

## APPLICATION NOTE

### STEREO PCM APPLICATION USER'S GUIDE FOR THE CS492X FAMILY

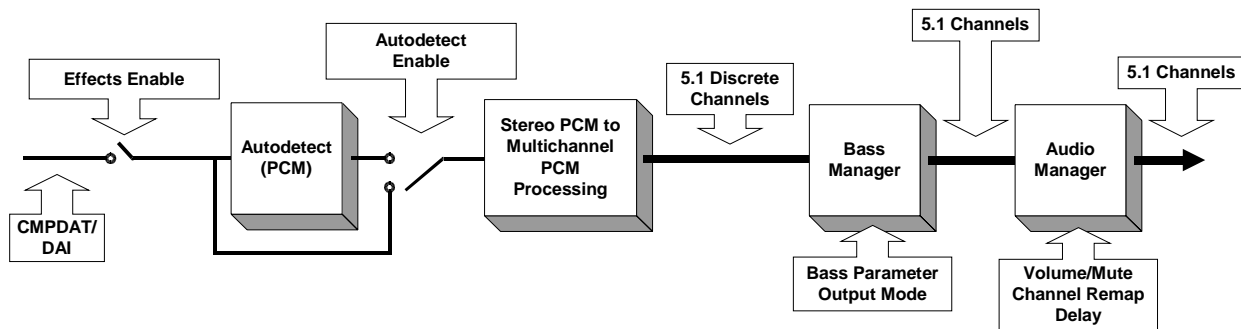
#### Contents

- Crystal Original Surround Description
- Software Naming Convention
- Stereo PCM Application Description
- Hardware Configurations Supported by the CS492X and the PCM Application Code
- Understanding Application Messaging (Including Write Sessions, Solicited and Unsolicited Read Message)
- How to Control Standard Application Modules such as:
  - Audio Manager
  - Generalized Bass Manager
- How to Control the Crystal Original Surround Application Modules such as:
  - Crystal Original Surround Effects Manager
  - Crystal Original Surround Effects PCM Processor
- How to Use the Applications: COEFMAKR.EXE and EFFECTS.EXE

#### Description

This document includes a brief description of hardware configuration and in depth descriptions of application messaging protocol, application control modules, and application configuration examples. The main body of this document covers all the commonly used features of each stereo PCM application.

Additional application notes that cover proprietary Stereo PCM processing such as: SRS CircleSurround™, Logic 7™, and Pacific Microsonics™ HDCD® are available as an appendix to this application note for approved licensees of each corresponding technology.



---

**TABLE OF CONTENTS**

<b>1. OVERVIEW</b> .....	<b>4</b>
1.1 Multichannel Decoder Family of Parts .....	4
1.2 Document Strategy .....	5
1.2.1 Hardware Documentation .....	5
1.3 Using the CS492X .....	6
1.4 Software Naming Convention .....	6
1.5 Stereo PCM Application Description .....	6
<b>2. HARDWARE CONFIGURATION</b> .....	<b>7</b>
2.1 Supported Input/Output Modes .....	7
<b>3. APPLICATION MESSAGING</b> .....	<b>8</b>
3.1 Write Session .....	8
3.2 Solicited Read Message Format .....	9
3.3 Unsolicited Read Message Format .....	10
<b>4. APPLICATION MODULES</b> .....	<b>11</b>
4.1 Audio Manager .....	12
4.1.1 Audio_Control: (Index 0x00) .....	14
4.2 Generalized Bass Manager .....	16
4.3 Unsolicited Messages (Read-Only) .....	22
4.3.1 Autodetect Operation .....	23
<b>5. APPENDIX A - CRYSTAL ORIGINAL SURROUND (C.O.S.) EFFECTS CODE</b> .....	<b>27</b>
5.1 Crystal Original Surround Description .....	27
5.2 Standard Application Modules .....	27
5.3 Crystal Original Surround Effects Manager .....	28
5.3.1 Effects_Mgr_Control .....	30
5.4 Crystal Original Surround Effects PCM Processor .....	31
5.4.1 PCM_Control .....	33
5.4.2 Basic Crystal Surround Effects Explanation .....	33
5.4.3 Creating Custom Effects Using Advanced Settings .....	37
5.4.3.1 FIR Section: Early Reflections .....	37
5.4.3.2 Comb Filter Section: After-reflections .....	38
5.4.3.3 All-Pass Filter Section: More After-reflections and Decorrelation .....	38
5.4.4 Crystal Original Surround Effects Modes .....	38

---

**Contacting Cirrus Logic Support**

For a complete listing of Direct Sales, Distributor, and Sales Representative contacts, visit the Cirrus Logic web site at:

<http://www.cirrus.com/corporate/contacts/>

---

SRS CircleSurround is a trademark of SRS Labs, Inc. Logic 7 is a trademark of Harman International Corporation. The HDCD logo, HDCD, High Definition Compatible Digital and Pacific Microsonics are either registered trademarks or trademarks of Pacific Microsonics, Inc. in the United States and/or other countries. HDCD technology provided under license from Pacific Microsonics, Inc. This product's design (and/or software) is covered by one or more of the following in the United States: 5,479,168; 5,638,074; 5,640,161; 5,808,574; 5,838,274; 5,854,600; 5,864,311; and in Australia: 669114; with other patents pending. The "Crystal Original Surround Logo" is a trademark and the "Crystal Logo" and the "Crystal Digital Sound Processing Logo" are registered trademarks of Cirrus Logic, Inc.

Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Advance product information describes products which are in development and subject to development changes. Cirrus Logic, Inc. has made best efforts to ensure that the information contained in this document is accurate and reliable. However, the information is subject to change without notice and is provided "AS IS" without warranty of any kind (express or implied). No responsibility is assumed by Cirrus Logic, Inc. for the use of this information, nor for infringements of patents or other rights of third parties. This document is the property of Cirrus Logic, Inc. and implies no license under patents, copyrights, trademarks, or trade secrets. No part of this publication may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc. Items from any Cirrus Logic website or disk may be printed for use by the user. However, no part of the printout or electronic files may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc. Furthermore, no part of this publication may be used as a basis for manufacture or sale of any items without the prior written consent of Cirrus Logic, Inc. The names of products of Cirrus Logic, Inc. or other vendors and suppliers appearing in this document may be trademarks or service marks of their respective owners which may be registered in some jurisdictions. A list of Cirrus Logic, Inc. trademarks and service marks can be found at <http://www.cirrus.com>.

5.4.5 Non - Reverberative Effects .....	39
5.4.5.1 Passthru .....	39
5.4.5.2 Mono .....	39
5.4.5.3 Classical .....	39
5.4.5.4 Panorama .....	40
5.4.5.5 Movie 1-4 .....	40
5.4.5.6 Music 1-4 .....	40
5.4.6 Reverberative Effects .....	40
5.4.6.1 t1r1a1i1.cfg - Room 1 .....	42
5.4.6.2 t2r3a1i2.cfg - Room 2 .....	42
5.4.6.3 t1r1a2i1.cfg - Theater 1 .....	42
5.4.6.4 t2r3a2i2.cfg -- Theater 2 .....	42
5.4.6.5 t1r1a3i1.cfg -- Hall 1 .....	42
5.4.6.6 t2r3a3i2.cfg -- Hall 2 .....	42
5.4.6.7 t2r3a4i2.cfg -- Stadium .....	42
5.4.7 Effects Configuration Files .....	42
5.5 COEFMAKR.EXE and EFFECTS.EXE .....	42

**LIST OF FIGURES**

Figure 1. Generic Stereo PCM Processing Block Diagram .....	6
Figure 2. Filter Topology .....	20
Figure 3. Bass Manager Processing Unit .....	20
Figure 4. Subwoofer Summing Module .....	21
Figure 5. Generic Autodetection Flow Chart .....	25
Figure 6. Typical Download and Configuration .....	26
Figure 7. Crystal Original Surround (C.O.S.) Block Diagram .....	27
Figure 8. Illustrated Basic Reverb Theory .....	34
Figure 9. Detailed Block Diagram of C.O.S. Effects Code .....	34
Figure 10. Block Diagram of C.O.S. Output Channel Mixer .....	36
Figure 11. Detailed Block Diagram of C.O.S. Reverberator Module .....	37

**LIST OF TABLES**

Table 1. Input/Output Configurations Supported by PCM Processing .....	7
Table 2. Write Message Format.....	8
Table 3. Read Command Message Format .....	9
Table 4. Read Response Message Format .....	9
Table 5. Unsolicited Read Message Format .....	10
Table 6. Audio Manager.....	12
Table 7. Bass Manager .....	16
Table 8. Unsolicited Messages .....	22
Table 9. Crystal Original Surround Effects Manager.....	28
Table 10. Crystal Original Surround Effects PCM Processor.....	31

## 1. OVERVIEW

The CS4923/4/5/6/7/8/9 (CS492X) is a family of system on a chip solutions for Multichannel audio decompression and digital signal processing. Since the part is RAM-based, a download of application software is required each time the CS492X is powered up.

These parts are generally targeted at two different market segments. The broadcast market where audio/video (A/V) synchronization is required, and the outboard decoder markets where audio/video synchronization is not required. The important differentiation is the format in which the data will be received by the CS492X. In systems where A/V synchronization is required from the CS492X, the incoming data is typically PES encoded. In an outboard decoder application the data typically comes in the IEC61937 format (as specified by the DVD consortium). An important point to remember is that the CS492X will support both environments, but different downloads are required depending on the input data type.

Broadcast applications include (but are not limited to) set top box applications, DVDs and digital TVs. Outboard decoder applications include stand-alone decoders and audio/video receivers. Often times a system may be a hybrid between an outboard decoder and a broadcast system depending on its functionality.

This users guide covers code that provides Stereo PCM Processing. All PCM processing codes, including C.O.S., SRS CircleSurround, HDCD, and Logic 7 may run on any of the CS492X Family.

### 1.1 Multichannel Decoder Family of Parts

#### ***CS4923 - Dolby Digital<sup>TM</sup> Audio Decoder***

The CS4923 is the original member of the family and is intended to be used if only Dolby Digital decoding is required. For Dolby Digital, post processing includes bass management, delays and Dolby Pro Logic decoding. Separate downloads

can also be used to support stereo to 5.1 channel effects processing and stereo MPEG decoding.

#### ***CS4924 - Dolby Digital<sup>TM</sup> Source Product Decoder***

The CS4924 is the stereo version of the CS4923 designed for source products such as DVD, HDTV, and set top boxes.

#### ***CS4925 - International Multichannel DVD Audio Decoder***

The CS4925 supports both Dolby Digital and MPEG-2 Multichannel formats. For both Dolby Digital and MPEG-2 Multichannel, post processing includes bass management and Dolby Pro Logic decoding. The Dolby Digital code and MPEG code take separate code downloads. Another code load can be used to support stereo to 5.1 channel effects processing.

#### ***CS4926 - DTS/Dolby<sup>®</sup> Multichannel Audio Decoder***

The CS4926 supports both Dolby Digital and DTS, or Digital Theater Surround. For Dolby Digital, post processing includes bass management and Dolby Pro Logic. The Dolby Digital code and DTS code take separate code downloads. Separate downloads can also be used to support stereo to 5.1 channel effects processing and stereo MPEG decoding.

#### ***CS4927 - MPEG-2 Multichannel Decoder***

The CS4927 supports MPEG-2 Multichannel decoding and should be used in applications where Dolby Digital decoding is not necessary. For MPEG-2 Multichannel decoding, post processing includes bass management and Dolby Pro Logic decoding. Another code load can be used to support stereo to 5.1 channel effects processing.

#### ***CS4928 - DTS Multichannel Decoder***

The CS4928 supports DTS Multichannel decoding and should be used in applications where Dolby Digital decoding is not necessary. For DTS

Multichannel decoding, post processing includes bass management. Separate downloads can also be used to support stereo to 5.1 channel effects processing and stereo MPEG decoding.

### ***CS4929 - AAC 2-Channel, (Low Complexity) and MPEG-2 Stereo Decoder***

The CS4929 is capable of decoding both 2-channel AAC (ADTS format) and MPEG-2 audio. The CS4929 supports both elementary and PES formats.

## **1.2 Document Strategy**

Multiple documents are needed to fully define, understand and implement the functionality of the CS492X. They can be split up into two basic groups: hardware and application code documentation. It should be noted that hardware and application code are co-dependent and one can not successfully use the part without an understanding of both. The 'ANXXX' notation denotes the application note number under which the respective user's guide was released.

### ***1.2.1 Hardware Documentation***

#### ***CS4923/4/5/6/7/8/9 Family Data Sheet***

This document describes the electrical characteristics of the device from timing to base functionality. This is the hardware designers tool to learn the part's electrical and systems requirements.

#### ***AN115 - CS4923/4/5/6/7/8/9 Hardware User's Guide***

describes the functional aspects of the device. An in depth description of communication, boot procedure, external memory and hardware configuration are given in this document. This document will be valuable to both the hardware designer and the system programmer.

## ***CS492X Application Code User's Guides***

The following application notes describe the application codes used with the CS492X. Whenever an application code user's guide is referred to, it should be assumed that one or more of the below documents are being referenced. The following list covers currently released application notes. This list will grow with each new application released. For a current list of released user's guides please see [www.crystal.com](http://www.crystal.com) and search for the part number.

### ***AN120 - Dolby Digital User's Guide for the CS4923/4/5/6***

This document covers the features available in the Dolby Digital code including delays, pink noise, bass management, Pro Logic, PCM pass through and Dolby Digital processing features. Optional appendices are available that document code for Virtual Dolby Digital<sup>™</sup>, and VMax<sup>™</sup>.

### ***AN121 - MPEG User's Guide for the CS4925***

This document covers the features available in the MPEG Multichannel code including delays, bass management, Pro Logic, and MPEG processing features.

### ***AN122 - DTS<sup>™</sup> User's Guide for the CS4926, CS4928***

This document covers the features available in the DTS code including bass management and DTS processing features.

### ***AN123 - Surround User's Guide for the CS492X***

This code covers the different Stereo PCM to surround effects processing code. Additional appendices are available that document SRS CircleSurround<sup>™</sup>, Logic 7<sup>™</sup>, and Pacific Microsonics<sup>™</sup> HDCD<sup>®</sup>.

**AN140 - Broadcast Systems Guide for the CS492X**

This guide describes all application code (e.g. Dolby Digital and MPEG) designed for broadcast systems such as HDTV and set-top box receivers. This document also provides a discussion of broadcast system considerations and dependencies.

**1.3 Using the CS492X**

No matter what application is being used on the chip, the following four steps are always followed to use the CS492X in system.

- 1) Reset and/or Download Code - Detailed information can be found in AN115.
- 2) Hardware Configuration - Detailed information can be found in AN115.
- 3) Application configuration - Detailed information can be found in the appropriate Application Code User's Guide.
- 4) Kickstart - This is the "Go" command to the CS492X once the system is properly configured. Detailed information can be found in the appropriate Application Code User's Guide.

**1.4 Software Naming Convention**

To keep up with the different applications, parts, hardware revisions and software revisions the following naming convention has been adopted for the CS492X software:

AAAACCRV.LD

where:

AAAA = three or four letter Application description (e.g. AC3\_)

CC = Chip suffix for hardware (e.g. for CS4923 CC=23)

R = ROM ID specification

V = actual version release of particular code

**1.5 Stereo PCM Application Description**

Each application that accepts Stereo PCM input is described in detail in its corresponding appendix. However, all these applications share the same general structure.

Figure 1 is a functional block diagram of this general structure. It gives an idea of the interaction between the various modules in the application.

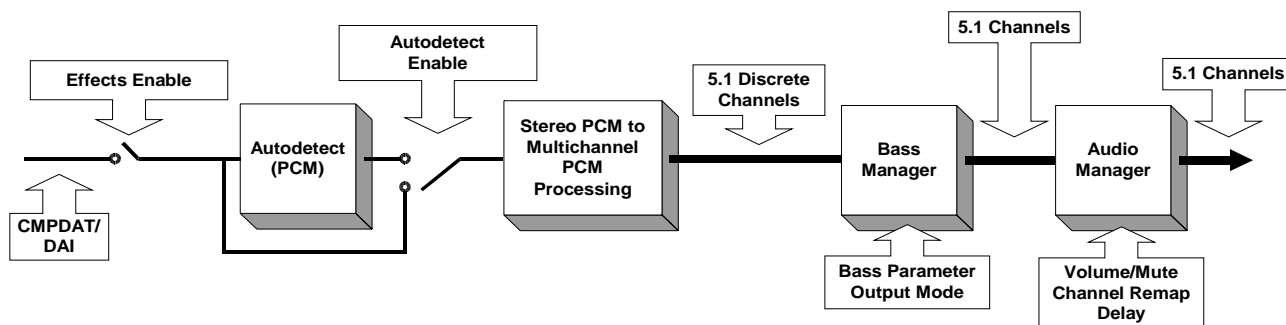


Figure 1. Generic Stereo PCM Processing Block Diagram

## 2. HARDWARE CONFIGURATION

After download or soft reset, and before kickstarting the application (please see “Audio Manager” for more information on kickstarting), the host has the option of changing the default hardware configuration. Address checking for serial communication and data type and format for digital data input and output can be changed through hardware configuration settings. The entire hardware configuration can only be changed immediately after download or soft reset.

### 2.1 Supported Input/Output Modes

The CS492X has two input ports and one output port. AN115 describes the digital audio formats supported by the ports and gives a description of the ports themselves. The capabilities of each port are presented in AN115 although all configurations are not supported by all applications. Each

Software Application User’s Guide specifies the exact input/output modes supported by the application.

Referring to AN115, the following input/output modes are supported by the PCM Processing application (C.O.S., Logic 7, SRS Circle Surround, and HDCD).

<b>I/O Configuration</b>	<b>Description</b>	<b>Supported Modes</b>
INPUTA	Input Data Type	0,1,2
INPUTB	Input Data Format	0,1
INPUTC	Input SCLK/Data Edge	0,1
OUTPUTA	Output Master/Slave Clock Setup	0,1
OUTPUTB	Output Data Format	0,1,2,3
OUTPUTC	Output MCLK Rate	0,1,2,3
OUTPUTD	Output SCLK Rate	0,1,2
OUTPUTE	Output SCLK/Data Edge	0,1

**Table 1. Input/Output Configurations Supported by PCM Processing**

### 3. APPLICATION MESSAGING

While using the CS492X it may be necessary to control or monitor the application to take full advantage of the rich feature set employed by the CS492X and its software. Application messaging allows the user to do this. Whether it be configuring the part for a new software mode, e.g. change from AC-3 decode to PCM pass through, or changing run-time parameters, e.g. controlling digital volume, the host will use application messaging to communicate with the CS492X.

The CS492X has a strict software protocol which must be used in conjunction with the hardware protocol discussed in AN115, the CS492X Hardware User’s Guide. This section will cover

both the format of the messages while “Application Modules” will cover different configuration modules available with the CS492X. It must be stressed that, for write and read transactions with the CS492X, the host must strictly adhere to the hardware and software protocols to insure successful communication.

#### 3.1 Write Session

A write session with the CS492X consists of one 6 byte message from the host to the CS492X. The write message consists of a command word followed by an associated data word.

Table 2 shows the format of a Write Message.

<b>Write Command Word:</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE[7:0]								INDEX[15:0]															

<b>Write Data Word:</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[23:0]																							

**Table 2. Write Message Format**

The fields should be specified as follows:

OPCODE[7:0] - 8 bit (1 byte) field containing opcode for targeted application module. This field will choose which module is written.

INDEX[15:0] - 16 bit (2 byte) field containing the index for the desired variable in the module chosen by *OPCODE*. This field will choose the actual variable to be altered.

DATA[23:0] - 24 bit (3 byte) data word to be written into the variable specified by *INDEX* in the module specified by *OPCODE*.



### 3.2 Solicited Read Message Format

A solicited read session consists of one 3-byte Read Request message from the host to the CS492X, followed by a 6 byte Read Response message from the CS4932X to the host.

The read request message simply consists of a Read Command word whose format is shown in Table 3.

After the host sends the read command word it should wait for the  $\overline{\text{INTREQ}}$  line to fall. See the AN115 for more on hardware communication with the CS492X.

After  $\overline{\text{INTREQ}}$  falls the host should read out the 6-byte Read Response message which consists of a 3 byte Read Response Command word followed by the requested data word. The format of the Read Response message is shown in Table 4.

<b>Read Command Word:</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE[7:0]								INDEX[15:0]															

**Table 3. Read Command Message Format**

The fields should be specified as follows:

OPCODE[7:0] - 8 bit (1 byte) field containing opcode for targeted application module. This field will choose which module is to be read from.

INDEX[15:0] - 16 bit (2 byte) field containing the index for the desired variable in the module chosen by *OPCODE*. This field will choose the actual variable to be read.

<b>Read Response Command Word:</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE[7:0]								INDEX[15:0]															

<b>Read Response Data Word</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[23:0]																							

**Table 4. Read Response Message Format**

The fields are specified as follows:

OPCODE[7:0] - 8 bit (1 byte) field containing the Read Response opcode for the targeted application module. This field will show which module was read.

INDEX[15:0] - 16 bit (2 byte) field containing the index for the desired variable in the module chosen by *OPCODE*. This field will show the actual variable that was read.

DATA[23:0] - 24 bit (3 byte) data word that was read from the variable specified by *INDEX* in the module specified by *OPCODE*.

### 3.3 Unsolicited Read Message Format

Unsolicited messages will typically be used in systems where  $\overline{\text{INTREQ}}$  can generate interrupts. These messages will come from the CS492X to indicate a change in the system that must be addressed. One example is when the part is in autodetect mode and detects a new stream. An unsolicited read message will be sent by the CS492X to indicate the new stream type.

The 6-byte unsolicited read messages from the CS492X consist of a 3-byte read command word which defines the type of unsolicited message (as described in “Unsolicited Messages (Read-Only)”) and a 3-byte associated data word that contains more information describing a system condition. Every time the existence of a message is detected (by sensing that  $\overline{\text{INTREQ}}$  has gone LOW), the host should read out the 6-byte read unsolicited message.

Table 5 shows the format of an Unsolicited Read Message.

<b>Unsolicited Read Command Word:</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE[7:0]								INDEX[15:0]															

<b>Unsolicited Read Data Word:</b>																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[23:0]																							

**Table 5. Unsolicited Read Message Format**

The fields should be specified as follows:

OPCODE[7:0] - 8 bit (1 byte) field containing opcode for the unsolicited read message

INDEX[15:0] - 16 bit (2 byte) field containing the index for the variable in the module designated by *OPCODE*.

DATA[23:0] - 24 bit (3 byte) data word that corresponds to the variable specified by *INDEX* in the module specified by *OPCODE*.

#### 4. APPLICATION MODULES

The block diagram on the cover page of this document accurately portrays the interaction between the various generic application modules of the PCM Processing code.

The following should be noted about all values in the application modules:

Variables marked by ‘*Default\**’ will only be initialized after download. These variables will retain their values after a soft reset or application restart.

Variables marked by ‘Default’ will be reinitialized to the values shown in this application note after download, soft reset or application restart.

Only those values which are presented as valid or which fall within the specified range should be written to the application module variables. If a value which falls outside the stated range is written

to a variable, functionality of the application is not guaranteed and erroneous output could result.

All bits that are not defined should be considered reserved and written with 0’s unless specified otherwise.

For variables such as volume and scaling factors, the real number range of 0.0-1.0 is written as 0x000000-0x7FFFFFFF. This range is linear, i.e. 1/4 volume, or -12 dB, is represented as 0x1FFFFFFF and is equivalent to a value of 0.25.

The formula for converting variable settings from a hexadecimal number into dB is as follows:

$$20 * \log (\text{variable setting}/\text{max value for variable})$$

e.g.  $20 * \log (0x1FFFFFFF/0x7FFFFFFF) == -12 \text{ dB}$

Numbers preceded by 0x should be interpreted as hex, and numbers followed by ‘b’ should be interpreted as binary. All values sent to the CS492X should be converted to hexadecimal. Likewise all values read from the part are in hexadecimal.

## 4.1 Audio Manager

Write Opcode = 0x88; Read Opcode = 0x09; Read Response Opcode = 0x89

Write Message = 0x8800HH 0xhhhhh

Read Request Message = 0x0900HH;

Read Response Message = 0x8900HH 0xhhhhh

0xHH = index, 0xhhhhh = data value

Index	Variable	Dataword Content
0x00	AUDIO_MGR_CONTROL	<p><b>Bit 14:</b> Startup_Autodetect_Bypass_Enable= 0/1= Disable/Enable bypassing of autodetection at start-up, i.e. for the first time after download/reset. (Valid only with Autodetect_Enable==1).            Note: If bit 14 is enabled, during readback this bit is reserved and may read 0 or 1.</p> <p><b>Bit 12:</b> Autodetect_Enable = 0/1= Disable/Enable autodetect functionality.</p> <p><b>Bit 4:</b> PLL_Enable = 0/1 = Disable/Enable Phase Locked Loop generation of DSP clock. It is mandatory that the PLL be enabled for Rev D CS492X.</p> <p><b>Bit 0:</b> Kickstart_Enable = 0/1 = Disable/Enable Kickstart of application.  <i>Default = 0x000000</i></p>
0x06	MASTER_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x07	L_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x08	C_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x09	R_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x0a	LS_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x0b	RS_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x0c	LFE_VOLUME	0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFF</i>
0x0d	MUTE	0/1 = Unmute/mute audio. This is a soft mute. <i>Default = 0</i>
0x0e	DAO0_CHANNEL <sup>A</sup>	0...5 = Channel type <sup>a</sup> for Digital Audio Output 0. Each channel must be mapped to one and only one unique output. <i>Default* = 0(L)</i>
0x0f	DAO1_CHANNEL <sup>A</sup>	0...5 = Channel type <sup>a</sup> for Digital Audio Output 1. Each channel must be mapped to one and only one unique output. <i>Default* = 2(R)</i>
0x10	DAO2_CHANNEL <sup>A</sup>	0...5 = Channel type <sup>a</sup> for Digital Audio Output 2. Each channel must be mapped to one and only one unique output. <i>Default* = 3(Ls)</i>
0x11	DAO3_CHANNEL <sup>A</sup>	0...5 = Channel type <sup>a</sup> for Digital Audio Output 3. Each channel must be mapped to one and only one unique output. <i>Default* = 4(Rs)</i>
0x12	DAO4_CHANNEL <sup>A</sup>	0...5 = Channel type <sup>a</sup> for Digital Audio Output 4. Each channel must be mapped to one and only one unique output. <i>Default* = 1(C)</i>
0x13	DAO5_CHANNEL <sup>A</sup>	0...5 = Channel type <sup>a</sup> for Digital Audio Output 5. Each channel must be mapped to one and only one unique output. <i>Default* = 5(LFE)</i>

**Table 6. Audio Manager**

<b>Index</b>	<b>Variable</b>	<b>Dataword Content</b>
0x16	LAST_UN SOLICITED_MESSAGE	Last Unsolicited Message Value (see "Unsolicited Messages (Read-Only)") <b>READ ONLY</b> Default=0x000000
0x17	PLL_REGISTER_1	Register 1 value for Phase Locked Loop (no Fs generation). Default* = 0x07f270 (CLKIN=12.288MHz, DSPCLK=60MHz)
0x18	PLL_REGISTER_2	Register 2 value for Phase Locked Loop (no Fs generation). Default* = 0x0001fe (CLKIN=12.288MHz, DSPCLK=60MHz)

**Table 6. Audio Manager (Continued)**

Notes: a. Mapping should be setup *before* kickstart of application and should always be complete and one-to-one, i.e. each DAO should be mapped to one and only Channel type. Channel type 0...5 corresponds to Channels L, C, R, Ls, Rs and LFE respectively.

### 4.1.1 Audio\_Control: (Index 0x00)

#### Bit\_15:

##### *Audio\_Config\_Change\_Notification\_Enable*

Setting this bit HIGH will cause the CS492X to generate interrupts when a change to a specific stream definition variable is occurs. Stream definition variables are embedded in the compressed bitstream.

Note: The host must wait at least 20ms before attempting to read any of the stream definition variables after receiving an Unsolicited Message indicating that there has been a change to one of the variables. Variable values read from the DSP before this time has passed may not be valid.

For the AC-3 code if any of *acmod*, *lfeon*, *dsurmod*, *dialnorm* or *timecod2* variables change in the incoming stream, CS492X will issue an unsolicited message.

For the DTS code if any of *amode*, *lff*, or *pcmr* variables change in the incoming stream, CS492X will issue an unsolicited message.

For the MP3 code if the *mode* variable changes in the incoming stream, CS492X will issue an unsolicited message.

For the MPEG code if any of *mode*, *center*, *surround*, or *lfe* variables change in the incoming stream, CS492X will issue an unsolicited message.

For the AAC code if the *channel\_configuration* variable changes in the incoming stream, CS492X will issue an unsolicited message.

- 0 - Disables Audio\_Config\_Change\_Notification
- 1 - Enables Audio\_Config\_Change\_Notification

#### Bit 14: *Startup\_Autodetect\_Bypass\_Enable*

Maintains the CS492X in autodetect mode, but bypasses autodetection (directly enters input data processing) at start-up, i.e. for the first time after download/reset. This bit should only be changed after download, soft reset, or Application Restart.

The state of autodetect should not be modified during run-time.

- 0 - Disables Startup\_Autodetect\_Bypass
- 1 - Enables Startup\_Autodetect\_Bypass

No unsolicited message is generated by CS492X if the processing is successful. However, if the data format is subsequently found to be the wrong type, autodetect messaging will be generated by CS492X as usual.

The above Startup\_Autodetect\_Bypass mode is useful in saving the autodetect latency when the controller already knows that the current input data format is decodable by the downloaded application. The controller is usually aware of this due to the autodetection message from the previous (most recent) downloaded application.

Note: If this mode is enabled, Bit 14 readback value is reserved (either 0 or 1 may be received and should be ignored).

#### Bit 12: *Autodetect\_Enable*

Puts the CS492X into autodetect mode (see “Unsolicited Messages (Read-Only)”)

This bit should only be changed after download, soft reset, or Application Restart. The state of autodetect should not be modified during run-time.

- 0 - Disables Autodetect
- 1 - Enables Autodetect

#### Bit 4: *PLL\_Enable*

Enables the use of the Phase Locked Loop (PLL) to generate internal DSPCLK from the desired external CLKIN.

- 0 - Disables PLL
- 1 - Enables PLL

The register values in PLL\_Register\_1 (0x17) and PLL\_Register\_2 (0x18) define the desired DSP clock and reference CLKIN used. The default values at download setup a 60 MHz DSPCLK from a 12.288 MHz external CLKIN. For a CLKIN of 11.2896 MHz, PLL\_Register\_1=0x24BC34 and PLL\_Register\_2 = 0x0001fe should be used. These

register values should be downloaded prior to Kickstart with PLL\_Enable.

**Bit 0:** *Kickstart Enable*

Puts CS492X into run-time mode. Indicates that all hardware and software configuration has been completed and CS492X can start the application.

- 0 - Application continues waiting for kickstart.
- 1 - Application is kickstarted

## 4.2 Generalized Bass Manager

Write Opcode = 0x94; Read Opcode = 0x15; Read Response Opcode = 0x95

Write = 0x9400HH 0xhhhhhh

Read Request = 0x1500HH;

Read Response = 0x9500HH 0xhhhhhh

0xHH = index, 0xhhhhhh = data value

Index	Variable	Datword Content
0x00	BASS_MGR_CONTROL	<b>Bit 16:</b> LFE_Only_Sub = 0/1 = Disable/Enable LFE Only to subwoofer. <b>Bit 14:</b> 4th_Order_LPF = 0/1 = Disable/Enable 4th order LPF at output of summer. Bit 12 must be set in addition to Bit 14 to enable the 4th order LPF. <b>Bit 12:</b> LPF = 0/1 = Disable/Enable LPF at output of summer. <b>Bit 8:</b> HPF_L = 0/1 = Disable/Enable L channel HPF. <b>Bit 7:</b> HPF_C = 0/1 = Disable/Enable C channel HPF. <b>Bit 6:</b> HPF_R = 0/1 = Disable/Enable R channel HPF. <b>Bit 5:</b> HPF_Ls = 0/1 = Disable/Enable Ls channel HPF. <b>Bit 4:</b> HPF_Rs = 0/1 = Disable/Enable Rs channel HPF. <b>Bit 0:</b> Bass_Mgr = 0/1 = Disable/Enable Bass Manager post-processing. Default = 0x000000
0x01	INPUT_L_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input L channel pass-through. <i>Default* = 0x7FFFFFFF (0 dB)</i>
0x02	INPUT_C_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input C channel input pass-through. <i>Default* = 0x7FFFFFFF (0 dB)</i>
0x03	INPUT_R_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input R channel input pass-through. <i>Default* = 0x7FFFFFFF (0 dB)</i>
0x04	INPUT_LS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input Ls channel pass-through. <i>Default* = 0x7FFFFFFF (0 dB)</i>
0x05	INPUT_RS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input Rs channel pass-through. <i>Default* = 0x7FFFFFFF (0 dB)</i>
0x06	INPUT_LFE_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE channel pass-through. <i>Default* = 0x7FFFFFFF (0 dB)</i>
0x07	SUM_OUTPUT_L_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to L channel output. <i>Default* = 0x000000 (-∞ dB)</i>

**Table 7. Bass Manager**



<b>Index</b>	<b>Variable</b>	<b>Dataword Content</b>
0x08	SUM_OUTPUT_C_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to C channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x09	SUM_OUTPUT_R_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to R channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x0a	SUM_OUTPUT_LS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to Ls channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x0b	SUM_OUTPUT_RS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to Rs channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x0c	LFE_L_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to L channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x0d	LFE_C_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to C channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x0e	LFE_R_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to R channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x0f	LFE_LS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to Ls channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x10	LFE_RS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to Rs channel output. <i>Default* = 0x000000 (-∞ dB)</i>
0x11	SUM_INPUT_L_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input L channel contribution to bass summer input. <i>Default* = 0x16C311 (-15 dB)</i>
0x12	SUM_INPUT_C_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input C channel contribution to bass summer input. <i>Default* = 0x16C311 (-15 dB)</i>
0x13	SUM_INPUT_R_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input R channel contribution to bass summer input. <i>Default* = 0x16C311 (-15 dB)</i>
0x14	SUM_INPUT_LS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input Ls channel contribution to bass summer input. <i>Default* = 0x16C311 (-15 dB)</i>

**Table 7. Bass Manager (Continued)**

<b>Index</b>	<b>Variable</b>	<b>Dataword Content</b>
0x15	SUM_INPUT_RS_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input Rs channel contribution to bass summer input. <i>Default* = 0x16C311 (-15 dB)</i>
0x16	SUM_INPUT_LFE_LEVEL	0x000000-0x7FFFFFFF = Level adjustment for input LFE channel contribution to bass summer. <i>Default* = 0x47FACD (-5 dB)</i>
0x17	LPF_48_B0	0x000000-0x7FFFFFFF = b0 coefficient for Low Pass Filter, Fs=48 KHz. <i>Default* = 0x014D4C</i>
0x18	LPF_48_B1	0x000000-0x7FFFFFFF = b1 coefficient for Low Pass Filter, Fs=48 KHz. <i>Default* = 0x014D4C</i>
0x19	LPF_48_A1	0x000000-0x7FFFFFFF = a1 coefficient for Low Pass Filter, Fs=48 KHz. <i>Default* = 0x7D6567</i>
0x1a	HPF_48_B0	0x000000-0x7FFFFFFF = b0 coefficient for High Pass Filter, Fs=48 KHz. <i>Default* = 0x7F7750</i>
0x1b	HPF_48_B1	0x000000-0x7FFFFFFF = b1 coefficient for High Pass Filter, Fs=48 KHz. <i>Default* = 0x8088B0</i>
0x1c	HPF_48_A1	0x000000-0x7FFFFFFF = a1 coefficient for High Pass Filter, Fs=48 KHz. <i>Default* = 0x7EEEE0</i>
0x1d	LPF_44_B0	0x000000-0x7FFFFFFF = b0 coefficient for Low Pass Filter, Fs=44 KHz. <i>Default* = 0x016A73</i>
0x1e	LPF_44_B1	0x000000-0x7FFFFFFF = b1 coefficient for Low Pass Filter, Fs=44 KHz. <i>Default* = 0x016A73</i>
0x1f	LPF_44_A1	0x000000-0x7FFFFFFF = a1 coefficient for Low Pass Filter, Fs=44 KHz. <i>Default* = 0x7D2B19</i>
0x20	HPF_44_B0	0x000000-0x7FFFFFFF = b0 coefficient for High Pass Filter, Fs=44 KHz. <i>Default* = 0x7F6B48</i>
0x21	HPF_44_B1	0x000000-0x7FFFFFFF = b1 coefficient for High Pass Filter, Fs=44 KHz. <i>Default* = 0x8094B8</i>
0x22	HPF_44_A1	0x000000-0x7FFFFFFF = a1 coefficient for High Pass Filter, Fs=44 KHz. <i>Default* = 0x7ED68F</i>
0x23	LPF_32_B0	0x000000-0x7FFFFFFF = b0 coefficient for Low Pass Filter, Fs=32 KHz. <i>Default* = 0x01F171</i>
0x24	LPF_32_B1	0x000000-0x7FFFFFFF = b1 coefficient for Low Pass Filter, Fs=32 KHz. <i>Default* = 0x01F171</i>
0x25	LPF_32_A1	0x000000-0x7FFFFFFF = a1 coefficient for Low Pass Filter, Fs=32 KHz. <i>Default* = 0x7C1D1F</i>

**Table 7. Bass Manager (Continued)**

---

<b>Index</b>	<b>Variable</b>	<b>Dataword Content</b>
0x26	HPF_32_B0	0x000000-0x7FFFFFFF = b0 coefficient for High Pass Filter, Fs=32 KHz. <i>Default* = 0x7F3365</i>
0x27	HPF_32_B1	0x000000-0x7FFFFFFF = b1 coefficient for High Pass Filter, Fs=32 KHz. <i>Default* = 0x80CC9B</i>
0x28	HPF_32_A1	0x000000-0x7FFFFFFF = a1 coefficient for High Pass Filter, Fs=32 KHz. <i>Default* = 0x7E66CA</i>

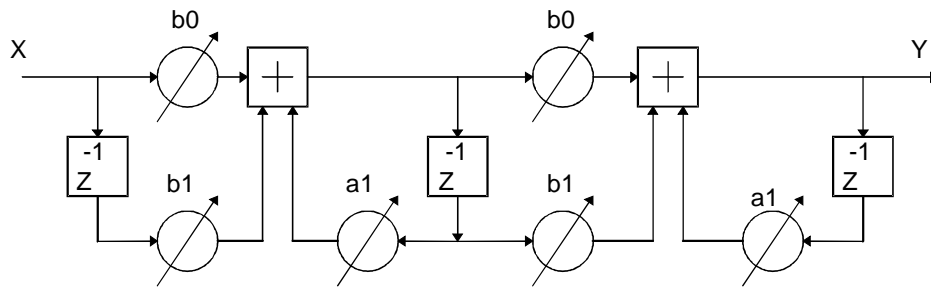
**Table 7. Bass Manager (Continued)**

The Bass Manager is best described with the use of a block diagram. Figure 2 shows the topology of the filter used for both the low pass and the high pass filters. This filter represents two first order filters cascaded to implement a second order filter. For the 4th order LPF two filters as in figure 2 are implemented back to back. Depending on the coefficient values, either a second order low pass or second order high pass can be implemented. The default 3 dB frequency for this filter is 100Hz. The corner frequency can be changed by downloading new values for the coefficients a1, b0 and b1. Note that since two first order filters are cascaded to obtain the second order filter, the corner frequency

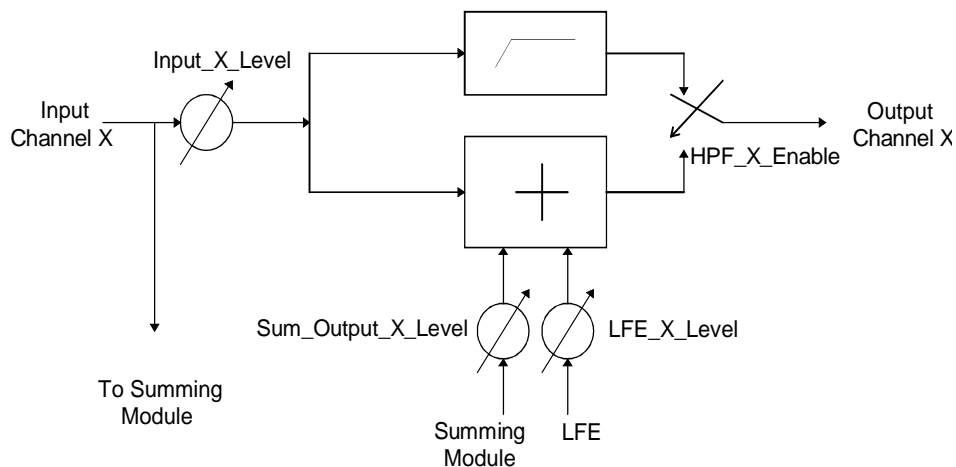
design of the first order prototype has to be adjusted to obtain the desired overall 3dB frequency when two such first order filters are cascaded. In effect, the first order filter should be designed to yield 1.5dB at the desired overall 3dB corner frequency.

Figure 3 shows the processing unit on each of the output channels (except the LFE) and Figure 4 shows the summing module used to redirect bass to the subwoofer channel

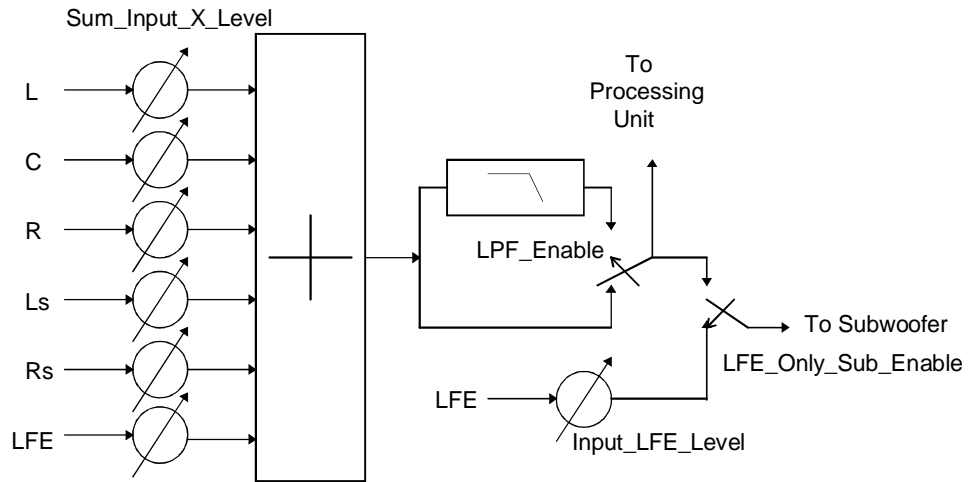
AN120 has examples of many typical bass management configurations.



**Figure 2. Filter Topology**



**Figure 3. Bass Manager Processing Unit**



**Figure 4. Subwoofer Summing Module**

### 4.3 Unsolicited Messages (Read-Only)

No Write Message. No Read Request.

Unsolicited Read Response = 0x8700HH 0xhhhhhh

0xHH = index, 0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	AUTODETECT_RESPONSE	<p><b>Bit 23</b> = Decodable_Stream_Flag= 0/1 = This stream is not/is decodable by the application (no need for new download if 1).</p> <p><b>Bits 22:6</b> = Reserved.</p> <p><b>Bit 5</b> = Non_IEC61937_Stream_Flag= 1/0 = This stream is not/is IEC61937 compressed data.</p> <p><b>If Non_IEC61937_Stream_Flag=1</b></p> <p><b>Bits 4:0</b> = Non_IEC61937 Stream Descriptor.</p> <p>0x00 = Silent Input Data (Out of Application Sync).</p> <p>0x01 = DTS Format-16 elementary stream.</p> <p>0x02 = DTS Format-14 elementary stream.</p> <p>0x03 = Linear PCM stream.</p> <p>0x04 = HDCD PCM stream (only available in HDCD application).</p> <p>0x05-0x1F = Reserved.</p> <p><b>If Non_IEC61937_Stream_Flag=0</b></p> <p><b>Bits 4:0</b> = IEC61937 Stream Descriptor = Identical to bits [4:0] of the Pc burst datatype descriptor in IEC61937 specification.</p> <p>Description of the data-type field of Pc reproduced below from IEC61937 specification (current as of 11/97):</p> <p>0x00 = <i>Never Reported. (Null data is ignored by CS492X).</i></p> <p>0x01 = AC-3 data.</p> <p>0x02 = Reserved.</p> <p>0x03 = <i>Never Reported. (Pause is ignored by CS492X).</i></p> <p>0x04 = MPEG-1 Layer 1 data.</p> <p>0x05 = MPEG-1 Layer 2 or 3 data or MPEG-2 without extension.</p> <p>0x06 = MPEG-2 data with extension.</p> <p>0x07 = Reserved.</p> <p>0x08 = MPEG-2 Layer 1 Low sampling frequency.</p> <p>0x09 = MPEG-2 Layer 2 or 3 Low sampling frequency.</p> <p>0x0A = Reserved.</p> <p>0x0B = DTS-1 data (512-sample bursts).</p> <p>0x0C = DTS-2 data (1024-sample bursts).</p> <p>0x0D = DTS-3 data (2048-sample bursts).</p> <p>0x0E - 0x1B = Reserved.</p> <p>0x1C = MPEG-2 AAC ADTS data.</p> <p>0x1D - 0x1F = Reserved.</p>
0x03	AUDIO_CONFIGURATION_CHANGE	<p><b>Bits 23:0</b> = Reserved.</p> <p>Please see Section 4.1.1. bit 15 for more information.</p>
0x10	PLL_OUT_OF_LOCK	<p><b>Bit 23</b> = 1.</p> <p><b>Bits 22:0</b> = Reserved.</p>

Table 8. Unsolicited Messages

### 4.3.1 Autodetect Operation

The sequence of events involving autodetection are described below from the host's perspective. This is a general example for the CS492X, when using MPEG code or DTS code application restart is not available and thus should not be used.

- 1) Host downloads CS492X with a tentative application code, for this example we will use AC3\_2\*\*\*.LD.
- 2) Host then configures the CS492X hardware appropriately and sets up application parameters as desired - including enable of the desired application. For this example we will say the code is configured for AC-3 decode.
- 3) Host then kickstarts CS492X with Autodetect enabled (see Section Audio Manager for details).
- 4) The autodetect module of the enabled application of the CS492X analyzes the input for a maximum of 500 ms of non-silent/non-pause data and determines the content of the input bitstream.
- 5) (a) If the enabled application can play the detected input (i.e. if AC-3 was detected in this case), then the CS492X issues an Unsolicited Message to the host indicating the datatype with `Decodable_Bitstream_Flag=1`. In our example of the AC-3 stream, the message would be `0x870000 0x800001`. CS492X then goes ahead and processes it according to the application parameters as setup in Step 2 above.  
(b) If the enabled application cannot play the detected input (say Non-IEC61937 LD DTS was detected), then the CS492X soft mutes the outputs, and issues an Unsolicited Message to the host indicating the datatype with `Decodable_Bitstream_Flag=0`. In our example, the message would be `0x870000 0x000021`.

On receiving this message, host repeats Steps 1 onwards but this time downloads the DTS code

to the CS492X (along with correct hardware configuration and application configuration for DTS with autodetect enabled). Subsequently, DTS will be detected within 500 mS and successfully played by the new DTS code, after sending the corresponding unsolicited message (`0x870000 0x800021`).

Note: This example assumes CS4926 or CS4928 are being used since only it can play DTS. In the case of DTS being detected on CS4923/4/5/7/9 or MPEG detected on CS4923/4/6/8/9, the host should display an appropriate user message to the front panel stating that the detected stream (DTS/MPEG) cannot be played on this receiver.

- 6) After the above steps and while CS492X is successfully playing the input bitstream (still AC-3 in our example), if the host receives external information that the input has been changed (e.g. the user selects a new source using the front panel buttons), then before switching the input data to the CS492X, the host should send an Application Restart message (see "Audio Manager") or a Soft Reset. If application restart is used, this effectively puts CS492X in Step 2, without changing the output hardware configuration (i.e. output clocks are not interrupted). The input configuration would need to be changed if it is different than the default. If Soft Reset is issued then the entire hardware configuration needs to be resent.

The host should then repeat Steps 2, 3, 4, 5a/b as described above after delivering the new input stream to the CS492X.

If the new input content is detected as unchanged (still AC3 in our example), the CS492X responds and continues processing it as in Step 5a. This situation will happen if the new stream selected by the user is also AC-3.

If the input content is detected as different (non-AC-3 in our example), the CS492X

responds as in Step 5b and continues monitoring the input stream for change in content.

- 7) During run-time, while successfully playing the input bitstream, the CS492X also simultaneously monitors the input. Note that the CS492X has only one active input. The 'input' is defined as the pin receiving data for which the application is configured (e.g. If the application is configured for PCM, the pin receiving PCM data is the input. If the application is configured for compressed data, the pin receiving compressed data is the input). As soon as the CS492X detects a change in the bitstream (no longer AC-3, in our original example), then the CS492X automatically reverts to Step 4., i.e. analyzes the input to determine the content. This is an automatic version of Step 6 above, but is intended to only cover the cases where the host is not aware of any possible upstream content changes. Whenever possible, the host should convey information about a possible change in input as in Step 6.

If the input content is detected as different (non-AC-3 in our example), the CS492X reverts to Step 5b.

For compressed data streams the code will also report the Unsolicited Message 0x870000 0x800020 to indicate an out-of-sync condition when the decoder loses sync with the incoming stream due to silent input data. This is an informative message and no action is needed by the host.

If the input content is detected as unchanged (still AC-3 in our example), CS492X continues

processing it like in Step 5a, without requiring any further action from the host. This situation could arise due to a pause or track change upstream in the source, like from a player. In the case of compressed data being played currently (like AC-3 in our example), the host could see an out-of-sync Unsolicited Message followed by an Unsolicited Message indicating AC-3 data. The presence of out-of-sync messages due to 'special' functions like pause or track change is completely dependent upon the audio source (such as the DVD player).

In the case of a PCM application that is currently active, if the silence is less than PCM\_Autodetect\_Silence\_Threshold (default 48000 samples, i.e. 1 Second at 48 KHz) before transitioning to new PCM, the CS492X continues to process the input data as if no change had occurred.

However, during PCM processing, if the silence is more than PCM\_Autodetect\_Silence\_Threshold, the CS492X jumps to a Silent Input Data state, and the output is muted (transparent due to silent input anyway). Transition to this Silent Input Data state is reported via an Unsolicited Message (0x870000 0x800020). This message is informative only, and no action should be taken by the host. CS492X is effectively in Step 4 above now, waiting to autodetect the input once non-silent data appears. Once input data is successfully detected, a corresponding Unsolicited Message is issued to the host and the process continues as above. Please see Figure 5, for an overview of the previously described flow.



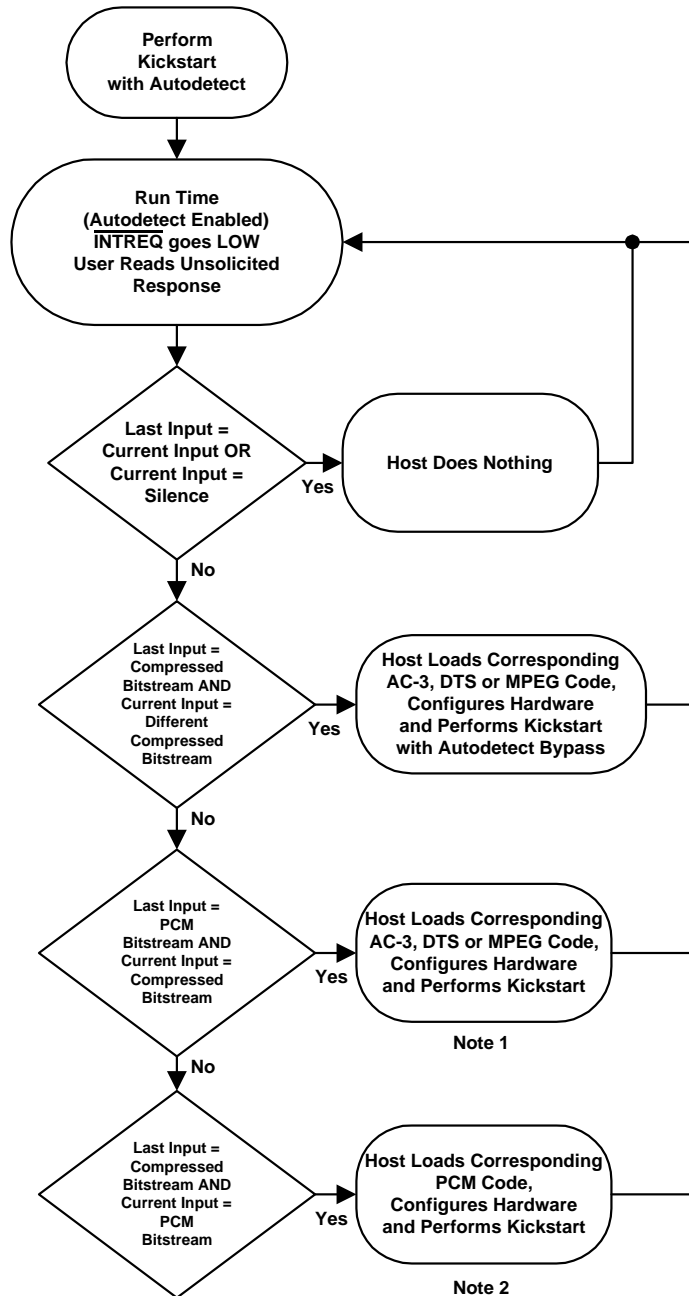
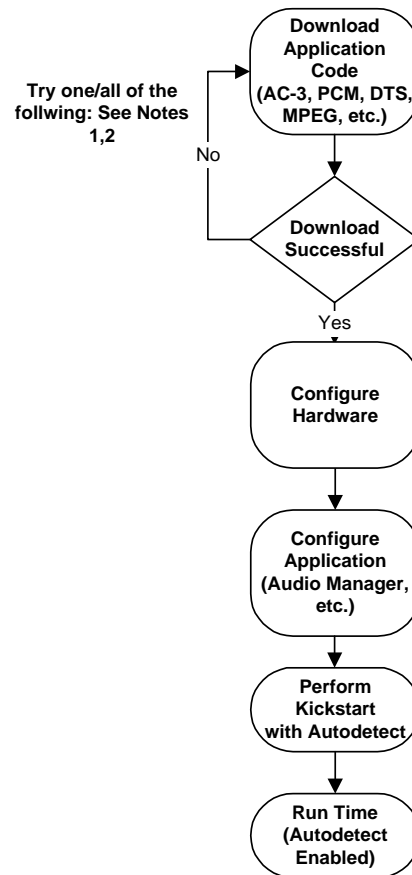


Figure 5. Generic Autodetection Flow Chart

- Notes:
1. Host need not load AC-3 code if AC-3 is already loaded and passing-through PCM. Host need only perform a Soft Reset, Configure the Hardware and perform a Kickstart OR perform an Application Restart, Configure the Inputs, and perform a Kickstart.
  2. Host need not reload AC-3 code if AC-3 is already loaded, as it will pass-through PCM. Host need only perform a Soft Reset, Configure the Hardware and perform a Kickstart OR perform an Application Restart, Configure the Inputs, and perform a Kickstart.

In summary the following is the recommended procedure when using autodetect. Please see Figure 6, to the right for an overview of the previously described summary.

- 1) Download code (or Soft Reset or Application Restart)
- 2) Hardware Configuration (for download or Soft Reset) / Input Configuration (for Application Restart)
- 3) Application Configuration (including enable of desired application)
- 4) Kickstart with autodetection enabled.



**Figure 6. Typical Download and Configuration**

- Notes:
1. Check .LD file version. Contact your FAE for the most recent version.
  2. Check to make sure .LD file is correct for chip being used.

## 5. APPENDIX A - CRYSTAL ORIGINAL SURROUND (C.O.S.) EFFECTS CODE

### 5.1 Crystal Original Surround Description

Crystal Original Surround Effects code was developed to provide stereo to 5.1 channel effects including delays, reverb and room effects. This section covers the software designed to run on the CS492X that provides Crystal Original Surround.

This appendix covers all code that is named with AAAA == EFF\_, CC==XX and R = X (the code was designed to be cross compatible across parts and ROM) as described in “Software Naming Convention”.

It is assumed that the reader is familiar effects processing. If not, one good reference in which many of the effects provided were derived from is:

“About This Reverberation Business”, James A. Moorer, *Computer Music Journal* 3(2): 13-28, 1979.

Figure 5 illustrates the various features of the Crystal Original Surround modules and their interaction with the standard application modules available in all Stereo PCM applications.

### 5.2 Standard Application Modules

All the standard features described in “Application Modules” are included in this application with no changes.

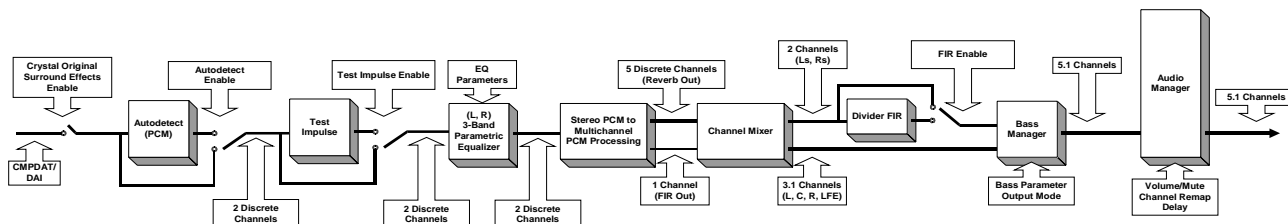


Figure 7. Crystal Original Surround (C.O.S.) Block Diagram

### 5.3 Crystal Original Surround Effects Manager

Write Opcode = 0x98; Read Opcode = 0x19; Read Response Opcode = 0x99

Write Message = 0x9800HH 0xhhhhh

Read Request Message = 0x1900HH;

Read Response Message = 0x9900HH 0xhhhhh

0xHH = index, 0xhhhhh = data value

Index	Variable	Dataword Content
0x00	EFFECTS_MGR_CONTROL	<b>Bit 8:</b> Test Impulse Enable = 0/1 = Disable/Enable replacement of L/R input with periodic impulse signal (repeated every 24000 samples). <b>Bit 4:</b> Deemphasis_Enable = 0/1 = Disable/Enable PCM Deemphasis. <b>Bit 0:</b> Effects Enable = 0/1 = Disable/Enable Surround Effects. <i>Default = 0x000000</i>
0x01	L_MONO_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x02	R_MONO_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x03	DIRECT_FIR_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x04	LPF_FIR_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x05	RESERVED	For future use.
0x06	RESERVED	For future use.
0x07	RESERVED	For future use.
0x08-0x27	T0, T1, ... T30, T31	0...5615 = Delay in samples for each of the 32 taps. <i>NOTE: Setting is in samples, and will thus produce different delays for different sampling frequencies.</i> <i>Default = 0</i>
0x28-0x47	C0, C1,... C31	-1.0 to 1.0 = Coefficients for each of the 32 taps. <i>Default = 0.0</i>
0x48	DIRECT_REVERB_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.875</i>
0x49	FIR_REVERB_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.125</i>
0x4a	REVERB_TIME (COMB_G)	0.0 to 1.0 = Reverberation time: unitless constant (minimum 0.0 to maximum 1.0). <i>Default = 0.5</i>
0x4b	REVERB_LIVENESS (COMB_H)	0.0 to 1.0 = High-frequency attenuation in reverberator: unitless constant. <i>Default = 0.2</i>
0x4c	L_L_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 1.0</i>
0x4d	R_L_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x4e	L_C_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x4f	R_C_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x50	L_R_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x51	R_R_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 1.0</i>
0x52	L_LS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>

**Table 9. Crystal Original Surround Effects Manager**

Index	Variable	Datavord Content
0x53	R_LS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x54	L_RS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x55	R_RS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x56	L_LFE_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x57	R_LFE_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.5</i>
0x58	FIR_L_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x59	REVERB_L_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x5a	FIR_C_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x5b	REVERB_C_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x5c	FIR_R_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x5d	REVERB_R_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x5e	FIR_LS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x5f	REVERB_LS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x60	FIR_RS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
0x61	REVERB_RS_OUT_MIXLEVEL	-1.0 to 1.0. <i>Default = 0.0</i>
	<b><u>ADVANCED SETTINGS</u></b>	
0x62	COMB_0_DELAY_SIZE	1...1103 samples. <i>Default=887</i>
0x63	COMB_1_DELAY_SIZE	1...1213 samples. <i>Default=947</i>
0x64	COMB_2_DELAY_SIZE	1...1321 samples. <i>Default=1069</i>
0x65	COMB_3_DELAY_SIZE	1...1433 samples. <i>Default=1297</i>
0x66	COMB_4_DELAY_SIZE	1...1543 samples. <i>Default=1439</i>
0x67	COMB_5_DELAY_SIZE	1...1657 samples. <i>Default=1571</i>
0x68	COMB_0_G1	0.0 to 1.0. <i>Default=0.46</i>
0x69	COMB_1_G1	0.0 to 1.0. <i>Default=0.48</i>
0x6a	COMB_2_G1	0.0 to 1.0. <i>Default=0.50</i>
0x6b	COMB_3_G1	0.0 to 1.0. <i>Default=0.52</i>
0x6c	COMB_4_G1	0.0 to 1.0. <i>Default=0.53</i>
0x6d	COMB_5_G1	0.0 to 1.0. <i>Default=0.55</i>
0x6e	COMB_0_MIXLEVEL	0.0 to 1.0. <i>Default=0.9</i>
0x6f	COMB_1_MIXLEVEL	0.0 to 1.0. <i>Default=0.9</i>
0x70	COMB_2_MIXLEVEL	0.0 to 1.0. <i>Default=0.9</i>
0x71	COMB_3_MIXLEVEL	0.0 to 1.0. <i>Default=0.9</i>
0x72	COMB_4_MIXLEVEL	0.0 to 1.0. <i>Default=0.9</i>
0x73	COMB_5_MIXLEVEL	0.0 to 1.0. <i>Default=0.9</i>
0x74	ALLPASS_G1	0.0 to 1.0. <i>Default=0.7</i>
0x75	ALLPASS_G2	0.0 to 1.0. <i>Default=0.51</i>

Table 9. Crystal Original Surround Effects Manager (Continued)

### 5.3.1 *Effects\_Mgr\_Control*

#### **Bit 8:** *Test Impulse Enable*

Disable/Enable replacement of L/R input with periodic impulse signal (repeated every 24000 samples). When using this mode Autodetection should be disabled so that the pulse generation is not dependent on the input. If autodetect is left enabled, the pulse will only be generated if PCM is presented at the active PCM input.

- 1 - Enables Test Impulse
- 0 - Disables Test Impulse

#### **Bit 4:** *Deemphasis\_Enable*

Disable/Enable PCM Deemphasis. The filter used is identical to CS4334 (50/15 uS).

- 1 - Enables Deemphasis
- 0 - Disables Deemphasis

#### **Bit 0:** *Effects Enable*

Disable/Enable Surround Effects.

- 1 - Enables Effects
- 0 - Disables Effects

Note: Surround Effects parameters are sampled by the CS492X only once just after kickstart (after a download/soft reset or application restart). These parameters cannot be varied at run-time. Therefore, to change modes, one has to issue a soft reset or application restart, reconfigure hardware settings, issue the new Effects messages and then kickstart.

### 5.4 Crystal Original Surround Effects PCM Processor

Write Opcode = 0x9C; Read Opcode = 0x1D; Read Response Opcode = 0x9D

Write = 0x9C00HH 0xhhhhhh

Read Request = 0x1D00HH

Read Response = 0x9D00HHh 0xhhhhhh

0xHH = index, 0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	PCM_CONTROL	<b>Bit 4:</b> PCM_Input_EQ_Enable = 0/1 = Disable/Enable 3-band EQ filters on input PCM. <b>Bit 0:</b> PCM_Divider_FIR_Enable = 0/1 = Disable/Enable FIR filter on PCM output of Ls/Rs channels. <i>Default = 0x000000</i>
0x01-0x10	DIV_C0, DIV_C1, DIV_C15	-1.0 to 1.0 = Coefficients for each of the 16 taps of the FIR filter. <i>Default* DIV_C0 = 1.0</i> <i>Default* DIV_C1-15 = 0.0</i>
0x11	PRE_EQ_ATTENUATION	-1.0 to 1.0 = Pre-EQ attenuation scale factor to setup desired headroom for EQ filtering. <i>Default* = 1.0</i>
0x12	EQ0_B2	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 0. <i>Default* = 1.0 (0x200000)</i>
0x13	EQ0_B1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 0. <i>Default* = 0.0</i>
0x14	EQ0_B0	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 0. <i>Default* = 0.0</i>
0x15	EQ0_A2	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 0. <i>Default* = 0.0</i>
0x16	EQ0_A1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 0. <i>Default* = 0.0</i>
0x17	EQ1_B2	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 1. <i>Default* = 1.0 (0x200000)</i>
0x18	EQ1_B1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 1. <i>Default* = 0.0</i>
0x19	EQ1_B0	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 1. <i>Default* = 0.0</i>
0x1a	EQ1_A2	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 1. <i>Default* = 0.0</i>
0x1b	EQ1_A1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 1. <i>Default* = 0.0</i>

**Table 10. Crystal Original Surround Effects PCM Processor**

<b>Index</b>	<b>Variable</b>	<b>Dataword Content</b>
0x1c	EQ2_B2	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 2. <i>Default* = 1.0 (0x200000)</i>
0x1d	EQ2_B1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 2. <i>Default* = 0.0</i>
0x1e	EQ2_B1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 2. <i>Default* = 0.0</i>
0x1f	EQ2_A2	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 2. <i>Default* = 0.0</i>
0x20	EQ2_A1	-4.0 to 4.0 in 3.21 format = Biquad coefficient for EQ Filter 2. <i>Default* = 0.0</i>
0x21	POST_EQ_GAIN	-32.0 to 32.0 in 6.18 format = Post-EQ gain scale factor to compensate for Pre-EQ attenuation and EQ filter gains and maximize digital dynamic range usage. <i>Default* = 1.0 (0x040000)</i>

**Table 10. Crystal Original Surround Effects PCM Processor (Continued)**



### 5.4.1 PCM\_Control

#### Bit 4: Input\_PCM\_EQ\_Enable

Switches ON/OFF EQ of input PCM.

- 1 - Enables Input PCM EQ
- 0 - Disables Input PCM EQ

When EQ is enabled, stereo PCM input to the Effects Module (after de-emphasis if enabled) is first attenuated by the Pre\_EQ\_Attenuation scale factor (specified in usual 1.23 format), and then processed by the cascade of three EQ biquads specified by the {EQ0...2: b2, b1, b0, a2, a1} coefficients. The output of the last EQ biquad is then scaled by the Post\_EQ\_Gain scale factor (specified in 6.18 format to allow for up to 30 dB of gain, i.e. scale factor of 32). The EQ-ed stereo pair output then forms the LR input for all further effects processing.

Note: All EQ parameters should be downloaded by the host controller before kickstart.

For an input  $x(n)$  the output  $y(n)$  of each EQ biquad filter is given by:

$$y(n) = x(n)*b0 + x(n-1)*b1 + x(n-2)*b2 + y(n-1)*a1 + y(n-2)*a2$$

Note: The coefficients b2, b1, b0, a2, a1 are all specified in 3.21 format to allow for coefficient magnitudes of up to 4.

#### Bit 0: PCM\_Divider\_FIR\_Enable

Switches ON/OFF the PCM Divider FIR.

- 1 - Enables FIR Filter on Ls and Rs PCM output
- 0 - Disables FIR Filter

PCM produced for output at the Ls and Rs channels according to the Effects setup is passed individually through 2 identical 16-tap FIR filters (assuming Control above has been turned ON). This forms the input to the Bass Manager (if it is turned ON).

For an input  $x(n)$ , the output  $y(n)$  of the FIR filter is defined by:

$$y(n) = x(n)*DIV\_C0 + x(n-1)*DIV\_C1 + x(n-2)*DIV\_C2 \dots + x(n-15)*DIV\_C15$$

The coefficients  $DIV\_C0...15$  are set by the host using messaging into the indices as specified above.

### 5.4.2 Basic Crystal Surround Effects Explanation

Crystal Original Surround Effects is based on public-domain research on acoustic reverberation. For a more detailed discussion of the properties of the implementation, see “About This Reverberation Business”, James A. Moorer, *Computer Music Journal* 3(2): 13-28, 1979.

Figure 8 is an illustration of basic reverb theory. In this figure, E-R should be interpreted as early reflections while A-R should be interpreted as after reflections. Figure 9 illustrates the various modules of the Crystal Original Surround Effects and the corresponding control parameters available to the user. This forms the Surround Effects Processor whose outputs (square nodes with bold legend) are available to the Output Channel Mixer. The Output Channel Mixer also receives the LR stereo pair input

If Deemphasis is enabled, the LR stereo input is first deemphasized (50/15 uS). This stage is not shown in figure 9 and is required only if the input LR PCM input has been preemphasized at the source.

The mono input for the surround processing is derived by mixing the input LR stereo pair with user parameters  $L\_Mono\_Mixlevel$  and  $R\_Mono\_Mixlevel$ . No internal protection or checking is applied to prevent overflow. This is true for all other summing stages in the application also. Therefore, the user settings should ensure that the sum of the magnitudes of summing mixlevels does not exceed unity.

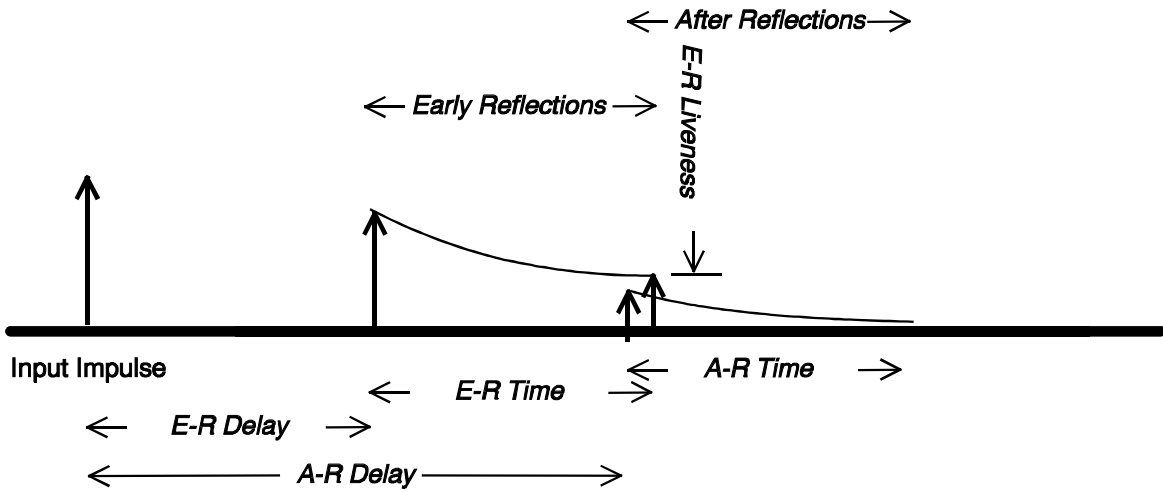


Figure 8. Illustrated Basic Reverb Theory

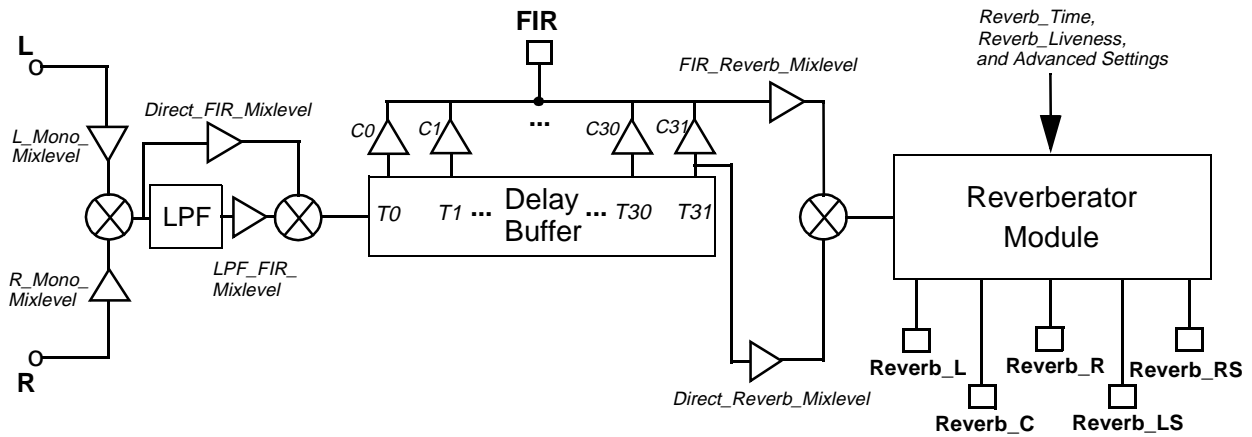


Figure 9. Detailed Block Diagram of C.O.S. Effects Code

In addition, an impulse generator (not shown in figure 9) is also provided for testing purposes. In this mode, the input L/R stereo pair is zeroed out and a unit amplitude impulse is inserted in both input channels every 24000 samples (half a second at  $F_s=48$  KHz).

The LPF at the input of the FIR delay structure is a first-order filter with a corner frequency of  $F_s/6$ , i.e. 8 KHz for  $F_s=48$  KHz and 7.35 KHz for  $F_s=44.1$  KHz. The variable mixlevels *Direct\_FIR\_Mixlevel* and *LPF\_FIR\_Mixlevel* allow one to select the amount of high frequencies to be passed into the FIR stage of the surround processor. This can be used to vary the amount of “brightness” of the surround effect.

The Delay Buffer can hold up to 5616 samples which corresponds to 117 mS at  $F_s=48$  KHz, and 127.35 mS at  $F_s=44$  KHz. Thirty two user-specifiable taps *T0-T31* are available from this delay buffer. The settings *T0-T31* specify the absolute number of samples of delay (0... 5615). The tap delays need not be in monotonic or any specific ordering, and can be arbitrarily sequenced. For example, *T0=4000* samples, *T1 = 300* samples, *T31 = 0* samples is a valid setting.

An FIR filter can be constructed using these delay buffer taps with coefficients *C0-C31* corresponding to the taps *T0-T31*. This FIR filter can be used to model the Early Reflections of a particular room. The FIR output is available to the Output Channel Mixer directly. Tap 31 is available as an input to the reverberator for further surround effects processing.

The mono input to the Reverberator Module is derived by mixing the output of Tap 31 of the delay buffer and the FIR output using user-specified parameters *Direct\_Reverb\_Mixlevel* and *FIR\_Reverb\_Mixlevel* respectively. Note that Tap 31 defaults to a delay of 0, i.e. the direct output of Variable LPF structure. Tap 31 can be adjusted to align the impulse response of the Reverberator Module (After Reflections) with the end of the Early Reflections produced by the FIR filter.

The Reverberator Module accepts user settings for *Reverb\_Time* and *Reverb\_Liveness*, which are both unitless constants. A larger value of *Reverb\_Time* increases the duration of the reverberation effect, i.e. the period of sustained echo. A larger value of *Reverb\_Liveness* decreases the high frequency content (increased high-frequency attenuation) in the reverberation. The Reverberator Module produces 5 outputs to the Output Channel Mixer, one for each output full-bandwidth channel. A more detailed treatment of the advanced settings possible in the reverberator, as well as information on creating custom effects modes is presented in Section 13.2

Figure 10 illustrates the Output Channel Mixer.

Note: *'Mixlevel' abbreviated as 'Mlv.'* due to lack of space. Each of the five full-bandwidth output channels (L, C, R, Ls, Rs) is derived as a user-specifiable mix of the input L/R stereo pair, the FIR output, and the corresponding output channel of the Reverberator Module. The Low Frequency Effects (LFE) channel output carries no surround processing, and is only a user-specifiable mix of the input L/R stereo pair.

Disabling the surround effects can be used for simple PCM pass through.

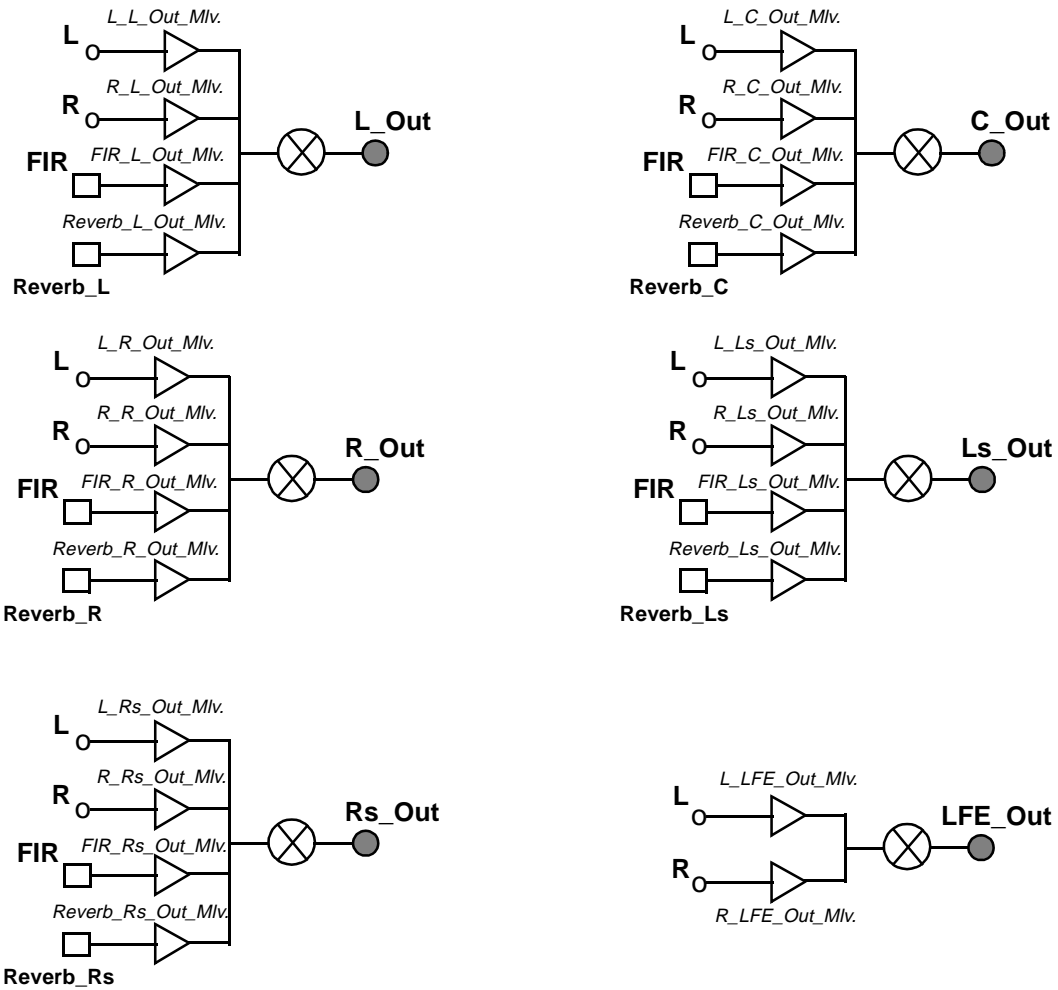


Figure 10. Block Diagram of C.O.S. Output Channel Mixer

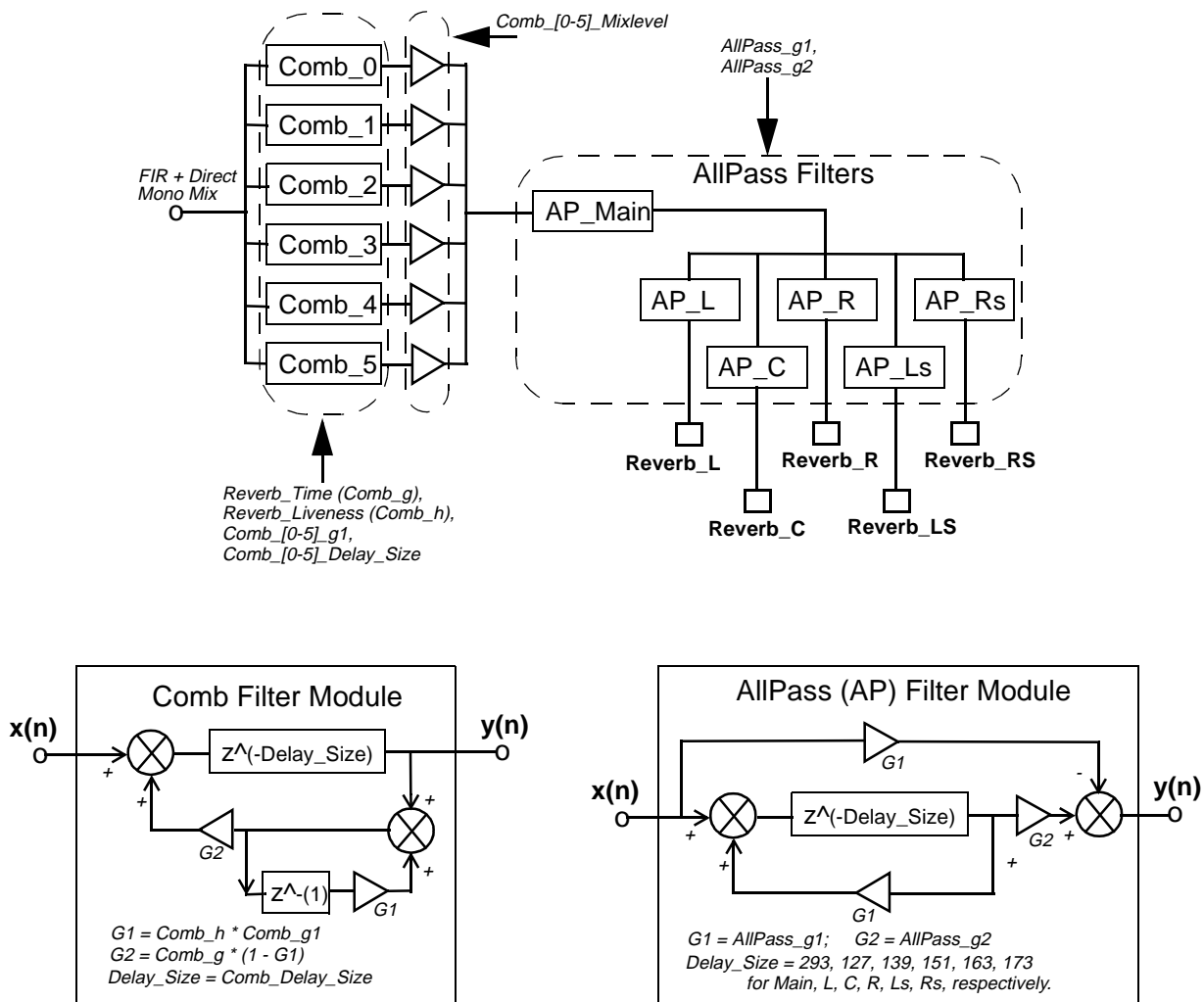
**5.4.3 Creating Custom Effects Using Advanced Settings**

This section goes into a technical description of the Reverberator module. Figure 11 shows a block diagram of the internal implementation of the reverberator module.

**5.4.3.1 FIR Section: Early Reflections**

The FIR filter shown in figure 9 allows one to simulate early reflections and model specific room acoustics. Given a particular set of FIR filter taps

and coefficients (generated by actual room measurements and/or listening experiments), various sizes of rooms can be derived by simply scaling the tap locations (not the coefficients). Other effects can be generated by moving back all of the taps to simulate a delay in the early reflections. The amplitude of the coefficients can be scaled according to the desired liveness of the room. If the amplitude of the coefficients does not fall off quickly, then the room will appear more live (later early reflections will be more present in the output).



**Figure 11. Detailed Block Diagram of C.O.S. Reverberator Module**

#### 5.4.3.2 Comb Filter Section: After-reflections

Following the early reflections simulated by the FIR, comb filters are used to simulate the dense after-reflections that are caused by walls and transitions thereof. Each comb filter reproduces the input signal with periodic delay and decaying amplitude. Furthermore, a first order low pass filter is included in the feedback to simulate the absorption of high frequencies by air. This produces a more realistic or ‘live’ reverberation that sounds less ‘tinny’.

Six combs are used in parallel to produce dense after-reflections. The delay sizes of the combs are chosen to be mutually prime to avoid coinciding echoes from different combs. It is recommended to set the delays of each comb as roughly 10% more than the previous one. However comb sizes with arrangements that are too regular can cause a distorted output.

The coefficients G1 and G2 for a comb should satisfy  $G1+G2<1$  for stability. The scheme for controlling the parameters in CS492X guarantees this. The individual default Comb\_[0-5]\_g1 can be changed. G1 and G2 are derived for each comb automatically as shown in Figure 11 using Comb\_g and Comb\_h as well as Comb\_[0-5]\_g1.

Increasing (or decreasing) the Reverb\_Time (Comb\_g) parameter increases (or decreases) the length of the after-reflection tail since it scales up (or down) the feedback in the comb.

Increasing (or decreasing) the Reverb\_Liveness (Comb\_h) parameter increases (or decreases) the amount of LPF action in the feedback, thus making the sound more (or less) dampened. This is particularly important to reduce the clicking effect when reverberating short impulsive sounds.

While experimenting, it may be found that a particular comb’s reverberation (among the six) is audibly predominant (buzzing sound). To overcome this, the corresponding Comb\_[0-

5]\_Mixlevel for that comb can be reduced to equalize its audible effect with the other combs. These can also be used to overcome overflow issues that may happen with different choices of g1, g, and h.

#### 5.4.3.3 All-Pass Filter Section: More After-reflections and Decorrelation

All-Pass filters are another means of simulating after-reflections. In addition, they are used here to decorrelate the different channel outputs from each other.

In the CS492X implementation, the delay sizes are fixed (as specified in figure 11) and all of the All-Pass filters share the same values of G1 and G2. For stability,  $G2 \leq 1 - G1^2$  should be guaranteed by the host if these values are changed.

Increasing (or decreasing) the G2 parameter increases (or decreases) the reverberation time (length of after-reflection tail) by scaling up (or down) the reverberated contribution to the output.

#### 5.4.4 Crystal Original Surround Effects Modes

This section describes some standard effects modes that are provided along with the application microcode. Each mode is setup by issuing an application restart message or soft reset (assuming the microcode is already downloaded), sending any required hardware configuration messages, and finally sending the corresponding configuration message set for each surround effects mode. For each mode described, the configuration settings are given. Although one can assume the default values as listed in Table 9, the provided message sets explicitly setup all the parameters in most cases. Thus they can serve as templates for any future custom development by the user.

Note: Surround Effects parameters are sampled by the CS492x only once just after kickstart (after a download/soft reset or application restart). These parameters cannot be varied at run-time. Therefore, to change modes, one has to issue

a soft reset or application restart, reconfigure hardware settings, issue the new Effects messages and then kickstart.

The included effects modes were all derived from one of two basic filter structures:

A basic 7-point FIR filter with taps at: T0-6 = 0, 955, 1699, 1867, 1987, 3355, 3821 samples. The corresponding coefficients are C0-6 = 0.9804, 1.0, 0.802, 0.6225, 0.7049, 0.2618, 0.2373.

A 19-point FIR filter with taps at: T0-18 = 0, 206, 1032, 1080, 1286, 1296, 1430, 2198, 2328, 2746, 2818, 2856, 2938, 3394, 3398, 3485, 3557, 3614, 3826 samples. The corresponding coefficients are C0-18 = 1.0, 0.841, 0.504, 0.491, 0.379, 0.380, 0.346, 0.289, 0.272, 0.192, 0.193, 0.217, 0.181, 0.180, 0.181, 0.176, 0.142, 0.167, 0.134.

By scaling these taps and coefficients, different effects can be created. By decreasing (or increasing) the coefficients, the liveness of the room can be increased (or decreased). Scaling the tap locations will increase/decrease the length of early reflections. Delaying the taps (offsetting all taps by the same amount of time) causes a delay in the early reflections.

A brief description of each surround effect mode is provided below. Note that the modes have been designed for Fs=48 KHz. When used for Fs=44.1 KHz, the change is largely inaudible and acceptable in most cases. If necessary, the user may scale the delay and reverb time parameters (if used) appropriately.

The effects can be broken into two categories, reverberative and non reverberative. Reverberative affects have late reflections while non reverberative effects. The next two sections list the different configuration examples sent with the code categorized by whether the effect is reverberative or not. Non-reverberative effects are much more subtle while reverberative effects are more extreme

and can sometimes interfere with the actual source material. The naming of each mode is very subjective and only is intended as a suggestion.

“COEFMAKR.EXE and EFFECTS.EXE” describes a program that can create effects configuration files for the user.

### 5.4.5 Non - Reverberative Effects

Non-reverberative effects provide a subtle effect through remapping and/or redirecting the stereo input. In this way the end user can fill all of the speakers in a surround system without significantly altering the content of the source. Delays are used in some modes to simulate a larger listening area.

#### 5.4.5.1 Passthru

This mode does not engage the Effects Manager and thus has no reverberation or early reflections. The default setup of Output Channel Mixer is used, i.e.

$$\begin{aligned} L\_Out &= L \\ R\_Out &= R \\ C\_Out &= Ls\_Out = Rs\_Out = LFE\_Out = 0.5 L + \\ &\quad 0.5 R \end{aligned}$$

#### 5.4.5.2 Mono

This mode has no reverberation or early reflections and only uses the Output Channel Mixer. The Output Channel Mixer is setup for:

$$\begin{aligned} L\_Out &= R\_Out = C\_Out = Ls\_Out = Rs\_Out = 0.5 \\ &\quad L + 0.5 R \\ LFE\_Out &= 0 \end{aligned}$$

#### 5.4.5.3 Classical

This mode has no reverberation and only uses FIR and the Output Channel Mixer. The mono mix is setup as (L+R)/2 with no LPF contribution and feeds FIR, a 5-tap filter.

The Output Channel Mixer is setup for:

$$L\_Out = L$$

$$R\_Out = R$$

$$C\_Out = Ls\_Out = Rs\_Out = FIR$$

$$LFE\_Out = 0$$

#### 5.4.5.4 *Panorama*

This mode has no reverberation and only uses FIR and the Output Channel Mixer. The mono mix is setup as  $(L-R)/2$  with no LPF contribution and feeds FIR, the same 5-tap filter as in Classical mode.

The Output Channel Mixer is setup for:

$$L\_Out = L - 0.31623 R$$

$$R\_Out = R - 0.31623 L$$

$$C\_Out = 0.5 L + 0.5 R$$

$$Ls\_Out = Rs\_Out = FIR$$

$$LFE\_Out = 0$$

#### 5.4.5.5 *Movie 1-4*

These modes have no reverberation and only uses FIR and the Output Channel Mixer. Each mode has a different FIR delay tap/coefficient setting, but all have the following generic setup. The mono mix is setup as  $(L+R)/2$  with no LPF contribution and feeds FIR, a 4 or 5-tap filter depending on the mode.

The Output Channel Mixer is setup for:

$$L\_Out = L$$

$$R\_Out = R$$

$$C\_Out = Ls\_Out = Rs\_Out = FIR$$

$$LFE\_Out = 0$$

#### 5.4.5.6 *Music 1-4*

These modes are identical to Movie1-4 modes above, except that the Output Channel Mixer is setup for  $C\_Out = 0$ .

### 5.4.6 *Reverberative Effects*

Reverberative effects combine the different filters discussed in sections “Basic Crystal Surround Effects Explanation” and “Creating Custom Effects Using Advanced Settings” to create a more obvious effect. These effects can be used to fill out the incoming source if so desired. Care should be taken as some reverberative effects can be so strong that they can interfere with the original stereo source signal. The effects listed below start with the subtle effects (Room) and graduate to very obvious (Stadium).

To simplify the use of the following configuration files, the following naming convention has been used:

**tErGaKiN.cfg**

**E** - This number represents the FIR filter type used for early reflections. Type 1 is a 7-Tap, and Type 2 is a 19-Tap, per discussion above. The 19-Tap filter will make the early reflections sound much thicker. This is useful when trying to create an echo pattern that sounds thick.

**G** - This represents the amount of Initial Delay and Early Reflection time. A shorter Early Reflection time will make the room size sound smaller, and a longer time will make the room sound larger. Initial Delay until Early Reflections will cause the room to sound as if the source is further from the walls.

**G == 1:** Early Reflections last for 40 ms, and the Initial Delay until Early Reflections is 20 ms.

**G == 2:** Early Reflections last for 80 ms (default), and the Initial Delay until Early Reflections is 10 ms.

**G == 3:** Early Reflections last for 115 ms, and there is no Initial Delay until Early Reflections (0 ms).



**K** - This represents the amount of time for After Reflections. The After Reflections produced by Reverb Module will produce a dense tail to the effect, and this parameter will influence the length of that tail.

- K == 1: Reverb time is set to 0.25.
- K == 2: Reverb time is set to 0.5 (default).
- K == 3: Reverb time is set to 0.8.
- K == 4: Reverb time is set to 0.98. This will produce a reverb time that is very large. NOTE: Only one example with K==4 is presented here (Stadium).

**N** - This represents the After Reflection (Reverb) Liveness. To absorb more of the high frequency content of the After Reflections (to sound less bright), the number should be higher. To increase the high frequency content (to sound more bright), the number should be lower.

- For N == 1:Reverb Liveness (After Reflection Liveness) is set to 0.2
- For N == 2:Reverb Liveness (After Reflection Liveness) is set to 0.8

To simplify the use of the filters, a description of some of the filters follows. Note that configuration files are supplied for many more variations of the parameters specified above. The supplied configuration files are intended to span a range of effects that can be created using the `coefmakr.exe` program. To create intermediate sounding effects, one could simply interpolate the parameters specified to the `coefmakr.exe` program and use the corresponding configuration files. "COEFMAKR.EXE and EFFECTS.EXE" has more details.

For all Reverberative Modes, the following mixlevels (default in `coefmakr.exe`) are used:

L\_Mono\_Mixlevel = 0.5  
R\_Mono\_Mixlevel = 0.5  
Direct\_FIR\_Mixlevel = 0.875  
LPF\_FIR\_Mixlevel = 0.125  
  
Reverb\_Direct\_Mixlevel = 1  
Reverb\_FIR\_Mixlevel = 0  
  
L\_to\_L\_Out = 0.875  
R\_to\_L\_Out = 0  
L\_to\_C\_Out = 0.4375  
R\_to\_C\_Out = 0.4375  
L\_to\_R\_Out = 0  
R\_to\_R\_Out = 0.875  
L\_to\_Ls\_Out = 0.125  
R\_to\_Ls\_Out = 0.125  
L\_to\_Rs\_Out = 0.125  
R\_to\_Rs\_Out = 0.125  
L\_to\_LFE\_Out = 0.5  
R\_to\_LFE\_Out = 0.5  
  
FIR\_to\_L\_Out = 0.0833  
Reverb\_to\_L\_Out = 0.0417  
FIR\_to\_C\_Out = 0.0417  
Reverb\_to\_C\_Out = 0.0213  
FIR\_to\_R\_Out = 0.0833  
Reverb\_to\_R\_Out = 0.0417  
FIR\_to\_Ls\_Out = 0.5  
Reverb\_to\_Ls\_Out = 0.25  
FIR\_to\_Rs\_Out = 0.5  
Reverb\_to\_Rs\_Out = 0.25

#### 5.4.6.1 *t1r1a1i1.cfg - Room 1*

This effect mode simulates the reflections from a small room. The room is not live, and has small Early and Late Reflection Times. The type 1 filter is used, providing a very mild effect.

#### 5.4.6.2 *t2r3a1i2.cfg - Room 2*

This effect mode simulates the reflections from a small room, but uses a larger FIR filter spread out over a larger time to thicken the sound of the early reflections.

#### 5.4.6.3 *t1r1a2i1.cfg - Theater 1*

This effect mode simulates the reflections as in a medium size theater. The Early Reflections will be short, but very present. The Late Reflections are bright, but are present for medium amount of time.

#### 5.4.6.4 *t2r3a2i2.cfg -- Theater 2*

The Early Reflections in this effect are dominant and long, allowing for a very thick sounding Early Reflection. This effect mode has a short Reverb Time but large Reverb Liveness, making the sound less bright.

#### 5.4.6.5 *t1r1a3i1.cfg -- Hall 1*

This effect has a lot of Early and Late Reflections, with a large Reverb Time, but a small Reverb Liveness. This effect will have a very large sound, and the high frequencies will be present in the After Reflections.

#### 5.4.6.6 *t2r3a3i2.cfg -- Hall 2*

This effect, similar to Hall1, has the same amount of Early and Late Reflections. In this effect mode, the higher frequencies are less present, resulting in a less bright sound in the reflections.

#### 5.4.6.7 *t2r3a4i2.cfg -- Stadium*

This effect, utilizing a maximum of Early and Late Reflections, has a long decay time. The sound is highly decorrelate, making the echoes sound

muddy. The long decay time provides for an effect of the sound bouncing off many hard walls, and a very bright tone is used. This effect is presented merely to demonstrate the extreme capability of the Crystal scheme, and may require modification before use in a practical system.

### 5.4.7 *Effects Configuration Files*

COEFMAKR.EXE is a program written to assist development of unique effects configurations. It is documented in “COEFMAKR.EXE and EFFECTS.EXE”. This program can be used to create configuration files for the effects modes presented in the previous sections. The configuration files have been omitted from this document to conserve space. The COEFMAKR.EXE program along with the configuration files are in a zip file available at [www.crystal.com](http://www.crystal.com) under the CS4923 page. The configuration file contains paired 3 byte words which are the actual commands to be sent to the CS4923. These messages correlate with the parameters in table 9. In addition comment lines marked by a # symbol are included to help explain the configuration messages. When porting the messages to code for a host microcontroller, the comment lines should be omitted and care should be taken to keep the paired 3 byte words together in the correct order. The configuration files also contain a “kickstart” as the last command (see “Audio Manager” for more information).

### 5.5 COEFMAKR.EXE and EFFECTS.EXE

COEFMAKR.EXE (Coefficient Maker) is a simple, easy to use program for generating configuration files used in conjunction with Crystal Effects code for the CS492X series of Surround Sound DSP's. Please contact your FAE for the latest version of this development software.

The program takes all of the configuration options on the command line, and all of these options can be displayed with the -? command line switch.

Note that since there are 56 command line options (a minimum of 1 must be specified), to display Mix Level options, the -x? switch must be used. For all Advanced Options, the -o? switch must be used.

EFFECTS.EXE is a GUI version of COEFMAKR.EXE which allows the user to edit effects configuration files and try them on one of the CRD4923 or CDB4923 application boards real time. The program mirrors the block diagrams shown in figures 9, 10 and 11 and allows the user controls to the parameters discussed in section 5.3.

Below is a short description of the files and what they contain:

- COEFMAKR.EXE - The main program for the coefficient maker.
- FIR.CFG - This file contains the templates for 2 basic FIR (Early Reflection) filters, which are

scaled by coefmakr into the FIR section of the effects code. Additional filters can be added to this file, or the file can be replaced by one of the same structure. However, if a different file is used, it must be specified in the command line (-c option). For COEFMAKR to run correctly, this file must reside in the same directory.

- MAKETEST.BAT - This batchfile will generate all of the default configuration files shown in section 5.3 (with reverb) using COEFMAKR.EXE.
- EFFECTS.EXE - GUI version of COEFMAKR.EXE

More files may need to be present. Please read the README.TXT file in the zip file from which the programs were extracted for more information on these programs.

\*\*\*\*\*

Command line options:

Basic Options:

-----

Usage: coefmakr [-qfile.out -cfilter.cfg -fx -dy -tz -rw -lb -mu -v -xelevel]

- qfile.out : file.out = Config file to be generated(default=file.out)
- cfilter.cfg : filter.cfg = Filter parameter file(default=fir.cfg)
- fb : b = Sampling Rate (32, 44, 48\* KHz)
- du : u = Initial Delay of Early Reflections (0\*,1,...,115ms)
- te : e = FIR Filter Type (1\* or 2)
- rg : g = Early Reflection Time (0,1,...,80\*,... ms)
- lh : h = Liveness of Early Reflection (0,...,1\*)
- mj : j = Delay Until After Reflections (0,1,...,100\*,...ms)
- ak : k = After Reflection Time (0,...,0.5\*,...,1)
- in : n = After Reflection Liveness (0,...,1\*)
- xp : p = Mix Level(s) to Change (? for options)
- os : s = Advanced Options (? for options)
- v = Run in non-verbose mode

-x Options:

-----

Usage of Mixlevel:

a = L\_Mono\_Mixlevel(-1..0.5\*..1)

b = R\_Mono\_Mixlevel(-1..0.5\*..1)

c = Direct\_FIR\_Mixlevel(-1..0.875\*..1) d = LPF\_FIR\_Mixlevel(-1..0.125\*..1)

e = Reverb\_Direct\_Mixlevel(-1..0.875\*..1)

f = Reverb\_FIR\_Mixlevel(-1..0\*..1)

g = L\_to\_L\_Out(-1..0.875\*..1)

h = R\_to\_L\_Out(-1..0\*..1)

---

i = L\_to\_C\_Out(-1..0.4375\*..1)  
j = R\_to\_C\_Out(-1..0.4375\*..1)  
k = L\_to\_R\_Out(-1..0\*..1)  
l = R\_to\_R\_Out(-1..0.875\*..1)  
m = L\_to\_Ls\_Out(-1..0.125\*..1)  
n = R\_to\_Ls\_Out(-1..0.125\*..1)  
o = L\_to\_Rs\_Out(-1..0.125\*..1)  
p = R\_to\_Rs\_Out(-1..0.125\*..1)  
q = L\_to\_LFE\_Out(-1..0.5\*..1)  
r = R\_to\_LFE\_Out(-1..0.5\*..1)  
s = FIR\_to\_L\_Out(-1..0.0625\*..1)  
t = Reverb\_to\_L\_Out(-1..0.0625\*..1)  
u = FIR\_to\_C\_Out(-1..0.03125\*..1)  
v = Reverb\_to\_C\_Out(-1..0.03125\*..1)  
w = FIR\_to\_R\_Out(-1..0.0625\*..1)  
x = Reverb\_to\_R\_Out(-1..0.0625\*..1)  
y = FIR\_to\_Ls\_Out(-1..0.375\*..1)  
z = Reverb\_to\_Ls\_Out(-1..0.375\*..1)  
1 = FIR\_to\_Rs\_Out(-1..0.375\*..1)  
2 = Reverb\_to\_Rs\_Out(-1..0.375\*..1)

-o Options:

-----

Usage of Advanced Options:

a = Comb\_0\_Delay\_Size(1.....1103\*)  
b = Comb\_1\_Delay\_Size(1.....1213\*)  
c = Comb\_2\_Delay\_Size(1.....1321\*)  
d = Comb\_3\_Delay\_Size(1.....1433\*)  
e = Comb\_4\_Delay\_Size(1.....1543\*)  
f = Comb\_5\_Delay\_Size(1.....1657\*)  
g = Comb\_0\_g1(0...0.0895\*...1)

---

h = Comb\_1\_g1(0...0.0950\*...1)  
i = Comb\_2\_g1(0...0.1002\*...1)  
j = Comb\_3\_g1(0...0.1055\*...1)  
k = Comb\_4\_g1(0...0.1106\*...1)  
l = Comb\_5\_g1(0...0.1160\*...1)  
m = Comb\_0\_Mixlevel(0...0.9\*...1)  
n = Comb\_1\_Mixlevel(0...0.9\*...1)  
o = Comb\_2\_Mixlevel(0...0.9\*...1)  
p = Comb\_3\_Mixlevel(0...0.9\*...1)  
q = Comb\_4\_Mixlevel(0...0.9\*...1)  
r = Comb\_5\_Mixlevel(0...0.9\*...1)  
s = Allpass\_g1(0...0.7\*...1)  
t = Allpass\_g2(0...0.51\*...1)

The following are the command line parameters to use with the COEFMAKR.EXE program to create the configuration files for the effects modes presented in “t1r1a1i1.cfg - Room 1” through “t2r3a4i2.cfg -- Stadium”.

```
coefmakr -qt1r1a1i1.cfg -r40 -d20 -a0.25 -v  
coefmakr -qt2r3a1i2.cfg -t2 -r115 -i0.6 -v  
coefmakr -qt1r1a2i1.cfg -r40 -d20 -v  
coefmakr -qt2r3a2i2.cfg -t2 -r115 -i0.6 -v  
coefmakr -qt1r1a3i1.cfg -r40 -d20 -a0.8 -v  
coefmakr -qt2r3a3i2.cfg -t2 -r115 -a0.8 -i0.6 -v  
coefmakr -qt2r3a4i2.cfg -r115 -l0.5 -m120 -a0.97 -i0.8 -v
```

• **Notes** •

---

SMART  
Analog™