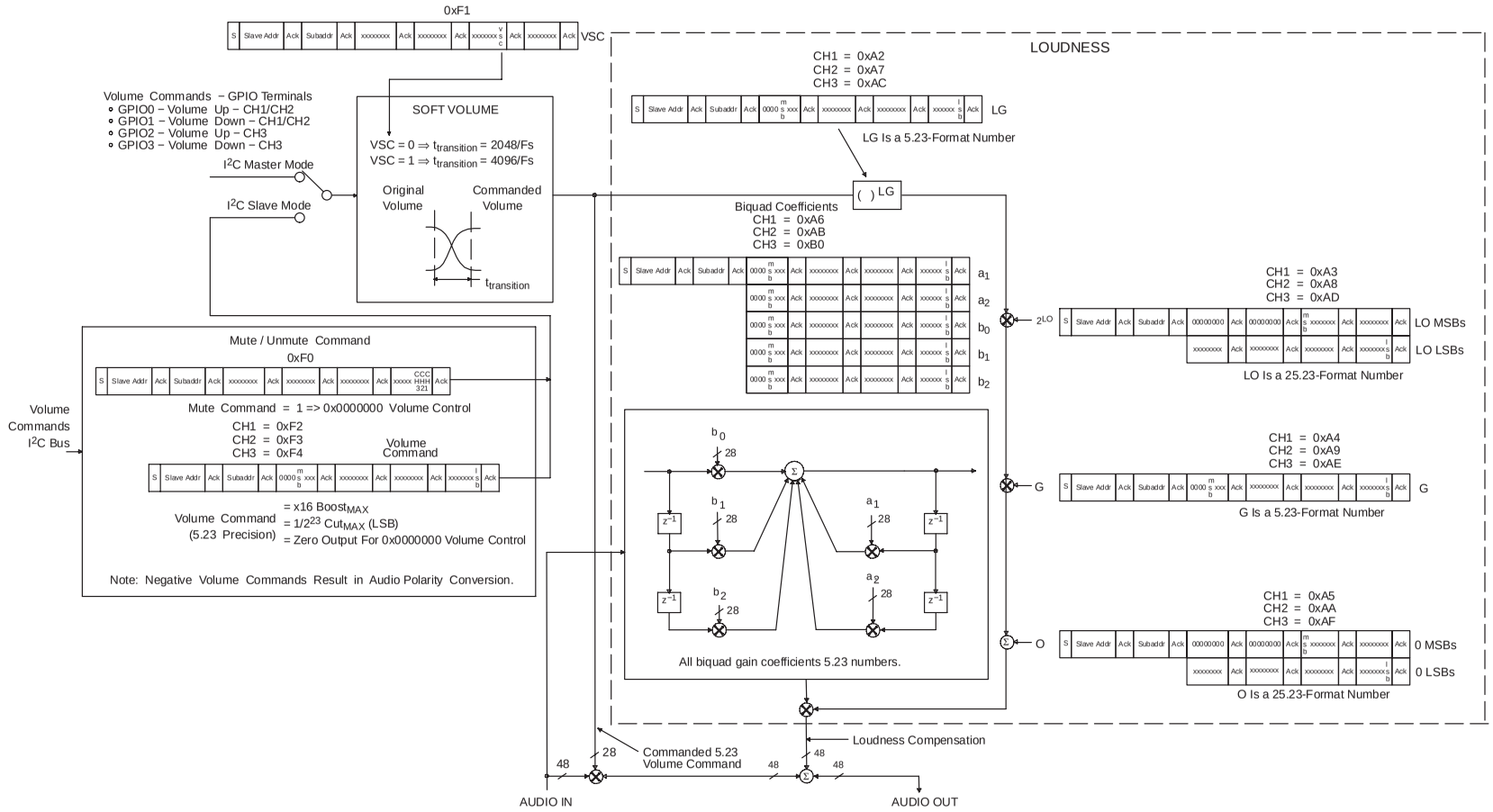


Figure 3–18. Detailed Block Diagram—Soft Volume and Loudness Compensation

If G is set to 0 and O is set to 0, loudness compensation is disabled. If G is set to 0 and O is set to 1, the biquad-filtered audio is directly added to the volume-level-adjusted audio. Typically, LG and LO are used to derive the desired loudness compensation function, G is used to turn loudness compensation on and off, and O is used to enable and disable the biquad filter output when automatic volume tracking is turned off.

3.6.3 Time Alignment and Reverb Delay Processing

The TAS3103A provides delay line facilities at two locations in the TAS3103A—in the 3D effects block (reverb delay), and at the output mix block (delay). There are three reverb delay blocks, one for each monaural channel, and these delay elements are typically used in implementing sound spatiality, a single-mix reverb, and other sound effects. There are also three delay blocks, again one for each monaural channel, and these delay elements are typically used for temporal channel alignment. The delay line facilities are implemented using a single 4K (4096) $\times$ 16-bit RAM resource. Each delay element implemented provides a one-sample delay (1/Fs). The size of each delay line can be programmed via the I^2C bus, or set by the EEPROM download in the I^2C master mode. The only restriction is that the total delay line resources programmed cannot exceed the capacity of the 4K $\times$ 16-bit memory bank.

Figure 3−19 illustrates how delay line structures are established within the 4K RAM memory. As seen in Figure 3−19, each delay line immediately begins where the previously implemented delay line leaves off. The actual placement of the pointers is computed by the resident microprocessor; the user needs only to enter the delay value required. However, consider the following four points when programming the lengths of the delay lines.

1. Each delay line of length L requires L+1 memory sample spaces.
2. Reverb delay lines require three memory words (48 bits) to implement a single delay element, as the reverb delay line operates in the 48-bit word structure of the digital audio processor (DAP).
3. Delay lines require two memory words (32 bits) to implement a single delay element, as the delay lines operate on the mixer outputs after 32-bit truncation has been applied.
4. There are five words of reserved memory space that must be preserved.

In Figure 3−19, the terms P_{CHx} refer to delay-line size assignments for the delay lines. The terms P_{Rx} see the delay line size assignments for the reverb delay lines. For the example shown in Figure 3−19, the delay for channel 3 is set to 0 and the reverb delay for channel 2 is set to 0. From these zero-valued settings, it is seen that a delay of 0 requires the use of one delay element. For the case of a delay of 0, the write transaction into the single delay element takes place before the read from the single delay element, thereby achieving a net delay of 0.

In making the delay-line length assignments, the only restriction is that the 4K memory resource not be exceeded. Figure 3−20 illustrates the computations required to determine the maximum delay-line length obtainable for five cases:

1. One reverb delay line
2. One delay line
3. Three equal-length reverb delay lines but no delay lines
4. Three equal-length delay lines but no reverb delay lines
5. Three equal-length delay lines and three equal-length reverb delay lines

All input crossbar mixers use signed 5.23-format mixer gain coefficients, and all are programmable via the I^2C bus. The 5.23 format provides a range of gain adjustment from 2^{-23} (−138 dB) to $2^4 - 1$ (23.5 dB).

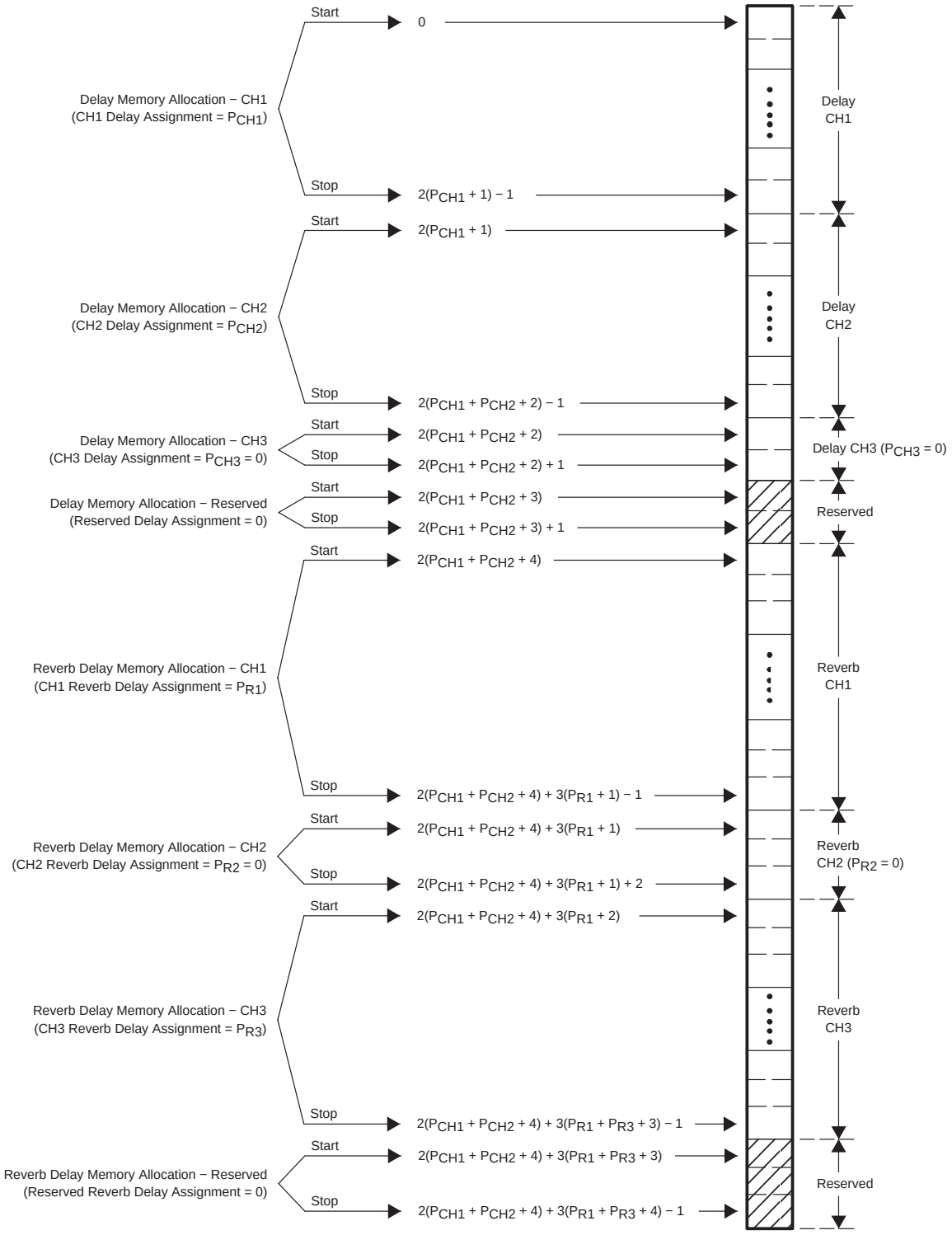
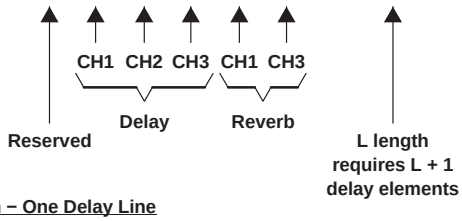


Figure 3−19. Delay-Line Memory Implementation

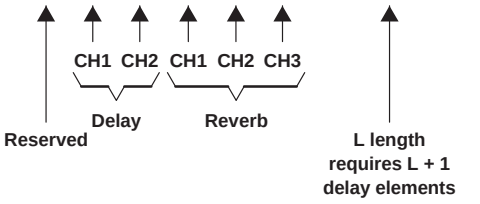
CASE 1: Maximum Length − One Reverb Delay Line

$$P_{Reverb\_max\_CH2} = [(4096 - 5 - 2 - 2 - 2 - 3 - 3) \div 3] - 1 = 1358\frac{2}{3} \rightarrow 1358$$



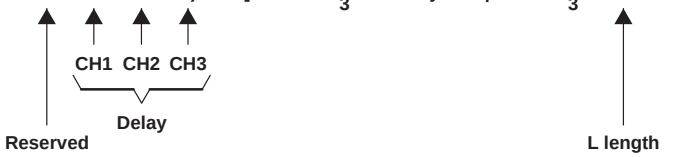
CASE 2: Maximum Length − One Delay Line

$$P_{Delay\_max\_CH3} = [(4096 - 5 - 2 - 2 - 3 - 3 - 3) \div 2] - 1 = 2038$$



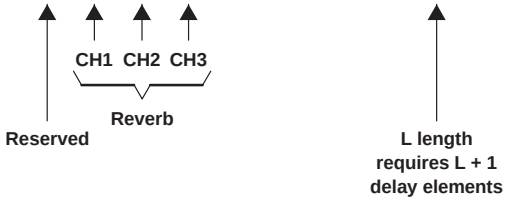
CASE 3: Maximum Length − Three Equal-Length Reverb Delay Lines

$$P_{Reverb\_max\_CH1,\ CH2,\ CH3} = |\{[(4096 - 5 - 2 - 2 - 2) \div 3] = 1361\frac{2}{3} \rightarrow 1361\} \div 3| = 453\frac{2}{3} - 1 = 452\frac{2}{3} \rightarrow 452$$



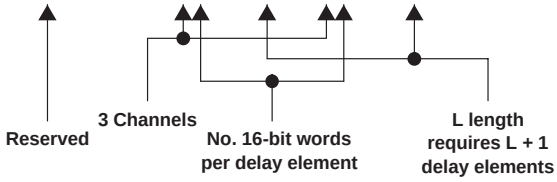
CASE 4: Maximum Length − Three Equal-Length Delay Lines

$$P_{Delay\_max\_CH1,\ CH2,\ CH3} = |\{[(4096 - 5 - 3 - 3 - 3) \div 2] = 2041\} \div 3| - 1 = 679\frac{1}{3} \rightarrow 679$$



CASE 5: Maximum Length − Three Equal-Length Delay Lines and Three Equal-Length Reverb Delay Lines

$$P_{Delay/Reverb\_max\_CH1,\ CH2,\ CH3} = (4096 - 5) = 4091 = 3[2(D + 1)] + 3[3(R + 1)]$$ Where D and R = No. delay and reverb elements/Channel



R takes $\frac{3}{2}$ the memory D does, so there should be three D elements for every two R elements.

Therefore, $D = \frac{3}{2}R$

$4091 = 3[2(\frac{3}{2}R + 1)] + 3[3(R + 1)] \rightarrow R = 226\frac{4}{9} \rightarrow \boxed{226}$

$D = \frac{3}{2}R \rightarrow D = \boxed{339}$

Figure 3−20. Maximum Delay-Line Lengths

Commands to reconfigure the reverb delay and delay lines should not be issued as stand-alone commands. When new delay assignments are issued, the content of the 4K memory resource used to implement the delay lines is not flushed. It takes a finite time for the memory to refill with samples in correspondence with its new assignments, and until this time has elapsed, audio samples can be output on the wrong channel. For this reason, it is recommended that all delay line assignment commands be preceded by a mute command, and followed by an unmute command.

CAUTION: No error flags are issued if the delay line assignments exceed the capacity of the 4K memory resource, but undefined and erratic behavior results if the delay line capacity is exceeded.

3.7 Dynamic Range Control (DRC)

DRC provides both compression and expansion capabilities over three separate and definable regions of audio signal levels. Programmable threshold levels set the boundaries of the three regions. Within each of the three regions, a distinct compression or expansion transfer function can be established and the slope of each transfer function is determined by programmable parameters. The offset (boost or cut) at the two boundaries defining the three regions can also be set by programmable offset coefficients. The DRC implements the composite transfer function by computing a 5.23-format gain coefficient from each sample output from the rms estimator. This gain coefficient is then applied to a mixer element, whose other input is the audio data stream. The mixer output is the DRC-adjusted audio data.

Two distinct DRC blocks are in the TAS3103A. One DRC services two monaural channels—CH1 and CH2. This DRC computes rms estimates of the audio data streams on both CH1 and CH2. The two estimates are then compared on a sample-by-sample basis, and the larger of the two is used to compute the compression/expansion gain coefficient. The gain coefficient is then applied to both CH1 and CH2 audio. The other DRC services only monaural channel CH3. This DRC also computes an rms estimate of the signal level on CH3, and this estimate is used to compute the compression/expansion gain coefficient applied to CH3 audio.

Figure 3−21 shows the positioning of the DRC block in the TAS3103A processing flow. As seen, the DRC input can come from either before or after soft volume control and loudness processing, or can be a weighted combination of both. The mixers feeding the DRC control the selection of which audio data stream or combination thereof, is input into the DRC. The mixers also provide a means of gaining or attenuating the signal level into the DRC. If the DRC setup is referenced to the 0-dB level at the TAS3103A input, the coefficient values for these mixers must be taken into account. Discussions and examples that follow further explore the role the mixers play in setting up the transfer function of the DRC.

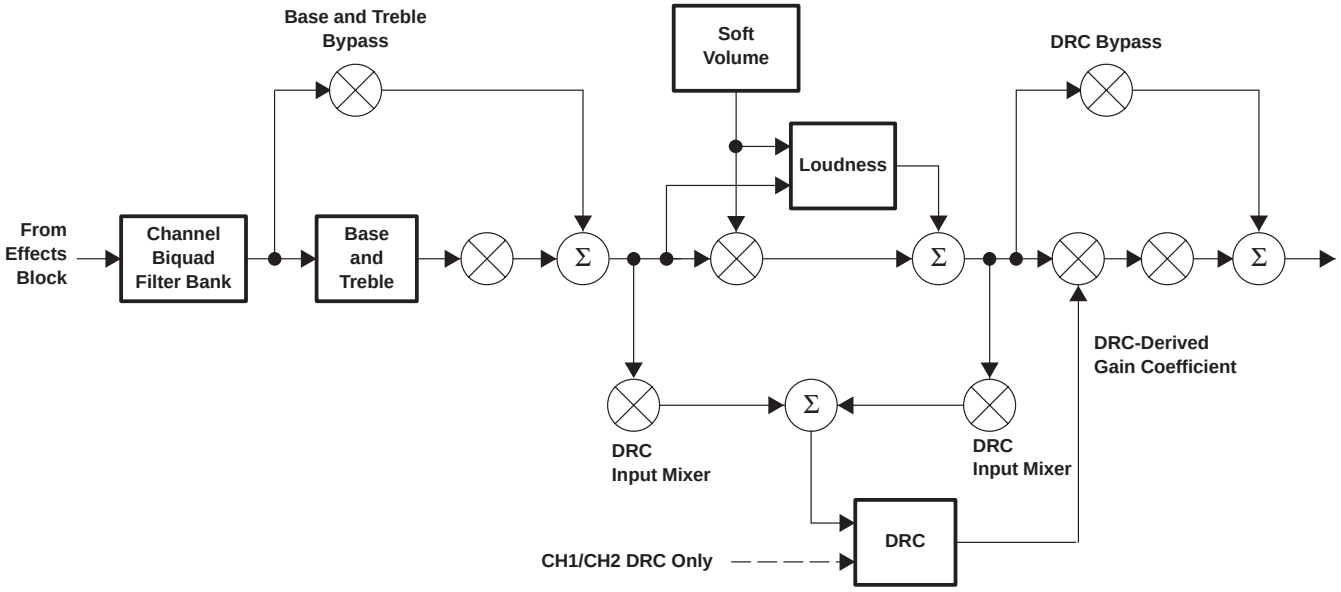


Figure 3−21. DRC Positioning in TAS3103A Processing Flow

Figure 3−22 illustrates a typical DRC transfer function.

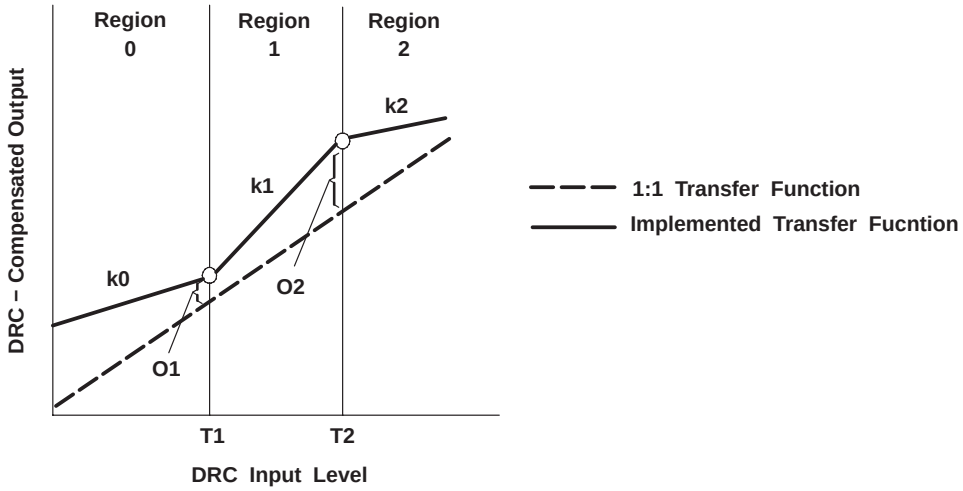


Figure 3−22. DRC Transfer Function Structure

The three regions shown in Figure 3−22 are defined by three sets of programmable coefficients:

- Thresholds T1 and T2—define region boundaries.
- Offsets O1 and O2—define the DRC gain coefficient settings at thresholds T1 and T2, respectively.
- Slopes k0, k1, and k2—define whether compression or expansion is to be performed within a given region. The magnitudes of the slopes define the degree of compression or expansion to be performed.

The three sets of parameters are all defined in logarithmic space and adhere to the following rules:

- The maximum input sample into the DRC is referenced at 0 dB. All values below this maximum value then have negative values in logarithmic (dB) space.
- The samples input into the DRC are 32-bit words and consist of the upper 32 bits of the 48-bit word format used by the digital audio processor (DAP). The 48-bit DAP word is derived from the 32-bit serial data received at the serial audio receive port by adding 8 bits of headroom above the 32-bit word and 8 bits of computational precision below the 32-bit word. If the audio processing steps between the SAP input and the DRC input result in no accumulative boost or cut, the DRC operates on the 8 bits of headroom and the 24 MSBs of the audio sample. Under these conditions, a 0-dB (maximum value) audio sample (0x7FFFFFFF) is seen at the DRC input as a –48-dB sample (8 bits $\times$ −6.02 dB/bit = −48 dB).
- Thresholds T1 and T2 define, in dB, the boundaries of the three regions of the DRC, as referenced to the rms value of the data into the DRC. Zero-valued threshold settings reference the maximum-valued rms input into the DRC and negative-valued thresholds reference all other rms input levels. Positive-valued thresholds have no physical meaning and are not allowed. In addition, zero-valued threshold settings are not allowed.

Although the DRC input is limited to 32-bit words, the DRC itself operates using the 48-bit word format of the DAP. The 32-bit samples input into the DRC are placed in the upper 32 bits of this 48-bit word space. This means that the threshold settings must be programmed as 48-bit (25.23-format) numbers.

CAUTION: Zero-valued and positive-valued threshold settings are not allowed and cause unpredictable behavior if used.

- Offsets O1 and O2 define, in dB, the attenuation (cut) or gain (boost) applied by the DRC-derived gain coefficient at the threshold points T1 and T2, respectively. Positive offsets are defined as cuts, and thus boost or gain selections are negative numbers. Offsets must be programmed as 48-bit (25.23-format) numbers.

- Slopes k0, k1, and k2 define whether compression or expansion is to be performed within a given region, and the degree of compression or expansion to be applied. Slopes are programmed as 28-bit (5.23-format) numbers.

3.7.1 DRC Implementation

Figure 3−23 shows the three elements comprising the DRC: (1) an rms estimator, (2) a compression/expansion coefficient computation engine, and (3) an attack/decay controller.

- RMS estimator—This DRC element derives an estimate of the rms value of the audio data stream into the DRC. For the DRC block shared by CH1 and CH2, two estimates are computed—an estimate of the CH1 audio data stream into the DRC, and an estimate of the CH2 audio data stream into the DRC. The outputs of the two estimators are then compared, sample-by-sample, and the larger valued sample is forwarded to the compression/expansion coefficient computation engine.

  Two programmable parameters, ae and (1 – ae), set the effective time window over which the rms estimate is made. For the DRC block shared by CH1 and CH2, the programmable parameters apply to both rms estimators. The time window over which the rms estimation is computed can be determined by:

$$t_{window} = \frac{-1}{Fs\ \ell n(1 - ae)} \quad \text{(seconds)}$$

- Compression/expansion coefficient computation—This DRC element converts the output of the rms estimator to a logarithmic number, determines the region where the input resides, and then computes and outputs the appropriate coefficient to the attack/decay element. Seven programmable parameters—T1, T2, O1, O2, k0, k1, and k2—define the three compression/expansion regions implemented by this element.
- Attack/decay control—This DRC element controls the transition time of changes in the coefficient computed in the compression/expansion coefficient computation element. Four programmable parameters define the operation of this element. Parameters ad and (1 − ad) set the decay or release time constant to be used for volume boost (expansion). Parameters aa and (1 − aa) set the attack time constant to be used for volume cuts. The transition time constants can be determined by:

$$t_a = \frac{-1}{Fs\ \ell n(1 - aa)} \quad \text{(seconds)} \qquad t_d = \frac{-1}{Fs\ \ell n(1 - ad)} \quad \text{(seconds)}$$

- In the foregoing formula, aa is a time constant. Use aa/5 for time constant T;
  $aa = 5\left[1 - e^{-1/Fs\ t_a}\right]$

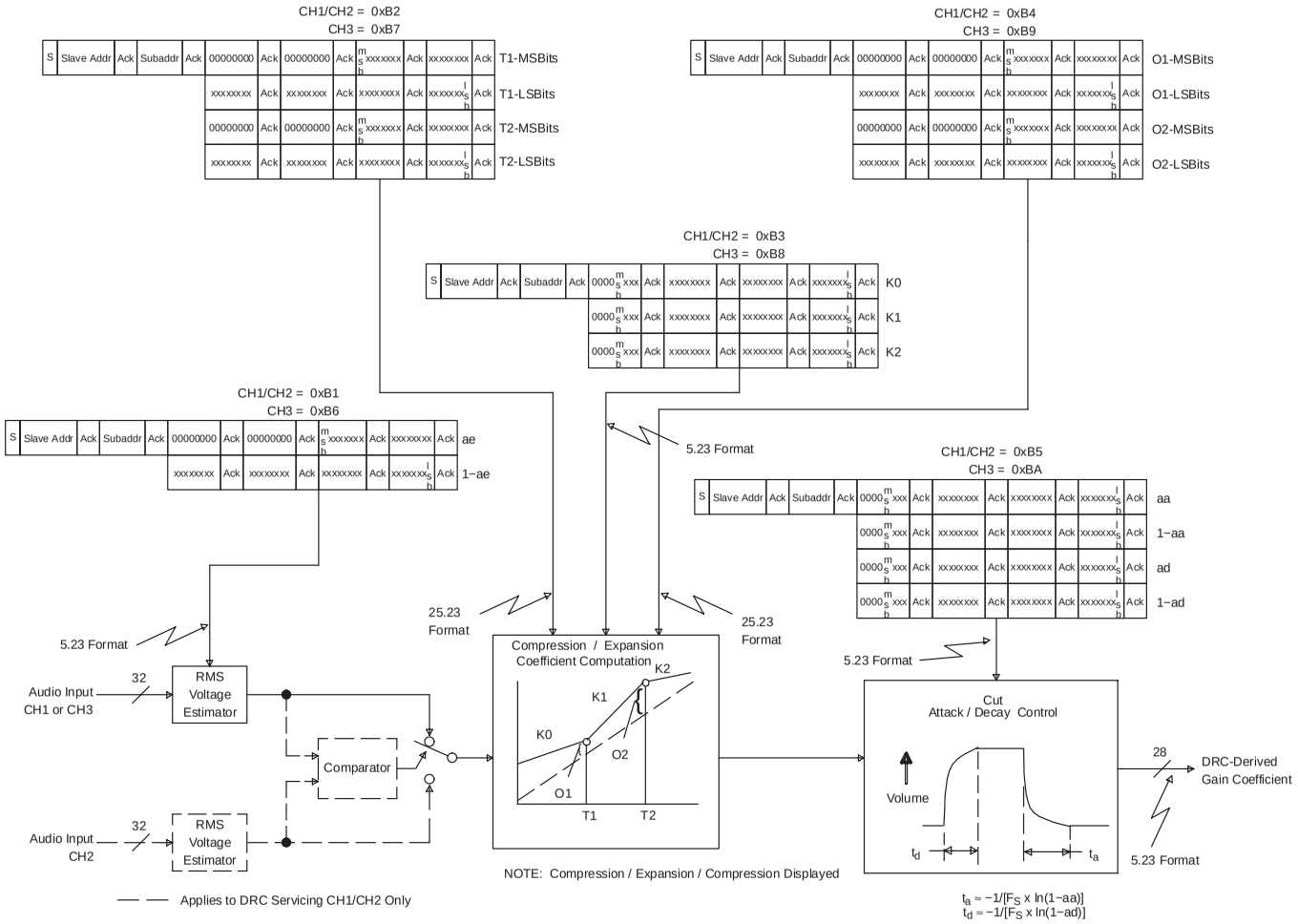


Figure 3−23. DRC Block Diagram

3.7.2 Compression/Expansion Coefficient Computation Engine Parameters

Seven programmable parameters are assigned to each DRC block: two threshold parameters, T1 and T2; two offset parameters, O1 and O2; and three slope parameters, k0, k1, and k2. The threshold parameters establish the three regions of the DRC transfer curve, the offsets anchor the transfer curve by establishing known gain settings at the threshold levels, and the slope parameters define whether a given region is a compression or an expansion region.

The audio input stream into the DRC must pass through DRC-dedicated programmable input mixers. These mixers are provided to scale the 32-bit input into the DRC to account for the positioning of the audio data in the 48-bit DAP word and the net gain or attenuation in signal level between the SAP input and the DRC. The selection of threshold values must take the gain (attenuation) of these mixers into account. The DRC implementation examples that follow illustrate the effect these mixers have on establishing the threshold settings.

T2 establishes the boundary between the high-volume region and the mid-volume region. T1 establishes the boundary between the mid-volume region and the low-volume region. Both thresholds are set in logarithmic space, and which region is active for any given rms estimator output sample is determined by the logarithmic value of the sample.

Threshold T2 serves as the fulcrum or pivot point in the DRC transfer function. O2 defines the boost (> 0 dB) or cut (< 0 dB) implemented by the DRC-derived gain coefficient for an rms input level of T2. If O2 = 0 dB, the value of the derived gain coefficient is 1 (0x0080 0000 in 5.23 format). k2 is the slope of the DRC transfer function for rms input levels above T2, and k1 is the slope of the DRC transfer function for rms input levels below T2 (and above T1). The labeling of T2 as the fulcrum stems from the fact that there cannot be a discontinuity in the transfer function at T2. However, the user can set the DRC parameters to realize a discontinuity in the transfer function at the boundary defined by T1. If no discontinuity is desired at T1, the value for the offset term O1 must obey the following equation.

$$O1_{\text{No Discontinuity}} = |T1 - T2| \times k1 + O2 \quad \text{For } (|T1| \geq |T2|)$$

T1 and T2 are the threshold settings in dB, k1 is the slope for region 1, and O2 is the offset in dB at T2. If the user chooses to select a value of O1 that does not obey the above equation, a discontinuity at T1 is realized.

Going down in volume from T2, the slope k1 remains in effect until the input level T1 is reached. If, at this input level, the offset of the transfer function curve from the 1:1 transfer curve does not equal O1, there is a discontinuity at this input level as the transfer function is snapped to the offset called for by O1. If no discontinuity is wanted, O1 and/or k1 must be adjusted so that the value of the transfer curve at the input level T1 is offset from the 1:1 transfer curve by the value O1. The examples that follow illustrate both continuous and discontinuous transfer curves at T1.

Going down in volume from T1, starting at the offset level O1, the slope k0 defines the compression/expansion activity in the lower region of the DRC transfer curve.

3.7.2.1 Threshold Parameter Computation

For thresholds,

$$T_{dB} = -6.0206T_{INPUT} = -6.0206T_{SUB\_ADDRESS\_ENTRY}$$

If, for example, it is desired to set T1 = –64 dB, then the subaddress entry required to set T1 to –64 dB is:

$$T1_{SUB\_ADDRESS\_ENTRY} = \frac{-64}{-6.0206} = 10.63$$

From Figure 3−23, it can be seen that T1 is entered as a 48-bit number in 25.23 format. Therefore:

T1 = 10.63 = 0_1010.1010_0001_0100_0111_1010_111
= 0x00000550A3D7 in 25.23 format

3.7.2.2 Offset Parameter Computation

The offsets set the boost or cut applied by the DRC-derived gain coefficient at the threshold point. An equivalent statement is that offsets represent the departure of the actual transfer function from a 1:1 transfer at the threshold point. Offsets are 25.23-formatted 48-bit logarithmic numbers. They are computed by the following equation.

$$O_{INPUT} = \frac{O_{DESIRED} + 24.0824\ dB}{6.0206}$$

Gains or boosts are represented as negative numbers; cuts or attenuation are represented as positive numbers. For example, to achieve a boost of 21 dB at threshold T1, the I^2C coefficient value entered for O1 must be:

$$O1_{INPUT} = \frac{-21\ dB + 24.0824\ dB}{6.0206} = 0.51197555$$

$$= 0.1000\_0011\_0001\_1101\_0100$$

$$= 0x00000041886A \text{ in } 25.23 \text{ format}$$

More examples of offset computations follow.

3.7.2.3 Slope Parameter Computation

In developing the equations used to determine the subaddress input value required to realize a given compression or expansion within a given region of the DRC, the following convention is adopted.

DRC Transfer = Input Increase : Output Increase

If the DRC realizes an output increase of n dB for every dB increase in the rms value of the audio into the DRC, a 1:n expansion is being performed. If the DRC realizes a 1-dB increase in output level for every n-dB increase in the rms value of the audio into the DRC, a n:1 compression is being performed.

For 1:n expansion, the slope k can be found by:

$k = n - 1$

For n:1 compression, the slope k can be found by: $k = \frac{1}{n} - 1$

In both expansion (1:n) and compression (n:1), n is implied to be greater than 1. Thus, for expansion:

$k = n - 1$ means $k > 0$ for $n > 1$. Likewise, for compression, $k = \frac{1}{n} - 1$ means $-1 < k < 0$ for $n > 1$. Thus, it appears that k must always lie in the range $k > -1$.

The DRC imposes no such restriction, and k can be programmed to values as negative as −15.999. To determine what results when such values of k are entered, it is first helpful to note that the compression and expansion equations for k are actually the same equation. For example, a 1:2 expansion is also a 0.5:1 compression.

$$0.5 \text{ Compression} \Rightarrow k = \frac{1}{0.5} - 1 = 1$$

$$1:2 \text{ Expansion} \Rightarrow k = 2 - 1 = 1$$

As can be seen, the same value for k is obtained either way. The ability to choose values of k less than −1 allows the DRC to implement negative-slope transfer curves within a given region. Negative-slope transfer curves are usually not associated with compression and expansion operations, but the definition of these operations can be expanded to include negative-slope transfer functions. For example, if k = −4

$$\text{Compression equation: } k = -4 = \frac{1}{n} - 1 \Rightarrow n = -\frac{1}{3} \Rightarrow -0.3333 : 1 \text{ compression}$$

$$\text{Expansion equation: } k = -4 = n - 1 \Rightarrow n = -3 \Rightarrow 1 : -3 \text{ expansion}$$

With k = −4, the output decreases 3 dB for every 1-dB increase in the rms value of the audio into the DRC. As the input increases in volume, the output decreases in volume.

3.7.3 DRC Compression/Expansion Implementation Examples

The following four examples illustrate the steps that must be taken to calculate the DRC compression/expansion coefficients for a specified DRC transfer function. The first example is an expansion/compression/expansion implementation without discontinuities in the transfer function and represents a typical application. This first example also illustrates one of the three modes of DRC saturation—32-bit dynamic range limitation saturation. The second example is a compression/expansion/compression implementation. There is no discontinuity at T1, and 32-bit dynamic range saturation occurs at low volume levels into the DRC. Example 2 also illustrates another form of DRC saturation—maximum gain saturation. Example 3 illustrates the concept of infinite compression. Also, in Example 3, 32-bit dynamic range saturation occurs at low volume levels and the third form of DRC saturation is illustrated—minimum gain saturation. Example 4 illustrates the ability of the DRC to realize a negative slope transfer function. This example also illustrates two of the three forms of saturation—32-bit dynamic range saturation at low volume levels and minimum gain saturation.

CAUTION:

The examples presented all exhibit some form of DRC saturation. This is not intended to imply that all (or most) DRC transfer implementations exhibit some form of saturation. Most practical implementations do not exhibit saturation. The examples are chosen to explain by example the three types of saturation that can be encountered. But the phenomenon of saturation can also be used to advantage in that it effectively provides a means to implement more than three zones or regions of operation. If saturation is intended, the regions exhibiting the transfer characteristic set by k0, k1, and k2 provide three regions, and the regions exhibiting saturation provide the additional regions of operation.

3.7.3.1 Example 1—Expansion/Compression/Expansion Transfer Function With 32-Bit Dynamic Range Saturation

For this example, the following transfer characteristics are chosen.

- Threshold point 2: T2 = −26 dB, O2 = 30 dB
- Threshold point 1: T1 = −101 dB, O1 = −7.5 dB
- Region 0 slope: k0 = 0.05 ≥ 1:1.05 Expansion
- Region 1 slope: k1 = −0.5 ≥ 2:1 Compression
- Region 2 slope: k2 = 0.1 ≥ 1:1.1 Expansion

The thresholds T1 and T2 are typically referenced, by the user, to the 0-dB signal level into the TAS3103A. But to determine the equivalent threshold point at the DRC input, it is necessary to take into account the processing gain (or loss) between the TAS3103A SAP input and the DRC. As an example, consider the processing gain structure shown in Figure 3−24. Inputting the data below the 8-bit headroom in the 48-bit DAP word and then routing only the upper 32 bits of the 48-bit word into the DRC, results in a 48-dB (8 bits x 6 dB/bit = 48 dB) attenuation of the signal level into the DRC. Channel processing gain and use of the dedicated mixer into the DRC can revise this apparent 48-dB attenuation in signal level into the DRC. In Figure 3−24, the 2^4 mixer gain into the DRC, coupled with a net channel gain of 0 dB, changes the net 48-dB attenuation of the signal level into the DRC to a net attenuation of 24 dB.

For slopes:

Region 0 = 1:1.05 Expansion	≥ k0 = 1.05 − 1 = 0.05 = 0 0000.0000 1100 1100 1100 1100 110 = 0x0066 666 in 5.23 format
Region 1 = 2:1 Compression	≥ k1 = 1/2 − 1 = -0.5 = 1 1111.1000 0000 0000 0000 0000 000 = 0xFC00 000 in 5.23 format
Region 2 = 1:1.1 Expansion	≥ k2 = 1.1 − 1 = 0.1 = 0 0000.0001 1001 1001 1001 1001 100 = 0x00CC CCC in 5.23 format

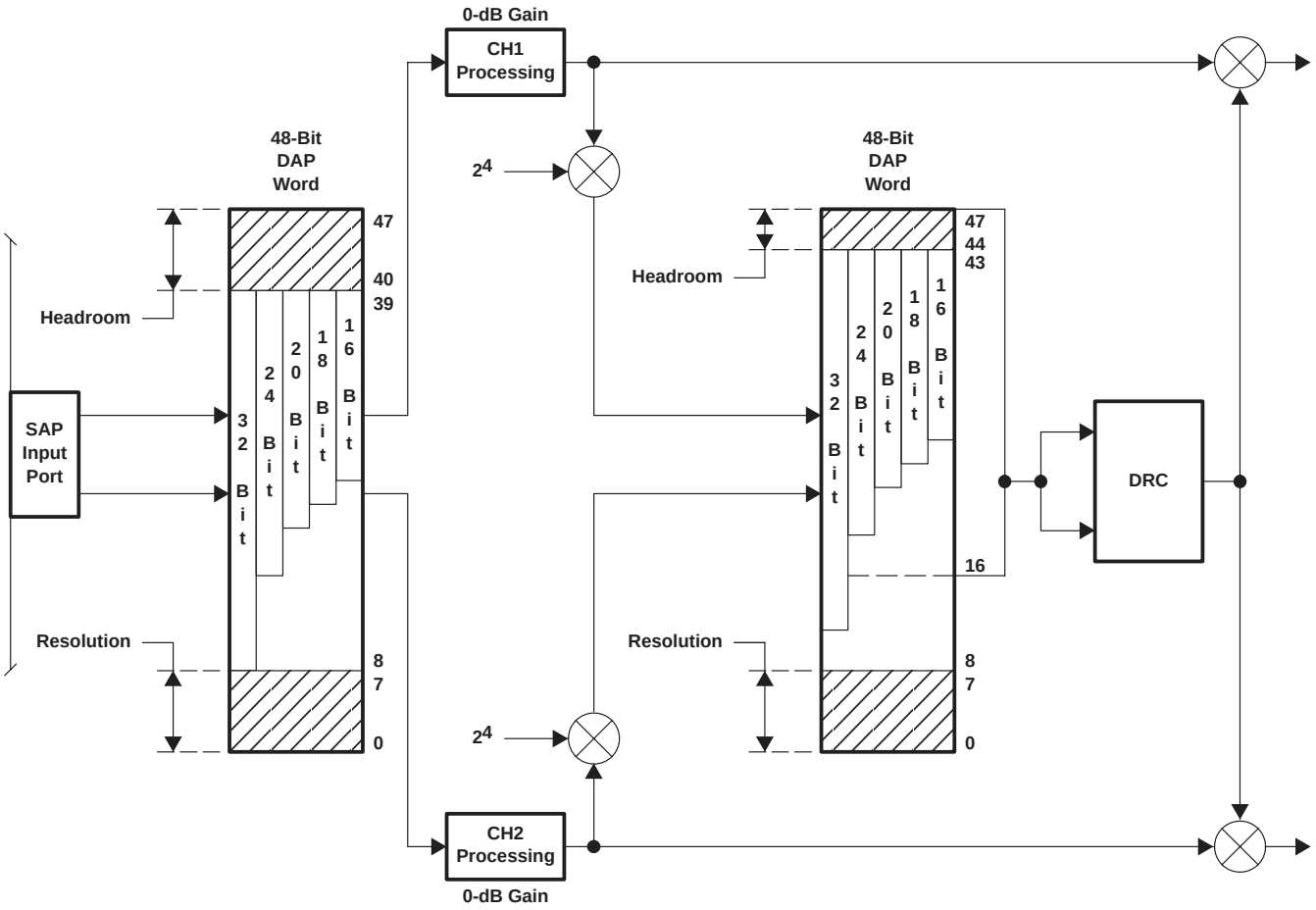


Figure 3−24. DRC Input Word Structure for 0-dB Channel Processing Gain

The resulting DRC transfer function for the above parameters is shown in Figure 3−25. The threshold T2 is set at the DRC rms input level of −50 dB, which corresponds to a −26-dB rms signal level at the SAP input. The DRC-compensated output at T2 is cut 30 dB with respect to the 1:1 transfer function (O2 = 30 dB). The threshold T1 is set at the DRC rms input level of −125 dB, which corresponds to a −101-dB rms signal level at the SAP input. The DRC-compensated output at T1 is boosted by 7.5 dB with respect to the 1:1 transfer function (O1 = −7.5 dB).

For thresholds, $T_{dB} = -6.0206T_{INPUT} = -6.0206T_{SUB\_ADDRESS\_ENTRY}$.

Therefore,

$T2 = -26\text{ dB} - 24\text{ dB} = -50\text{ dB} \geq T2_{INPUT}$ = −50/−6.0206 = 8.30482
= 0 1000.0100 1110 0000 1000 1010 111
= 0x0000 0427 0457 in 25.23 format

$T1 = -101\text{ dB} - 24\text{ dB} = -125\text{ dB} \geq T1_{INPUT}$ = −125/−6.0206 = 20.76205
= 1 0100.1100 0011 0001 0101 1011 010
= 0x0000 0A61 8ADA in 25.23 format

For offsets, $O_{INPUT} = \frac{1}{6.0206}\left[O_{dB} + 24.0824\right]$.

Therefore, $O2_{INPUT} = \frac{1}{6.0206}[30 + 24.0824] = 8.982892$
= 0 1000.1111 1011 1001 1110 1100 111
= 0x0000 047D CF67 in 25.23 format

$$O1_{INPUT} = \frac{1}{6.0206}[-7.5 + 24.0824] = 2.754277$$
$$= 0\ 0010.1100\ 0001\ 0001\ 1000\ 0100\ 110$$
$$= 0x0000\ 0160\ 8C26 \text{ in } 25.23 \text{ format}$$

For input levels above the T2 threshold, the transfer function exhibits a 1:1.1 expansion. For input levels below T2, the transfer function exhibits a 2:1 compression. Also, by definition, it is seen that there is no discontinuity in the transfer function at T2. When the 2:1 compression curve in region 1 intersects the T1 threshold level, the output level is 7.5 dB above the 1:1 transfer, an offset value identical to O1. Thus, there is no discontinuity at T1. For input levels below T1, the transfer function exhibits a 1:1.05 expansion.

DRC rms input levels below −192 dB fall below the 32-bit precision of the DRC input (32 bits × −6 dB/bit = −192 dB). This means that for levels below −192 dB, the DRC sees a constant input level of 0, and thus the computed DRC gain coefficient remains fixed at the value computed when the input was at −192 dB. The transfer function then has a 1:1 slope below the –192-dB input level and is offset from the 1:1 transfer curve by the offset present at the –192-dB input level.

The change from a 1:1.05 expansion to a 1:1 transfer below −192 dB is the result of 32-bit dynamic range saturation at the DRC input. This type of saturation always occurs at a DRC input level of −192 dB. However, the input level at which this type of saturation occurs depends on the channel gain. For this example, the saturation occurs at an input level of −168 dB (−192-dB DRC input + 48 dB 8-bit headroom –24-dB mixer gain into DRC).

3.7.3.2 Example 2—Compression/Expansion/Compression Transfer Function With Maximum Gain Saturation and 32-Bit Dynamic Range Saturation

The transfer function parameters for this example are given in Table 3−5. In setting the threshold levels it is assumed that the net processing gain between the SAP input and the DRC is 0 dB. This is the same as Example 1 except that the gain of the mixer into the DRC is set to 1 instead of 2^4. Because of the 8-bit headroom in the 48-bit DAP word, the upper eight bits of the 32-bit DRC input word are zero, resulting in 0-dB signal levels at the SAP input being seen as –48-dB signal levels at the DRC.

Figure 3−25 shows the DRC transfer function resulting from the parameters given in Table 3−5. At threshold level T2 (−70 dB), O2 specifies a boost of 30 dB. But the signed, 5.23-formatted gain coefficient only provides a 24-dB boost capability (5 integer bits = Sxxxx ≥ 2^4 × 6 dB/octave = 24 dB). Internally, the DRC operates in 48-bit space and thus computes a 30-dB boost. But the 5.23-formatted gain coefficient saturates or clips at 24 dB. The transfer curve thus resides 24 dB above the 1:1 transfer curve at T2.

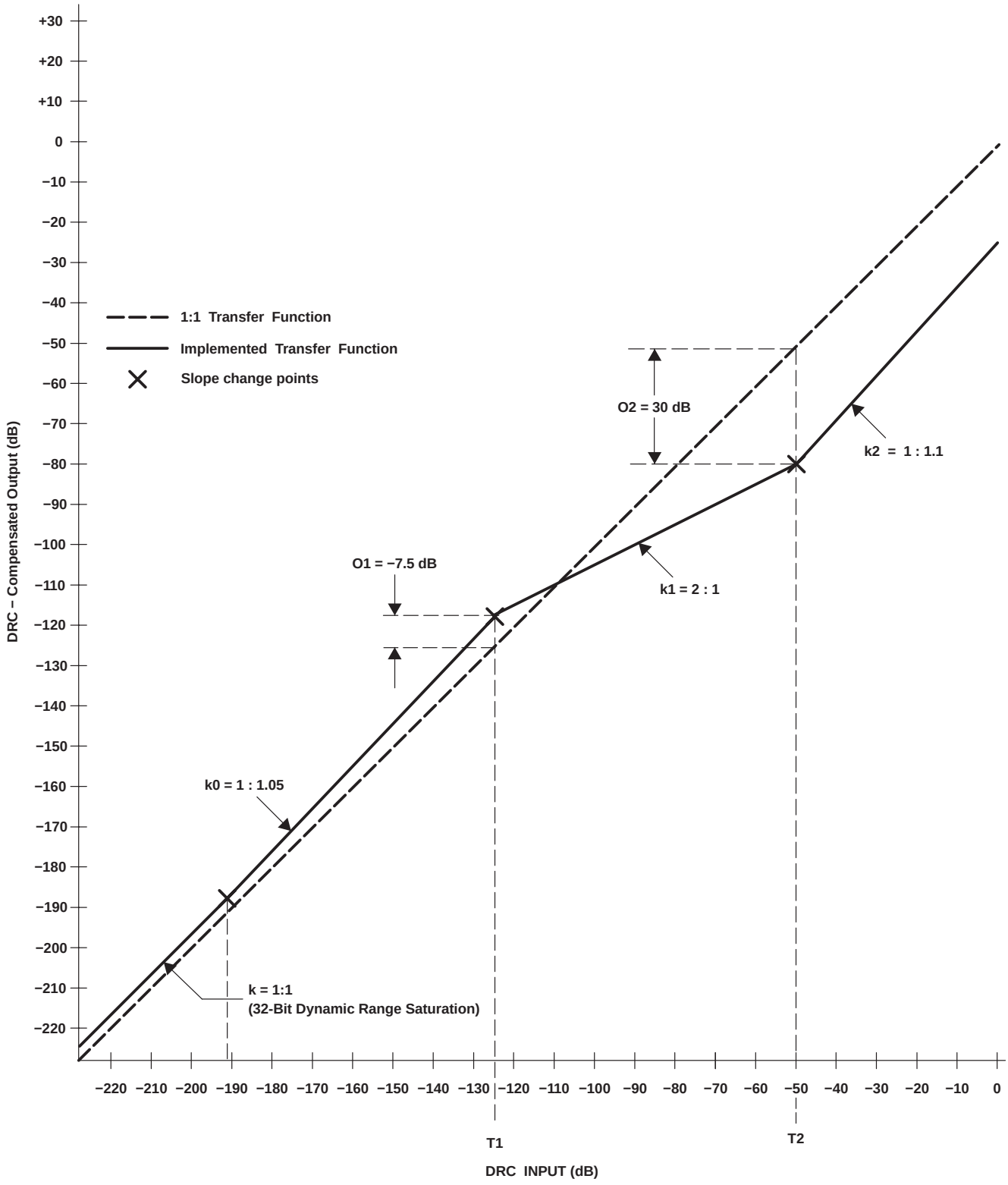


Figure 3−25. DRC Transfer Curve—Example 1

The transfer curve remains a constant 24 dB above the 1:1 transfer curve for input levels above and below T2 until the computed DRC gain coefficient falls within the dynamic range of a 5.23-format number. For input levels above T2, k2 implements a 5:1 compression. At an input level 7.5 dB above T2 (−62.5 dB), the DRC transfer curve has risen 7.5/5 = 1.5 dB. The boost at this point is 30 dB – (7.5 dB – 1.5 dB) = 24 dB. The DRC has come out of gain saturation. For input levels above −62.5 dB, the transfer curve exhibits 5:1 compression.

Table 3−5. DRC Example 2 Parameters

DRC PARAMETER	REQUIRED (SPECIFIED) VALUE (NET $GAIN_{SAP\ Input\text{-}DRC}$ = 0 dB)	I^2C COEFFICIENT VALUE
T1 and T2	$-148.7\ dB_{Input} \geq -172.7\ dB_{DRC}$	−172.7/−6.0206 = 28.684849 = 0x0000 0E57 $A91F_{25.23\ format}$
O2	−20 dB	(−20 + 24.0824)/6.0206 = 0.678072 = 0x0000 0056 $CB0F_{25.23\ format}$
O1	10 dB	(10 + 24.0824)/6.0206 = 5.660964 = 0x0000 02D4 $9A78_{25.23\ format}$
k2	∞:1 compression	$(1/\infty) - 1 = -1 = 0xF800\ 000_{5.23\ format}$
k1	1:1 transfer	$(1/1) - 1 = 0 = 0x0000\ 000_{5.23\ format}$
k0	2:1 compression	$(1/2) - 1 = -0.5 = 0xFC00\ 000_{5.23\ format}$

For input levels below T2, k1 implements a 1:2 expansion. With a 1:2 expansion in effect, the transfer curve has dropped 12 dB at an input level 6 dB below T2. The boost at this level is 30 dB − (12 dB − 6 dB) = 24 dB. The DRC gain coefficient has again come out of saturation. For input levels below −76 dB and above −150 dB, the transfer curve exhibits a 1:2 expansion.

At T1 (–150 dB), the transfer curve is 50 dB below the 1:1 transfer curve. Because O1 = 50 dB, there is no discontinuity in the transfer function. For inputs below −150 dB, k0 implements a 2:1 compression. The change from a 2:1 compression to a 1:1 transfer at −192 dB is due to 32-bit dynamic range saturation at the DRC input.

3.7.3.3 Example 3—1:1 Transfer/Infinite Compression With Minimum Gain Saturation, 32-Bit Dynamic Range Saturation, and Equal Threshold Settings (T1 = T2)

The DRC transfer function parameters for this example are given in Table 3−6. In addition to illustrating minimum gain saturation, this example also illustrates the operation of the DRC when T1 and T2 are set equal.

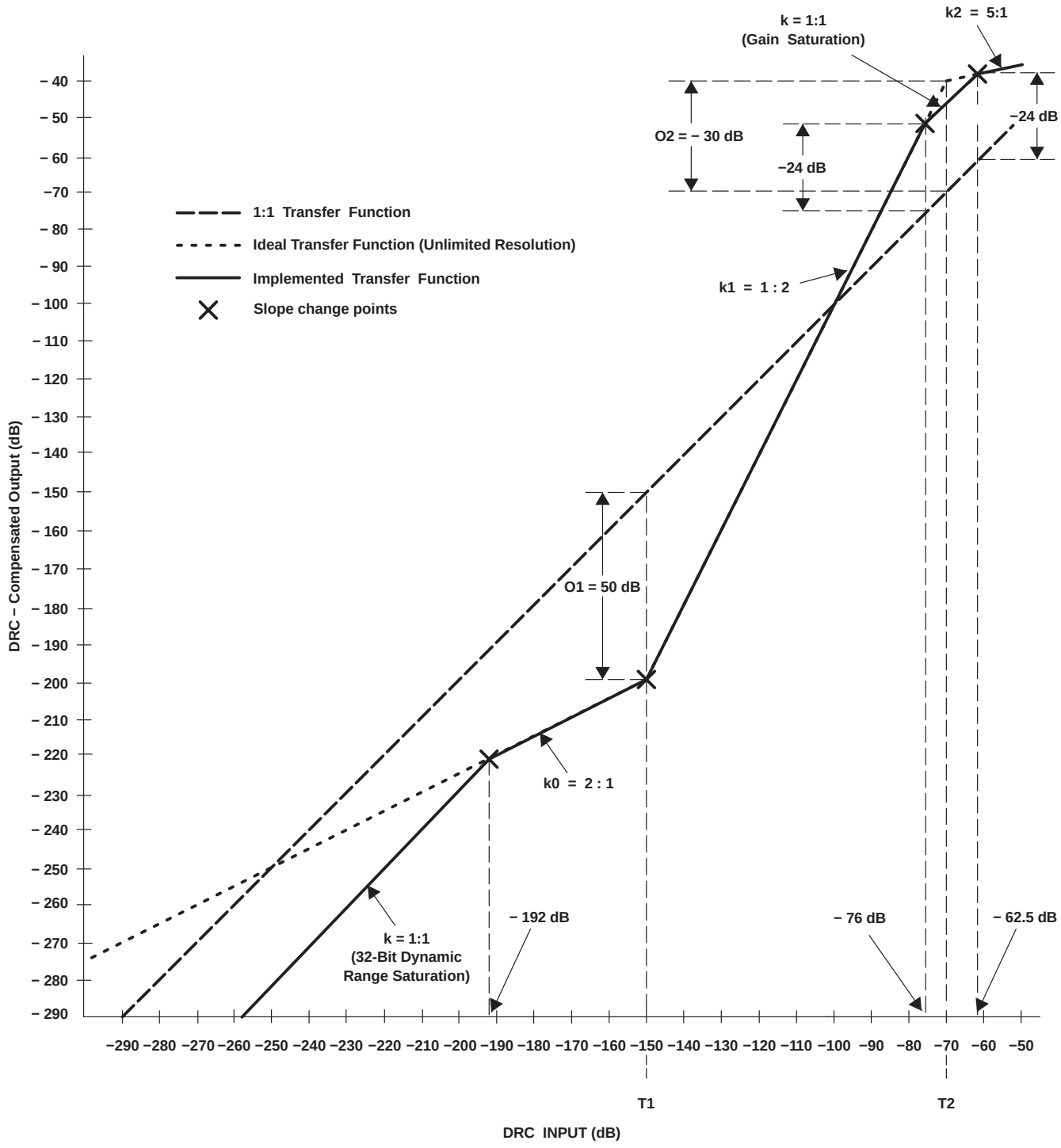


Figure 3−26. DRC Transfer Curve—Example 2

When T1 and T2 are set equal, the following questions arise:

- If O1 ≠ O2, what roles do O1 and O2 have?
- Which slope parameter, k0 or k1, has control of the transfer function for input levels below the common threshold point?
- Does k2 control the transfer function for inputs above the common threshold point?

This example addresses and answers those questions.

Table 3−6. DRC Example 3 Parameters

DRC PARAMETER	REQUIRED (SPECIFIED) VALUE (NET $GAIN_{SAP\ Input\text{-}DRC}$ = 0 dB)	I^2C COEFFICIENT VALUE
T1 and T2	$-148.7\ dB_{Input} \geq -172.7\ dB_{DRC}$	−172.7/−6.0206 = 28.684849 = 0x0000 0E57 A91F$_{25.23\ format}$
O2	−20 dB	(−20 + 24.0824)/6.0206 = 0.678072 = 0x0000 0056 CB0F$_{25.23\ format}$
O1	10 dB	(10 + 24.0824)/6.0206 = 5.660964 = 0x0000 02D4 9A78$_{25.23\ format}$
k2	∞:1 compression	(1/∞) − 1 = −1 = 0xF800 000$_{5.23\ format}$
k1	1:1 transfer	(1/1) − 1 = 0 = 0x0000 000$_{5.23\ format}$
k0	2:1 compression	(1/2) − 1 = −0.5 = 0xFC00 000$_{5.23\ format}$

For this example it is assumed that a net processing gain of 2^4 (24 dB) is realized from the SAP input and the DRC (which is identical to the net processing gain assumed for Example 1). The 2^4 gain results in reducing the 8-bit headroom in the 48-bit DAP word to a headroom of four bits. The 32-bit data into the DRC then resides in bits 27:0, which means that the data level into the DRC is down 24 dB with respect to the input level at the SAP. Data input into the TAS3103A (SAP) at a level of −148.7 dB is seen as a −148.7 dB − 24 dB = −172.7-dB signal at the DRC. T1 and T2 must be set to −172.7 dB to realize a common threshold point at an incoming signal level of −148.7 dB.

Figure 3−27 shows the transfer function resulting from entering the I^2C coefficient values given in Table 3−6. At the T1/T2 threshold, a discontinuity of 30 dB is observed. For inputs above the threshold, the transfer curve is horizontal (infinite compression), and the horizontal line starts 20 dB above the 1:1 transfer curve at the threshold point. Thus, for cases when T1 = T2, O2 governs the offset with regard to the starting point of the transfer curve above the common threshold point and k2 determines the slope of the transfer curve. For inputs below the common threshold point, the transfer curve exhibits a 2:1 compression and starts 10 dB below the 1:1 transfer curve. Thus, O1 sets the offset at the threshold point for the transfer curve at and below the common threshold point and k0 determines the slope of this curve. Slope parameter k1 plays no role when T1 = T2. The value of 0 (1:1 transfer) used in this example for k1 could be changed to any value and the resulting transfer function would not be altered. The change from a 2:1 compression to a 1:1 transfer at −192 dB is due to 32-bit dynamic range saturation at the DRC input.

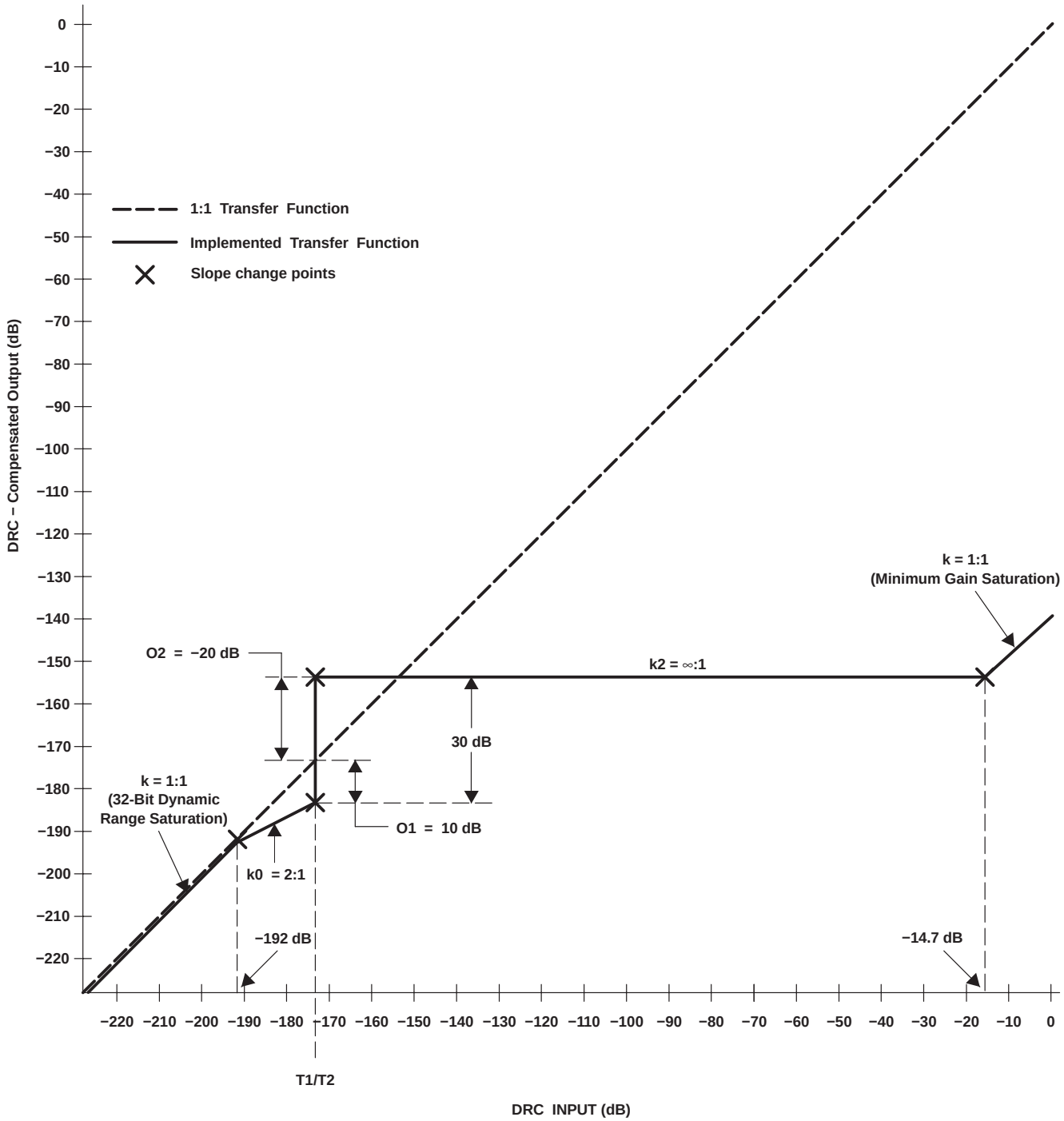


Figure 3−27. DRC Transfer Curve—Example 3

The horizontal slope of the transfer curve above the common threshold point does not remain horizontal indefinitely. At a point 158 dB above the common threshold point (−14.7-dB DRC input level), the transfer function has gone from a boost of 20 dB to a cut of 138 dB. A cut of 138 dB is the maximum cut possible for a 5.23-format gain coefficient ($2^{-23} \geq 23$ octaves $\times$ 6 dB/octave = 138 dB). Thus, at a DRC input level of −14.7 dB, minimum gain saturation has been reached. For inputs above this saturation point, the DRC-derived gain coefficient remains constant at the minimum gain value (2^{-23}), and the transfer function exhibits a 1:1 transfer slope.

3.7.3.4 Example 4—Expansion/Cut/Expansion With Gain Saturation and 32-Bit Dynamic Range Saturation

The three previous examples restricted the slope factor k to lie in the range $k \geq -1$. This example illustrates the transfer characteristic obtained using a value of k less than −1. For this example it is assumed that the net processing gain into the DRC is 0 dB. This means that the 8-bit headroom in the 48-bit DAP processing word structure does not contain data. Because the DRC receives the upper 32 bits of this 48-bit word, data at the DRC is down 48 dB (8 bits $\times$ 6 dB/bit = 48 dB) with respect to the signal level at the SAP input (TAS3103A input). The transfer function parameters for this example are given in Table 3−7. Figure 3−28 shows the transfer function resulting from entering the I^2C coefficient values given in Table 3−7.

At the threshold point T2 (−70 dB), the transfer function is 100 dB below the 1:1 transfer slope (O2 = 100 dB). For input levels above T2, the transfer function exhibits a 1:1.4 expansion. For input levels below T2, the transfer function exhibits a negative slope; for every dB the input decreases, the output increases by 1 dB. At an input level 62 dB below T2 (−132 dB), the transfer curve has risen 62 dB, for a net boost of 124 dB. The transfer curve at this input level is 24 dB above the 1:1 transfer curve. This boost value puts the DRC-derived gain coefficient into gain saturation. For input levels below −132 dB, the gain coefficient remains constant at maximum gain and the transfer function exhibits a 1:1 transfer slope, parallel to the 1:1 transfer curve but 24 dB above it.

At T1 (–150 dB), the transfer curve snaps back to the 1:1 transfer curve because O1 = 0 dB. The DRC gain coefficient is no longer in gain saturation and for inputs below −150 dB, the transfer function exhibits a 1:1.5 expansion. The change from a 1:1.5 expansion to a 1:1 transfer below −192 dB is the result of 32-bit dynamic range saturation.

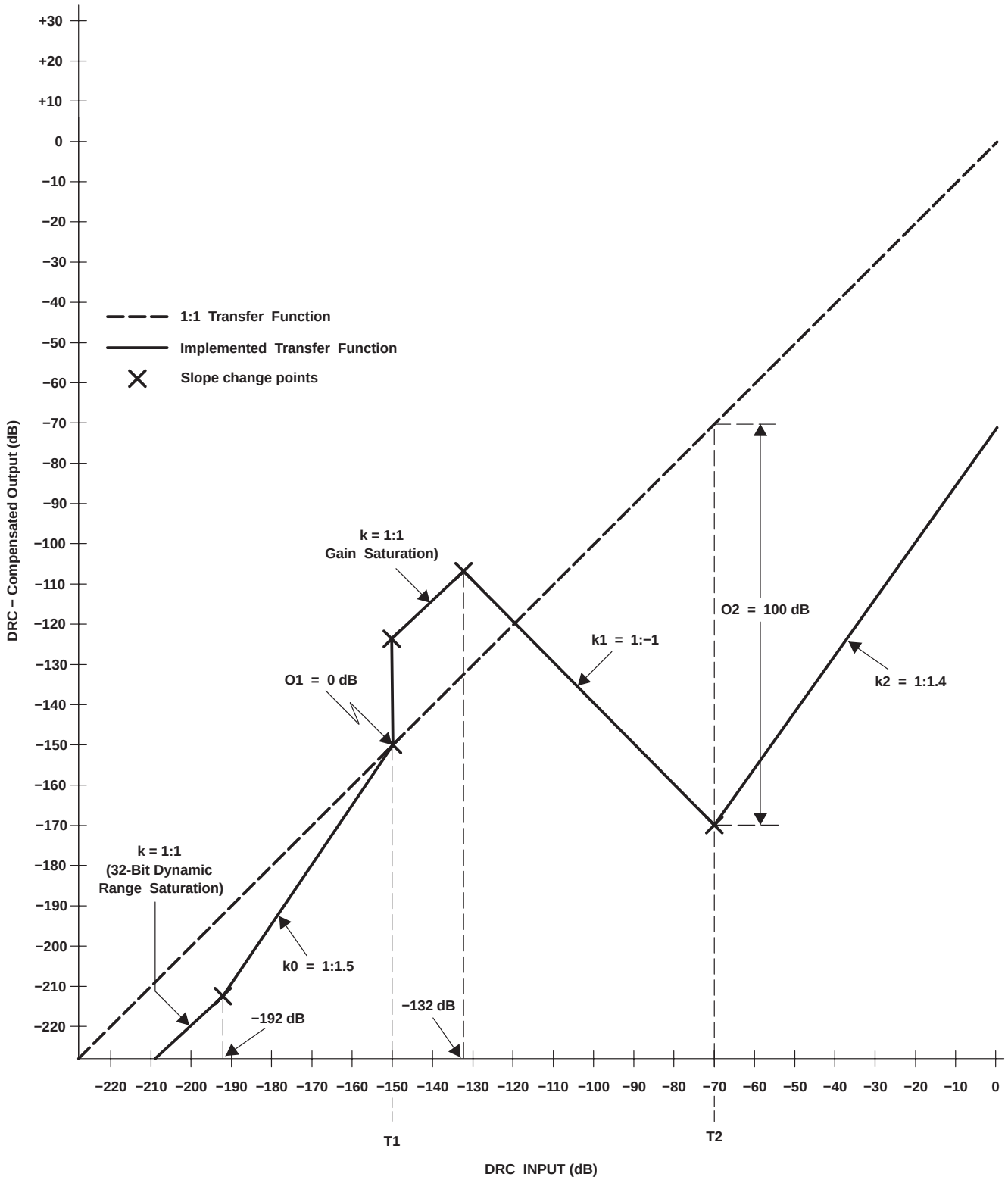


Figure 3−28. DRC Transfer Curve—Example 4

Table 3−7. DRC Example 4 Parameters

DRC PARAMETER	REQUIRED (SPECIFIED) VALUE (NET $GAIN_{SAP\ Input-DRC}$ = 0 dB)	I^2C COEFFICIENT VALUE
T2	$-22\ dB_{Input} \geq -70\ dB_{DRC}$	−70/−6.0206 = 11.626748 = $0x0000\ 05D0\ 3948_{25.23\ format}$
T1	$-102\ dB_{Input} \geq -150\ dB_{DRC}$	−150/−6.0206 = 24.91446 = $0x0000\ 0C75\ 0D09_{25.23\ format}$
O2	100 dB	(100 + 24.0824)/6.0206 = 20.609640 = $0x0000\ 0A4E\ 08B0_{25.23\ format}$
O1	0 dB	(0 + 24.0824)/6.0206 = 4.000000 = $0x0000\ 0200\ 0000_{25.23\ format}$
k2	1:1.4 expansion	1.4 − 1 = 0.4 = $0x0333\ 333_{5.23\ format}$
k1	1:−1 transfer	(1/−1) − 1 = −1 −1 = −2 = $0XF000000_{5.23\ format}$
k0	1:1.5 expansion	1.5 − 1 = 0.5 = $0x0400\ 000_{5.23\ format}$

3.8 Spectrum Analyzer/VU Meter

The TAS3103A contains an I^2C bus programmable function block that can serve as either a spectrum analyzer or a volume unit (VU) meter. Figure 3−29 shows the structure of this function block and lists the I^2C subaddress of the parameters that control it.

The block consists of 10 biquad filters, each followed by an rms estimator and a logarithmic converter. Two nodes provide input to the block, with each node servicing five of the 10 biquad filters. Audio from input node s can either come exclusively from channel 1, channel 2, channel 3, or from a gain-weighted combination of these channels. Audio from input node t can also come exclusively from either channel 1, channel 2, or channel 3, or from a gain-weighted combination of these channels. The spectrum analyzer then can be used to divide the audio frequency band into 10 frequency bins to examine the spectrum of the audio data stream on channel 1, channel 2, channel 3, or any combination of these channels. The spectrum analyzer can also be used to divide the audio frequency band into five frequency bins to examine the spectral content of two of the channels independently.

The VU meter is a special case of the spectrum analyzer that uses only the outputs from biquad 5 and biquad 6. Typically, for the VU meter, one channel would be routed to biquad 5 (node s) and a different channel would be routed to biquad 6 (node t). Each biquad filter would be assigned a band-pass transfer function that encompasses most of the audio band, or the filter could be configured as a pass-through device to see the full spectral band. The two outputs then would be a measure of the energy on the two channels. Other options for the VU meter are also available. For example, by properly setting the coefficients on biquad 5 and biquad 6, the concurrent measurement of bass and treble volume levels on a single channel could be made.

Mixer and summation elements preceding the two input nodes s and t provide a means of adjusting the spectrum analyzer and VU meter outputs relative to the incoming audio data stream. The spectrum analyzer and VU meter outputs are unsigned 5.3-format, base-2 logarithmic numbers. The integer part of the number designates the most-significant bit [in the 48-bit digital audio processor(DAP) word] occupied by the magnitude of the rms estimate of the biquad filter output. A value of 31 means the magnitude of the rms estimate occupies bit 47 of the 48-bit DAP word (bit 48 is the sign bit, and using the absolute value of the biquad filter output in determining the rms estimate makes this bit always 0 in value). A value of 30 means the magnitude of the rms estimate occupies bit 46 and this pattern continues with a value of 1 signifying the magnitude of the rms estimate occupies bit 17. A value of 0 signifies that the magnitude of the rms estimate is below bit 17. The fractional digits in the 5.3-formatted number are simply the three bits below the most-significant data bit. If the rms estimate lies below bit 16 of the 48-bit DAP word, the spectrum analyzer/VU meter output is 0.0. Figure 3−27 gives examples of logarithmic outputs for different 48-bit rms estimate values.

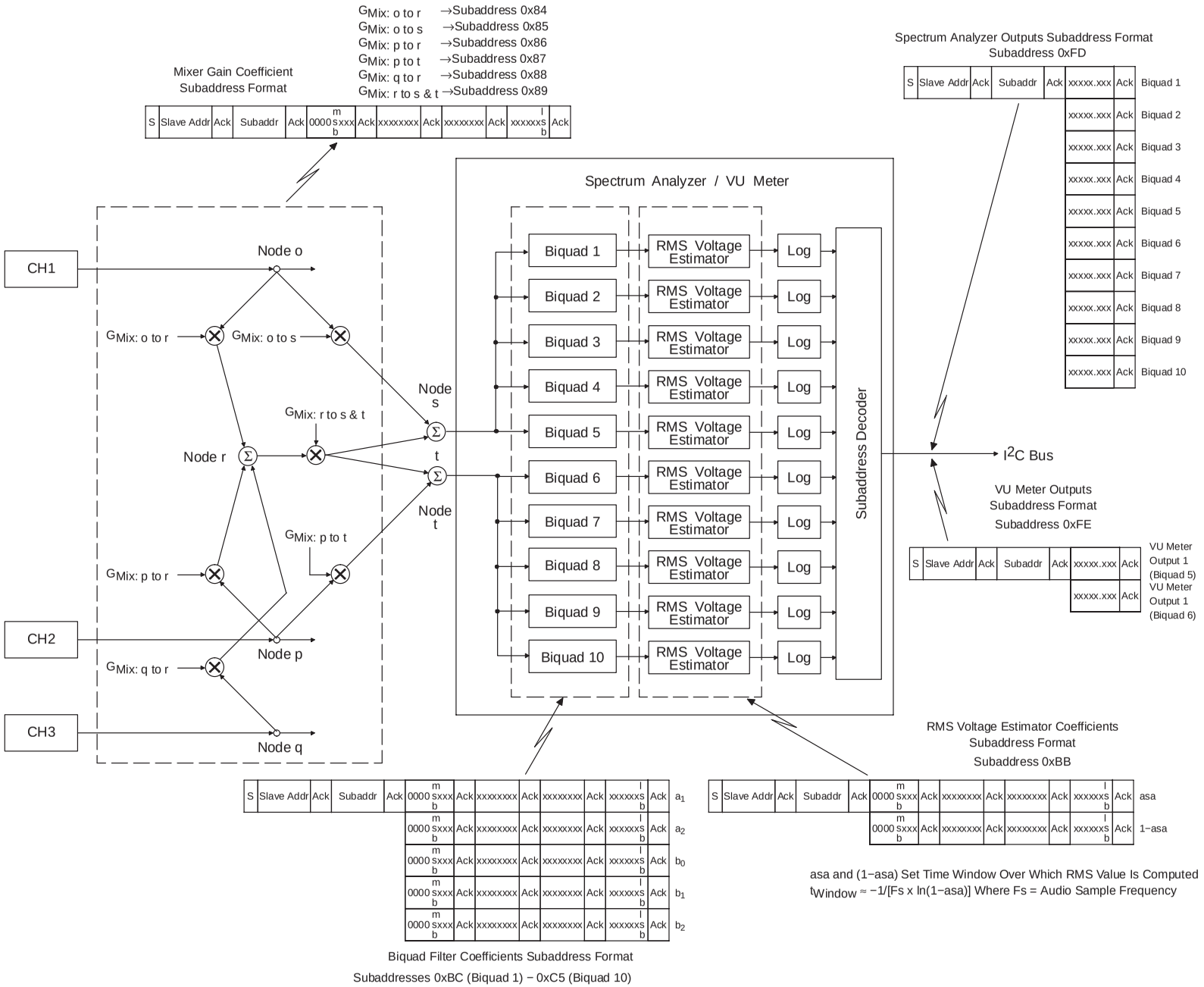


Figure 3−29. Spectrum Analyzer/VU Meter Block Diagram

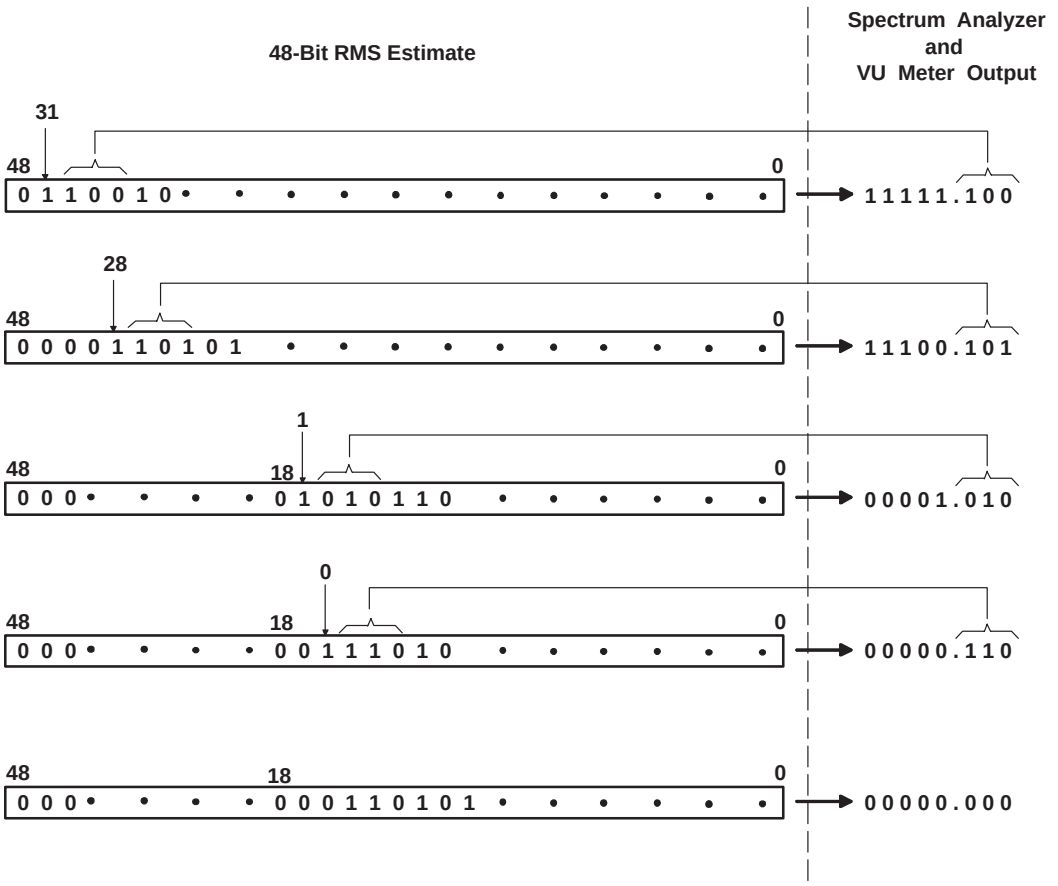


Figure 3−30. Logarithmic Number Conversions—Spectrum Analyzer/VU Meter

The time window over which the rms estimate is conducted is programmable via the I^2C bus (subaddress 0xBB). The time window for a given set of coefficients is approximately:

$$t_{Window} \approx -\frac{1}{Fs\ \ell n(1 - asa)}$$

Where Fs is the audio sample rate and asa is a 5.23-format number. The variable asa (and 1 – asa) must be kept within the range of greater than zero and less than one. The time constant programmed applies to all 10 rms estimate blocks.

CAUTION: The spectrum analyzer and VU meter functions are only accessible in the I^2C slave mode.

3.9 Dither

The TAS3103A provides a dither block for adding triangular or quadratic (sum of two uncorrelated triangular distributions) distributed noise to the processed audio data stream prior to routing to the output serial audio port (SAP). Each of the three monaural channels in the TAS3103A has its own dedicated dither data stream that is statistically independent from the dither data streams used by the other two monaural channels. The statistical distribution of the dither data stream, triangular or quadratic, is selectable, but the selection made applies to the dither data streams for all three monaural channels. Each monaural channel is also assigned a mixer for adjusting the level at which the dither data stream is inserted into the audio data stream.

Figure 3−31 is a detailed block diagram of the dither block. Five subaddresses are used to fully configure the dither blocks and the associated channel mixers. In the I^2C master mode, these dither parameters are set by the EEPROM content and cannot be subsequently changed.

3.9.1 Dither Seeds

The dither circuit consists of two linear feedback shift registers—LFSR1 and LFSR2. The dither seed subaddress (0xC7) consists of a byte-wide seed for LFSR1 (bits 7:0) and a byte-wide seed for LFSR2 (bits 15:8). The seeds serve to define the starting point of each LFSR sequence, but not the feedback structure itself. Each linear feedback shift register (LFSR) is a 26-bit structure that runs off the digital audio processor (DAP) clock. For a maximum DAP clock frequency of 135.168 MHz [12.288 MHz (MCLKI) $\times$ 11 (PLL multiplier)], the 26-bit LFSR has a cycle time of 496.5 ms (2^{26}/135.168 MHz). LFSR1 and LFSR2 use the same feedback structure but different taps for outputting. As long as the seeds for the two LFSRs are not $\pm$13 counts apart (in which case the two different sets of output taps would correlate), any set of seed values for LFSR1 and LFSR2 is suitable.

When two or more TAS3103As are powered by the same supply, a concern as to whether or not there is correlation between the dither data streams on different chips arises. At power turn on, a TAS3103A does not begin the dither process until reset is deactivated. If the TAS3103As are reset by the internal power-good signals from the internal regulators, the chip-to-chip variance of the logic voltage threshold point at which the power-good signal is deactivated ensures, with a high probability, that the dither data streams between chips are uncorrelated. However, when an external logic-driven reset is applied to the TAS3103As, the probability of correlation between the dither data streams on different chips after the reset is removed significantly increases. For this reason, the dither seeds have been made programmable via the I^2C bus.

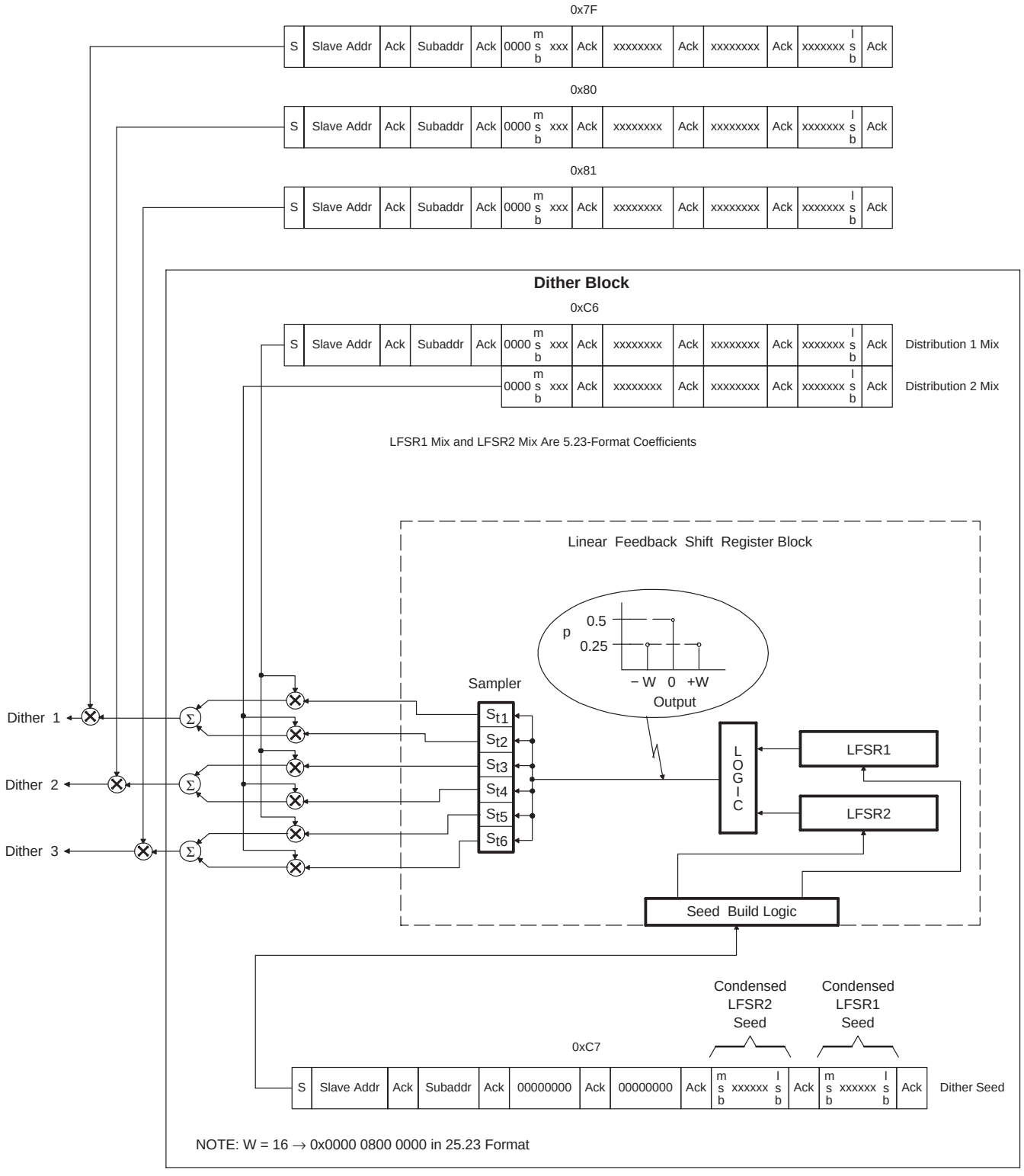


Figure 3−31. Dither Data Block Diagram

When updating multiple TAS3103As with dither seeds, timing should be taken into account. The recommended seed update process is to load all TAS3103As with their seed values in less time than the minimum LFSR cycle time of 496.5 ms, and use the same set of seeds for all TAS3103As. Each TAS3103A immediately begins running, starting at the state set by the new seed, on receiving the new seed. The sequential delivery, in time, of the seeds to the multiple TAS3103As ensures that the TAS3103As do not all start with their new seeds at the same time, and the completion of the process in less than 496.5 ms ensures that previously programmed TAS3103As are not repeating the cycle at the same time that another TAS3103A is being programmed with the same seed set—causing correlation between the dither data streams of the two TAS3103As.

CAUTION: The state of the digital audio processor may prevent the loading of a new I^2C-commanded dither seed value. Anytime a new seed is loaded into the TAS3103A via an I^2C write transaction to address 0xC7, it must be followed by an I^2C read transaction to address 0xC7 to verify that the new seed value was accepted. If the new seed was not accepted, the write-read sequence must be repeated.

3.9.2 Dither Mix Options

In Figure 3−31, it is seen that the two LFSRs are logically combined to produce a triangular probability distribution. This distribution is then sampled at six different points in time by the DAP processing clock to create six statistically independent data streams. Sampling the LSFR outputs at different points in time within an audio sample period (1/LRCLK) ensures that the six dither data streams are uncorrelated. The six uncorrelated dither data streams are then routed through mixers. Each pair of mixer outputs is then applied to a summation block. It is noted in Figure 3−31 that each of the three mixer pairs has the same set of coefficients (set by I^2C subaddress 0xC6). If the coefficients for both mixers are set to 1, a quadratic distribution is obtained. If either coefficient is set to 0, a triangular distribution is obtained. If both coefficients are set to 0, dither is disabled.

3.9.3 Dither Gain Mixers

Figure 3−31 shows the peak magnitude of the triangular distribution to be ±16 in the 48-bit DAP word (25.23 format). The peak magnitude of the quadratic dither data is twice this, or ±32. Figure 3−32 shows the position of this peak magnitude value in reference to the DAP 48-bit data word. Table 3−8 lists the mixer gains required to position the dither data stream at the LSB of the output data word for different data sample word sizes.

Table 3−8. Mixer Gain Setting for LSB Dither Data Insertion

DISTRIBUTION	MIXER GAIN COEFFICIENT				
	32-BIT SAMPLE	24-BIT SAMPLE	20-BIT SAMPLE	18-BIT SAMPLE	16-BIT SAMPLE
Triangular	2^{-19}	2^{-11}	2^{-7}	2^{-5}	2^{-3}
Quadratic	2^{-20}	2^{-12}	2^{-8}	2^{-6}	2^{-4}

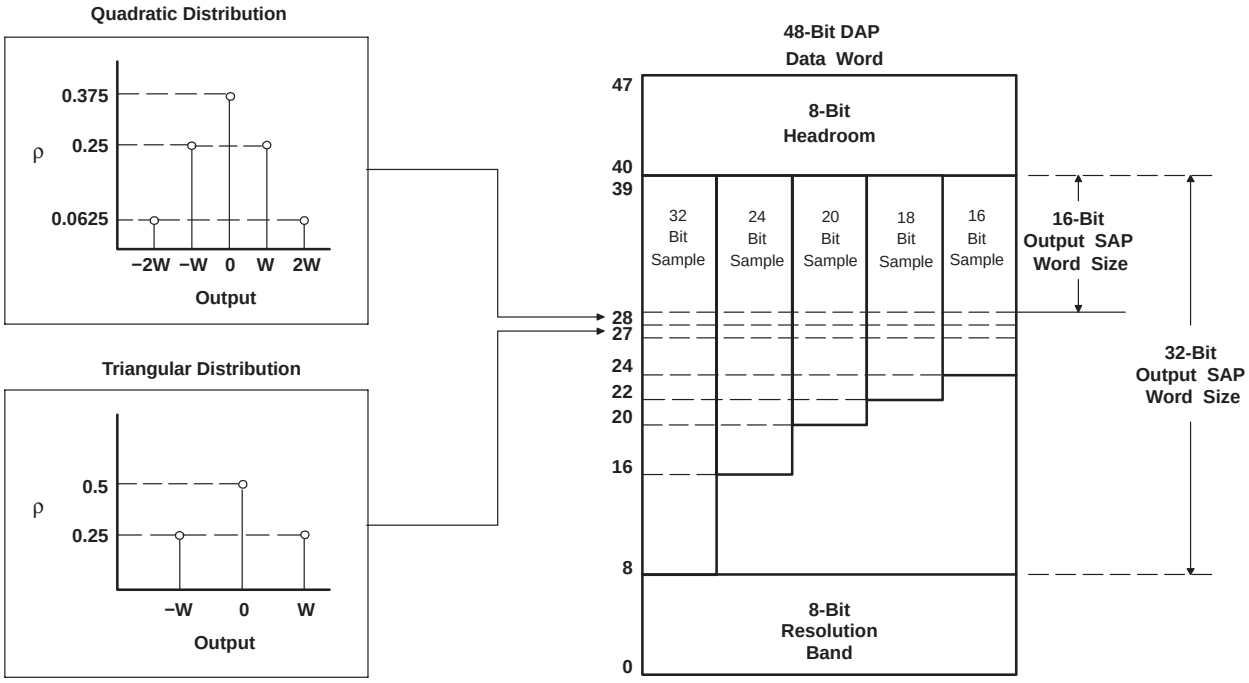
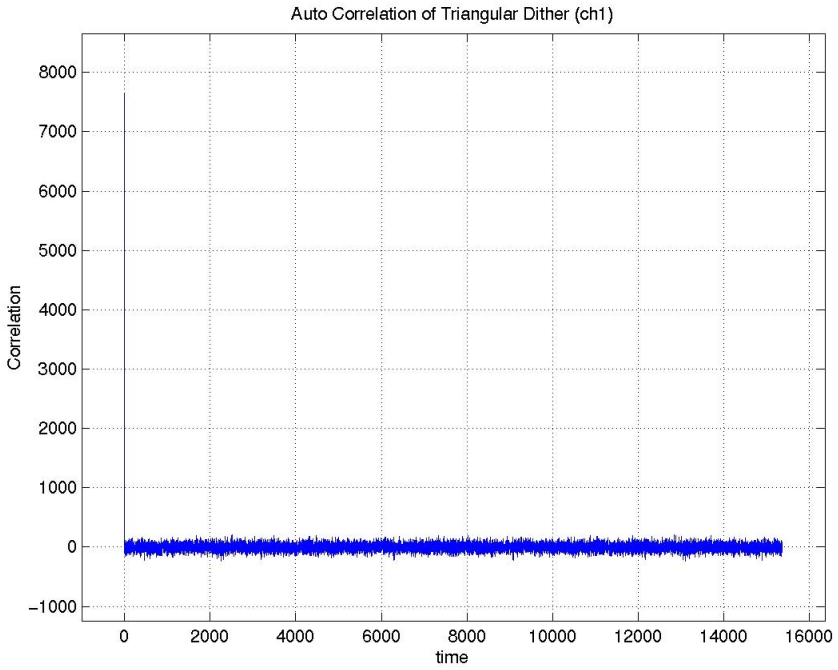


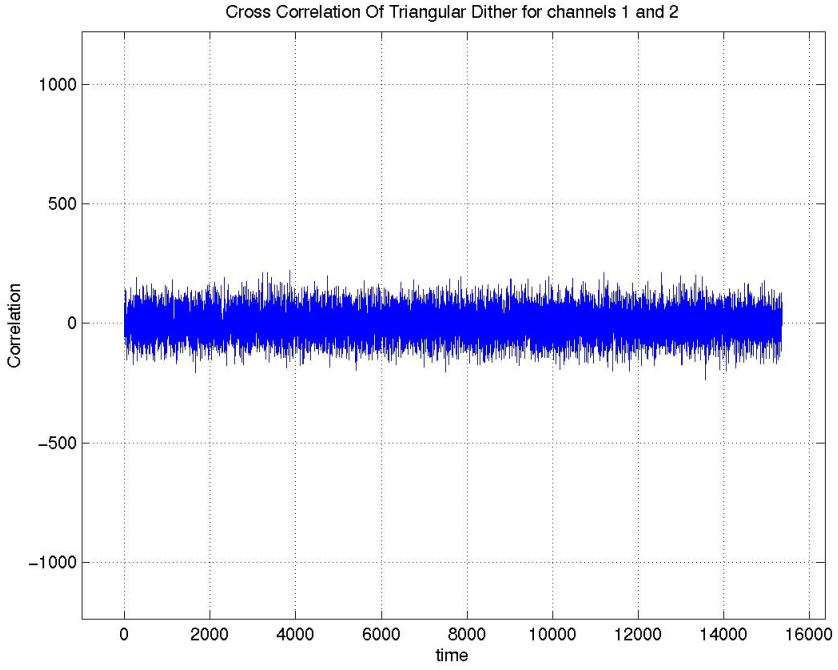
Figure 3−32. Dither Data Magnitude (Gain = 1)

3.9.4 Dither Statistics

Figure 3−33 presents plots of the autocorrelation and channel-to-channel correlation properties of the dither data stream when configured as triangular distributed noise. Figure 3−33(a) is the circular autocorrelation of 16K samples of dither data collected from the TAS3103A. The audio signal level was set to zero, and the dither data stream was inserted at the LSB of the output word. The autocorrelation contains a single line of value 8000, at the point of correlation, and random noise terms of approximately ±150 counts in value. The value 8000 agrees with the selection of the triangular distribution—50% of the 16K dither output samples are of value ±1. Figure 3−33(b) is the circular correlation of 16K samples of dither data from CH1 and 16K samples of dither data from CH2. No points of correlation are in this plot, verifying that the two data streams are uncorrelated. The random noise terms are again approximately ±150 counts in value.



(a) Auto-Correlation Plot − CH1

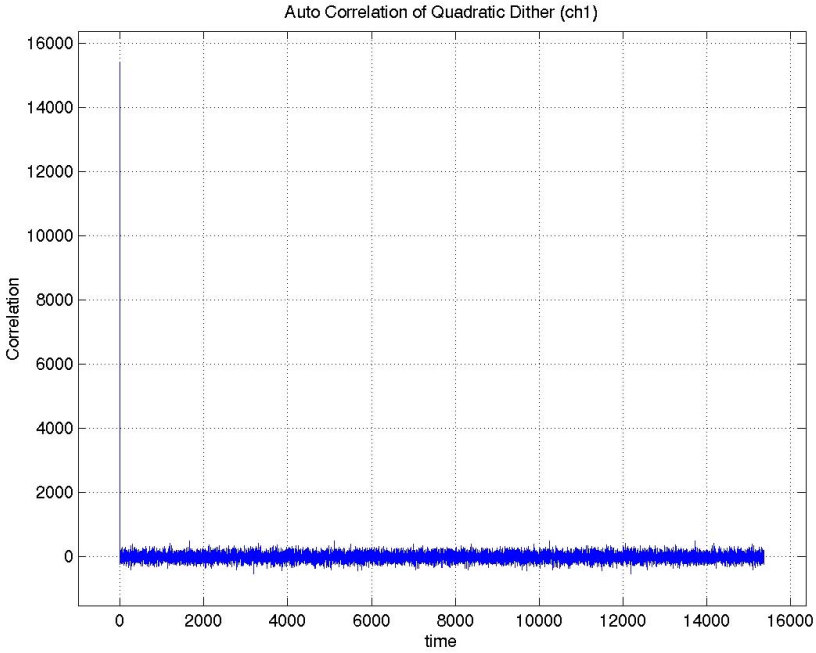


(b) Correlation Plot − CH1 and CH2

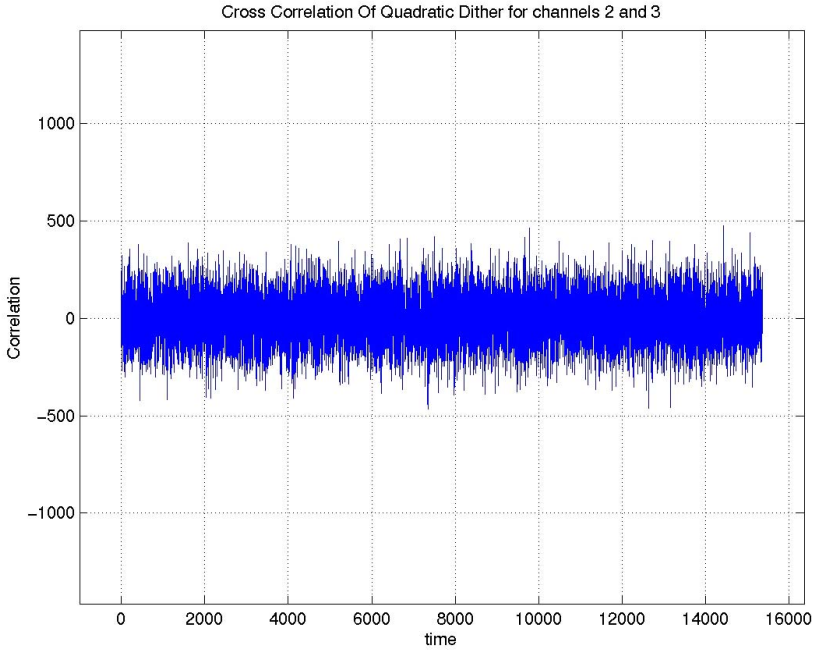
Figure 3−33. Triangular Dither Statistics, Case 1

Figure 3−34 presents plots of the autocorrelation and channel-to-channel correlation properties of the dither data stream when configured as quadratic distributed noise. Figure 3−34(a) is the circular autocorrelation of 16K samples of dither data collected from the TAS3103A. The audio signal level was set to zero and the dither data stream was inserted at the LSB+1 level of the output word. The autocorrelation contains a single line of value 16,000 at the point of correlation and random noise terms of approximately ±300 counts in value. The value 16,000 agrees with the

selection of the quadratic distribution—50% of the 16K dither output samples are of value ±1 ($0.5 \times 1^2 \times 16{,}000 = 8000$) and 12.5% of the 16K dither output samples are ±2 ($0.125 \times 2^2 \times 16{,}000 = 8000$). Figure 3−34(b) is the circular correlation of 16K samples of dither data from CH2 and 16K samples of dither data from CH3. No points of correlation are in this plot, verifying that the two data streams are uncorrelated. The random noise terms are approximately ±300 counts in value, as the dither data pattern was inserted at the LSB+1 bit of the output word instead of the LSB bit, as was the case for triangular dither.



(a) Auto-Correlation Plot − CH1



(b) Correlation Plot − CH2 and CH3

Figure 3−34. Triangular Dither Statistics, Case 2

3.10 Output Crossbar Mixers

The TAS3103A has three serial output ports—SDOUT1, SDOUT2, and SDOUT3. Each serial output port is assigned two processing nodes within the TAS3103A. One of the two nodes sources the left stereo data sample, and the other node sources the right stereo data sample. Figure 3−35 shows the assignment of these internal nodes to the serial output ports. Two cases are shown in Figure 3−35—discrete mode and TDM mode. The discrete mode connections are straightforward, but the TDM connections are considerably more involved in order to support the different one-chip and two-chip TDM modes. See *Input and Output Serial Audio Port (SAPs)*, Section 2.1, for more discussion on the TDM modes.

The purpose of the output crossbar is to give each of the three monaural channels in the TAS3103A access to any of the six internal processing nodes (U, V, W, X, Y, and Z) that supply data to the three serial output ports. This flexibility in the routing of the monaural channel outputs to the serial output ports, coupled with the flexibility in the routing of the serial input ports to the monaural channels, fully decouples the input data from the output data. A given process flow and output data topology can be obtained from any ordering of data into the TAS3103A.

Figure 3−36 shows the output crossbar mixer topology. Each monaural channel feeds six mixers. The six mixers, in turn, feed the six output nodes U, V, W, X, Y, and Z. A given monaural channel can thus be connected to either the left or right side of SDOUT1, SDOUT2, and SDOUT3.

The mixers, although capable of performing boost (gain) and cut (attenuation) on the outgoing audio, are typically used to facilitate on/off switching (a 5.23−format coefficient value of 0x0800000 turns the mixer on and a coefficient value of 0x0000000 turns the mixer off). The audio data streams at the input to these mixers include dither, and any boost or cut in the audio at this point affects the dither levels as well.

Node r in Figure 3−36 provides a means of outputting a post-processed sum of the audio on channel 1 and channel 2. This capability could be used to generate a center audio component from L and R components being processed on channels 1 and 2. This would allow channel 3 to be a subwoofer channel. Node r could also be used to create a subwoofer channel, assuming an active subwoofer with filtering capability is receiving the subwoofer output. This option would then free channel 3 for center channel processing.

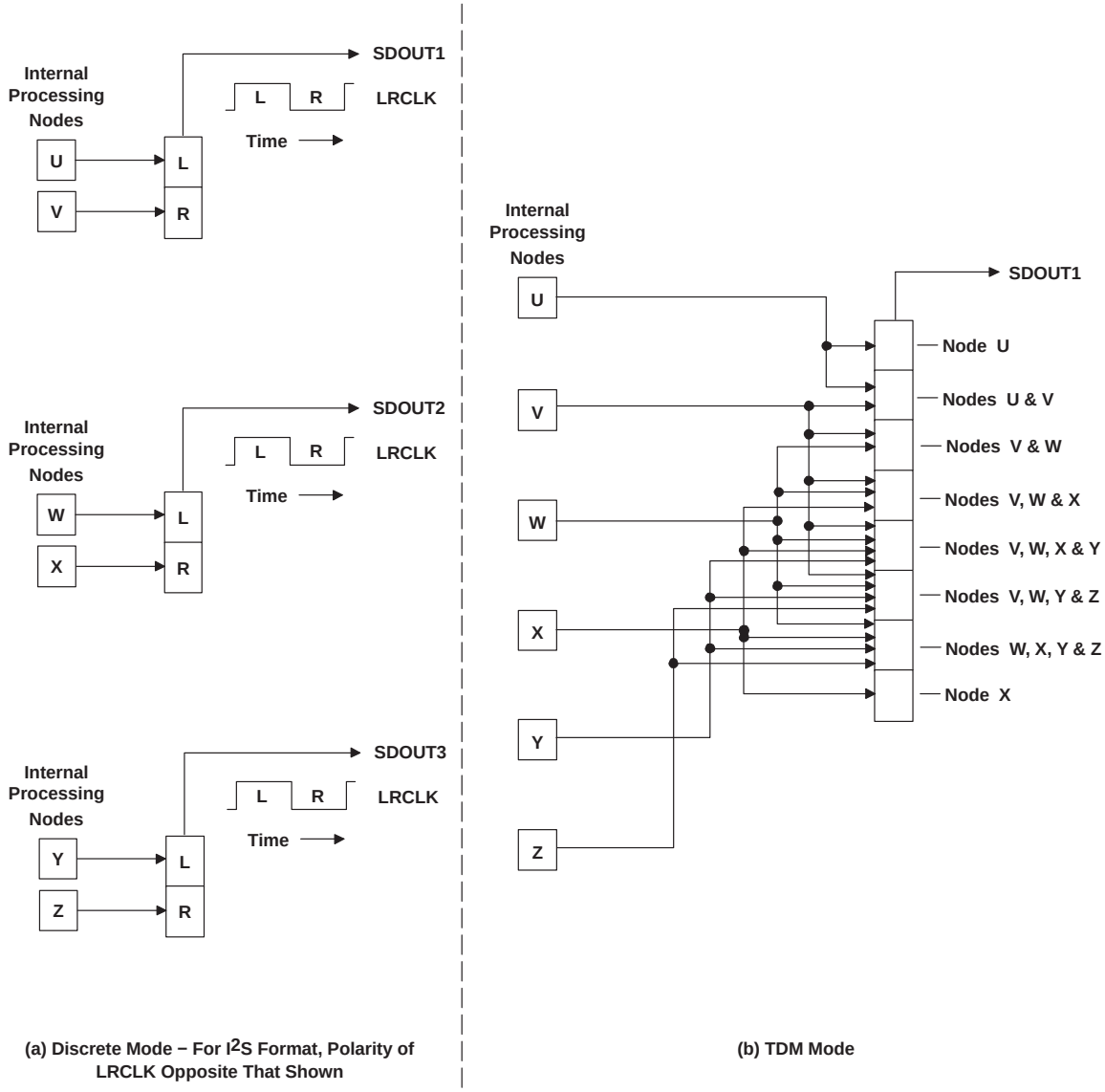


Figure 3−35. Processing Node to Serial Output Port Topology

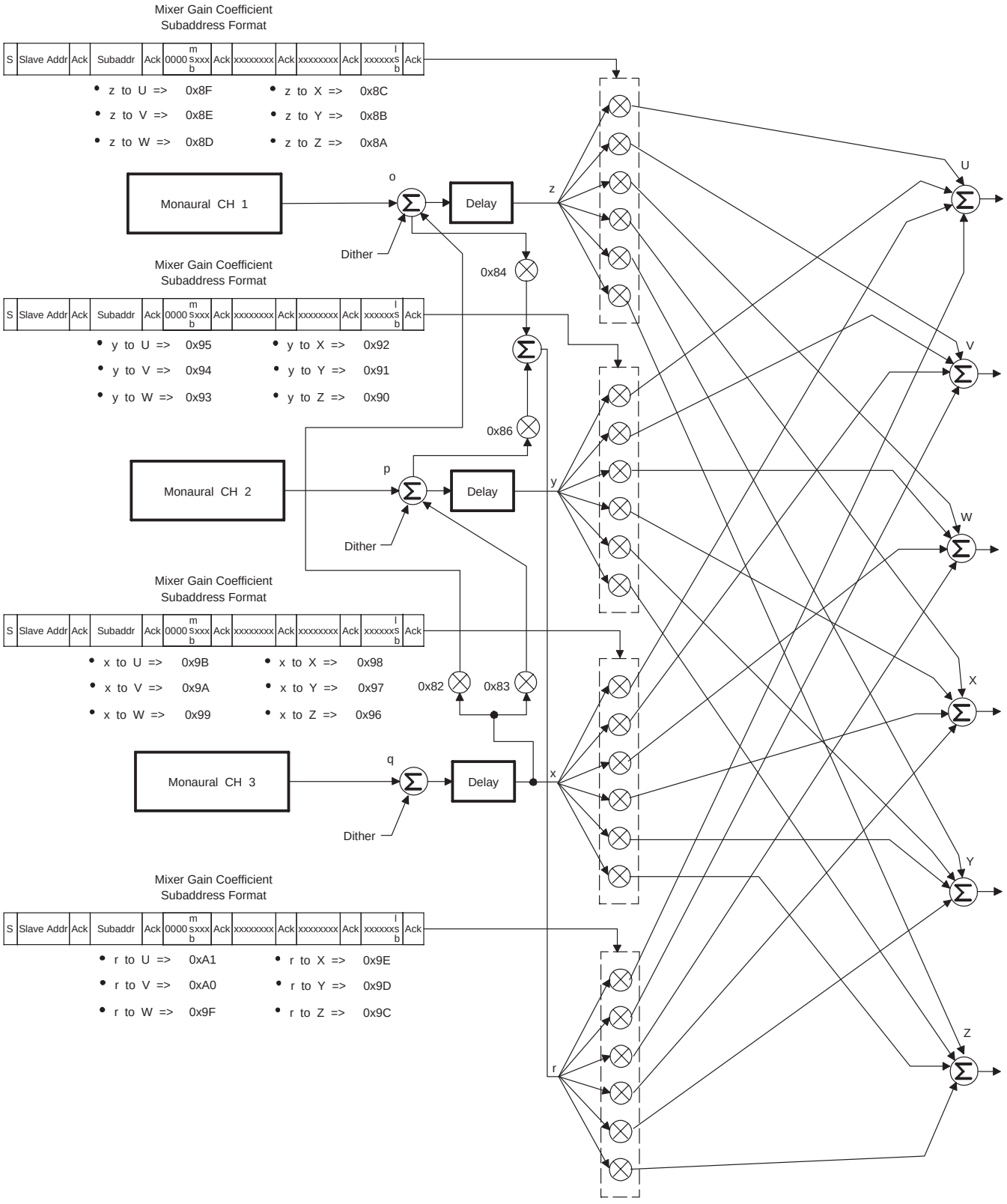


Figure 3−36. Output Crossbar Mixer Topology

4 Electrical Specifications

4.1 Absolute Maximum Ratings Over Operating Temperature Ranges (unless otherwise noted)†

Supply voltage range: VDDS . . . . . −0.5 to 3.8 V
A_VDDS . . . . . −0.5 to 3.8 V
Input voltage range, V_I: 3.3-V LVCMOS . . . . . −0.5 V to VDDS + 0.5 V
1.8-V LVCMOS . . . . . −0.5 V to AVDD(1) + 0.5 V
Output voltage range, V_O: 3.3-V LVCMOS . . . . . −0.5 V to VDDS + 0.5 V
1.8-V LVCMOS . . . . . −0.5 V to DVDD(2) + 0.5 V
1.8-V LVCMOS . . . . . −0.5 V to AVDD(3) + 0.5 V
Input clamp current, I_{IK} (V_I < 0 or V_I > DVDD) . . . . . ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > DVDD) . . . . . ±20 mA
Operating free-air temperature . . . . . 0°C to 70°C
Storage temperature range, T_{stg} . . . . . −65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 260°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. AVDD is a 1.8-V supply derived from a regulator in the TAS3103A chip. Pin XTALI is the only TAS3103A input that is referenced to this 1.8-V logic supply. The absolute maximum rating listed is for reference; only a crystal should be connected to XTALI.
2. DVDD is a 1.8-V supply derived from regulators internal to the TAS3103A chip. DVDD is routed to pin 29 (DVDD_BYPASS_CAP) to provide access to external filter capacitors, but should not be used to source power to external devices.
3. Pin XTALO is the only TAS3103A output that is derived from the internal 1.8-V logic supply AVDD. The absolute maximum rating listed is for reference; only a crystal should be connected to XTALO. AVDD is also routed to pin 6 (AVDD_BYPASS_CAP) to provide access to external filter capacitors, but should not be used to source power to external devices.

DISSIPATION RATING TABLE (High-k Board, 105°C Junction)

PACKAGE	$T_A \le 25°C$ POWER RATING	DERATING FACTOR ABOVE $T_A = 25°C$	$T_A = 85°C$ POWER RATING
DBT	1.094 W	13.68 mW/°C	0.478 mW

4.2 Recommended Operating Conditions

		MIN	NOM	MAX	UNITS
Digital supply voltage, VDDS		3	3.3	3.6	V
Analog supply voltage, A_VDDS		3	3.3	3.6	V
High-level input voltage, V_{IH}	3.3-V LVCMOS	0.7 VDDS		VDDS	V
	1.8-V LVCMOS (XTALI)	0.7 AVDD		AVDD	
Low-level input voltage, V_{IL}	3.3-V LVCMOS	0		0.3 VDDS	V
	1.8-V LVCMOS (XTALI)	0		0.3 AVDD	
Input voltage, V_I	3.3-V LVCMOS	0		VDDS	V
	1.8-V LVCMOS (XTALI)	0		AVDD	
Output voltage, V_O	3.3-V LVCMOS	0.8 VDDS		VDDS	V
	1.8-V LVCMOS (XTALO)	0.8 AVDD		AVDD	
Operating ambient air temperature range, T_A	Commercial	0		70	°C
Operating junction temperature range, T_J	Commercial	0		105	°C

4.3 Electrical Characteristics Over Recommended Operating Conditions (Unless Otherwise Noted)

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNITS
V_{OH}	High-level output voltage	3.3-V LVCMOS	I_{OH} = −4 mA	0.8 VDDS			V
		1.8-V LVCMOS (XTALO)	I_{OH} = −0.55 mA	0.8 AVDD			
V_{OL}	Low-level output voltage	3.3-V LVCMOS	I_{OL} = 4 mA			0.22 VDDS	V
		1.8-V LVCMOS (XTALO)	I_{OL} = 0.75 mA			0.22 AVDD	
I_{OZ}	High-impedance output current	3.3-V LVCMOS				±20	µA
I_{IL}	Low-level input current(1)	3.3-V LVCMOS	$V_I = V_{IL}$			±20	µA
		1.8-V LVCMOS (XTALI)	$V_I = V_{IL}$			±20	
I_{IH}	High-level input current(2)	3.3-V LVCMOS	$V_I = V_{IH}$			±20	µA
		1.8-V LVCMOS (XTALI)	$V_I = V_{IH}$			±20	
I_{DVDD}	Digital supply current		DSP clock = 135 MHz, LRCLK = 96 kHz, MCLKI/XTALI = 12.228 MHz		75		mA
			DSP clock = 67.5 MHz, LRCLK = 48 kHz, MCLKI/XTALI = 12.228 MHz		44		
			DSP clock = 33.75 MHz, LRCLK = 24 kHz, MCLKI/XTALI = 12.228 MHz		25		
			LRCLK = 48 kHz, MCLKI/XTALI = 12.288 MHz, Power down enabled		3.5		
			No LRCLK, SCLK. MCLKI/XTALI = 12.288 MHz, Power down enabled		2.2		
			No LRCLK, SCLK, or MCLKI/XTALI, Power down enabled		2		
I_{A_DVDD}	Analog supply current		DSP clock = 135 MHz, LRCLK = 96 kHz, MCLKI/XTALI = 12.228 MHz		2.9		mA
			DSP clock = 67.5 MHz, LRCLK = 48 kHz, MCLKI/XTALI = 12.228 MHz		2.7		
			DSP clock = 33.75 MHz, LRCLK = 24 kHz, MCLKI/XTALI = 12.228 MHz		2.4		
			LRCLK = 48 kHz, MCLKI/XTALI = 12.288 MHz, Power down enabled		1.5		
			No LRCLK, SCLK, or MCLKI/XTALI, Power down enabled		1.5		

NOTES: 1. Value given is for those input pins that connect to an internal pullup resistor as well as an input buffer. For inputs that have a pulldown resistor or no resistor, I_{IL} is ±1 µA.
2. Value given is for those input pins that connect to an internal pulldown resistor as well as an input buffer. For inputs that have a pullup resistor or no resistor, I_{IH} is ±1 µA.

4.4 TAS3103A Timing Characteristics

4.4.1 Master Clock Signals Over Recommended Operating Conditions (Unless Otherwise Noted)

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNITS
$f_{(XTALI)}$	Frequency, XTALI ($1/t_{c(1)}$)			2.8		20	MHz
$f_{(MCLKI)}$	Frequency, MCLKI ($1/t_{c(2)}$)			2.8		25	MHz
$t_{w(MCLKI)}$	Pulse duration, MCLKI high, see Note 1			H_{MCLKI} − 25	H_{MCLKI}	H_{MCLKI} + 25	ns
MCLKI jitter						±5	ns
$f_{(MCLKO)}$	Frequency, MCLKO ($1/t_{c(3)}$)			2.8		25	MHz
$t_{r(MCLKO)}$	Rise time, MCLKO		C_L = 30 pF			9.5	ns
$t_{f(MCLKO)}$	Fall time, MCLKO		C_L = 30 pF			9.5	ns
$t_{w(MCLKO)}$	Pulse duration, MCLKO high, see Note 4				H_{MCLKO}		ns
MCLKI jitter	XTALI master clock source				80		ps
	MTALI master clock source, see Note 5						
$t_{d(MI-MO)}$	Delay time, MCLKI rising edge to MCLKO rising edge	MCKLO = MCLKI	See Note 2			17	ns
		MCLKO < MCLKI	See Note 2 and Note 3			17	ns

NOTES: 1. H_{MCLKI} = 1 / 2MCLKI
2. Only applies when MCLKI is selected as master source clock.
3. Also applies to MCLKO falling edge when MCLKO = MCLKI/2 or MCLKI/4
4. H_{MCLKO} = 1 / 2MCLKO. MCLKO has the same duty cycle as MCLKI when MCLKO = MCLKI. When MCLKO = 0.5 MCLKI or 0.25 MCLKI, the duty cycle of MCLKO is typically 50%.
5. When MCLKO is derived from MCLKI, MCLKO jitter = MCLKI jitter

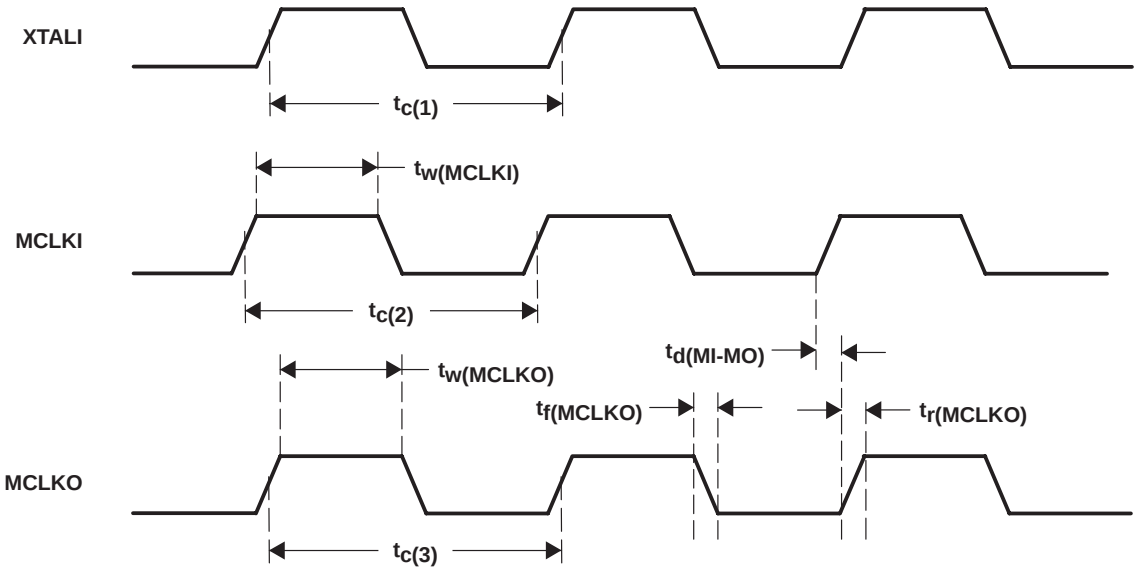


Figure 4−1. Master Clock Signals Timing Waveforms

4.4.2 Control Signals Over Recommended Operating Conditions (Unless Otherwise Noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNITS
$t_{w1(L)}$	Pulse duration, $\overline{RST}$ low		10			ns
t_{pd1}	Propagation delay, PWRDN high to power-down state asserted	See Note 1				µs
t_{pd2}	Propagation delay, PWRDN low to power-down state deasserted	See Note 2				µs

NOTES: 1. The maximum worst-case value for t_{pd1} is given by

$$t_{pd1\_worst\_case} = \frac{4096 + GPIOFSCOUNT}{LRCLK} + \frac{80}{Microprocessor\_Clock}$$

2. t_{pd2} is determined by the time it takes the internal digital PLL to reach a locked condition, which, in turn, is governed by the MCLKI/XTALI frequency and the PLL output frequency. For a 135-MHz PLL output and an MCLKI value of 24.576 MHz, t_{pd2} is typically 25 µs. For an 11.264-MHz PLL output clock and a 1.024-MHz MCLKI/XTALI master clock, t_{pd2} is typically 360 µs.

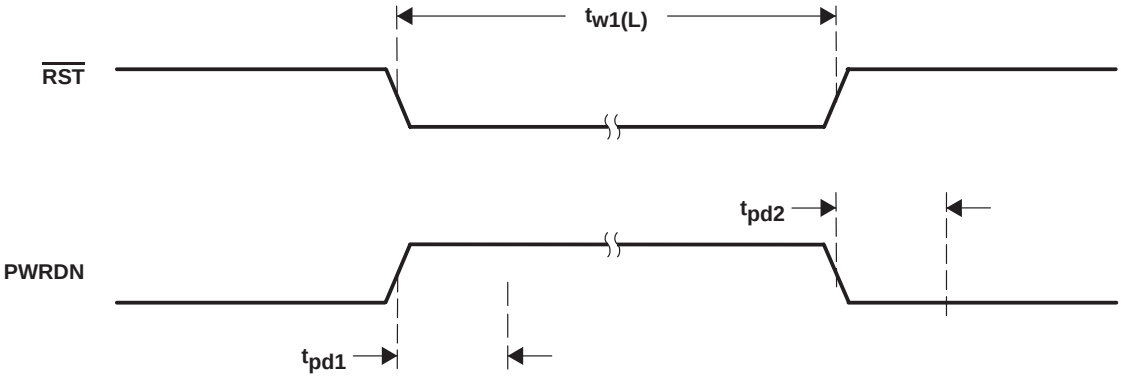


Figure 4−2. Control Signals Timing Waveforms

4.4.3 Serial Audio Port Slave-Mode Signals Over Recommended Operating Conditions (unless otherwise noted)

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNITS
f_{LRCLK}	Frequency, LRCLK (Fs)			8		96	kHz
$t_{w(SCLKIN)}$	Pulse duration, SCLKIN high		See Note 2	0.25 H_{SCLKIN}	HSCLKIN	0.75 H_{SCLKIN}	ns
f_{SCLKIN}	Frequency, SCLKIN		See Note 1	32 Fs		25	MHz
t_{cyc}	Cycle time, SCLKIN		See Note 1	40		1/32 Fs	ns
t_{pd1}	Propagation delay, SCLKIN falling edge to SDOUT					15.1	ns
t_{su1}	Setup time, LRCLK to SCLKIN rising edge			6.6			ns
t_{h1}	Hold time, LRCLK from SCLKIN rising edge			0			ns
t_{su2}	Setup time, SDIN to SCLKIN rising edge			1.15			ns
t_{h2}	Hold time, SDIN from SCLKIN rising edge			2.3			ns
t_{pd2}	Propagation delay, SCLKIN falling edge to SCLKOUT2 falling edge	SCLKOUT2 = SCLKIN				12.4	ns
		SCLKOUT2 < SCLKIN				12.5	ns

NOTES: 1. Typical duty cycle is 50/50.
2. $H_{SCLKIN} = 1/2f_{SCLKIN}$

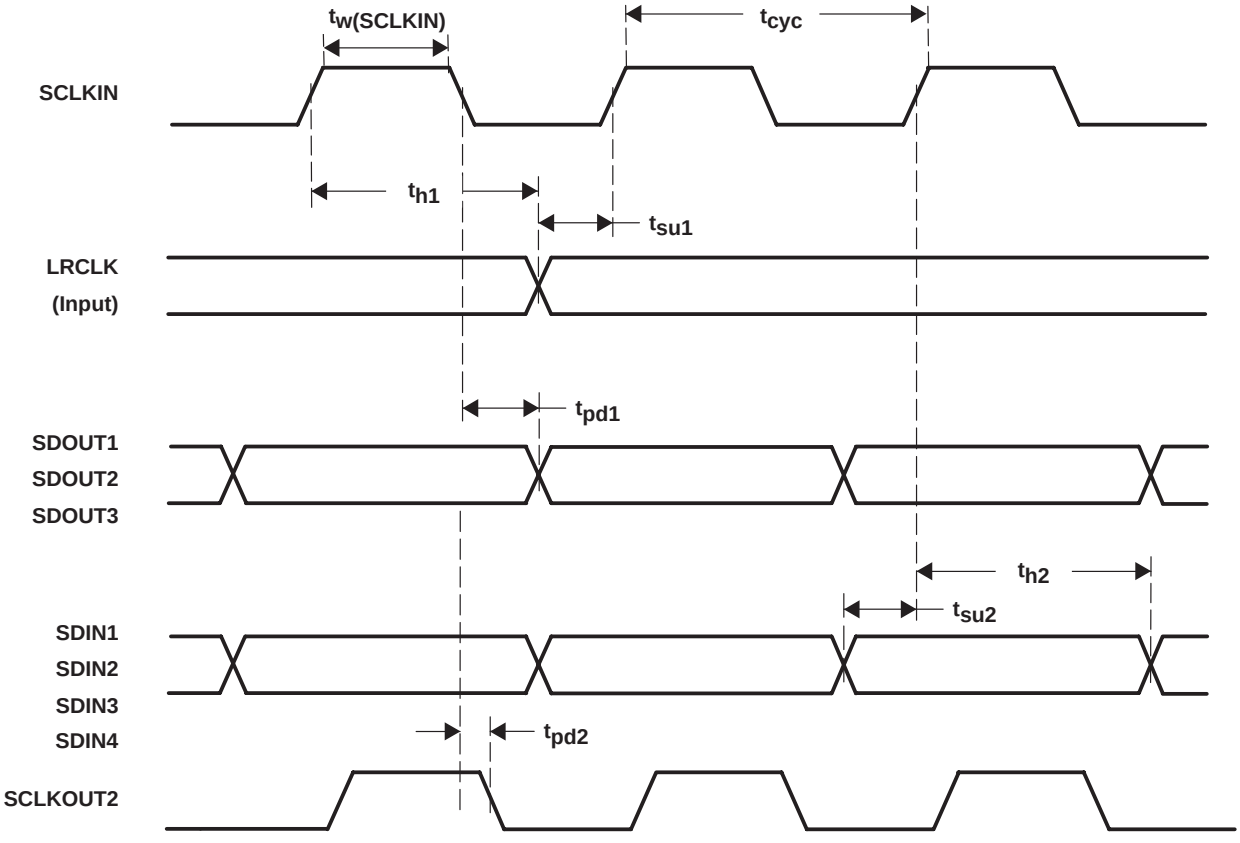


Figure 4−3. Serial Audio Port Slave-Mode Timing Waveforms

4.4.4 Serial Audio Port Master-Mode Signals Over Recommended Operating Conditions (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNITS
$f_{(LRCLK)}$	Frequency LRCLK		8		96	kHz
$t_{r(LRCLK)}$	Rise time, LRCLK	C_L = 30 pF			11.4	ns
$t_{f(LRCLK)}$	Fall time, LRCLK	C_L = 30 pF			11.2	ns
$f_{(SCLKOUT)}$	Frequency ($1/t_{cyc}$), SCLKOUT1/SCLKOUT2	See Note 1	32 Fs		25	MHz
$t_{r(SCLKOUT)}$	Rise time, SCLKOUT1/SCLKOUT2	C_L = 30 pF			9.5	ns
$t_{f(SCLKOUT)}$	Fall time, SCLKOUT1/SCLKOUT2	C_L = 30 pF			9.8	ns
$t_{pd1(SCLKOUT1)}$	Propagation delay, SCLKOUT1 falling edge to LRCLK edge				4.1	ns
$t_{pd1(SCLKOUT2)}$	Propagation delay, SCLKOUT2 falling edge to LRCLK edge				4.3	ns
t_{pd2}	Propagation delay, SCLKOUT2 falling edge to SDOUT				3.4	ns
t_{su}	Setup time, SDIN to SCLKOUT1 rising edge		18.4			ns
t_h	Hold time, SDIN from SCLKOUT1 rising edge		23			ns
t_{sk}	Skew time, SCLKOUT1 to SCLKOUT2			0.8	3	ns

NOTES: 1. :Typical duty cycle is 50/50.

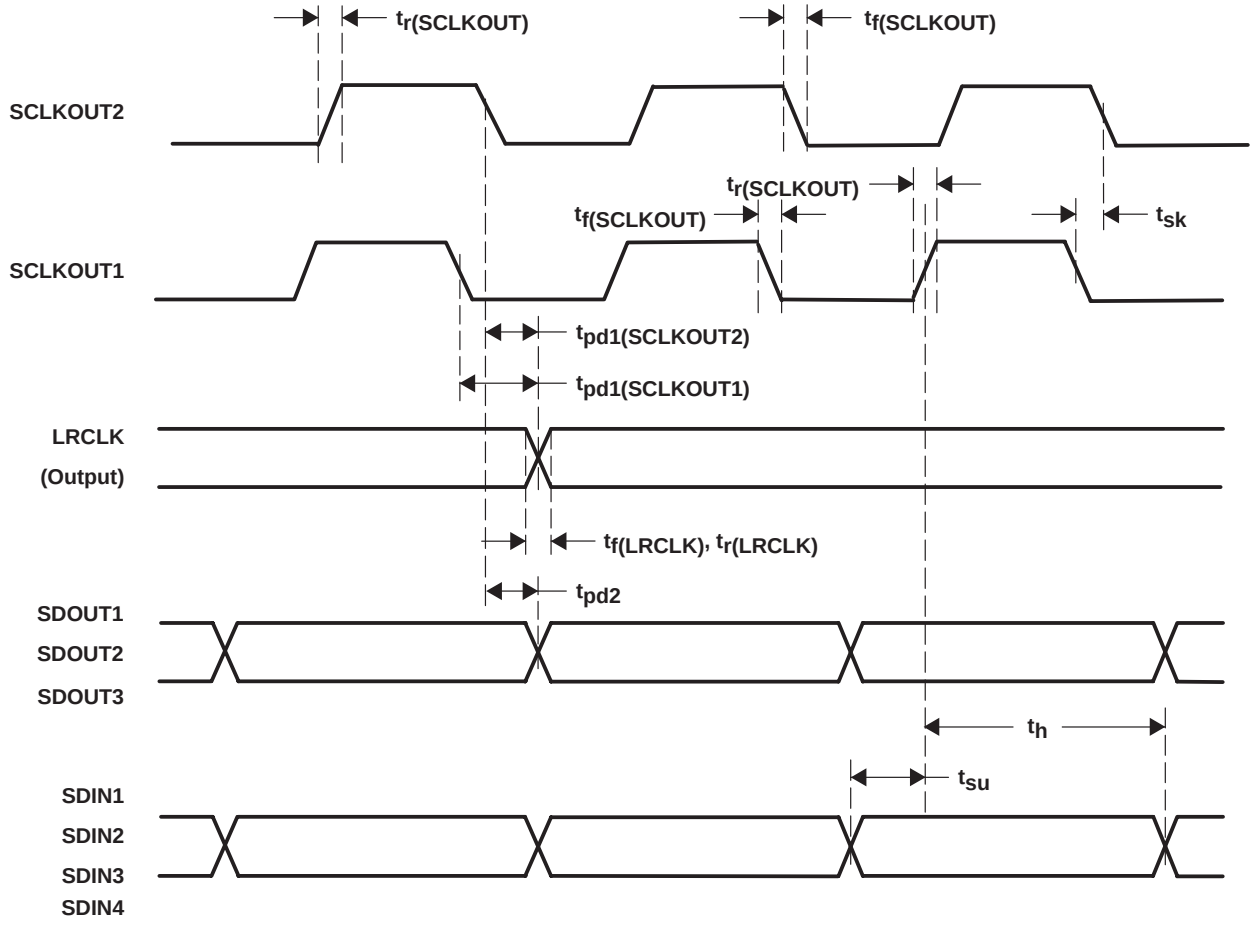


Figure 4−4. Serial Audio Port Master-Mode Timing Waveforms

4.4.5 Characteristics of the SDA and SCL I/O Stages for F/S-Mode I^2C-Bus Devices

PARAMETER		TEST CONDITIONS	STANDARD MODE		FAST MODE		UNITS
			MIN	MAX	MIN	MAX	
V_{IL}	LOW-level input voltage		–0.5	0.3 V_{DD}	–0.5	0.3 V_{DD}	V
V_{IH}	HIGH-level input voltage		0.7 V_{DD}		0.7 V_{DD}		V
V_{hys}	Hysteresis of inputs		N/A	N/A	0.05 V_{DD}		V
V_{OL1}	LOW level output voltage (open drain or open collector)	3 mA sink current			0	0.4	V
t_{of}	Output fall time from V_{IHmin} to V_{ILmax}	Bus capacitance from 10 pF to 400 pF		250	7 + 0.1 C_b (2)	250	ns
I_i	Input current, each I/O pin		–10	10	–10 (3)	10 (3)	μA
t_{SP}	Pulse width of spikes which must be suppressed by the input filter		N/A	N/A	0	0 (1)	ns
C_i	Capacitance, each I/O pin			10		10	pF

NOTES: 1. SCL and SDA do not have a 50-ns glitch filter.
NOTES: 2. C_b = capacitance of one bus line in pF. The output fall time is faster than the standard I^2C specification.
NOTES: 3. The I/O pins of fast-mode devices must not obstruct the SDA and SCL lines if V_{DD} is switched off.

Caution: The TAS3103A does not have 50-ns glitch filtering on the SCL and SDA inputs.

Caution: SDA does not have the standard I2C specification 300-ns internal hold time. SDA must be valid by the rising and falling edges of SCL.

4.4.6 Characteristics of the SDA and SCL I/O Stages for F/S-Mode I²C-Bus Devices, (1)

PARAMETER		STANDARD MODE		FAST MODE		UNITS
		MIN	MAX	MIN	MAX	
f_{SCL}	SCL clock frequency	0	100	0	400	kHz
t_{HD-STA}	Hold time (repeated) START condition. After this period, the first clock pulse is generated.	4		0.6		µs
t_{LOW}	LOW period of the SCL clock	4.7		1.3		µs
t_{HIGH}	HIGH period of the SCL clock	4		0.6		µs
t_{SU-STA}	Setup time for repeated START	4.7		0.6		µs
t_{SU-DAT}	Data setup time	250		100		µs
t_{HD-DAT}	Data hold time (2)(3)	0	3.45	0	0.9	µs
t_r	Rise time of both SDA and SCL signals		1000	20 + 0.1 C_b (4)	300	ns
t_f	Fall time of both SDA and SCL signals		300	20 + 0.1 C_b (4)	300	ns
t_{SU-STO}	Setup time for STOP condition	4		0.6		µs
t_{BUF}	Bus free time between a STOP and START condition	4.7		1.3		µs
C_B	Capacitive load for each bus line		400		400	pF
V_{nL}	Noise margin at the LOW level for each connected device (including hysteresis)	0.1 V_{DD}		0.1 V_{DD}		V
V_{nH}	Noise margin at the HIGH level for each connected device (including hysteresis)	0.2 V_{DD}		0.2 V_{DD}		V

NOTES: 1. All values referred to V_{IHmin} and V_{ILmax} levels (see Section 4.4.5).
NOTES: 2. Note that SDA does not have the standard I²C specification 300-ns internal hold time. SDA must be valid by the rising and falling edges of SCL. TI recommends that a minimum 2-kΩ pullup resistor be used to avoid potential timing issues.
NOTES: 3. A fast-mode I²C-bus device can be used in a standard-mode I²C-bus system, but the requirement $t_{SU-DAT} \geq 250$ ns must then be met. This is automatically the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_{r\,max} + t_{SU-DAT} = 1000 + 250 = 1250$ ns (according to the standard-mode I²C-bus specification) before the SCL line is released.
NOTES: 4. C_b = total capacitance of one bus line in pF.

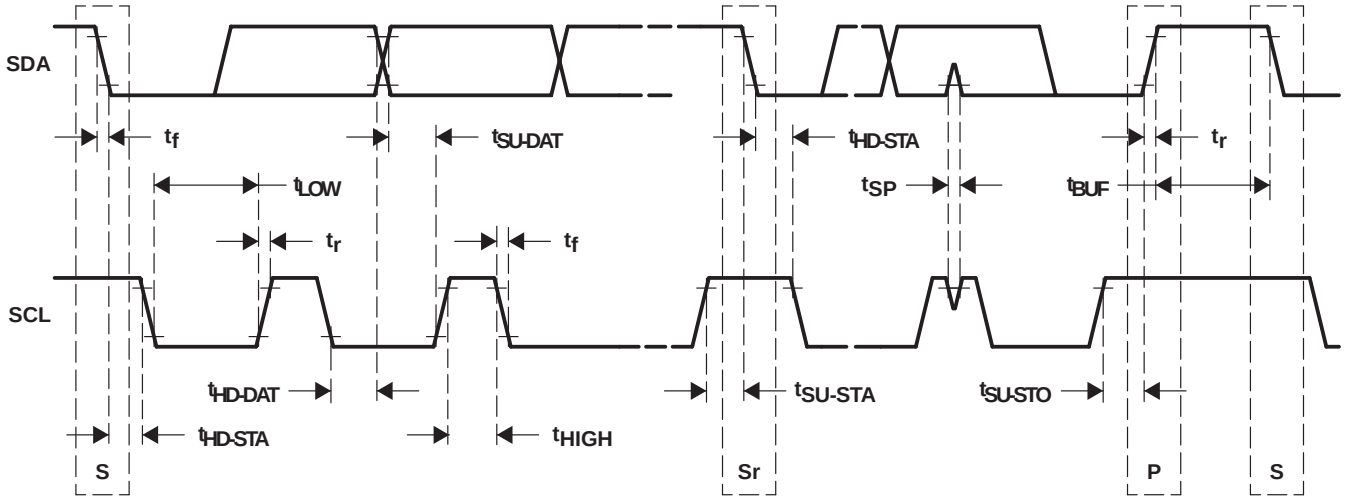


Figure 4–5. I²C Start and Stop Conditions Timing Waveforms

Appendix A

A.1 I2C Subaddress Table

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x00	Starting I2C check word	1	SCW(31:24), SCW(23:16), SCW(15:8), SCW(7:0)	0x81, 0x42, 0x24, 0x18
0x01	Mix A to a	1	u(31:28)A_a(27:24), A_a(23:16), A_a(15:8), A_a(7:0)	0x00, 0x80, 0x00, 0x00
0x02	Mix A to b	1	u(31:28)A_b(27:24), A_b(23:16), A_b(15:8), A_b(7:0)	0x00, 0x00, 0x00, 0x00
0x03	Mix A to c	1	u(31:28)A_c(27:24), A_c(23:16), A_c(15:8), A_c(7:0)	0x00, 0x00, 0x00, 0x00
0x04	Mix A to d	1	u(31:28)A_d(27:24), A_d(23:16), A_d(15:8), A_d(7:0)	0x00, 0x00, 0x00, 0x00
0x05	Mix A to e	1	u(31:28)A_e(27:24), A_e(23:16), A_e(15:8), A_e(7:0)	0x00, 0x00, 0x00, 0x00
0x06	Mix A to f	1	u(31:28)A_f(27:24), A_f(23:16), A_f(15:8), A_f(7:0)	0x00, 0x00, 0x00, 0x00
0x07	Mix B to a	1	u(31:28)B_a(27:24), B_a(23:16), B_a(15:8), B_a(7:0)	0x00, 0x00, 0x00, 0x00
0x08	Mix B to b	1	u(31:28)B_b(27:24), B_b(23:16), B_b(15:8), B_b(7:0)	0x00, 0x80, 0x00, 0x00
0x09	Mix B to c	1	u(31:28)B_c(27:24), B_c(23:16), B_c(15:8), B_c(7:0)	0x00, 0x00, 0x00, 0x00
0x0A	Mix B to d	1	u(31:28)B_d(27:24), B_d(23:16), B_d(15:8), B_d(7:0)	0x00, 0x00, 0x00, 0x00
0x0B	Mix B to e	1	u(31:28)B_e(27:24), B_e(23:16), B_e(15:8), B_e(7:0)	0x00, 0x00, 0x00, 0x00
0x0C	Mix B to f	1	u(31:28)B_f(27:24), B_f(23:16), B_f(15:8), B_f(7:0)	0x00, 0x00, 0x00, 0x00
0x0D	Mix C to a	1	u(31:28)C_a(27:24), C_a(23:16), C_a(15:8), C_a(7:0)	0x00, 0x00, 0x00, 0x00
0x0E	Mix C to b	1	u(31:28)C_b(27:24), C_b(23:16), C_b(15:8), C_b(7:0)	0x00, 0x00, 0x00, 0x00
0x0F	Mix C to c	1	u(31:28)C_c(27:24), C_c(23:16), C_c(15:8), C_c(7:0)	0x00, 0x00, 0x00, 0x00
0x10	Mix C to d	1	u(31:28)C_d(27:24), C_d(23:16), C_d(15:8), C_d(7:0)	0x00, 0x00, 0x00, 0x00
0x11	Mix C to e	1	u(31:28)C_e(27:24), C_e(23:16), C_e(15:8), C_e(7:0)	0x00, 0x00, 0x00, 0x00
0x12	Mix C to f	1	u(31:28)C_f(27:24), C_f(23:16), C_f(15:8), C_f(7:0)	0x00, 0x40, 0x00, 0x00
0x13	Mix D to a	1	u(31:28)D_a(27:24), D_a(23:16), D_a(15:8), D_a(7:0)	0x00, 0x00, 0x00, 0x00
0x14	Mix D to b	1	u(31:28)D_b(27:24), D_b(23:16), D_b(15:8), D_b(7:0)	0x00, 0x00, 0x00, 0x00
0x15	Mix D to c	1	u(31:28)D_c(27:24), D_c(23:16), D_c(15:8), D_c(7:0)	0x00, 0x00, 0x00, 0x00
0x16	Mix D to d	1	u(31:28)D_d(27:24), D_d(23:16), D_d(15:8), D_d(7:0)	0x00, 0x00, 0x00, 0x00
0x17	Mix D to e	1	u(31:28)D_e(27:24), D_e(23:16), D_e(15:8), D_e(7:0)	0x00, 0x00, 0x00, 0x00
0x18	Mix D to f	1	u(31:28)D_f(27:24), D_f(23:16), D_f(15:8), D_f(7:0)	0x00, 0x40, 0x00, 0x00
0x19	Mix E to a	1	u(31:28)E_a(27:24), E_a(23:16), E_a(15:8), E_a(7:0)	0x00, 0x00, 0x00, 0x00
0x1A	Mix E to b	1	u(31:28)E_b(27:24), E_b(23:16), E_b(15:8), E_b(7:0)	0x00, 0x00, 0x00, 0x00
0x1B	Mix E to c	1	u(31:28)E_c(27:24), E_c(23:16), E_c(15:8), E_c(7:0)	0x00, 0x00, 0x00, 0x00
0x1C	Mix E to d	1	u(31:28)E_d(27:24), E_d(23:16), E_d(15:8), E_d(7:0)	0x00, 0x00, 0x00, 0x00
0x1D	Mix E to e	1	u(31:28)E_e(27:24), E_e(23:16), E_e(15:8), E_e(7:0)	0x00, 0x00, 0x00, 0x00
0x1E	Mix E to f	1	u(31:28)E_f(27:24), E_f(23:16), E_f(15:8), E_f(7:0)	0x00, 0x00, 0x00, 0x00
0x1F	Mix F to a	1	u(31:28)F_a(27:24), F_a(23:16), F_a(15:8), F_a(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x20	Mix F to b	1	u(31:28)F_b(27:24), F_b(23:16), F_b(15:8), F_b(7:0)	0x00, 0x00, 0x00, 0x00
0x21	Mix F to c	1	u(31:28)F_c(27:24), F_c(23:16), F_c(15:8), F_c(7:0)	0x00, 0x00, 0x00, 0x00
0x22	Mix F to d	1	u(31:28)F_d(27:24), F_d(23:16), F_d(15:8), F_d(7:0)	0x00, 0x00, 0x00, 0x00
0x23	Mix F to e	1	u(31:28)F_e(27:24), F_e(23:16), F_e(15:8), F_e(7:0)	0x00, 0x00, 0x00, 0x00
0x24	Mix F to f	1	u(31:28)F_f(27:24), F_f(23:16), F_f(15:8), F_f(7:0)	0x00, 0x00, 0x00, 0x00
0x25	Mix a to c	1	u(31:28)a_c(27:24), a_c(23:16), a_c(15:8), a_c(7:0)	0x00, 0x00, 0x00, 0x00
0x26	Mix b to c	1	u(31:28)b_c(27:24), b_c(23:16), b_c(15:8), b_c(7:0)	0x00, 0x00, 0x00, 0x00
0x27	Mix a to g	1	u(31:28)a_g(27:24), a_g(23:16), a_g(15:8), a_g(7:0)	0x00, 0x00, 0x00, 0x00
0x28	Mix b to h	1	u(31:28)b_h(27:24), b_h(23:16), b_h(15:8), b_h(7:0)	0x00, 0x00, 0x00, 0x00
0x29	Mix a to d via BQ and Rev/D	1	u(31:28)a_d(27:24), a_d(23:16), a_d(15:8), a_d(7:0)	0x00, 0x80, 0x00, 0x00
0x2A	Mix a to e via BQ and Rev/D	1	u(31:28)a_e(27:24), a_e(23:16), a_e(15:8), a_e(7:0)	0x00, 0x00, 0x00, 0x00
0x2B	Mix b to d via BQ and Rev/D	1	u(31:28)b_d(27:24), b_d(23:16), b_d(15:8), b_d(7:0)	0x00, 0x00, 0x00, 0x00
0x2C	Mix b to e via BQ and Rev/D	1	u(31:28)b_e(27:24), b_e(23:16), b_e(15:8), b_e(7:0)	0x00, 0x80, 0x00, 0x00
0x2D	Mix g to d via BQ	1	u(31:28)g_d(27:24), g_d(23:16), g_d(15:8), g_d(7:0)	0x00, 0x00, 0x00, 0x00
0x2E	Mix g to e via BQ	1	u(31:28)g_e(27:24), g_e(23:16), g_e(15:8), g_e(7:0)	0x00, 0x00, 0x00, 0x00
0x2F	Mix h to d via BQ	1	u(31:28)h_d(27:24), h_d(23:16), h_d(15:8), h_d(7:0)	0x00, 0x00, 0x00, 0x00
0x30	Mix h to e via BQ	1	u(31:28)h_e(27:24), h_e(23:16), h_e(15:8), h_e(7:0)	0x00, 0x00, 0x00, 0x00
0x31	Mix c to d via BQ	1	u(31:28)c_d(27:24), c_d(23:16), c_d(15:8), c_d(7:0)	0x00, 0x00, 0x00, 0x00
0x32	Mix c to e via BQ	1	u(31:28)c_e(27:24), c_e(23:16), c_e(15:8), c_e(7:0)	0x00, 0x00, 0x00, 0x00
0x33	Mix f to g and h	1	u(31:28)f_gh(27:24), f_gh(23:16), f_gh(15:8), f_gh(7:0)	0x00, 0x00, 0x00, 0x00
0x34	a_de path, biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x35	a_de path, biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x36	a_de path, biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x37	a_de path, biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x38	b_de path, biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x39	b_de path, biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x3A	b_de path, biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x3B	b_de path, biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x3C	g_de path, biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x3D	g_de path, biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x3E	g_de path, biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x3F	g_de path, biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x40	h_de path, biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x41	h_de path, biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x42	h_de path, biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x43	h_de path, biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x44	c_de path, biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x45	c_de path, biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x46	c_de path, biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x47	c_de path, biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x48	f_CH3 path, biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x49	f_CH3 path, biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x4A	f_CH3 path, biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x4B	f_CH3 path, biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x4C	a_de path, reverberation (reverb) gain Rg0	2	u(31:28)Rg0(27:24), Rg0(23:16), Rg0(15:8), Rg0(7:0)	0x00, 0x80, 0x00, 0x00
	a_de path, reverb gain Rg1		u(31:28)Rg1(27:24), Rg1(23:16), Rg1(15:8), Rg1(7:0)	0x00, 0x00, 0x00, 0x00
0x4D	b_de path, reverb gain Rg0	2	u(31:28)Rg0(27:24), Rg0(23:16), Rg0(15:8), Rg0(7:0)	0x00, 0x80, 0x00, 0x00
	b_de path, reverb gain Rg1		u(31:28)Rg1(27:24), Rg1(23:16), Rg1(15:8), Rg1(7:0)	0x00, 0x00, 0x00, 0x00
0x4E	f_CH3 path, reverb gain Rg0	2	u(31:28)Rg0(27:24), Rg0(23:16), Rg0(15:8), Rg0(7:0)	0x00, 0x80, 0x00, 0x00
	f_CH3 path, reverb gain Rg1		u(31:28)Rg1(27:24), Rg1(23:16), Rg1(15:8), Rg1(7:0)	0x00, 0x00, 0x00, 0x00
0x4F	CH1 biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x50	CH1 biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x51	CH1 biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x52	CH1 biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x53	CH1 biquad 5	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x54	CH1 biquad 6	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x55	CH1 biquad 7	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x56	CH1 biquad 8	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x57	CH1 biquad 9	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x58	CH1 biquad 10	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x59	CH1 biquad 11	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x5A	CH1 biquad 12	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x5B	CH2 biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x5C	CH2 biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x5D	CH2 biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x5E	CH2 biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x5F	CH2 biquad 5	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x60	CH2 biquad 6	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x61	CH2 biquad 7	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x62	CH2 biquad 8	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x63	CH2 biquad 9	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x64	CH2 biquad 10	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x65	CH2 biquad 11	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x66	CH2 biquad 12	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x67	CH3 biquad 1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x68	CH3 biquad 2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x69	CH3 biquad 3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x6A	CH3 biquad 4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x6B	CH3 biquad 5	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x6C	CH3 biquad 6	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x6D	CH3 biquad 7	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x6E	CH3 biquad 8	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x6F	CH3 biquad 9	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x70	CH3 biquad 10	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x71	CH3 biquad 11	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x72	CH3 biquad 12	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0x73	Bass and treble bypass 1	2	u(31:28)BTby1(27:24), BTby1(23:16), BTby1(15:8), BTby1(7:0)	0x00, 0x80, 0x00, 0x00
	Bass and treble inline 1		u(31:28)BT1(27:24), BT1(23:16), BT1(15:8), BT1(7:0)	0x00, 0x00, 0x00, 0x00
0x74	Bass and treble bypass 2	2	u(31:28)BTby2(27:24), BTby2(23:16), BTby2(15:8), BTby2(7:0)	0x00, 0x80, 0x00, 0x00
	Bass and treble inline 2		u(31:28)BT2(27:24), BT2(23:16), BT2(15:8), BT2(7:0)	0x00, 0x00, 0x00, 0x00
0x75	Bass and treble bypass 3	2	u(31:28)BTby3(27:24), BTby3(23:16), BTby3(15:8), BTby3(7:0)	0x00, 0x80, 0x00, 0x00
	Bass and treble inline 3		u(31:28)BT3(27:24), BT3(23:16), BT3(15:8), BT3(7:0)	0x00, 0x00, 0x00, 0x00
0x76	Mix u to i	1	u(31:28)u_i(27:24), u_i(23:16), u_i(15:8), u_i(7:0)	0x00, 0x00, 0x00, 0x00
0x77	Mix v to k	1	u(31:28)v_k(27:24), v_k(23:16), v_k(15:8), v_k(7:0)	0x00, 0x00, 0x00, 0x00
0x78	Mix w to m	1	u(31:28)w_m(27:24), w_m(23:16), w_m(15:8), w_m(7:0)	0x00, 0x00, 0x00, 0x00
0x79	Mix j to i	1	u(31:28)j_i(27:24), j_i(23:16), j_i(15:8), j_i(7:0)	0x00, 0x00, 0x00, 0x00
0x7A	Mix l to k	1	u(31:28)l_k(27:24), l_k(23:16), l_k(15:8), l_k(7:0)	0x00, 0x00, 0x00, 0x00
0x7B	Mix n to m	1	u(31:28)n_m(27:24), n_m(23:16), n_m(15:8), n_m(7:0)	0x00, 0x00, 0x00, 0x00
0x7C	Mix j to o via DRC mult	2	u(31:28)j_o(27:24), j_o(23:16), j_o(15:8), j_o(7:0)	0x00, 0x00, 0x00, 0x00
	DRC bypass 1		u(31:28)DRC_{by1}(27:24), DRC_{by1}(23:16), DRC_{by1}(15:8), DRC_{by1}(7:0)	0x00, 0x80, 0x00, 0x00
0x7D	Mix l to p via DRC mult	2	u(31:28)l_p(27:24), l_p(23:16), l_p(15:8), l_p(7:0)	0x00, 0x00, 0x00, 0x00
	DRC bypass 2		u(31:28)DRC_{by2}(27:24), DRC_{by2}(23:16), DRC_{by2}(15:8), DRC_{by2}(7:0)	0x00, 0x80, 0x00, 0x00
0x7E	Mix n to q via DRC mult	2	u(31:28)n_q(27:24), n_q(23:16), n_q(15:8), n_q(7:0)	0x00, 0x00, 0x00, 0x00
	DRC bypass 3		u(31:28)DRC_{by3}(27:24), DRC_{by3}(23:16), DRC_{by3}(15:8), DRC_{by3}(7:0)	0x00, 0x80, 0x00, 0x00
0x7F	Mix dither1 to o	1	u(31:28)Dth1_o(27:24), Dth1_o(23:16), Dth1_o(15:8), Dth1_o(7:0)	0x00, 0x00, 0x00, 0x00
0x80	Mix dither2 to p	1	u(31:28)Dth2_p(27:24), Dth2_p(23:16), Dth2_p(15:8), Dth2_p(7:0)	0x00, 0x00, 0x00, 0x00
0x81	Mix dither3 to q	1	u(31:28)Dth3_q(27:24), Dth3_q(23:16), Dth3_q(15:8), Dth3_q(7:0)	0x00, 0x00, 0x00, 0x00
0x82	Mix delay3 to o	1	u(31:28)Dth3_o(27:24), Dth3_o(23:16), Dth3_o(15:8), Dth3_o(7:0)	0x00, 0x00, 0x00, 0x00
0x83	Mix delay3 to p	1	u(31:28)Dth3_p(27:24), Dth3_p(23:16), Dth3_p(15:8), Dth3_p(7:0)	0x00, 0x00, 0x00, 0x00
0x84	Mix o to r	1	u(31:28)o_r(27:24), o_r(23:16), o_r(15:8), o_r(7:0)	0x00, 0x40, 0x00, 0x00
0x85	Mix o to s	1	u(31:28)o_s(27:24), o_s(23:16), o_s(15:8), o_s(7:0)	0x00, 0x00, 0x00, 0x00
0x86	Mix p to r	1	u(31:28)p_r(27:24), p_r(23:16), p_r(15:8), p_r(7:0)	0x00, 0x40, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0x87	Mix p to t	1	u(31:28)p_t(27:24), p_t(23:16), p_t(15:8), p_t(7:0)	0x00, 0x00, 0x00, 0x00
0x88	Mix q to r	1	u(31:28)q_r(27:24), q_r(23:16), q_r(15:8), q_r(7:0)	0x00, 0x00, 0x00, 0x00
0x89	Mix r to s and t	1	u(31:28)r_st(27:24), r_st(23:16), r_st(15:8), r_st(7:0)	0x00, 0x80, 0x00, 0x00
0x8A	Mix z to Z	1	u(31:28)z_Z(27:24), z_Z(23:16), z_Z(15:8), z_Z(7:0)	0x00, 0x00, 0x00, 0x00
0x8B	Mix z to Y	1	u(31:28)z_Y(27:24), z_Y(23:16), z_Y(15:8), z_Y(7:0)	0x00, 0x00, 0x00, 0x00
0x8C	Mix z to X	1	u(31:28)z_X(27:24), z_X(23:16), z_X(15:8), z_X(7:0)	0x00, 0x00, 0x00, 0x00
0x8D	Mix z to W	1	u(31:28)z_W(27:24), z_W(23:16), z_W(15:8), z_W(7:0)	0x00, 0x00, 0x00, 0x00
0x8E	Mix z to V	1	u(31:28)z_V(27:24), z_V(23:16), z_V(15:8), z_V(7:0)	0x00, 0x00, 0x00, 0x00
0x8F	Mix z to U	1	u(31:28)z_U(27:24), z_U(23:16), z_U(15:8), z_U(7:0)	0x00, 0x80, 0x00, 0x00
0x90	Mix y to Z	1	u(31:28)y_Z(27:24), y_Z(23:16), y_Z(15:8), y_Z(7:0)	0x00, 0x00, 0x00, 0x00
0x91	Mix y to Y	1	u(31:28)y_Y(27:24), y_Y(23:16), y_Y(15:8), y_Y(7:0)	0x00, 0x00, 0x00, 0x00
0x92	Mix y to X	1	u(31:28)y_X(27:24), y_X(23:16), y_X(15:8), y_X(7:0)	0x00, 0x00, 0x00, 0x00
0x93	Mix y to W	1	u(31:28)y_W(27:24),y_W(23:16), y_W(15:8), y_W(7:0)	0x00, 0x00, 0x00, 0x00
0x94	Mix y to V	1	u(31:28)y_V(27:24), y_V(23:16), y_V(15:8), y_V(7:0)	0x00, 0x80, 0x00, 0x00
0x95	Mix y to U	1	u(31:28)y_U(27:24), y_U(23:16), y_U(15:8), y_U(7:0)	0x00, 0x00, 0x00, 0x00
0x96	Mix x to Z	1	u(31:28)x_Z(27:24), x_Z(23:16), x_Z(15:8), x_Z(7:0)	0x00, 0x00, 0x00, 0x00
0x97	Mix x to Y	1	u(31:28)x_Y(27:24), x_Y(23:16), x_Y(15:8), x_Y(7:0)	0x00, 0x00, 0x00, 0x00
0x98	Mix x to X	1	u(31:28)x_X(27:24), x_X(23:16), x_X(15:8), x_X(7:0)	0x00, 0x00, 0x00, 0x00
0x99	Mix x to W	1	u(31:28)x_W(27:24), x_W(23:16), x_W(15:8), x_W(7:0)	0x00, 0x80, 0x00, 0x00
0x9A	Mix x to V	1	u(31:28)x_V(27:24), x_V(23:16), x_V(15:8), x_V(7:0)	0x00, 0x00, 0x00, 0x00
0x9B	Mix x to U	1	u(31:28)x_U(27:24), x_U(23:16), x_U(15:8), x_U(7:0)	0x00, 0x00, 0x00, 0x00
0x9C	Mix r to Z	1	u(31:28)r_Z(27:24), r_Z(23:16), r_Z(15:8), r_Z(7:0)	0x00, 0x00, 0x00, 0x00
0x9D	Mix r to Y	1	u(31:28)r_Y(27:24), r_Y(23:16), r_Y(15:8), r_Y(7:0)	0x00, 0x00, 0x00, 0x00
0x9E	Mix r to X	1	u(31:28)r_X(27:24), r_X(23:16), r_X(15:8), r_X(7:0)	0x00, 0x80, 0x00, 0x00
0x9F	Mix r to W	1	u(31:28)r_W(27:24), r_W(23:16), r_W(15:8), r_W(7:0)	0x00, 0x00, 0x00, 0x00
0xA0	Mix r to V	1	u(31:28)r_V(27:24), r_V(23:16), r_V(15:8), r_V(7:0)	0x00, 0x00, 0x00, 0x00
0xA1	Mix r to U	1	u(31:28)r_U(27:24), r_U(23:16), r_U(15:8), r_U(7:0)	0x00, 0x00, 0x00, 0x00
0xA2	CH1 loudness $\log_2$ G	1	u(31:28)LG(27:24), LG(23:16), LG(15:8), LG(7:0)	0x00, 0x40, 0x00, 0x00
0xA3	CH1 loudness $\log_2$ O	2	u(31:24), u(23:16), $LO_{47:40}$(15:8), $LO_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$LO_{31:24}$(31:24), $LO_{23:16}$(23:16), $LO_{15:8}$(15:8), $LO_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x00
0xA4	CH1 loudness G	1	u(31:28)G(27:24), G(23:16), G(15:8), G(7:0)	0x00, 0x00, 0x00, 0x00
0xA5	CH1 loudness O	2	u(31:24), u(23:16), $O_{47:40}$(15:8), $O_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O_{31:24}$(31:24), $O_{23:16}$(23:16), $O_{15:8}$(15:8), $O_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0xA6	CH1 loudness biquad	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xA7	CH2 loudness $\log_2$ G	1	u(31:28)LG(27:24), LG(23:16), LG(15:8), LG(7:0)	0x00, 0x40, 0x00, 0x00
0xA8	CH2 loudness $\log_2$ O	2	u(31:24), u(23:16), $LO_{47:40}$(15:8), $LO_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$LO_{31:24}$(31:24), $LO_{23:16}$(23:16), $LO_{15:8}$(15:8), $LO_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x00
0xA9	CH2 loudness G	1	u(31:28)G(27:24), G(23:16), G(15:8), G(7:0)	0x00, 0x00, 0x00, 0x00
0xAA	CH2 loudness O	2	u(31:24), u(23:16), $O_{47:40}$(15:8), $O_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O_{31:24}$(31:24), $O_{23:16}$(23:16), $O_{15:8}$(15:8), $O_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x00
0xAB	CH2 loudness biquad	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xAC	CH3 loudness $\log_2$ G	1	u(31:28)LG(27:24), LG(23:16), LG(15:8), LG(7:0)	0x00, 0x40, 0x00, 0x00
0xAD	CH3 loudness $\log_2$ O	2	u(31:24), u(23:16), $LO_{47:40}$(15:8), $LO_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$LO_{31:24}$(31:24), $LO_{23:16}$(23:16), $LO_{15:8}$(15:8), $LO_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x00
0xAE	CH3 loudness G	1	u(31:28)G(27:24), G(23:16), G(15:8), G(7:0)	0x00, 0x00, 0x00, 0x00
0xAF	CH3 loudness O	2	u(31:24), u(23:16), $O_{47:40}$(15:8), $O_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O_{31:24}$(31:24), $O_{23:16}$(23:16), $O_{15:8}$(15:8), $O_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x00
0xB0	CH3 loudness biquad	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xB1	CH1/CH2 DRCE ae	2	u(31:28)ae(27:24), ae(23:16), ae(15:8), ae(7:0)	0x00, 0x80, 0x00, 0x00
	CH1/CH2 DRCE 1-ae		u(31:28)1-ae(27:24), 1-ae(23:16), 1-ae(15:8), 1-ae(7:0)	0x00, 0x00, 0x00, 0x00
0xB2	CH1/CH2 DRCE T1	4	u(31:24), u(23:16), $T1_{47:40}$(15:8), $T1_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$T1_{31:24}$(31:24), $T1_{23:16}$(23:16), $T1_{15:8}$(15:8), $T1_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x01
	CH1/CH2 DRCE T2		u(31:24), u(23:16), $T2_{47:40}$(15:8), $T2_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$T2_{31:24}$(31:24), $T2_{23:16}$(23:16), $T2_{15:8}$(15:8), $T2_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x01
0xB3	CH1/CH2 k0’	3	u(31:28)k0’(27:24), k0’(23:16), k0’(15:8), k0’(7:0)	0x00, 0x00, 0x00, 0x00
	CH1/CH2 k1’		u(31:28)k1’(27:24), k1’(23:16), k1’(15:8), k1’(7:0)	0x00, 0x00, 0x00, 0x00
	CH1/CH2 k2’		u(31:28)k2’(27:24), k2’(23:16), k2’(15:8), k2’(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0xB4	CH1/CH2 DRCE O1	4	u(31:24), u(23:16), $O1_{47:40}$(15:8), $O1_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O1_{31:24}$(31:24), $O1_{23:16}$(23:16), $O1_{15:8}$(15:8), $O1_{7:0}$(7:0)	0x01, 0xFF, 0xFF, 0xFF
	CH1/CH2 DRCE O2		u(31:24), u(23:16), $O2_{47:40}$(15:8), $O2_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O2_{31:24}$(31:24), $O2_{23:16}$(23:16), $O2_{15:8}$(15:8), $O2_{7:0}$(7:0)	0x01, 0xFF, 0xFF, 0xFF
0xB5	CH1/CH2 DRCE aa	4	u(31:28)aa(27:24), aa(23:16), aa(15:8), aa(7:0)	0x00, 0x80, 0x00, 0x00
	CH1/CH2 DRCE 1-aa		u(31:28)1-aa(27:24), 1-aa(23:16), 1-aa(15:8), 1-aa(7:0)	0x00, 0x00, 0x00, 0x00
	CH1/CH2 DRCE ad		u(31:28)ad(27:24), ad(23:16), ad(15:8), ad(7:0)	0x00, 0x80, 0x00, 0x00
	CH1/CH2 DRCE 1-ad		u(31:28)1-ad(27:24), 1-ad(23:16), 1-ad(15:8), 1-ad(7:0)	0x00, 0x00, 0x00, 0x00
0xB6	CH3 DRCE ae	2	u(31:28)ae(27:24), ae(23:16), ae(15:8), ae(7:0)	0x00, 0x80, 0x00, 0x00
	CH3 DRCE 1-ae		u(31:28)1-ae(27:24), 1-ae(23:16), 1-ae(15:8), 1-ae(7:0)	0x00, 0x00, 0x00, 0x00
0xB7	CH3 DRCE T1	4	u(31:24), u(23:16), $T1_{47:40}$(15:8), $T1_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$T1_{31:24}$(31:24), $T1_{23:16}$(23:16), $T1_{15:8}$(15:8), $T1_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x01
	CH3 DRCE T2		u(31:24), u(23:16), $T2_{47:40}$(15:8), $T2_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$T2_{31:24}$(31:24), $T2_{23:16}$(23:16), $T2_{15:8}$(15:8), $T2_{7:0}$(7:0)	0x00, 0x00, 0x00, 0x01
0xB8	CH3 k0’	3	u(31:28)k0’(27:24), k0’(23:16), k0’(15:8), k0’(7:0)	0x00, 0x00, 0x00, 0x00
	CH3 k1’		u(31:28)k1’(27:24), k1’(23:16), k1’(15:8), k1’(7:0)	0x00, 0x00, 0x00, 0x00
	CH3 k2’		u(31:28)k2’(27:24), k2’(23:16), k2’(15:8), k2’(7:0)	0x00, 0x00, 0x00, 0x00
0xB9	CH3 DRCE O1	4	u(31:24), u(23:16), $O1_{47:40}$(15:8), $O1_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O1_{31:24}$(31:24), $O1_{23:16}$(23:16), $O1_{15:8}$(15:8), $O1_{7:0}$(7:0)	0x01, 0xFF, 0xFF, 0xFF
	CH3 DRCE O2		u(31:24), u(23:16), $O2_{47:40}$(15:8), $O2_{39:32}$(7:0)	0x00, 0x00, 0x00, 0x00
			$O2_{31:24}$(31:24), $O2_{23:16}$(23:16), $O2_{15:8}$(15:8), $O2_{7:0}$(7:0)	0x01, 0xFF, 0xFF, 0xFF
0xBA	CH3 DRCE aa	4	u(31:28)aa(27:24), aa(23:16), aa(15:8), aa(7:0)	0x00, 0x80, 0x00, 0x00
	CH3 DRCE 1-aa		u(31:28)1-aa(27:24), 1-aa(23:16), 1-aa(15:8), 1-aa(7:0)	0x00, 0x00, 0x00, 0x00
	CH3 DRCE ad		u(31:28)ad(27:24), ad(23:16), ad(15:8), ad(7:0)	0x00, 0x80, 0x00, 0x00
	CH3 DRCE 1-ad		u(31:28)1-ad(27:24), 1-ad(23:16), 1-ad(15:8), 1-ad(7:0)	0x00, 0x00, 0x00, 0x00
0xBB	Spectrum analyzer asa	2	u(31:28)asa(27:24), asa(23:16), asa(15:8), asa(7:0)	0x00, 0x80, 0x00, 0x00
	Spectrum analyzer 1-asa		u(31:28)1-asa(27:24), 1-asa(23:16), 1-asa(15:8), 1-asa(7:0)	0x00, 0x00, 0x00, 0x00
0xBC	Spectrum analyzer BQ1	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0xBD	Spectrum analyzer BQ2	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xBE	Spectrum analyzer BQ3	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xBF	Spectrum analyzer BQ4	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xC0	Spectrum analyzer BQ5	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xC1	Spectrum analyzer BQ6	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xC2	Spectrum analyzer BQ7	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xC3	Spectrum analyzer BQ8	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0xC4	Spectrum analyzer BQ9	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xC5	Spectrum analyzer BQ10	5	u(31:28)a1(27:24), a1(23:16), a1(15:8), a1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)a2(27:24), a2(23:16), a2(15:8), a2(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b0(27:24), b0 (23:16), b0(15:8), b0(7:0)	0x00, 0x80, 0x00, 0x00
			u(31:28)b1(27:24), b1(23:16), b1(15:8), b1(7:0)	0x00, 0x00, 0x00, 0x00
			u(31:28)b2(27:24), b2(23:16), b2(15:8), b2(7:0)	0x00, 0x00, 0x00, 0x00
0xC6	Dither LFSR1 mix	2	u(31:28)LFSR1(27:24), LFSR1(23:16), LFSR1(15:8), LFSR1(7:0)	0x00, 0x80, 0x00, 0x00
	Dither LFSR2 mix		u(31:28)LFSR2(27:24), LFSR2(23:16), LFSR2(15:8), LFSR2(7:0)	0x00, 0x80, 0x00, 0x00
0xC7	Dither seed	1	u(31:24), u(23:16), LFSR2_SEED(15:8), LFSR1_SEED(7:0)	0x00, 0x00, 0x22, 0x49
0xC8	Factory test	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xC9	Factory test	2	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
			u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xCA	Mix G to g	1	u(31:28)G_g(27:24), G_g(23:16), G_g(15:8), G_g(7:0)	0x00, 0x00, 0x00, 0x00
0xCB	Mix G to f	1	u(31:28)G_f(27:24), G_f(23:16), G_f(15:8), G_f(7:0)	0x00, 0x00, 0x00, 0x00
0xCC	Mix G to Y	1	u(31:28)G_Y(27:24), G_Y(23:16), G_Y(15:8), G_Y(7:0)	0x00, 0x00, 0x00, 0x00
0xCD	Mix H to h	1	u(31:28)H_h(27:24), H_h(23:16), H_h(15:8), H_h(7:0)	0x00, 0x00, 0x00, 0x00
0xCE	Mix H to f	1	u(31:28)H_f(27:24), H_f(23:16), H_f(15:8), H_f(7:0)	0x00, 0x00, 0x00, 0x00
0xCF	Mix H to Z	1	u(31:28)H_Z(27:24), H_Z(23:16), H_Z(15:8), H_Z(7:0)	0x00, 0x00, 0x00, 0x00
0xD0	Mix d to aa	1	u(31:28)d_aa(27:24), d_aa(23:16), d_aa(15:8), d_aa(7:0)	0x00, 0x00, 0x00, 0x00
0xD1	Mix e to aa	1	u(31:28)e_aa(27:24), e_aa(23:16), e_aa(15:8), e_aa(7:0)	0x00, 0x00, 0x00, 0x00
0xD2	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD3	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD4	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD5	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD6	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD7	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD8	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xD9	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xDA	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xDB	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xDC	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xDD	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A

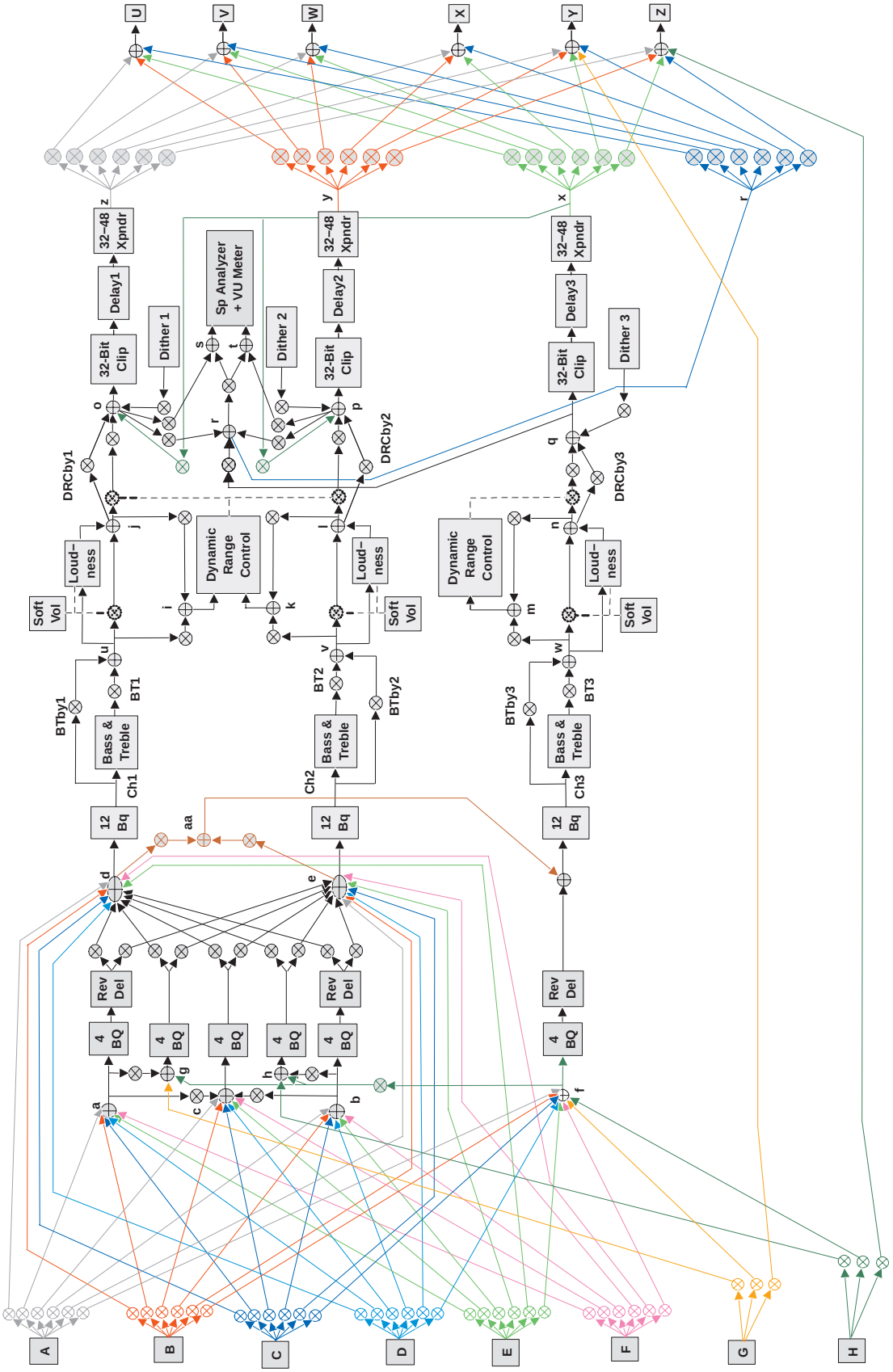
SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0xDE	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xDF	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE0	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE1	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE2	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE3	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE4	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE5	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE6	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE7	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE8	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xE9	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xEA	Reserved (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xEB	Watchdog timer enable	1	u(31:24), u(23:16), u(15:8), u(7:1)R1(0)	0x00, 0x00, 0x00, 0x01
0xEC	Factory test (2)	1	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xED	Factory test (2)	2	u(31:24), u(23:16), u(15:8), u(7:0)	N/A
			u(31:24), u(23:16), u(15:8), u(7:0)	N/A
0xEE	GPIO port I/O value	1	u(31:24), u(23:16), u(15:8), u(7:4)GPIO_in_out(3:0)	0x00, 0x00, 0x00, 0x0X(1)
0xEF	GPIO parameters	1	u(31:24), u(23:20)GPIODIR(19:16), GPIOFSCOUNT(15:8), GPIO_samp_int(7:0)	0x00, 0x0F, 0x6E, 0x6D
0xF0	Master mute/unmute	1	u(31:24), u(23:16), u(15:8), u(7:3)CH3M_U(2)CH2M_U(1)CH1M_U(0)	0x00, 0x00, 0x00, 0x00
0xF1	Vol, T and B slew rates	1	u(31:24), u(23:16), u(15:9)VSC(8), TBLC(7:0)	0x00, 0x00, 0x00, 0x40
0xF2	CH1 volume (5.23 precision)	1	u(31:28)Vol1(27:24), Vol1(23:16), Vol1(15:8), Vol1(7:0)	0x00, 0x00, 0x00, 0x00
0xF3	CH2 volume (5.23 precision)	1	u(31:28)Vol2(27:24), Vol2(23:16), Vol2(15:8), Vol2(7:0)	0x00, 0x00, 0x00, 0x00
0xF4	CH3 volume (5.23 precision)	1	u(31:28)Vol3(27:24), Vol3(23:16), Vol3(15:8), Vol3(7:0)	0x00, 0x00, 0x00, 0x00
0xF5	Bass filter set (1-5)	1	u(31:24), u(23:19)CH3Bs(18:16), u(15:11) CH2Bs(10:8), u(7:3)CH1Bs(2:0),	0x00, 0x04, 0x04, 0x04
0xF6	Bass filter index	1	u(31:24), CH3Bf(23:16), CH2Bf(15:8), CH1Bf(7:0)	0x00, 0x72, 0x72, 0x72
0xF7	Treble filter set (1-5)	1	u(31:24), u(23:19)CH3Ts(18:16), u(15:11) CH2Ts(10:8), u(7:3)CH1Ts(2:0),	0x00, 0x04, 0x04, 0x04
0xF8	Treble filter index	1	u(31:24), CH3Tf(23:16), CH2Tf(15:8), CH1Tf(7:0)	0x00, 0x72, 0x72, 0x72
0xF9	I^2S command word	1	MLRCLK(31:24), SCLK(23:16), DWFMT(15:8), IOM(7:0)	0x01, 0x01, 0x09, 0x11
0xFA	Delay/reverb times−CH1	3	u(31:28)D1(27:24), D1(23:16), u(15:12)R1(11:8), R1(7:0)	0x00, 0x00, 0x00, 0x00
	Delay/reverb times−CH2		u(31:28)D2(27:24), D2(23:16), u(15:12)R2(11:8), R2(7:0)	0x00, 0x00, 0x00, 0x00
	Delay/reverb times−CH3		u(31:28)D3(27:24), D3(23:16), u(15:12)R3(11:8), R3(7:0)	0x00, 0x00, 0x00, 0x00
0xFB	I^2C M and N	1	u(31:24), u(23:16), u(15:8), u(7)M(6:3)N(2:0)	0x00, 0x00, 0x00, 0x41
0xFC	Ending I^2C check word	1	ECW(31:24), ECW(23:16), ECW(15:8), ECW(7:0)	0x81, 0x42, 0x24, 0x18

SUBADDRESS (0xSS)	REGISTER NAME	NUMBER OF 4-BYTE WORDS	CONTENTS (u Indicates Unused Bits)	INITIALIZATION VALUE
0xFD	Spectrum analyzer output 1	2.5	SA1(7:0)	(Always data dependent)
	Spectrum analyzer output 2		SA2(7:0)	(Always data dependent)
	Spectrum analyzer output 3		SA3(7:0)	(Always data dependent)
	Spectrum analyzer output 4		SA4(7:0)	(Always data dependent)
	Spectrum analyzer output 5		SA5(7:0)	(Always data dependent)
	Spectrum analyzer output 6		SA6(7:0)	(Always data dependent)
	Spectrum analyzer output 7		SA7(7:0)	(Always data dependent)
	Spectrum analyzer output 8		SA8(7:0)	(Always data dependent)
	Spectrum analyzer output 9		SA9(7:0)	(Always data dependent)
	Spectrum analyzer output 10		SA10(7:0)	(Always data dependent)
0xFE	VU meter output 1 (SA5)	0.5	SA5(7:0)	(Always data dependent)
	VU meter output 2 (SA6)		SA6(7:0)	(Always data dependent)
0xFF	Flag register	0.25	u(7:1)VolBusy(0), 1 = busy	N/A

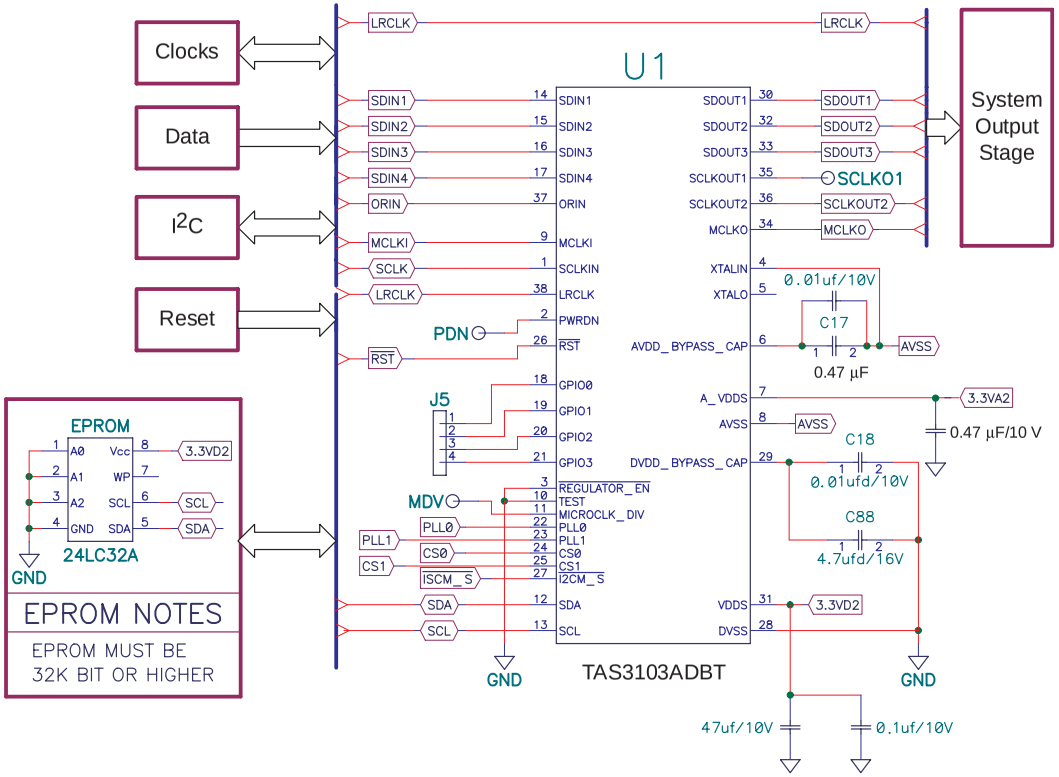
NOTE 1: GPIO ports are initialized to be read ports. The initial input values read then are dependent on what is connected to the GPIO pins. If a given GPIO pin is left unconnected, the internal pullup results in a logic 1 being read.

NOTE 2: Do not write to reserved of factory-test subaddresses.

A.2 TAS3103A Firmware Block Diagram



A.3 TAS3103A Simplified Application Schematic Diagram



NOTE: 0.01-µF capacitors must be placed as close as possible to the device pins. All other capacitors should be placed after the 0.01-µF so that the distance between the capacitors and the device pins is mini-mized.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)
TAS3103ADBT	ACTIVE	TSSOP	DBT	38	50	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
TAS3103ADBTG4	ACTIVE	TSSOP	DBT	38	50	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
TAS3103ADBTR	ACTIVE	TSSOP	DBT	38	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
TAS3103ADBTRG4	ACTIVE	TSSOP	DBT	38	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.