

APPLICATION NOTE

MPEG USER'S GUIDE FOR THE CS4925 AND CS4927

Contents

- MPEG Multi-Channel Description

-

- Software Naming Convention
- Document Strategy
- Hardware Configurations Supported by the CS492X and the MPEG Code
- Understanding Application Messaging (Including Write Sessions, Solicited and Unsolicited Read Message)
- How to Control Application Modules such as:
 - Audio Manager
 - MPEG Manager
 - Pro Logic Manager
 - Generalized Bass Manager
- Application Configuration Examples

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 features included in the standard MPEG Multi-Channel application for IEC61937 compliant bitstreams.

This document covers code supported by CS492X revision D silicon.

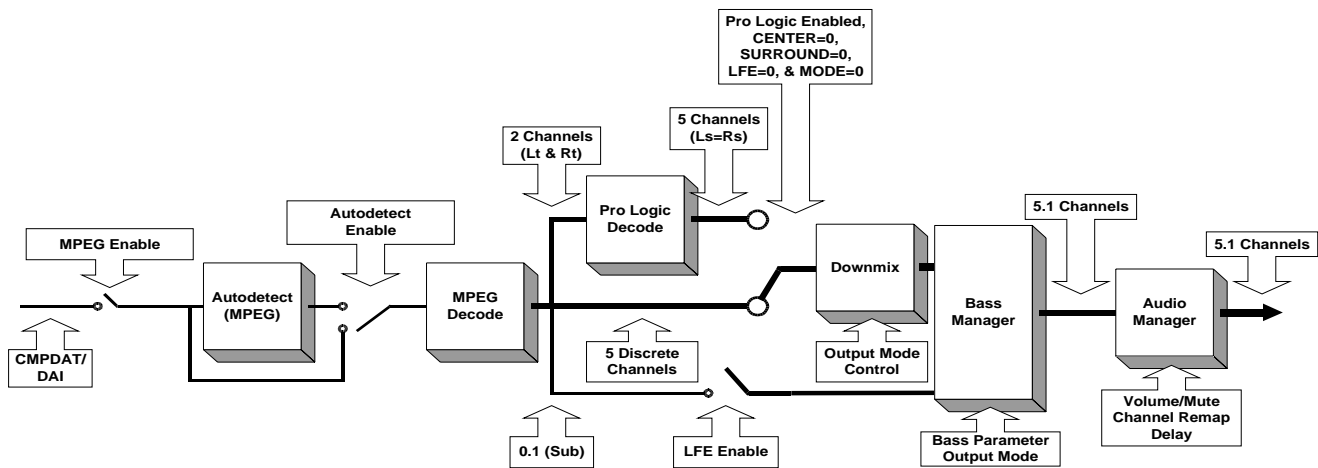


TABLE OF CONTENTS

1. OVERVIEW	4
1.1 Multi-Channel Decoder Family of Parts	4
1.2 Document Strategy	5
1.2.1 Hardware Documentation	5
1.2.2 CS4923/4/5/6/7/8/9 Application Code User's Guides	5
1.3 Using the CS4923/4/5/6/7/8/9	6
1.4 Software Naming Convention	6
1.5 MPEG Description	6
2. HARDWARE CONFIGURATION	7
2.1 Supported Input/Output Modes	7
3. APPLICATION MESSAGING	8
3.1 Indexed Module Communication Protocol	8
3.1.1 Write Session	8
3.1.2 Solicited Read Message Format	8
3.2 Unsolicited Read Message Format	9
4. APPLICATION MODULES	10
4.1 Audio Manager	11
4.1.1 Audio_Control: (Index 0x00)	12
4.2 MPEG Manager	14
4.2.1 MPEG_Control	15
4.2.2 MPEG Stream Information	15
4.3 Pro Logic Manager	16
4.4 Generalized Bass Manager	17
4.5 Unsolicited Messages (Read-Only)	21
4.5.1 Autodetect Operation	22
4.5.2 Special Considerations For Autodetect	23
5. APPLICATION CONFIGURATION EXAMPLES	25
5.1 MPEG decode with Autodetect	25

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/>

AC-3, Dolby Digital and Dolby Pro Logic are registered trademarks of Dolby Laboratories Licensing Corporation.
DTS is a registered trademark of DTS, Inc.
The MPEG Logo is a registered trademark of Philips Electronics.

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>.

LIST OF FIGURES

Figure 1. MPEG Functional Block Diagram	6
Figure 2. Filter Topology	20
Figure 3. Bass Manager Processing Unit	20
Figure 4. Subwoofer Summing Module	20
Figure 5. Autodetect Flow Chart	24
Figure 6. Typical Download and Configuration	24
Figure 7. Pseudocode Example to Configure Part MPEG Decode with Autodetect .	25

LIST OF TABLES

Table 1. Input/Output Configurations Supported by MPEG.....	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	11
Table 7. MPEG Manager.....	14
Table 8. Pro Logic Manager	16
Table 9. Bass Manager	17
Table 10. Unsolicited Messages.....	21
Table 11. Enabling MPEG decode with Autodetect.....	25

1. OVERVIEW

The CS4923/4/5/6/7/8/9 is a family of system on a chip solutions for multi-channel audio decompression and digital signal processing. Since the part is RAM-based, a download of application software is required each time the CS4923/4/5/6/7/8/9 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 CS4923/4/5/6/7/8/9. In systems where A/V synchronization is required from the CS4923/4/5/6/7/8/9, 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 CS4923/4/5/6/7/8/9 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 user's guide covers code that provides MPEG-1, layer 2 (Stereo) and MPEG-2, layer 2 Multi-Channel and Stereo decoding for IEC61937 compliant bitstreams.

IEC61937 is a protocol for delivering compressed audio over a consumer IEC60958 or S/PDIF interface.

1.1 Multi-Channel Decoder Family of Parts

CS4923 - Dolby Digital[®] 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 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 Multi-Channel DVD Audio Decoder.

The CS4925 supports both Dolby Digital and MPEG-2 multi-channel formats. For both Dolby Digital and MPEG-2 multi-channel, 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 Multi-Channel 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 Multi-Channel Decoder.

The CS4927 supports MPEG-2 multi-channel decoding and should be used in applications where Dolby Digital decoding is not necessary. For MPEG-2 multi-channel 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 Multi-Channel Decoder.

The CS4928 supports DTS multi-channel decoding and should be used in applications where Dolby Digital decoding is not necessary. For DTS multi-channel decoding, post processing includes bass manage-

ment. 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 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 CS4923/4/5/6/7/8/9. 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.

1.2.2 CS4923/4/5/6/7/8/9 Application Code User’s Guides

The following application notes describe the application codes used with the CS4923/4/5/6/7/8/9. Whenever an application code user’s guide is re-

ferred 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 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[™], QSURround[™] and VMAx[™].

AN121 - MPEG User’s Guide for the CS4925. This document covers the features available in the MPEG Multi-Channel code including delays, bass management, Pro Logic, and MPEG processing features.

AN122 - DTS 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 CS4923/4/5/6/7/8. This code covers the different Stereo PCM to surround effects processing code. Optional appendices are available that document Crystal Original Surround, Circle Surround and Lexicon Logic.

AN140 - Broadcast Systems Guide for the CS4923/4/5/6/7/8/9. This guide describes all application code (e.g. Dolby Digital, MPEG, AAC) 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 CS4923/4/5/6/7/8/9

No matter what application is being used on the chip, the following four steps are always followed to use the CS4923/4/5/6/7/8/9 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 CS4923/4/5/6/7/8/9 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 MPEG Description

MPEG is a digital audio compression algorithm developed by the Moving Pictures Experts Group for up to 5.1 channels of audio. This user's guide covers the software designed to run on the CS4925/7 that decodes MPEG. This user's guide covers all code that is named with AAAA == MPG_ or MPS_, and CC==25 or 27 as described in the previous section.

Along with MPEG this document will describe how to enable Dolby Pro Logic Decode and bass management among other features. It is assumed that the reader is familiar with the features of MPEG as specified in:

- *ISO/IEC 13813-3*, MPEG Audio Coding Standard

Figure 1 is a functional block diagram of the application code and gives an idea of the interaction between the various application modules.

For this document CS4925/7 has been replaced with CS492X for readability. Unless otherwise specified CS492X should be interpreted as applying to both the CS4925 and CS4927.

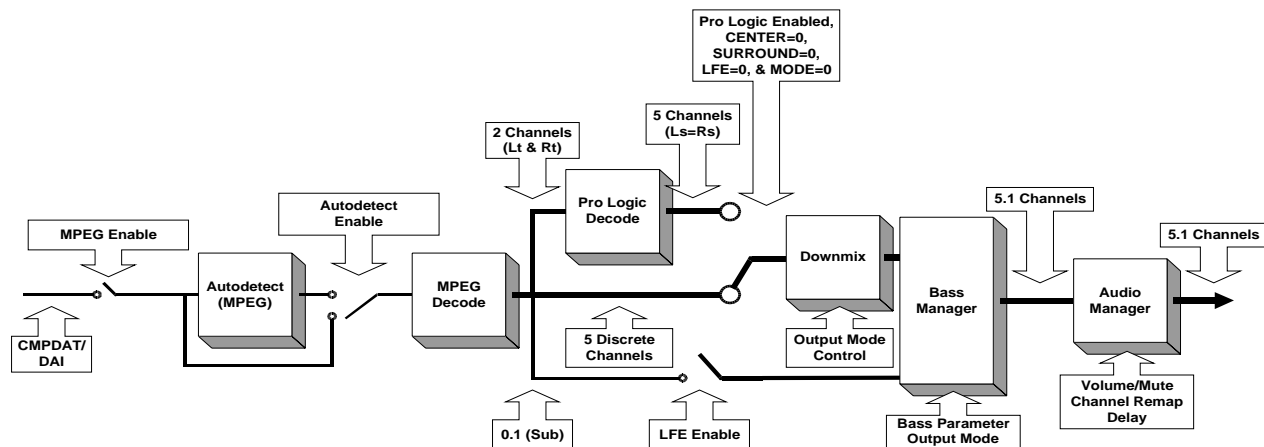


Figure 1. MPEG Functional Block Diagram

2. HARDWARE CONFIGURATION

After download or soft reset, and before kickstarting the application (please see Section 4.1, *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, Table 1 shows the input/output modes are supported by the MPEG application:

I/O Configuration	Description	Supported Modes
INPUTA	Input Data Type	0,1
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 MPEG

3. APPLICATION MESSAGING

While using the CS4925/7 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 after download, e.g. enabling MPEG decode, or changing run-time parameters, e.g. controlling digital volume, the host will use application messaging to communicate with the CS492X.

While communicating with the CS492X using indexed modules, a strict software protocol must be used in conjunction with the hardware protocol discussed in the CS4923/4/5/6/7/8/9 Hardware User's Guide. This section will cover both the format of the messages and the different configuration modules available with the CS492X. It must be stressed that the host must strictly adhere to the hardware and software protocols to insure successful communication.

3.1 Indexed Module Communication Protocol

Each indexed module of the application can be thought of as a block of software registers or variables. The index identifies a unique variable within the module. When the opcode for a module and an

index are combined, a unique variable can be read or written.

This section covers how to communicate with the CS492X using indexed modules. The software protocol is presented for the following types of messages: Write, Solicited Read, Read Response, and Unsolicited Read Response.

When the protocol presented in this section is used with the application modules in Section 4, *Application Modules*, the host will be able to fully configure the application running on the CS492X.

3.1.1 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.

3.1.2 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 CS492X to the host.

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

Write Command Word:

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]															

Write Data Word:

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]																							

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*.

Table 2. Write Message Format

After the host sends the read command word it should wait for the $\overline{\text{INTREQ}}$ line to fall. See the CS4923/4/5/6/7/8/9 Hardware User's Guide 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.

3.2 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 CS4925/7 to indicate a change in the system that must be addressed. One example is when the part is in autode-

tect mode and detects a new stream. An unsolicited read message will be sent by the CS4925/7 to indicate the new stream type.

The 6-byte unsolicited read messages from the CS4925/7 consist of a 3-byte read command word which defines the type of unsolicited message (as described in *Unsolicited Messages (Read-Only)*, Section 4.5) 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 $\overline{\text{INTREQ}}$ is low), the host should read out the 6- byte read unsolicited message.

Table 5 shows the format of an unsolicited read message.

Read Command Word:

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]															

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.

Table 3. Read Command Message Format

Read Response Command Word:

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]															

Read Response Data Word:

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]																							

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*.

Table 4. Read Response Message Format

Unsolicited Read Command Word:

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]															

Unsolicited Read Data Word:

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]																							

OPCODE[7:0] - 8 bit (1 byte) field containing opcode for the Unsolicited Messages module.

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*.

Table 5. Unsolicited Read Message Format

4. APPLICATION MODULES

The block diagram on the cover page of this document accurately portrays the interaction between the various application modules of the MPEG 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 0x1FFFFFF and is equivalent to a value of 0.25.
- The formula for converting variable settings from a hexadecimal number into dB is as follows:

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

$$\text{e.g. } 20 * \log (0x1FFFFFF/0x7FFFFFF) == -12 \text{ dB}$$

Numbers preceded by 0x should be interpreted as hex, and numbers followed by ‘b’ should be interpreted as binary. All values sent to the CS4923/4/5/6/7/8/9 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 0xhhhhhh

Read Request Message = 0x0900HH;

Read Response Message = 0x8900HH 0xhhhhhh

0xHH = index

0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	Audio_Mgr_Control	<p>Bit 15: Audio_Config_Change_Notification_Enable = 0/1= Disable/Enable unsolicited message notification of any change in audio configuration of input bitstream (Valid only for AC-3[®] application). See “Unsolicited Messages (Read-Only)” on page 21 for more details</p> <p>Bit 14: 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>Bit 12: Autodetect_Enable = 0/1= Disable/Enable autodetect functionality.</p> <p>Bit 4: 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>Bit 0: Kickstart_Enable = 0/1 = Disable/Enable Kickstart of application. <i>Default = 0x000000</i></p>
0x04	Ls_Delay	0..15 = The left surround delay (Ls_Delay) +15ms. In other words, if Ls_Delay is set to 10, the Left Surround delay will be set to 25ms. Note: Ls_Delay will only have an effect when Pro Logic is enabled AND when the input stream is MPEG 2/0 encoded. <i>Default* = 0</i>
0x05	Rs_Delay	0..15 = The right surround delay (Rs_Delay) +15ms. In other words, if Rs_Delay is set to 10, the Right Surround delay will be set to 25ms. Note: Rs_Delay will only have an effect when Pro Logic is enabled AND when the input stream is MPEG 2/0 encoded. <i>Default* = 0</i>
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 ^a	0...5 = Channel type ^a 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 ^a	0...5 = Channel type ^a for Digital Audio Output 1. Each channel must be mapped to one and only one unique output. <i>Default* = 2(R)</i>

Table 6. Audio Manager

Index	Variable	Dataword Content
0x10	DAO2_Channel ^a	0...5 = Channel type ^a 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 ^a	0...5 = Channel type ^a 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 ^a	0...5 = Channel type ^a 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 ^a	0...5 = Channel type ^a for Digital Audio Output 5. Each channel must be mapped to one and only one unique output. <i>Default* = 5(LFE)</i>
0x16	Last_Unsolicited_Message	Last Unsolicited Message Value (see Section 4.5) READ ONLY Default=0x000000
0x17	PLL_Register_1	Register 1 value for Phase Locked Loop (no Fs generation). <i>Default* = 0x07f270 (CLKIN=12.288MHz, DSPCLK=60MHz)</i>
0x18	PLL_Register_2	Register 2 value for Phase Locked Loop (no Fs generation). <i>Default* = 0x0001fe (CLKIN=12.288MHz, DSPCLK=60MHz)</i>

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*

Makes the CS492X sensitive to changes in the Audio configuration reported in the input bit-stream. Whenever there is a change in the MODE, EXT_BIT_STREAM_PRESENT, CENTER, SURROUND or LFE variables, the CS492X will issue an unsolicited message (see Unsolicited Messages (Read-Only), Section 4.5).

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.

Notes: 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), Section 4.5). 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 DSP-CLK 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 MPEG Manager

Write Opcode = 0x8C; Read Opcode = 0x0D; Read Response Opcode = 0x8D

Write Message = 0x8C00HH 0xhhhhhh

Read Request Message = 0x0D00HH;

Read Response Message = 0x8D00HH 0xhhhhhh

0xHH = index

0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	MPEG_Control	<p>Bits 7:4: Output_Mode = 0...7</p> <p>0 = 2/0 Lt, Rt Surround Encoded.</p> <p>1 = Reserved.</p> <p>2 = 2/0 L, R</p> <p>3 = 3/0 L, C, R</p> <p>4 = Reserved.</p> <p>5 = Reserved.</p> <p>6 = 2/2 L, R, Ls, Rs (with Pro Logic decode, Ls = Rs = S - 3 dB)</p> <p>7 = 3/2 L, C, R, Ls, Rs (with Pro Logic decode, Ls = Rs = S - 3 dB)</p> <p>Bit 0: MPEG_Enable = 0/1 = Disable/Enable MPEG decoding.</p> <p>Default = 0x000070 [MPG_code]</p> <p>Default = 0x000000 [MPS_code]</p> <p>NOTE: Undesired Channel outputs should be explicitly muted by setting the corresponding Ch_Vol (see Audio Manager) to 0.0</p>
0x01	ID	0 - Extension to lower frequencies, 1 - ISO/IEC 11172-3 or ISO/IEC 13818-3 READ-ONLY
0x02	LAYER	2 bits to indicate which layer is used READ-ONLY
0x03	PROTECTION_BIT	0 - stream is CRC protected, 1 - not CRC protected READ-ONLY
0x04	BITRATE_INDEX	4-bit index to a bitrate table READ-ONLY
0x05	SAMPLING_FREQUENCY	2-bit index to a sampling frequency table READ-ONLY
0x06	PADDING_BIT	0 - no padding, 1 - padding exists READ-ONLY
0x07	PRIVATE_BIT	reserved for non-ISO purposes READ-ONLY
0x08	MODE	2 bits to indicate the audio configuration mode (for 2 channels only) READ-ONLY
0x09	MODE_EXTENSION	2 bits to indicate which sub-bands are in intensity stereo READ-ONLY
0x0a	COPYRIGHT	0 - no copyright, 1 - copyright protected bitstream READ-ONLY
0x0b	ORIGHOME	0 - bitstream is a copy, 1 - bitstream is an original READ-ONLY
0x0c	EMPHASIS	2 bits to indicate which type of de-emphasis is used READ-ONLY
0x0d	EXT_BIT_STREAM_PRESENT	0 - extension bit stream not present, 1 - ext. present READ-ONLY
0x0e	N_AD_BYTES	8 bits: number of bytes used in MPEG-1 base frame for ancillary data field if an extension bit stream exists READ-ONLY

Table 7. MPEG Manager

Index	Variable	Dataword Content
0x0f	CENTER	2 bits to indicate presence and bandwidth of center channel READ-ONLY
0x10	SURROUND	2 bits to indicate whether surround channels are present or if secondary stereo exists READ-ONLY
0x11	LFE	0 - lfe not present, 1 - lfe present READ-ONLY
0x12	AUDIO_MIX	1 bit to indicate whether the signal is mixed for large listening rooms READ-ONLY
0x13	DEMATRIX_PROCEDURE	2 bits to indicate which de-matrix procedure is used READ-ONLY
0x14	NO_OF_MULTI_LINGUAL_CH	3 bits: number of multi-lingual channels present in the multi-channel extension READ-ONLY
0x15	MULTI_LINGUAL_FS	1 bit to indicate if fs is the same as in the main audio READ-ONLY
0x16	MULTI_LINGUAL_LAYER	1 bit to indicate which layer is used READ-ONLY
0x17	COPYRIGHT_ID_BIT	1 bit: part of a 72-bit copyright identification number READ-ONLY
0x18	COPYRIGHT_ID_START	0 - no start of copyright identification in this frame, 1 - start of copyright identification in this audio frame READ-ONLY

Table 7. MPEG Manager (Continued)

4.2.1 MPEG_Control

Bits 7:4 *Output_Mode*

These bits set up the output speaker downmix configuration. When configuring the Output _Mode, first the individual volumes of each undesired channel should be set to 0. Next, the desired output mode should be specified. This should be done before kickstart of the application. For example, if output mode 3/0 (L, C, R) is chosen, Ls and Rs volume must explicitly be set to 0 in the Audio Manager module to guarantee no output on those channels.

0x0 = 2/0 L, R Dolby Surround compatible

0x2 = 2/0 L, R

0x3 = 3/0 L, C, R

0x6 = 2/2 L, R, Ls, Rs (with Pro Logic decode, Ls = Rs = S - 3 dB)

0x7 = 3/2 L, C, R, Ls, Rs (with Pro Logic decode, Ls = Rs = S - 3 dB)

4.2.2 MPEG Stream Information

Table 7 lists out a complete set of READ-ONLY variables that are reported to the host by the MPEG Decoder module. These can be read out using the appropriate Read Request, and Read Response sessions.

For example, to read the MODE value, host should issue the 3-byte Read Request 0x0D0008 and then read the 6-byte Read Response 0x8D0008, 0x00000h, where the dataword (latter 3 bytes) indicates the reported value of MODE. Please refer to the MPEG Specification ISO/IEC 11172-3 and ISO/IEC 13818-3 for more information on these stream variables.

4.3 Pro Logic Manager

Write Opcode = 0x8E; Read Opcode = 0x0F; Read Response Opcode = 0x8F

Write = 0x8E00HH 0xhhhhh

Read Request = 0x0F00HH;

Read Response = 0x8F00HH 0xhhhhh

0xHH = index

0xhhhhh = data value

Index	Variable	Dataword Content
0x00	ProLogic_Control	Bit 8: Autobalance_Enable = 0/1= Disable/Enable Autobalance. Bit 4: Btype_NR_Enable = 0/1= Disable/Enable B-Type Noise Reduction. Bit 0: ProLogic_Enable = 0/1 = Disable/Enable Pro Logic decoding. Default = 0x000000

Table 8. Pro Logic Manager

Phantom modes are enabled by selecting output modes which do not include a center channel output. The center channel audio is mixed into the left and right speakers. Phantom modes would include:

Output_Mode = 0x6 2/2 L, R, Ls, Rs (Ls = Rs = S - 3dB)

Wide mode is used for audio outputs which are connected to a full-bandwidth speaker. Wide mode for a particular channel can be obtained by disabling the high pass filter for that channel within the bass manager module.

4.4 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	Dataword Content
0x00	Bass_Mgr_Control	Bit 16: LFE_Only_Sub = 0/1 = Disable/Enable LFE Only to subwoofer. Bit 14: 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. Bit 12: LPF = 0/1 = Disable/Enable LPF at output of summer. Bit 8: HPF_L = 0/1 = Disable/Enable L channel HPF. Bit 7: HPF_C = 0/1 = Disable/Enable C channel HPF. Bit 6: HPF_R = 0/1 = Disable/Enable R channel HPF. Bit 5: HPF_Ls = 0/1 = Disable/Enable Ls channel HPF. Bit 4: HPF_Rs = 0/1 = Disable/Enable Rs channel HPF. Bit 0: 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. Default* = 0x7FFFFFFF (0 dB)
0x02	Input_C_Level	0x000000-0x7FFFFFFF = Level adjustment for input C channel input pass-through. Default* = 0x7FFFFFFF (0 dB)
0x03	Input_R_Level	0x000000-0x7FFFFFFF = Level adjustment for input R channel input pass-through. Default* = 0x7FFFFFFF (0 dB)
0x04	Input_Ls_Level	0x000000-0x7FFFFFFF = Level adjustment for input Ls channel pass-through. Default* = 0x7FFFFFFF (0 dB)
0x05	Input_Rs_Level	0x000000-0x7FFFFFFF = Level adjustment for input Rs channel pass-through. Default* = 0x7FFFFFFF (0 dB)
0x06	Input_LFE_Level	0x000000-0x7FFFFFFF = Level adjustment for input LFE channel pass-through. Default* = 0x7FFFFFFF (0 dB)
0x07	Sum_Output_L_Level	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to L channel output. Default* = 0x000000 (-∞ dB)
0x08	Sum_Output_C_Level	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to C channel output. Default* = 0x000000 (-∞ dB)
0x09	Sum_Output_R_Level	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to R channel output. Default* = 0x000000 (-∞ dB)
0x0a	Sum_Output_Ls_Level	0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to Ls channel output. Default* = 0x000000 (-∞ dB)

Table 9. Bass Manager

Index	Variable	Dataword Content
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>
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>

Table 9. Bass Manager (Continued)

Index	Variable	Dataword Content
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* = 0x7EEEA0</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>
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 9. 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 100 Hz. 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 ob-

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

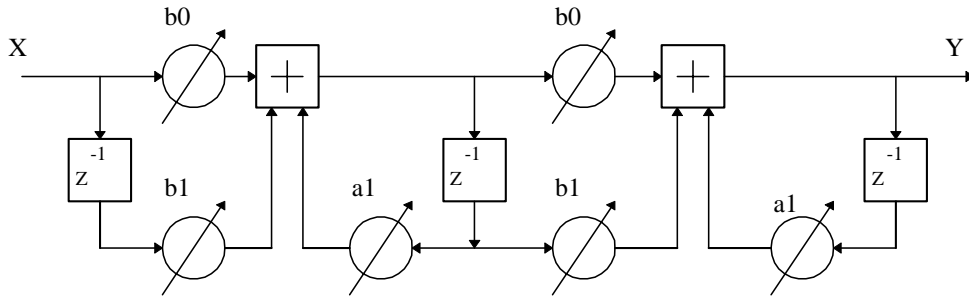


Figure 2. Filter Topology

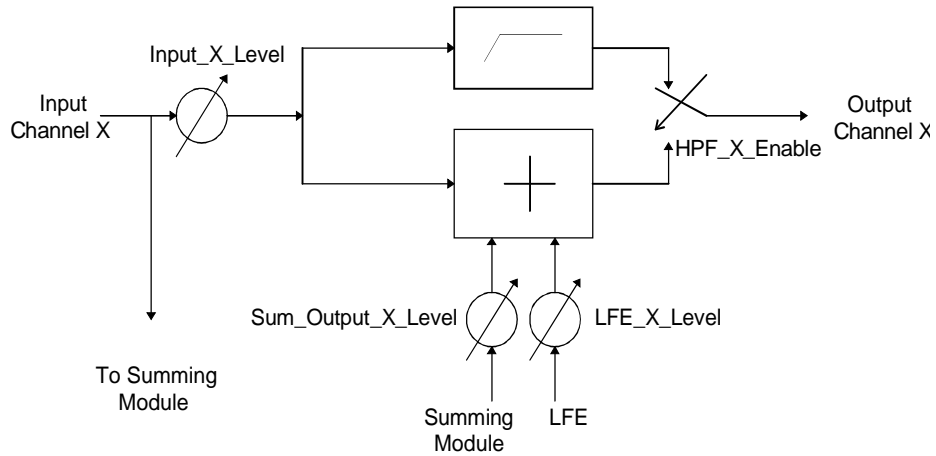


Figure 3. Bass Manager Processing Unit

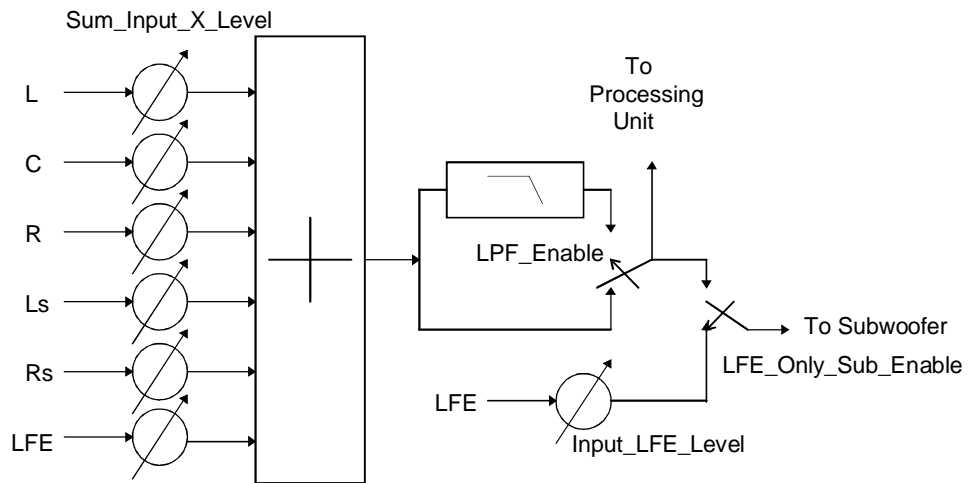


Figure 4. Subwoofer Summing Module

4.5 Unsolicited Messages (Read-Only)

No Write Message. No Read Request.

Unsolicited Read Response = 0x8700HH 0xhhhhh

0xHH = index,

0xhhhhh = data value

Index	Variable	Dataword Content
0x00	Autodetect_Response	<p>Bit 23 = Decodable_Stream_Flag= 0/1 = This stream is not/is decodable by the application (no need for new download if 1).</p> <p>Bits 22:6 = Reserved.</p> <p>Bit 5 = Non_IEC61937_Stream_Flag= 1/0 = This stream is not/is IEC61937 compressed data.</p> <p>If Non_IEC61937_Stream_Flag=1</p> <p>Bits 4:0 = 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>If Non_IEC61937_Stream_Flag=0</p> <p>Bits 4:0 = 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 spec.(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 - 0x1F = Reserved</p>
0x03	Audio_Configuration_Changed	<p>Bits 23:0 = Reserved. The DSP will send out this message to indicate that one of the following has changed in the MPEG stream: MODE, EXT_BIT_STREAM_PRESENT, CENTER, SURROUND, LFE. It is the host's responsibility to check exactly what has changed.</p>
0x10	PLL_Out_Of_Lock	<p>Bit 23 = 1</p> <p>Bits 22:0 = Reserved.</p>

Table 10. Unsolicited Messages

4.5.1 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_****.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 4.1, *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 Section 4.1, *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 AC-3 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 runtime, 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.

4.5.2 Special Considerations For Autodetect

Autodetect should only be enabled or disabled when sending the kickstart command. Changing the state of Autodetect at any other time can produce unpredictable results as it is a function of the input and the application that is currently enabled. If the host needs to change the state of Autodetect at run time for some reason, a soft reset should be sent.

For the MPEG code, MPEG decode must be enabled *before* the autodetect enable is sent with the kickstart. If the Autodetect function senses MPEG

data, then the appropriate Unsolicited Message is sent and decode will start automatically without any direction from the host.

In summary, the following is the recommended procedure when using autodetect. Please see Figure 6, 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)
- 3) Application Configuration (including enable of desired application)
- 4) Kickstart with autodetection enabled.

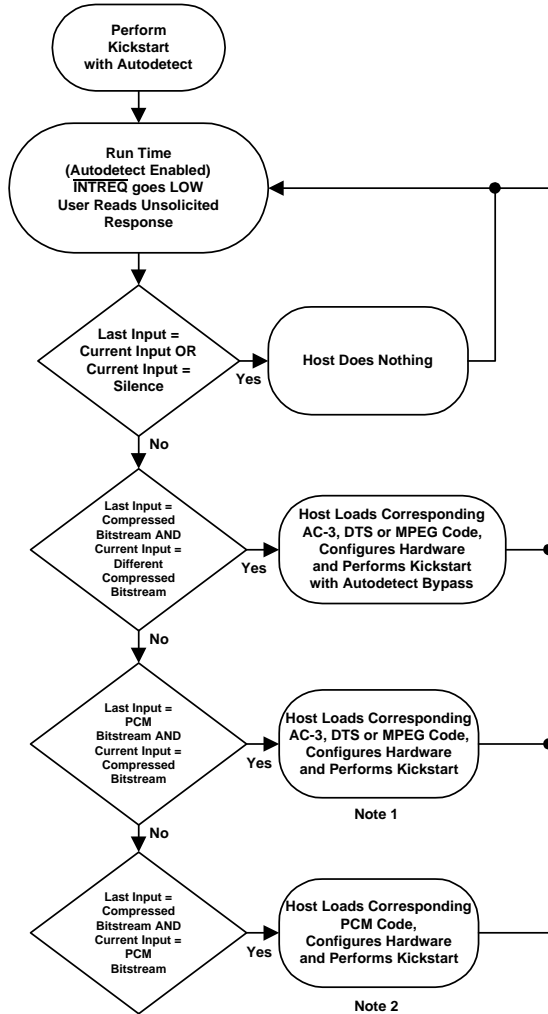


Figure 5. Autodetect Flow Chart

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.

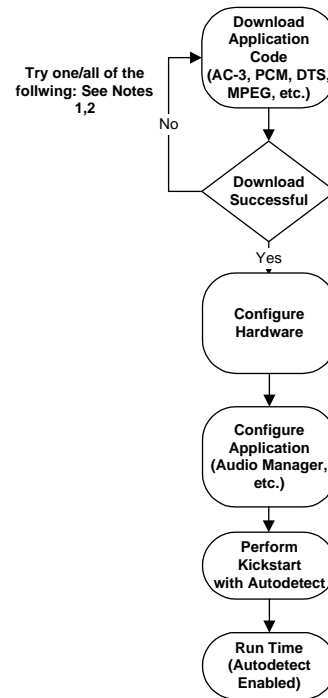


Figure 6. Typical Download and Configuration

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. APPLICATION CONFIGURATION EXAMPLES

This section covers various application modes available with the CS4925/7. Although many application modes are possible, only the most common modes have been presented here as examples. This should not be considered the limit of what modes to run the part in as the system designer can configure the CS4925/7 to fit the needs of the system.

Table 11 in the following section provides the message to be sent and what the message is doing. Care should be taken to guarantee that the correct value is sent directly after the associated opcode and in-

dex word. The command to kickstart the application should always be sent last.

5.1 MPEG decode with Autodetect

In this mode the input should be compressed MPEG. The data will be taken from the compressed input port as designated by the hardware configuration message. Table 11 gives a description of MPEG decode with Autodetect. If the input on this port is not MPEG the Autodetect function will notify the host as described in section 4.5.

Figure 7 example of pseudocode to configure either part for basic MPEG decode. This pseudocode can be used as a template for other configurations by swapping out the message array and the size.

Module	Index	Description	Opcode & Index	Value
MPEG	MPEG Control	Output Mode 3/2 MPEG Enabled	0x8C0000	0x000071
Audio Manager	Audio Manager Control	Autodetect Enabled PLL Enabled Kickstart Application	0x880000	0x001011

Table 11. Enabling MPEG decode with Autodetect

```
void MPEG_Config()
{
    unsigned char MPEG_config_message[] =
        {0x8C, 0x00, 0x00,
         0x00, 0x00, 0x71,
         0x88, 0x00, 0x00,
         0x00, 0x10, 0x11};

    Write_*(MPEG_config_message,12); /* Replace * with I2C or SPI depending on protocol */
}
```

Figure 7. Pseudocode Example to Configure Part MPEG Decode with Autodetect

SMART
Analog™



LittleDiode supplies new, hard to find or obsolete electronic components and semiconductors all over the world.

With over two million different components listed you are sure to find the part you need.

Feel free to visit us today at our online store:

LittleDiode.com

Looking forward to providing you with the best possible service.