

## Application Note

# DOLBY DIGITAL™ USER'S GUIDE FOR THE CS4923/4/5/6 FAMILY

### Contents

- Dolby Digital Description



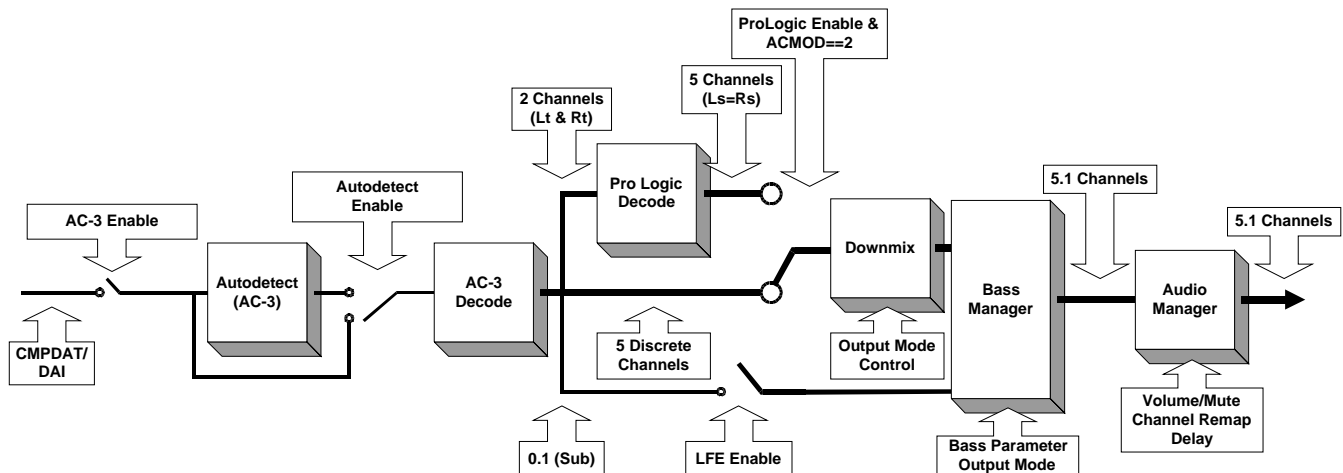
- Software Naming Convention
- Document Strategy
- Hardware Configurations Supported by the CS4923/4/5/6 and the AC-3® Code
- Understanding Application Messaging
- How to Control Application Modules such as:
  - Audio Manager
  - AC-3 Manager
  - Pro Logic Manager
  - Generalized Bass Manager
- Application Configuration Examples

### Description

This document includes a brief description of hardware configuration and in depth descriptions of the application messaging protocol, application control modules, and application configuration examples. The main body of this document covers all the features included in the standard Dolby Digital application for IEC61937 compliant bitstreams.

Special versions of the Dolby Digital decoder exist with different features, such as Virtual Dolby Digital™, QSurround™ and Harman VMAx VirtualTheater™. For each such version of Dolby Digital decode, a separate appendix covers the incremental changes with respect to this application note and are available to licensed customers.

This document covers code supported by CS4923/4/5/6 rev D silicon.



---

**TABLE OF CONTENTS**

|   |           |
|---|-----------|
| <b>1.0 OVERVIEW</b>                                     | <b>4</b>  |
| 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 Dolby Digital Description                           | 6         |
| <b>2.0 HARDWARE CONFIGURATION</b>                       | <b>8</b>  |
| 2.1 Supported Input/Output Modes                        | 8         |
| <b>3.0 APPLICATION MESSAGING</b>                        | <b>9</b>  |
| 3.1 Indexed Module Communication Protocol               | 9         |
| 3.1.1 Write Session                                     | 9         |
| 3.1.2 Solicited Read Message Format                     | 9         |
| 3.2 Unsolicited Read Message Format                     | 10        |
| <b>4.0 APPLICATION MODULES</b>                          | <b>11</b> |
| 4.1 Audio Manager                                       | 12        |
| 4.1.1 Audio_Control: (Index 0x00)                       | 14        |
| 4.1.2 Noise_Control (Index 0x01):                       | 15        |
| 4.1.3 Sampling Frequency Code (Index 0x14):             | 16        |
| 4.2 AC-3 Manager  | 17        |
| 4.2.1 Output_Mode_Control (Index 0x01)                  | 20        |
| 4.2.2 AC-3 Stream Information:                          | 20        |
| 4.2.3 Special Considerations For AC-3                   | 20        |
| 4.3 Pro Logic Manager                                   | 21        |
| 4.4 Generalized Bass Manager                            | 22        |
| 4.4.1 Dolby Digital Bass Management Configurations      | 26        |
| 4.5 PCM Processor                                       | 34        |
| 4.5.1 Karaoke Capable Function                          | 36        |
| 4.5.2 Special Considerations For PCM                    | 36        |
| 4.6 Unsolicited Messages (Read-Only)                    | 37        |
| 4.6.1 Autodetect Operation                              | 38        |
| 4.6.2 Special Considerations For Autodetect             | 40        |
| <b>5.0 APPLICATION CONFIGURATION EXAMPLES</b>           | <b>42</b> |
| 5.1 AC-3 decode with Autodetect                         | 42        |

---

**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, Dolby Pro Logic, the Dolby Digital Logo, Dolby and the double-D symbol are registered trademarks of Dolby Laboratories Licensing Corporation. DTS is a registered trademark of DTS, Inc. QSurround is a trademark of QSound Labs, Inc. Virtual Dolby Digital is a trademark of Dolby Laboratories Licensing Corporation. VMAx is a trademark of Harman International Industries, 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>.

|   |    |
|---|----|
| 5.2 AC-3 with Pro Logic Decode and Autodetect .....     | 43 |
| 5.3 Pro Logic decode of 48 kHz input PCM .....          | 43 |
| 5.4 PCM Pass-through of 44.1 kHz input PCM .....        | 43 |
| 5.5 Pink/White Noise Generation .....                   | 44 |
| 5.5.1 Special Considerations For Pink/White Noise ..... | 44 |
| 5.6 Using the PLL .....                                 | 45 |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 1. AC3 Code configured for AC3 input .....                                | 7  |
| Figure 2. AC-3 Code Configured For PCM .....                                     | 7  |
| Figure 3. Filter Topology .....  | 25 |
| Figure 4. Bass Manager Processing Unit .....                                     | 25 |
| Figure 5. Subwoofer Summing Module .....   | 25 |
| Figure 6. Autodetect Flow Chart .....  | 41 |
| Figure 7. Autodetect Flow Chart .....  | 41 |
| Figure 8. Typical Download and Configuration .....                               | 41 |
| Figure 9. Pseudocode Example to Configure Part AC-3 Decode with Autodetect ..... | 42 |

## LIST OF TABLES

|   |    |
|---|----|
| Table 1. Input/Output Configurations Supported by AC-3 .....        | 8  |
| Table 2. Input/Output Configurations Supported by PCM .....         | 8  |
| Table 3. Write Message Format .....                                 | 9  |
| Table 4. Read Command Message Format .....                          | 10 |
| Table 5. Read Response Message Format.....                          | 10 |
| Table 6. Unsolicited Read Message Format .....                      | 11 |
| Table 7. Audio Manager .....  | 12 |
| Table 8. AC-3 Manager .....   | 17 |
| Table 9. Pro Logic Manager .....                                    | 21 |
| Table 10. Bass Manager .....  | 22 |
| Table 11. Dolby Configuration 0 .....                               | 26 |
| Table 12. Dolby Configuration 1 .....                               | 27 |
| Table 13. Dolby Configuration 2 .....                               | 28 |
| Table 14. Dolby Alternative Configuration 2 .....                   | 29 |
| Table 15. Dolby Configuration 3 (Normal - No Sub Out) .....         | 30 |
| Table 16. Dolby Configuration 3 (Sub Out) .....                     | 31 |
| Table 17. Dolby DVD Configuration (Bass to L/R) .....               | 32 |
| Table 18. Dolby DVD Configuration (Bass to Subwoofer) .....         | 33 |
| Table 19. PCM Processor .....                                       | 34 |
| Table 20. Unsolicited Messages.....                                 | 37 |
| Table 21. Enabling AC-3 Decode with Autodetect .....                | 42 |
| Table 22. Enabling AC-3 with Pro Logic Decode with Autodetect ..... | 43 |
| Table 23. Enabling Pro Logic Decode of 48 kHz PCM .....             | 43 |
| Table 24. Enabling Pro Logic Decode of 44.1 kHz PCM .....           | 44 |
| Table 25. Enabling Pink/White Noise Generation .....                | 44 |
| Table 26. Pink/White Noise Generation for All 5.1 Channel .....     | 44 |
| Table 27. Configuring the PLL for 11.2896 MHz CLKIN.....            | 45 |

## 1.0 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 users guide covers code that provides AC-3 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 in-

cludes bass management, delays and Dolby Pro Logic<sup>®</sup> 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<sup>®</sup>/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 management. Separate downloads can also be used to sup-

port 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](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, Q-Surround 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 CS4923/4/5/6 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 Dolby Digital Description

Dolby Digital, also known as AC-3, is a digital audio compression algorithm developed by Dolby Laboratories for up to 5.1 channels of audio. This users guide covers the software designed to run on the CS4923/4/5/6 that decodes Dolby Digital. This user's guide covers all code that is named with

AAAA == AC3\_, and CC== 23, 24, 25, 26 as described in the previous section.

Other versions of AC-3 decoders with different feature sets (3D virtualizers etc.) are covered in corresponding appendices to this document. Each appendix specifies code name, and the changes with respect to this document.

Along with Dolby Digital, this document will describe how to use the code for Dolby Pro Logic Decode, pink/white noise generation, bass management and PCM pass through among other features. It is assumed that the reader is familiar with the requirements and features of Dolby Digital as specified in:

- *Dolby AC-3 Specification*: Digital Audio Compression Standard (AC-3), Advanced Television Systems Committee, Dec. 20, 1995
- *Licensee Information Manual*: Dolby Digital Multichannel Digital Audio Decoding for Consumer Products, Licensee Information Manual, Version 2.0, April, 1997

The block diagram in Figure 1 is a functional block diagram of the application code when configured for Dolby Digital. It gives an idea of the interaction between the various application modules when AC-3 is enabled.

Figure 2 is a functional block diagram of the application code when configured for PCM pass-through. It gives an idea of the interaction between the various application modules when PCM is enabled.

AC-3 and PCM should be considered as separate applications when using this code

This document uses Dolby Digital and AC-3 interchangeably. For this document CS4923/4/5/6 has been replaced in certain places with CS4923/4/5/6 for readability. Unless otherwise specified, CS4923/4/5/6 should be interpreted as applying to the CS4923, CS4924, CS4925 and CS4926.

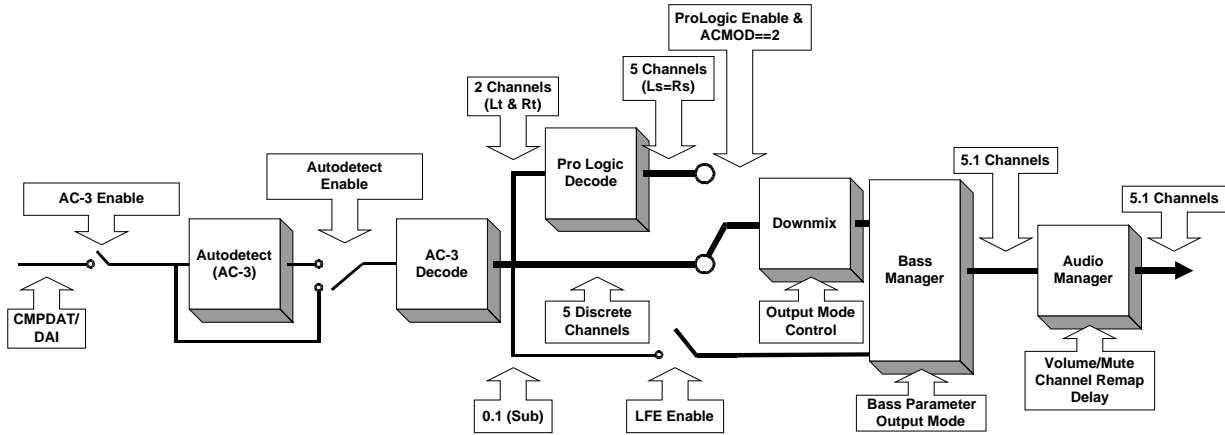


Figure 1. AC3 Code configured for AC3 input

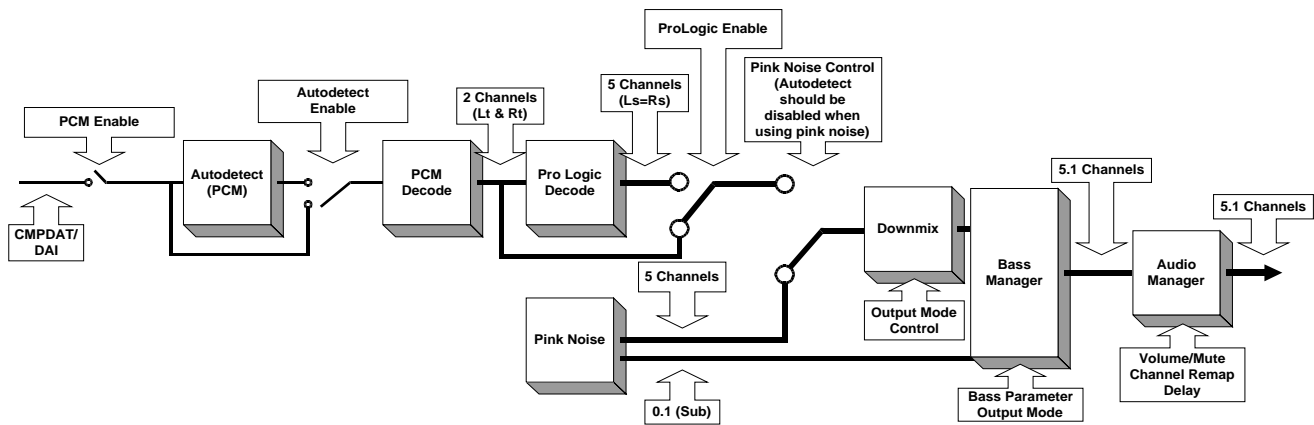


Figure 2. AC-3 Code Configured For PCM

## 2.0 HARDWARE CONFIGURATION

After download or soft reset, and before kickstarting the application, the host has the option of changing the default hardware configuration (see Section 4.1, *Audio Manager*, for more information on kickstarting). 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 CS4923/4/5/6 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 supported by the MPEG application. Table 2 shows the input/output modes supported by the PCM pass-through portion of the 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 AC-3**

| I/O Configuration | Description                     | Supported Modes |
|-------------------|---------------------------------|-----------------|
| 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 2. Input/Output Configurations Supported by PCM**



### 3.0 APPLICATION MESSAGING

While using the CS4923/4/5/6 it may be necessary to control or monitor the application to take full advantage of the rich feature set employed by the CS4923/4/5/6 and its software. Application messaging allows the user to do this. Whether it be configuring the part after download, e.g. enabling AC3 decode, or changing run-time parameters, e.g. controlling digital volume, the host will use application messaging to communicate with the CS4923/4/5/6.

While communicating with the CS4923/4/5/6 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 CS4923/4/5/6. 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 CS4923/4/5/6 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 4.0, *Application Modules*, the host will be able to fully configure the application running on the CS4923/4/5/6.

##### 3.1.1 Write Session

A write session with the CS4923/4/5/6 consists of one 6 byte message from the host to the CS4923/4/5/6. The write message consists of a command word followed by an associated data word. Table 3 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 CS4923/4/5/6, followed by a 6 byte Read Response message from the CS4923/4/5/6 to the host.

The read request message simply consists of a Read command word whose format is shown in Table 4.

*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 3. 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 CS4923/4/5/6.

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 5.

### 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 CS4923/4/5/6 to indicate a change in the system that must be ad-

ressed. One example is when the part is in autodefect mode and detects a new stream. An unsolicited read message will be sent by the CS4923/4/5/6 to indicate the new stream type.

The 6-byte unsolicited read messages from the CS4923/4/5/6 consist of a 3-byte read command word which defines the type of unsolicited message (as described in section 4.6) 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 6 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 4. 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 5. 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 6. Unsolicited Read Message Format**

#### 4.0 APPLICATION MODULES

Figures 1 and 2 accurately portray the interaction between the various application modules of the AC3 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:

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

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

Numbers preceded by 0x should be interpreted as hex, and numbers followed by ‘b’ should be interpreted as binary. All values sent to the CS4923/4/5/6 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><b>Bit 15:</b> 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).<br/>See “Unsolicited Messages (Read-Only)” on page 37 for more details</p> <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).<br/>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 8:</b> Application_Restart_Enable = 0/1= Disable/Enable restart of software application.</p> <p><b>Bit 4:</b> PLL_Enable = 0/1 = Disable/Enable Phase Locked Loop generation of DSP clock. It is <b>MANDATORY</b> that the PLL be enabled for Rev D CS4923/4/5/6.</p> <p><b>Bit 0:</b> Kickstart_Enable = 0/1 = Disable/Enable Kickstart of application.<br/>Default = 0x000000</p> |
| 0x01  | Noise_Control     | <p><b>Bit 5:</b> Level_Select = 0/1 = Default output levels/10.5dB boosted output level. Note that this bit <b>MUST</b> be disabled for Dolby testing.</p> <p><b>Bit 4:</b> White_Noise_Enable = 0/1 = Pink/White Noise output.</p> <p><b>Bits 3:0</b> = Output_Channel = 0...5 = L, C, R, Ls, Rs, LFE channel output. Disabled if &gt; 5.<br/>NOTE: Noise can only be enabled in PCM pass-through mode.<br/>Default = 0x7FFFFFFF</p>  |
| 0x02  | C_Delay           | 0...5 = mS delay. <i>Default* = 0</i>  |
| 0x04  | Ls_Delay          | 0...15 = mS delay (automatic +15 for Pro Logic). <i>Default* = 0</i>   |
| 0x05  | Rs_Delay          | 0...15 = mS delay (automatic +15 for Pro Logic). <i>Default* = 0</i>   |
| 0x06  | Master_Volume     | 0x800001-0x7FFFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFFF</i>  |
| 0x07  | L_Volume          | 0x800001-0x7FFFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFFF</i>  |
| 0x08  | C_Volume          | 0x800001-0x7FFFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFFF</i>  |
| 0x09  | R_Volume          | 0x800001-0x7FFFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFFF</i>  |
| 0x0a  | Ls_Volume         | 0x800001-0x7FFFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFFF</i>  |
| 0x0b  | Rs_Volume         | 0x800001-0x7FFFFFFF (-1.0 to 1.0). <i>Default* = 0x7FFFFFFF</i>  |

**Table 7. Audio Manager**

| Index | Variable                         | Dataword Content  |
|-------|----------------------------------|---|
| 0x0c  | LFE_Volume                       | 0x800001-0x7FFFFFF (-1.0 to 1.0). <i>Default*</i> = 0x7FFFFFF   |
| 0x0d  | Mute                             | 0/1 = Unmute/mute audio. This is a soft mute. <i>Default</i> = 0  |
| 0x0e  | DAO0_Channel <sup>1</sup>        | 0...5 = Channel type <sup>1</sup> for Digital Audio Output 0. Each channel must be mapped to one and only one unique output. <i>Default*</i> = 0(L)   |
| 0x0f  | DAO1_Channel <sup>1</sup>        | 0...5 = Channel type <sup>1</sup> for Digital Audio Output 1. Each channel must be mapped to one and only one unique output. <i>Default*</i> = 2(R)   |
| 0x10  | DAO2_Channel <sup>1</sup>        | 0...5 = Channel type <sup>1</sup> for Digital Audio Output 2. Each channel must be mapped to one and only one unique output. <i>Default*</i> = 3(Ls)  |
| 0x11  | DAO3_Channel <sup>1</sup>        | 0...5 = Channel type <sup>1</sup> for Digital Audio Output 3. Each channel must be mapped to one and only one unique output. <i>Default*</i> = 4(Rs)  |
| 0x12  | DAO4_Channel <sup>1</sup>        | 0...5 = Channel type <sup>1</sup> for Digital Audio Output 4. Each channel must be mapped to one and only one unique output. <i>Default*</i> = 1(C)   |
| 0x13  | DAO5_Channel <sup>1</sup>        | 0...5 = Channel type <sup>1</sup> for Digital Audio Output 5. Each channel must be mapped to one and only one unique output. <i>Default*</i> = 5(LFE)   |
| 0x14  | Sampling_Frequency_Code          | 0..2 = Sampling Frequency code to be set by host in case of PCM-only input applications. Irrelevant for non-PCM processing. This sampling frequency will be used for Fs dependent filters such as those in the bass manager.<br>0 = 48 KHz<br>1 = 44.1 KHz<br>2 = 32 KHz<br><i>Default*</i> = 0   |
| 0x15  | PCM_Autodetect_Silence_Threshold | 0..8388606 = Number of samples of silence after which CS4923/4/5/6 will declare Out-Of-PCM while playing PCM with autodetection enabled.<br>8388607 (0x7ffff) = Disable silence detection when playing PCM.<br>CS4923/4/5/6 will never declare Out-Of-PCM once it starts playing PCM (equivalent to infinite threshold).<br>NOTE: Valid only for PCM pass-through.<br><i>Default*</i> = 48000 |
| 0x16  | Last_Unsolicited_Message         | Last Unsolicited Message Value (see Section 4.6) <b>READ ONLY</b><br><i>Default</i> =0x000000   |
| 0x17  | PLL_Register_1                   | Register 1 value for Phase Locked Loop (no Fs generation).<br><i>Default*</i> = 0x07f270 (CLKIN=12.288MHz, DSPCLK=60MHz)  |
| 0x18  | PLL_Register_2                   | Register 2 value for Phase Locked Loop (no Fs generation).<br><i>Default*</i> = 0x0001fe (CLKIN=12.288MHz, DSPCLK=60MHz)  |

Notes: 1. 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.

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

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

**Bit\_15** *Audio\_Config\_Change\_Notification\_Enable*  
Makes the CS4923/4/5/6 sensitive to changes in the Audio configuration reported in the input bitstream. If any of *acmod*, *lfeon*, *dsurmod*, or *dialnorm* variables change, CS4923/4/5/6 will issue an unsolicited message (see *Unsolicited Messages (Read-Only)*, Section 4.6).

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

**Bit 14** *Startup\_Autodetect\_Bypass\_Enable*  
Maintains the CS4923/4/5/6 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 CS4923/4/5/6 if the processing is successful. However, if the data format is subsequently found to be the wrong type, autodetect messaging will be generated by CS4923/4/5/6 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 CS4923/4/5/6 into autodetect mode (see Section 4.6, *Unsolicited Messages (Read-Only)*, Section 4.6). 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 8** *Application\_Restart\_Enable*  
Used along with *Kickstart\_Enable* Restarts the downloaded software application.

0 - Disables Application\_Restart  
1 - Enables Application\_Restart

This is intended to allow change of input unit settings on the fly without resetting the output unit (which may cause clicks on the DAC if in master mode). This also allows on-the-fly changes of processing parameters (such as filter coefficients) without resetting the output unit. Application Restart effectively places the CS4923/4/5/6 in a start-up state as though it were issued a hardware reset + soft reset, except that the preset output and control port configuration (address checking for serial control) is maintained. Except for issuing output and host initialization messages, after an Application\_Restart, the host should treat the CS4923/4/5/6 as it would at start-up, i.e. issue input unit initialization, setup application parameters and then kickstart.

Notes: Just as with Soft Reset, all *Default\** settings are untouched by Application Restart.

**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 CS4923/4/5/6 into run-time mode. Indicates that all hardware and software configuration has been completed and CS4923/4/5/6 can start the application.

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

#### **4.1.2 Noise\_Control (Index 0x01):**

Switches ON Pink/White Noise in Channels L, C, R, Ls, Rs and LFE. Pink/White noise will not be enabled unless the application code is configured for PCM and the Output Channel is set to a value in the range 0x0 - 0x5.

Autodetect must NOT be enabled when using Pink Noise.

**Bit 5**     *Level\_Select*  
Sets up output level.

0 - Default level (Dolby-compatible)  
1 - 10.5 dB boosted with respect to default level.

**Bit 4**     *White\_Noise\_Enable*  
Enables white noise if set (otherwise output is Pink noise).

0 - Pink Noise  
1 - White Noise

**Bits 3:0**   *Output\_Channel*  
Selects channel for noise output.

0x0- Left  
0x1- Center  
0x2 - Right  
0x3 - Left Surround (This value controls both Ls and Rs in Pro Logic Mode)  
0x4 - Right Surround (This value has no effect in Pro Logic mode)  
0x5 - LFE  
0x6..0xf - disabled

### ***4.1.3 Sampling Frequency Code (Index 0x14):***

This parameter controls the sampling frequency for PCM applications. This must be set by the host during PCM applications, including PCM Pro Logic, for proper operation of filters. For compressed input, such as AC-3, the decoder application will setup sampling frequency information based on the information parsed from the bitstream so this field should not be set.

**Bits 2:0** *Sampling\_Frequency\_Code*

0x0 - 48 kHz  
0x1 - 44.1 kHz  
0x2 - 32 kHz

For 96 kHz PCM support, the host should set the Fs to 48 kHz and download the correct 96 kHz coefficients in the bass manager in the place of the 48 kHz coefficients.



## 4.2 AC-3 Manager

*Write Opcode = 0x8A; Read Opcode = 0x0B; Read Response Opcode = 0x8B*

*Write = 0x8A00HH 0xhhhhh*

*Read Request = 0x0B00HH;*

*Read Response = 0x8B00HH 0xhhhhh*

*0xHH = index*

*0xhhhhh = data value*

| Index | Variable            | Dataword Content  |
|-------|---------------------|---|
| 0x00  | AC3_Control         | <p><b>Bit 4:</b> LFE_Enable = 0/1 = Disable/Enable LFE channel decoding (if it exists in the stream)</p> <p><b>Bit 0:</b> AC3_Enable = 0/1 = Disable/Enable AC-3 decoding.</p> <p>Default = 0x000000</p>  |
| 0x01  | Output_Mode_Control | <p><b>Bit 8:</b> Karaoke_Capable_Enable = 0/1 = Enable/Disable Karaoke capable down-mixing (see Section 4.5 for Karaoke Capable user settings).</p> <p><b>Bits 5:4:</b> Dualmode = 0...3 = -u setting of Dolby C decoder. Valid only if acmod=0 (dual mono).</p> <p>00b = Stereo</p> <p>01b = Left Mono</p> <p>10b = Right Mono</p> <p>11b = Mixed-Mono.</p> <p><b>Bits 3:0:</b> Output_Mode = 0...7 = -o setting of Dolby C decoder</p> <p>0x0 = 2/0 L, R Dolby Surround compatible</p> <p>0x1 = 1/0 C</p> <p>0x2 = 2/0 L, R</p> <p>0x3 = 3/0 L, C, R</p> <p>0x4 = 2/1 L, R, S (Ls = S)</p> <p>0x5 = 3/1 L, C, R, S (Ls = S)</p> <p>0x6 = 2/2 L, R, Ls, Rs (with Pro Logic decode, Ls = Rs = S - 3 dB)</p> <p>0x7 = 3/2 L, C, R, Ls, Rs (with Pro Logic decode, Ls = Rs = S - 3 dB)</p> <p>Default* = 0x000007</p> <p><b>NOTE:</b> Undesired Channel outputs must be explicitly muted by setting the corresponding Ch_Vol (see Audio Manager) to 0.0</p> |
| 0x02  | Compression_Control | <p><b>Bits 1:0:</b> 0...3 = -k setting of Dolby C decoder</p> <p>00b = Custom Mode 0 (Analog Dialnorm)</p> <p>01b = Custom Mode 1 (Digital Dialnorm)</p> <p>10b = Line Out Mode (This mode is typical for most AVR systems - however the choice of compression is entirely up to the system designer)</p> <p>11b = RF Remodulation Mode</p> <p>Default* = 1</p>   |

**Table 8. AC-3 Manager**

| <b>Index</b> | <b>Variable</b> | <b>Dataword Content</b>   |
|--------------|-----------------|---|
| 0x03         | Cut_x           | 0x000000-0x7FFFFFFF = Compression cut scale factor = -x setting of Dolby C decoder. Cut_x is a scaling factor used on the absolute compression limits. The absolute compression limits are set during the encoding process. <i>Default* = 0</i>   |
| 0x04         | Boost_y         | 0x000000-0x7FFFFFFF = Compression boost scale factor = -y setting of Dolby C decoder. Boost_y is a scaling factor used on the absolute compression limits. The absolute compression limits are set during the encoding process. <i>Default* = 0</i>   |
| 0x05         | fscod           | <b>Bits 1:0:</b> Current Sample Rate Code (fscod) value.<br>00b = 48kHz<br>01b = 44.1kHz<br>10b = 32kHz<br>11b = Reserved<br><b>READ-ONLY</b>   |
| 0x06         | bsmod           | <b>Bits 3:0:</b> Current Bit Stream (bsmod) value.<br>0x0 = main audio service: complete main (CM)<br>0x1 = main audio service: music and effects (ME)<br>0x2 = associated service: visually impaired (VI)<br>0x3 = associated service: hearing impaired (HI)<br>0x4 = associated service: dialogue (D)<br>0x5 = associated service: commentary (C)<br>0x6 = associated service: emergency (E)<br>0x7:<br>if ACMOD == 1: associated service: voice over (VO)<br>if ACMOD == 010b to 111b: main audio service: karaoke<br><b>READ-ONLY</b> |
| 0x07         | acmod           | <b>Bits 3:0:</b> Current Audio Coding Mode (acmod) value<br>0x0 = 1+1 Ch1, Ch2<br>0x1 = 1/0 C<br>0x2 = 2/0 L, R<br>0x3 = 3/0 L, C, R<br>0x4 = 2/1 L, R, S<br>0x5 = 3/1 L, C, R, S<br>0x6 = 2/2 L, R, Ls, Rs<br>0x7 = 3/2 L, C, R, Ls, Rs<br><b>READ-ONLY</b>  |

**Table 8. AC-3 Manager (Continued)**

| Index | Variable           | Dataword Content  |
|-------|--------------------|---|
| 0x08  | dsurmod            | <b>Bits 1:0:</b> Current Dolby Surround Mode (dsurmod value) valid only when (acmod=2).<br>00b = not indicated<br>01b = NOT Dolby Surround encoded<br>10b = Dolby Surround Encoded<br>11b = Reserved<br>READ-ONLY |
| 0x09  | clev               | 0x000000-0x7FFFFFFF = Current center mixing level (interpreted stream value). Valid only if 3 front channels exist. <b>READ-ONLY</b>  |
| 0x0a  | slev               | 0x000000-0x7FFFFFFF = Current surround mixing level (interpreted stream value). Valid only if a surround channel exists. <b>READ-ONLY</b>   |
| 0x0b  | lfeon              | <b>Bit 0:</b> Current lfeon value<br>0 = LFE channel is off<br>1 = LFE channel is on<br>READ-ONLY   |
| 0x0c  | dialnorm           | 0...31 = Current dialnorm value. <b>READ-ONLY</b>   |
| 0x0d  | compr              | 0xFFFFFFFF = Nonexistent in stream. 0x00-0xFF = Current compr value. <b>READ-ONLY</b>   |
| 0x0e  | dialnorm2          | 0...31 = Current dialnorm2 value. Valid only in 1+1 dual mono mode (acmod=0). <b>READ-ONLY</b>  |
| 0x0f  | compr2             | 0xFFFFFFFF = Nonexistent in stream. 0x00-0xFF = Current compr2 value. Valid only in 1+1 dual mono mode (acmod=0). <b>READ-ONLY</b>  |
| 0x10  | bsid               | 0...31 = Current bsid value. <b>READ-ONLY</b>   |
| 0x11  | langcod            | 0xFFFFFFFF = Nonexistent in stream. 0...255 = Current langcod value. <b>READ-ONLY</b>   |
| 0x12  | mixlevel_roomtyp   | 0xFFFFFFFF = Nonexistent in stream. 0x00...0x7F: <b>Bits 6:2:</b> current mixlevel. <b>Bits 1:0:</b> current roomtyp. <b>READ-ONLY</b>  |
| 0x13  | langcod2           | 0xFFFFFFFF = Nonexistent in stream. 0...255 = Current langcod value. Valid only in 1+1 dual mono mode (acmod=0). <b>READ-ONLY</b>   |
| 0x14  | mixlevel2_roomtyp2 | 0xFFFFFFFF = Nonexistent in stream. 0x00...0x7F: <b>Bits 6:2:</b> current mixlevel2. <b>Bits 1:0:</b> current roomtyp2. Valid only in 1+1 dual mono mode (acmod=0). <b>READ-ONLY</b>                              |
| 0x15  | copyrightb         | 0/1 = Current copyrightb value. <b>READ-ONLY</b>  |
| 0x16  | origbs             | 0/1 = Current origbs value. <b>READ-ONLY</b>  |
| 0x17  | frmsizcod          | Current frame size code. <b>READ-ONLY</b>   |

Table 8. AC-3 Manager (Continued)

### 4.2.1 *Output\_Mode\_Control (Index 0x01)*

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

**Bit 8** *Karaoke\_Capable\_Enable*

If set, downmixing of Karaoke streams (bsmod=7, acmod=2...7) will be performed according to Karaoke Capable decoder specification by employing user settings (see Section 4.5, *PCM Processor*, for more details of the user settings).

If *Karaoke\_Capable\_Enable* is not set, then CS4923/4/5/6 defaults to a Karaoke Aware downmix using levels indicated in the stream.

**Bits 5:4** *Dualmode = 0...3 = -u setting of Dolby C decoder*  
Valid only if acmod=0 (dual mono).

00b = Stereo  
01b = Left Mono  
10b = Right Mono  
11b = Mixed-Mono.

**Bits 3:0** *Output\_Mode = 0...7 = -o setting of Dolby C decoder*

0x0 = 2/0 L, R Dolby Surround compatible  
0x1 = 1/0 C  
0x2 = 2/0 L, R  
0x3 = 3/0 L, C, R  
0x4 = 2/1 L, R, S (Ls = S)  
0x5 = 3/1 L, C, R, S (Ls = S)  
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 *AC-3 Stream Information:*

Table 8 lists all READ-ONLY variables that are reported to the host by the AC-3 Decoder module. These can be read out using the appropriate Read Request, Read Response sessions.

For example, to read the acmod value, host should issue the 3-byte Read Request 0x0b0007 and then read the 6-byte Read Response 0x9b0007, 0x00000h, where the dataword (latter 3 bytes) indicates the reported value of acmod. Please refer to the Dolby AC-3 Specification for more information on these stream variables.

### 4.2.3 *Special Considerations For AC-3*

When using the AC-3 code it should be noted that AC-3 and PCM are completely independent applications. The host must treat them as such and only enable either AC-3 or PCM at any one time. Transitions from AC-3 to PCM or vice-versa should be done with a soft reset followed by complete hardware configuration and application configuration. Conversely, an Application Restart message could be used which would only require that a new input port configuration and application configuration be sent -- output clocks would not be affected. For more information on Application Restart, please see Section 4.1, *Audio Manager*.

### 4.3 Pro Logic Manager

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

*Write = 0x8E00HH 0xhhhhhh*

*Read Request = 0x0F00HH;*

*Read Response = 0x8F00HH 0xhhhhhh*

*0xHH = index*

*0xhhhhhh = data value*

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

**Table 9. Pro Logic Manager**

Note that control of the output mode (which channels are configured to provide audio) is done through the Output\_Mode\_Control index (0x01) of the AC-3 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 = 0x4 2/1 L, R, S (Ls = S)

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*

| <b>Index</b> | <b>Variable</b>    | <b>Datword Content</b>  |
|--------------|--------------------|---|
| 0x00         | Bass_Mgr_Control   | <b>Bit 16:</b> LFE_Only_Sub = 0/1 = Disable/Enable LFE Only to subwoofer.<br><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.<br><b>Bit 12:</b> LPF = 0/1 = Disable/Enable LPF at output of summer.<br><b>Bit 8:</b> HPF_L = 0/1 = Disable/Enable L channel HPF.<br><b>Bit 7:</b> HPF_C = 0/1 = Disable/Enable C channel HPF.<br><b>Bit 6:</b> HPF_R = 0/1 = Disable/Enable R channel HPF.<br><b>Bit 5:</b> HPF_Ls = 0/1 = Disable/Enable Ls channel HPF.<br><b>Bit 4:</b> HPF_Rs = 0/1 = Disable/Enable Rs channel HPF.<br><b>Bit 0:</b> Bass_Mgr = 0/1 = Disable/Enable Bass Manager post-processing.<br>Default = 0x000000 |
| 0x01         | Input_L_Level      | 0x000000-0x7FFFFFFF = Level adjustment for input L channel pass-through.<br>Default* = 0x7FFFFFFF (0 dB)  |
| 0x02         | Input_C_Level      | 0x000000-0x7FFFFFFF = Level adjustment for input C channel input pass-through.<br>Default* = 0x7FFFFFFF (0 dB)  |
| 0x03         | Input_R_Level      | 0x000000-0x7FFFFFFF = Level adjustment for input R channel input pass-through.<br>Default* = 0x7FFFFFFF (0 dB)  |
| 0x04         | Input_Ls_Level     | 0x000000-0x7FFFFFFF = Level adjustment for input Ls channel pass-through.<br>Default* = 0x7FFFFFFF (0 dB)   |
| 0x05         | Input_Rs_Level     | 0x000000-0x7FFFFFFF = Level adjustment for input Rs channel pass-through.<br>Default* = 0x7FFFFFFF (0 dB)   |
| 0x06         | Input_LFE_Level    | 0x000000-0x7FFFFFFF = Level adjustment for input LFE channel pass-through.<br>Default* = 0x7FFFFFFF (0 dB)  |
| 0x07         | Sum_Output_L_Level | 0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to L channel output.<br>Default* = 0x000000 (-∞ dB)  |
| 0x08         | Sum_Output_C_Level | 0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to C channel output.<br>Default* = 0x000000 (-∞ dB)  |

**Table 10. Bass Manager**

| <b>Index</b> | <b>Variable</b>     | <b>Dataword Content</b>  |
|--------------|---------------------|--|
| 0x09         | Sum_Output_R_Level  | 0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to R channel output.<br><i>Default* = 0x000000 (-∞ dB)</i>  |
| 0x0a         | Sum_Output_Ls_Level | 0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to Ls channel output.<br><i>Default* = 0x000000 (-∞ dB)</i> |
| 0x0b         | Sum_Output_Rs_Level | 0x000000-0x7FFFFFFF = Level adjustment for bass summer output contribution to Rs channel output.<br><i>Default* = 0x000000 (-∞ dB)</i> |
| 0x0c         | LFE_L_Level         | 0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to L channel output.<br><i>Default* = 0x000000 (-∞ dB)</i>           |
| 0x0d         | LFE_C_Level         | 0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to C channel output.<br><i>Default* = 0x000000 (-∞ dB)</i>           |
| 0x0e         | LFE_R_Level         | 0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to R channel output.<br><i>Default* = 0x000000 (-∞ dB)</i>           |
| 0x0f         | LFE_Ls_Level        | 0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to Ls channel output.<br><i>Default* = 0x000000 (-∞ dB)</i>          |
| 0x10         | LFE_Rs_Level        | 0x000000-0x7FFFFFFF = Level adjustment for input LFE contribution to Rs channel output.<br><i>Default* = 0x000000 (-∞ dB)</i>          |
| 0x11         | Sum_Input_L_Level   | 0x000000-0x7FFFFFFF = Level adjustment for input L channel contribution to bass summer input.<br><i>Default* = 0x16C311 (-15 dB)</i>   |
| 0x12         | Sum_Input_C_Level   | 0x000000-0x7FFFFFFF = Level adjustment for input C channel contribution to bass summer input.<br><i>Default* = 0x16C311 (-15 dB)</i>   |
| 0x13         | Sum_Input_R_Level   | 0x000000-0x7FFFFFFF = Level adjustment for input R channel contribution to bass summer input.<br><i>Default* = 0x16C311 (-15 dB)</i>   |
| 0x14         | Sum_Input_Ls_Level  | 0x000000-0x7FFFFFFF = Level adjustment for input Ls channel contribution to bass summer input.<br><i>Default* = 0x16C311 (-15 dB)</i>  |
| 0x15         | Sum_Input_Rs_Level  | 0x000000-0x7FFFFFFF = Level adjustment for input Rs channel contribution to bass summer input.<br><i>Default* = 0x16C311 (-15 dB)</i>  |

**Table 10. Bass Manager (Continued)**

| <b>Index</b> | <b>Variable</b>     | <b>Datavord Content</b>  |
|--------------|---------------------|--|
| 0x16         | Sum_Input_LFE_Level | 0x000000-0x7FFFFFFF = Level adjustment for input LFE channel contribution to bass summer.<br>Default* = 0x47FACD (-5 dB) |
| 0x17         | LPF_48_b0           | 0x000000-0x7FFFFFFF = b0 coefficient for Low Pass Filter, Fs=48 KHz.<br>Default* = 0x014D4C                              |
| 0x18         | LPF_48_b1           | 0x000000-0x7FFFFFFF = b1 coefficient for Low Pass Filter, Fs=48 KHz.<br>Default* = 0x014D4C                              |
| 0x19         | LPF_48_a1           | 0x000000-0x7FFFFFFF = a1 coefficient for Low Pass Filter, Fs=48 KHz.<br>Default* = 0x7D6567                              |
| 0x1a         | HPF_48_b0           | 0x000000-0x7FFFFFFF = b0 coefficient for High Pass Filter, Fs=48 KHz.<br>Default* = 0x7F7750                             |
| 0x1b         | HPF_48_b1           | 0x000000-0x7FFFFFFF = b1 coefficient for High Pass Filter, Fs=48 KHz.<br>Default* = 0x8088B0                             |
| 0x1c         | HPF_48_a1           | 0x000000-0x7FFFFFFF = a1 coefficient for High Pass Filter, Fs=48 KHz.<br>Default* = 0x7EEEA0                             |
| 0x1d         | LPF_44_b0           | 0x000000-0x7FFFFFFF = b0 coefficient for Low Pass Filter, Fs=44 KHz.<br>Default* = 0x016A73                              |
| 0x1e         | LPF_44_b1           | 0x000000-0x7FFFFFFF = b1 coefficient for Low Pass Filter, Fs=44 KHz.<br>Default* = 0x016A73                              |
| 0x1f         | LPF_44_a1           | 0x000000-0x7FFFFFFF = a1 coefficient for Low Pass Filter, Fs=44 KHz.<br>Default* = 0x7D2B19                              |
| 0x20         | HPF_44_b0           | 0x000000-0x7FFFFFFF = b0 coefficient for High Pass Filter, Fs=44 KHz.<br>Default* = 0x7F6B48                             |
| 0x21         | HPF_44_b1           | 0x000000-0x7FFFFFFF = b1 coefficient for High Pass Filter, Fs=44 KHz.<br>Default* = 0x8094B8                             |
| 0x22         | HPF_44_a1           | 0x000000-0x7FFFFFFF = a1 coefficient for High Pass Filter, Fs=44 KHz.<br>Default* = 0x7ED68F                             |
| 0x23         | LPF_32_b0           | 0x000000-0x7FFFFFFF = b0 coefficient for Low Pass Filter, Fs=32 KHz.<br>Default* = 0x01F171                              |
| 0x24         | LPF_32_b1           | 0x000000-0x7FFFFFFF = b1 coefficient for Low Pass Filter, Fs=32 KHz.<br>Default* = 0x01F171                              |
| 0x25         | LPF_32_a1           | 0x000000-0x7FFFFFFF = a1 coefficient for Low Pass Filter, Fs=32 KHz.<br>Default* = 0x7C1D1F                              |
| 0x26         | HPF_32_b0           | 0x000000-0x7FFFFFFF = b0 coefficient for High Pass Filter, Fs=32 KHz.<br>Default* = 0x7F3365                             |
| 0x27         | HPF_32_b1           | 0x000000-0x7FFFFFFF = b1 coefficient for High Pass Filter, Fs=32 KHz.<br>Default* = 0x80CC9B                             |
| 0x28         | HPF_32_a1           | 0x000000-0x7FFFFFFF = a1 coefficient for High Pass Filter, Fs=32 KHz.<br>Default* = 0x7E66CA                             |

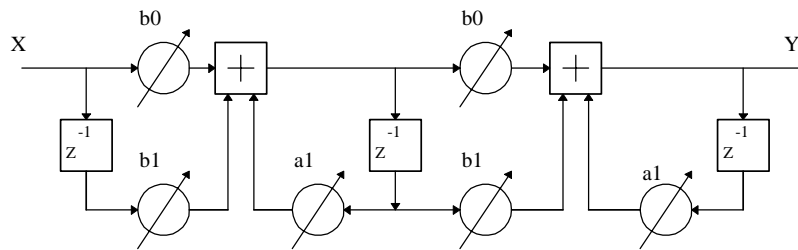
**Table 10. Bass Manager (Continued)**



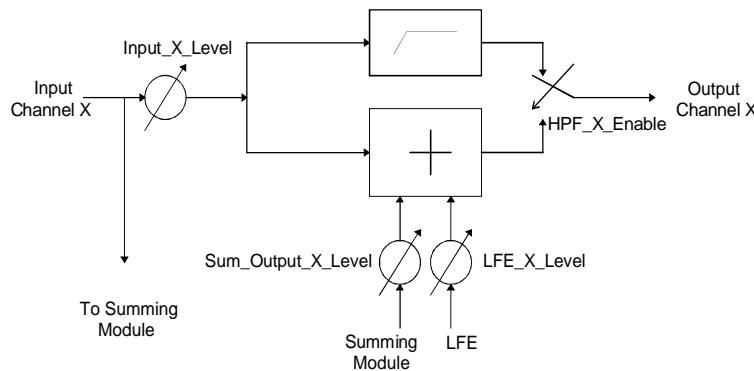
The Bass Manager is best described with the use of a block diagram. Figure 3 shows the topology of the filter used for both the low pass and the high pass filters. This filter represents two first order filters cascaded to implement a second order filter. For the 4th order LPF two filters as in Figure 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 obtain 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.

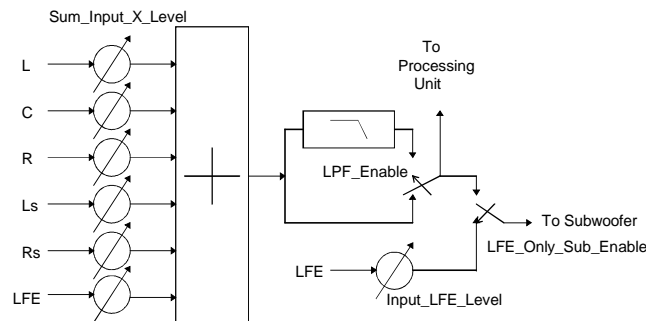
Figure 4 shows the processing unit on each of the output channels (except the LFE) and Figure 5 shows the summing module used to redirect bass to the subwoofer channel.



**Figure 3. Filter Topology**



**Figure 4. Bass Manager Processing Unit**



**Figure 5. Subwoofer Summing Module**

#### 4.4.1 Dolby Digital Bass Management Configurations

Tables 11 through 18 show the values that need to be written for the different Dolby Digital Bass Management configurations. These are in no way inclusive of all of the different modes possible,

they are just presented to show the different configurations required for Dolby Digital Certification. Different configurations are required depending on the level of certification that is desired. Any questions regarding certification should be directed towards Dolby Laboratories.

| Bass Manager Variables | Values  | Write Opcode | HEX Value  |
|------------------------|---|--------------|------------|
| Bass_Mgr_Control       | LFE_Only_Sub Disabled<br>LPF Disabled<br>HPF_L Disabled<br>HPF_C Disabled<br>HPF_R Disabled<br>HPF_Ls Disabled<br>HPF_Rs Disabled<br>Bass_Mgr Enabled | 0x940000     | 0x000001   |
| Input_L_Level          | 0 dB  | 0x940001     | 0x7FFFFFFF |
| Input_C_Level          | 0 dB  | 0x940002     | 0x7FFFFFFF |
| Input_R_Level          | 0 dB  | 0x940003     | 0x7FFFFFFF |
| Input_Ls_Level         | 0 dB  | 0x940004     | 0x7FFFFFFF |
| Input_Rs_Level         | 0 dB  | 0x940005     | 0x7FFFFFFF |
| Input_LFE_Level        | 0 dB  | 0x940006     | 0x7FFFFFFF |
| Sum_Output_L_Level     | -∞ dB   | 0x940007     | 0x000000   |
| Sum_Output_C_Level     | -∞ dB   | 0x940008     | 0x000000   |
| Sum_Output_R_Level     | -∞ dB   | 0x940009     | 0x000000   |
| Sum_Output_Ls_Level    | -∞ dB   | 0x94000A     | 0x000000   |
| Sum_Output_Rs_Level    | -∞ dB   | 0x94000B     | 0x000000   |
| LFE_L_Level            | -∞ dB   | 0x94000C     | 0x000000   |
| LFE_C_Level            | -∞ dB   | 0x94000D     | 0x000000   |
| LFE_R_Level            | -∞ dB   | 0x94000E     | 0x000000   |
| LFE_Ls_Level           | -∞ dB   | 0x94000F     | 0x000000   |
| LFE_Rs_Level           | -∞ dB   | 0x940010     | 0x000000   |
| Sum_Input_L_Level      | -15 dB  | 0x940011     | 0x16C311   |
| Sum_Input_C_Level      | -15 dB  | 0x940012     | 0x16C311   |
| Sum_Input_R_Level      | -15 dB  | 0x940013     | 0x16C311   |
| Sum_Input_Ls_Level     | -15 dB  | 0x940014     | 0x16C311   |
| Sum_Input_Rs_Level     | -15 dB  | 0x940015     | 0x16C311   |
| Sum_Input_LFE_Level    | -5 dB   | 0x940016     | 0x47FACD   |

**Table 11. Dolby Configuration 0**

| Bass Manager Variables | Values  | Write Opcode | HEX Value  |
|------------------------|---|--------------|------------|
| Bass_Mgr_Control       | LFE_Only_Sub Disabled<br>LPF Enabled<br>HPF_L Enabled<br>HPF_C Enabled<br>HPF_R Enabled<br>HPF_Ls Enabled<br>HPF_Rs Enabled<br>Bass_Mgr Enabled | 0x940000     | 0x0011F1   |
| Input_L_Level          | 0 dB  | 0x940001     | 0x7FFFFFFF |
| Input_C_Level          | 0 dB  | 0x940002     | 0x7FFFFFFF |
| Input_R_Level          | 0 dB  | 0x940003     | 0x7FFFFFFF |
| Input_Ls_Level         | 0 dB  | 0x940004     | 0x7FFFFFFF |
| Input_Rs_Level         | 0 dB  | 0x940005     | 0x7FFFFFFF |
| Input_LFE_Level        | 0 dB  | 0x940006     | 0x7FFFFFFF |
| Sum_Output_L_Level     | -∞ dB   | 0x940007     | 0x000000   |
| Sum_Output_C_Level     | -∞ dB   | 0x940008     | 0x000000   |
| Sum_Output_R_Level     | -∞ dB   | 0x940009     | 0x000000   |
| Sum_Output_Ls_Level    | -∞ dB   | 0x94000A     | 0x000000   |
| Sum_Output_Rs_Level    | -∞ dB   | 0x94000B     | 0x000000   |
| LFE_L_Level            | -∞ dB   | 0x94000C     | 0x000000   |
| LFE_C_Level            | -∞ dB   | 0x94000D     | 0x000000   |
| LFE_R_Level            | -∞ dB   | 0x94000E     | 0x000000   |
| LFE_Ls_Level           | -∞ dB   | 0x94000F     | 0x000000   |
| LFE_Rs_Level           | -∞ dB   | 0x940010     | 0x000000   |
| Sum_Input_L_Level      | -15 dB  | 0x940011     | 0x16C311   |
| Sum_Input_C_Level      | -15 dB  | 0x940012     | 0x16C311   |
| Sum_Input_R_Level      | -15 dB  | 0x940013     | 0x16C311   |
| Sum_Input_Ls_Level     | -15 dB  | 0x940014     | 0x16C311   |
| Sum_Input_Rs_Level     | -15 dB  | 0x940015     | 0x16C311   |
| Sum_Input_LFE_Level    | -5 dB   | 0x940016     | 0x47FACD   |

**Table 12. Dolby Configuration 1**

| <b>Bass Manager Variables</b> | <b>Values</b>   | <b>Write Opcode</b> | <b>HEX Value</b> |
|-------------------------------|---|---------------------|------------------|
| Bass_Mgr_Control              | LFE_Only_Sub Disabled<br>LPF Enabled<br>HPF_L Disabled<br>HPF_C Enabled<br>HPF_R Disabled<br>HPF_Ls Enabled<br>HPF_Rs Enabled<br>Bass_Mgr Enabled | 0x940000            | 0x0010B1         |
| Input_L_Level                 | -12 dB  | 0x940001            | 0x2026F2         |
| Input_C_Level                 | ADJ, default = 0 dB   | 0x940002            | 0x7FFFFFFF       |
| Input_R_Level                 | -12 dB  | 0x940003            | 0x2026F2         |
| Input_Ls_Level                | ADJ, default = 0 dB   | 0x940004            | 0x7FFFFFFF       |
| Input_Rs_Level                | ADJ, default = 0 dB   | 0x940005            | 0x7FFFFFFF       |
| Input_LFE_Level               | 0 dB  | 0x940006            | 0x7FFFFFFF       |
| Sum_Output_L_Level            | -1.5 dB   | 0x940007            | 0x6BB2D5         |
| Sum_Output_C_Level            | -∞ dB   | 0x940008            | 0x000000         |
| Sum_Output_R_Level            | -1.5 dB   | 0x940009            | 0x6BB2D5         |
| Sum_Output_Ls_Level           | -∞ dB   | 0x94000A            | 0x000000         |
| Sum_Output_Rs_Level           | -∞ dB   | 0x94000B            | 0x000000         |
| LFE_L_Level                   | -∞ dB   | 0x94000C            | 0x000000         |
| LFE_C_Level                   | -∞ dB   | 0x94000D            | 0x000000         |
| LFE_R_Level                   | -∞ dB   | 0x94000E            | 0x000000         |
| LFE_Ls_Level                  | -∞ dB   | 0x94000F            | 0x000000         |
| LFE_Rs_Level                  | -∞ dB   | 0x940010            | 0x000000         |
| Sum_Input_L_Level             | -∞ dB   | 0x940011            | 0x000000         |
| Sum_Input_C_Level             | -15 dB  | 0x940012            | 0x16C311         |
| Sum_Input_R_Level             | -∞ dB   | 0x940013            | 0x000000         |
| Sum_Input_Ls_Level            | -15 dB  | 0x940014            | 0x16C311         |
| Sum_Input_Rs_Level            | -15 dB  | 0x940015            | 0x16C311         |
| Sum_Input_LFE_Level           | -5 dB   | 0x940016            | 0x47FACD         |

**Table 13. Dolby Configuration 2**

| <b>Bass Manager Variables</b> | <b>Values</b>   | <b>Write Opcode</b> | <b>HEX Value</b> |
|-------------------------------|---|---------------------|------------------|
| Bass_Mgr_Control              | LFE_Only_Sub Disabled<br>LPF Enabled<br>HPF_L Disabled<br>HPF_C Enabled<br>HPF_R Disabled<br>HPF_Ls Enabled<br>HPF_Rs Enabled<br>Bass_Mgr Enabled | 0x940000            | 0x0010B1         |
| Input_L_Level                 | 0 dB  | 0x940001            | 0x7FFFFFFF       |
| Input_C_Level                 | 0 dB  | 0x940002            | 0x7FFFFFFF       |
| Input_R_Level                 | 0 dB  | 0x940003            | 0x7FFFFFFF       |
| Input_Ls_Level                | 0 dB  | 0x940004            | 0x7FFFFFFF       |
| Input_Rs_Level                | 0 dB  | 0x940005            | 0x7FFFFFFF       |
| Input_LFE_Level               | 0 dB  | 0x940006            | 0x7FFFFFFF       |
| Sum_Output_L_Level            | -∞ dB   | 0x940007            | 0x000000         |
| Sum_Output_C_Level            | -∞ dB   | 0x940008            | 0x000000         |
| Sum_Output_R_Level            | -∞ dB   | 0x940009            | 0x000000         |
| Sum_Output_Ls_Level           | -∞ dB   | 0x94000A            | 0x000000         |
| Sum_Output_Rs_Level           | -∞ dB   | 0x94000B            | 0x000000         |
| LFE_L_Level                   | -∞ dB   | 0x94000C            | 0x000000         |
| LFE_C_Level                   | -∞ dB   | 0x94000D            | 0x000000         |
| LFE_R_Level                   | -∞ dB   | 0x94000E            | 0x000000         |
| LFE_Ls_Level                  | -∞ dB   | 0x94000F            | 0x000000         |
| LFE_Rs_Level                  | -∞ dB   | 0x940010            | 0x000000         |
| Sum_Input_L_Level             | -∞ dB   | 0x940011            | 0x000000         |
| Sum_Input_C_Level             | -15 dB  | 0x940012            | 0x16C311         |
| Sum_Input_R_Level             | -∞ dB   | 0x940013            | 0x000000         |
| Sum_Input_Ls_Level            | -15 dB  | 0x940014            | 0x16C311         |
| Sum_Input_Rs_Level            | -15 dB  | 0x940015            | 0x16C311         |
| Sum_Input_LFE_Level           | -5 dB   | 0x940016            | 0x47FACD         |

**Table 14. Dolby Alternative Configuration 2**

| Bass Manager Variables | Values  | Write Opcode | HEX Value |
|------------------------|---|--------------|-----------|
| Bass_Mgr_Control       | LFE_Only_Sub Disabled<br>LPF Enabled<br>HPF_L Disabled<br>HPF_C Enabled<br>HPF_R Disabled<br>HPF_Ls Disabled<br>HPF_Rs Disabled<br>Bass_Mgr Enabled | 0x940000     | 0x001081  |
| Input_L_Level          | -8 dB   | 0x940001     | 0x32F52C  |
| Input_C_Level          | -8 dB   | 0x940002     | 0x32F52C  |
| Input_R_Level          | -8 dB   | 0x940003     | 0x32F52C  |
| Input_Ls_Level         | -8 dB   | 0x940004     | 0x32F52C  |
| Input_Rs_Level         | -8 dB   | 0x940005     | 0x32F52C  |
| Input_LFE_Level        | -8 dB   | 0x940006     | 0x32F52C  |
| Sum_Output_L_Level     | -4.5 dB   | 0x940007     | 0x4C3EA7  |
| Sum_Output_C_Level     | -∞ dB   | 0x940008     | 0x000000  |
| Sum_Output_R_Level     | -4.5 dB   | 0x940009     | 0x4C3EA7  |
| Sum_Output_Ls_Level    | -∞ dB   | 0x94000A     | 0x000000  |
| Sum_Output_Rs_Level    | -∞ dB   | 0x94000B     | 0x000000  |
| LFE_L_Level            | -8 dB   | 0x94000C     | 0x32F52C  |
| LFE_C_Level            | -∞ dB   | 0x94000D     | 0x000000  |
| LFE_R_Level            | -8 dB   | 0x94000E     | 0x32F52C  |
| LFE_Ls_Level           | -8 dB   | 0x94000F     | 0x32F52C  |
| LFE_Rs_Level           | -8 dB   | 0x940010     | 0x32F52C  |
| Sum_Input_L_Level      | -∞ dB   | 0x940011     | 0x000000  |
| Sum_Input_C_Level      | -8 dB   | 0x940012     | 0x32F52C  |
| Sum_Input_R_Level      | -∞ dB   | 0x940013     | 0x000000  |
| Sum_Input_Ls_Level     | -∞ dB   | 0x940014     | 0x000000  |
| Sum_Input_Rs_Level     | -∞ dB   | 0x940015     | 0x000000  |
| Sum_Input_LFE_Level    | -∞ dB   | 0x940016     | 0x000000  |

**Table 15. Dolby Configuration 3 (Normal - No Sub Out)**

| Bass Manager Variables | Values  | Write Opcode | HEX Value |
|------------------------|---|--------------|-----------|
| Bass_Mgr_Control       | LFE_Only_Sub Disabled<br>LPF Enabled<br>HPF_L Disabled<br>HPF_C Enabled<br>HPF_R Disabled<br>HPF_Ls Disabled<br>HPF_Rs Disabled<br>Bass_Mgr Enabled | 0x940000     | 0x011081  |
| Input_L_Level          | -4 dB   | 0x940001     | 0x50C335  |
| Input_C_Level          | -4 dB   | 0x940002     | 0x50C335  |
| Input_R_Level          | -4 dB   | 0x940003     | 0x50C335  |
| Input_Ls_Level         | -4 dB   | 0x940004     | 0x50C335  |
| Input_Rs_Level         | -4 dB   | 0x940005     | 0x50C335  |
| Input_LFE_Level        | -4 dB   | 0x940006     | 0x50C335  |
| Sum_Output_L_Level     | -4.5 dB   | 0x940007     | 0x4C3EA7  |
| Sum_Output_C_Level     | -∞ dB   | 0x940008     | 0x000000  |
| Sum_Output_R_Level     | -4.5 dB   | 0x940009     | 0x4C3EA7  |
| Sum_Output_Ls_Level    | -∞ dB   | 0x94000A     | 0x000000  |
| Sum_Output_Rs_Level    | -∞ dB   | 0x94000B     | 0x000000  |
| LFE_L_Level            | -∞ dB   | 0x94000C     | 0x000000  |
| LFE_C_Level            | -∞ dB   | 0x94000D     | 0x000000  |
| LFE_R_Level            | -∞ dB   | 0x94000E     | 0x000000  |
| LFE_Ls_Level           | -∞ dB   | 0x94000F     | 0x000000  |
| LFE_Rs_Level           | -∞ dB   | 0x940010     | 0x000000  |
| Sum_Input_L_Level      | -∞ dB   | 0x940011     | 0x000000  |
| Sum_Input_C_Level      | -4 dB   | 0x940012     | 0x50C335  |
| Sum_Input_R_Level      | -∞ dB   | 0x940013     | 0x000000  |
| Sum_Input_Ls_Level     | -∞ dB   | 0x940014     | 0x000000  |
| Sum_Input_Rs_Level     | -∞ dB   | 0x940015     | 0x000000  |
| Sum_Input_LFE_Level    | -∞ dB   | 0x940016     | 0x000000  |

**Table 16. Dolby Configuration 3 (Sub Out)**

| Bass Manager Variables | Values   | Write Opcode | HEX Value  |
|------------------------|--|--------------|------------|
| Bass_Mgr_Control       | LFE_Only_Sub Enabled<br>LPF Enabled<br>HPF_L Disabled<br>HPF_C Enabled<br>HPF_R Disabled<br>HPF_Ls Enabled<br>HPF_Rs Enabled<br>Bass_Mgr Enabled | 0x940000     | 0x0110B1   |
| Input_L_Level          | 0 dB   | 0x940001     | 0x7FFFFFFF |
| Input_C_Level          | 0 dB   | 0x940002     | 0x7FFFFFFF |
| Input_R_Level          | 0 dB   | 0x940003     | 0x7FFFFFFF |
| Input_Ls_Level         | 0 dB   | 0x940004     | 0x7FFFFFFF |
| Input_Rs_Level         | 0 dB   | 0x940005     | 0x7FFFFFFF |
| Input_LFE_Level        | -5 dB  | 0x940006     | 0x47FACC   |
| Sum_Output_L_Level     | 0 dB   | 0x940007     | 0x7FFFFFFF |
| Sum_Output_C_Level     | -∞ dB  | 0x940008     | 0x000000   |
| Sum_Output_R_Level     | 0 dB   | 0x940009     | 0x7FFFFFFF |
| Sum_Output_Ls_Level    | -∞ dB  | 0x94000A     | 0x000000   |
| Sum_Output_Rs_Level    | -∞ dB  | 0x94000B     | 0x000000   |
| LFE_L_Level            | -∞ dB  | 0x94000C     | 0x000000   |
| LFE_C_Level            | -∞ dB  | 0x94000D     | 0x000000   |
| LFE_R_Level            | -∞ dB  | 0x94000E     | 0x000000   |
| LFE_Ls_Level           | -∞ dB  | 0x94000F     | 0x000000   |
| LFE_Rs_Level           | -∞ dB  | 0x940010     | 0x000000   |
| Sum_Input_L_Level      | -∞ dB  | 0x940011     | 0x000000   |
| Sum_Input_C_Level      | -4.5 dB  | 0x940012     | 0x4C3EA7   |
| Sum_Input_R_Level      | -∞ dB  | 0x940013     | 0x000000   |
| Sum_Input_Ls_Level     | -4.5 dB  | 0x940014     | 0x4C3EA7   |
| Sum_Input_Rs_Level     | -4.5 dB  | 0x940015     | 0x4C3EA7   |
| Sum_Input_LFE_Level    | -∞ dB  | 0x940016     | 0x000000   |

**Table 17. Dolby DVD Configuration (Bass to L/R)**



| Bass Manager Variables | Values  | Write Opcode | HEX Value  |
|------------------------|---|--------------|------------|
| Bass_Mgr_Control       | LFE_Only_Sub Disabled<br>LPF Enabled<br>HPF_L Disabled<br>HPF_C Enabled<br>HPF_R Disabled<br>HPF_Ls Enabled<br>HPF_Rs Enabled<br>Bass_Mgr Enabled | 0x940000     | 0x0010B1   |
| Input_L_Level          | 0 dB  | 0x940001     | 0x7FFFFFFF |
| Input_C_Level          | 0 dB  | 0x940002     | 0x7FFFFFFF |
| Input_R_Level          | 0 dB  | 0x940003     | 0x7FFFFFFF |
| Input_Ls_Level         | 0 dB  | 0x940004     | 0x7FFFFFFF |
| Input_Rs_Level         | 0 dB  | 0x940005     | 0x7FFFFFFF |
| Input_LFE_Level        | 0 dB  | 0x940006     | 0x7FFFFFFF |
| Sum_Output_L_Level     | -∞ dB   | 0x940007     | 0x000000   |
| Sum_Output_C_Level     | -∞ dB   | 0x940008     | 0x000000   |
| Sum_Output_R_Level     | -∞ dB   | 0x940009     | 0x000000   |
| Sum_Output_Ls_Level    | -∞ dB   | 0x94000A     | 0x000000   |
| Sum_Output_Rs_Level    | -∞ dB   | 0x94000B     | 0x000000   |
| LFE_L_Level            | -∞ dB   | 0x94000C     | 0x000000   |
| LFE_C_Level            | -∞ dB   | 0x94000D     | 0x000000   |
| LFE_R_Level            | -∞ dB   | 0x94000E     | 0x000000   |
| LFE_Ls_Level           | -∞ dB   | 0x94000F     | 0x000000   |
| LFE_Rs_Level           | -∞ dB   | 0x940010     | 0x000000   |
| Sum_Input_L_Level      | -∞ dB   | 0x940011     | 0x000000   |
| Sum_Input_C_Level      | -15 dB  | 0x940012     | 0x16C310   |
| Sum_Input_R_Level      | -∞ dB   | 0x940013     | 0x000000   |
| Sum_Input_Ls_Level     | -15 dB  | 0x940014     | 0x16C310   |
| Sum_Input_Rs_Level     | -15 dB  | 0x940015     | 0x16C310   |
| Sum_Input_LFE_Level    | -5 dB   | 0x940016     | 0x47FACC   |

**Table 18. Dolby DVD Configuration (Bass to Subwoofer)**

## 4.5 PCM Processor

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

*Write = 0x9C00HH 0xhhhhh*

*Read Request = 0x1D00HH;*

*Read Response = 0x9D00HH 0xhhhhh*

*0xHH = index*

*0xhhhhh = data value*

| <b>Index</b> | <b>Variable</b>  | <b>Dataword Content</b>  |
|--------------|------------------|--|
| 0x00         | PCM_Control      | <p><b>Bit 4:</b> Deemphasis_Enable = 0/1 = Disable/Enable PCM Deemphasis.</p> <p><b>Bit 0:</b> PCM_Enable = 0/1 = Disable/Enable PCM. This bit should only be enabled if AC-3 is disabled.</p> <p>Default = 0x000000</p> |
| 0x01         | K_Cpl_mlev_1_F   | <p>M level for Karaoke capable 1 front output channel.</p> <p>0 &lt;= K_Cpl_mlev_1_F &lt;= 0x7ffff.</p> <p>*Default = 0x1e2a53</p>   |
| 0x02         | K_Cpl_v1lev_1_F  | <p>V1 level for Karaoke capable 1 front output channel.</p> <p>0 &lt;= K_Cpl_v1lev_1_F &lt;= 0x7ffff.</p> <p>*Default = 0x1e2a53</p>   |
| 0x03         | K_Cpl_v2lev_1_F  | <p>V2 level for Karaoke capable 1 front output channel.</p> <p>0 &lt;= K_Cpl_v2lev_1_F &lt;= 0x7ffff.</p> <p>*Default = 0x1e2a53</p>   |
| 0x04         | K_Cpl_mllev_2_F  | <p>m L level for Karaoke capable 2 front output channels.</p> <p>0 &lt;= K_Cpl_mllev_2_F &lt;= 0x7ffff.</p> <p>*Default = 0x34b4b3</p>   |
| 0x05         | K_Cpl_v1llev_2_F | <p>V1 L level for Karaoke capable 2 front output channels.</p> <p>0 &lt;= K_Cpl_v1llev_2_F &lt;= 0x7ffff.</p> <p>*Default = 0x4b4b4a</p>   |
| 0x06         | K_Cpl_v2llev_2_F | <p>V2 L level for Karaoke capable 2 front output channels.</p> <p>0 &lt;= K_Cpl_v2llev_2_F &lt;= 0x7ffff.</p> <p>*Default = 0x0</p>  |
| 0x07         | K_Cpl_mrlev_2_F  | <p>M R level for Karaoke capable 2 front output channels.</p> <p>0 &lt;= K_Cpl_mrlev_2_F &lt;= 0x7ffff.</p> <p>*Default = 0x34b4b3</p>   |
| 0x08         | K_Cpl_v1rlev_2_F | <p>V1 R level for Karaoke capable 2 front output channels.</p> <p>0 &lt;= K_Cpl_v1rlev_2_F &lt;= 0x7ffff.</p> <p>*Default = 0x0</p>  |

**Table 19. PCM Processor**

| <b>Index</b> | <b>Variable</b>  | <b>Dataword Content</b>   |
|--------------|------------------|---|
| 0x09         | K_Cpl_v2rlev_2_F | V2 R level for Karaoke capable 2 front output channels.<br>0 <= K_Cpl_v2rlev_2_F <= 0x7ffff.<br>*Default = 0x4b4b4a |
| 0x0a         | K_Cpl_mcpan_3_F  | M C level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_mcpan_3_F <= 0x7ffff.<br>*Default = 0x7ffff    |
| 0x0b         | K_Cpl_v1cpan_3_F | V1 C level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_v1cpan_3_F <= 0x7ffff.<br>*Default = 0x0      |
| 0x0c         | K_Cpl_v2cpan_3_F | V2 C level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_v2cpan_3_F <= 0x7ffff.<br>*Default = 0x0      |
| 0x0d         | K_Cpl_mlpan_3_F  | M L level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_mlpan_3_F <= 0x7ffff.<br>*Default = 0x0        |
| 0x0e         | K_Cpl_v1lpan_3_F | V1 L level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_v1lpan_3_F <= 0x7ffff.<br>*Default = 0x7ffff  |
| 0x0f         | K_Cpl_v2lpan_3_F | V2 L level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_v2lpan_3_F <= 0x7ffff.<br>*Default = 0x0      |
| 0x10         | K_Cpl_mrpan_3_F  | M R level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_mrpan_3_F <= 0x7ffff.<br>*Default = 0x0        |
| 0x11         | K_Cpl_v1rpan_3_F | V1 R level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_v1rpan_3_F <= 0x7ffff.<br>*Default = 0x0      |
| 0x12         | K_Cpl_v2rpan_3_F | V2 R level for Karaoke capable 3 front output channels.<br>0 <= K_Cpl_v2rpan_3_F <= 0x7ffff.<br>*Default = 0x7ffff  |

**Table 19. PCM Processor (Continued)**

### 4.5.1 Karaoke Capable Function

Karaoke Capable functionality is enabled when Bit 8 of Output\_Mode\_Control in AC-3\_Manager is set to 1.

If Karaoke\_Capable\_Enable is not set, then CS4923/4/5/6 defaults to a Karaoke Aware downmix using levels indicated in the stream.

In a karaoke bit stream (bsmod=7, acmod=2...7), 3 special channels have to be downmixed according to user specification. They are M (guide melody), and V1, V2 (vocal tracks one and two).

In Karaoke Aware mode (default mode where Karaoke\_Capable\_Enable=0), CS4923/4/5/6 downmixes these channels according to the clev/slev values indicated in the bitstream. This Karaoke Aware downmix reproduces all the program content in the front speakers according to the programmed levels and audio configuration of the bitstream.

In Karaoke Capable mode, the downmix coefficients are setup by the user according to the number of output front channels: 1, 2, or 3. Independent settings are available for each of these 3 cases as indicated in Table 19. The following equations define the downmixing performed for each of the 3 cases:

1 front channel:

$$Ck = mlev * M + v1lev * V1 + v2lev * V2.$$

$$Co = 0.707 * L + Ck + 0.707 * R.$$

2 front channels:

$$Lk = mlev * M + v1llev * V1 + v2llev * V2.$$

$$Rk = mrlev * M + v1rlev * V1 + v2rlev * V2.$$

$$Lo = L + Lk.$$

$$Ro = R + Rk.$$

3 front channels:

$$Ck = mcpan * M + v1cpan * V1 + v2cpan * V2.$$

$$Lk = mlpan * M + v1lpan * V1 + v2lpan * V2.$$

$$Rk = mrpan * M + v1rpan * V1 + v2rpan * V2.$$

$$Lo = L + Lk.$$

$$Co = Ck.$$

$$Ro = R + Rk.$$

### 4.5.2 Special Considerations For PCM

When using the AC-3 code it should be noted that AC-3 and PCM are completely independent applications. The host must treat them as such and only enable either AC-3 or PCM at any one time. Transitions from AC-3 to PCM or vice-versa should be done with a soft reset followed by complete hardware configuration and application configuration. Conversely, an Application Restart message could be used which would only require that a new input port configuration and application configuration be sent -- output clocks and address checking for serial control modes will not be affected.

#### 4.6 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><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>If Non_IEC61937_Stream_Flag=1</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>If Non_IEC61937_Stream_Flag=0</p> <p><b>Bits 4:0</b> = IEC61937 Stream Descriptor = Identical to bits [4:0] of the Pc burst datatype descriptor in IEC61937 specification. 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 CS4923/4/5/6)</i></p> <p>0x01 = AC-3 data</p> <p>0x02 = Reserved.</p> <p>0x03 = <i>Never Reported. (Pause is ignored by CS4923/4/5/6)</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_C_hanged | <p><b>Bits 23:0</b> = Reserved. The DSP will send out this message to indicate that one of the following has changed in the AC3 stream: ACMOD, LFEON, DSURMOD, DIALNORM. It is the host's responsibility to check exactly what has changed.</p>   |
| 0x10  | PLL_Out_Of_Lock              | <p><b>Bit 23</b> = 1.</p> <p><b>Bits 22:0</b> = Reserved.</p>   |

**Table 20. Unsolicited Messages**

### 4.6.1 Autodetect Operation

The sequence of events involving autodetection are described below from the host's perspective. This is a general example for the CS4923/4/5/6, when using MPEG code or DTS code application restart is not available and thus should not be used.

- 1) Host downloads CS4923/4/5/6 with a tentative application code, for this example we will use AC3\_\*\*\*\*.LD.
- 2) Host then configures the CS4923/4/5/6 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 CS4923/4/5/6 with Autodetect enabled (see Section 4.1, *Audio Manager*, for details).
- 4) The autodetect module of the enabled application of the CS4923/4/5/6 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 CS4923/4/5/6 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`. CS4923/4/5/6 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 CS4923/4/5/6 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 CS4923/4/5/6 (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 CS4923/4/5/6 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 CS4923/4/5/6, 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 CS4923/4/5/6 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 re-sent.

The host should then repeat Steps 2, 3, 4, 5a/b as described above after delivering the new input stream to the CS4923/4/5/6.

If the new input content is detected as unchanged (still AC3 in our example), the CS4923/4/5/6 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 CS4923/4/5/6 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 CS4923/4/5/6 also simultaneously monitors the input. Note that the CS4923/4/5/6 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 CS4923/4/5/6 detects a change in the bitstream (no longer AC-3, in our original example), then the CS4923/4/5/6 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 CS4923/4/5/6 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), CS4923/4/5/6 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 CS4923/4/5/6 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 CS4923/4/5/6 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. CS4923/4/5/6 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 6, for an overview of the previously described flow.

#### **4.6.2 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 or Application Restart should be sent.

The application must be enabled before sending the autodetect enable with kickstart. Particularly, for the AC-3 code, either AC-3 or PCM must be enabled *before* the autodetect message is sent with the kickstart. If the Autodetect function senses the data for the enabled application, then the appropriate Unsolicited Message is sent and decode (or PCM

pass-through) will start automatically without any direction from the host.

In summary, the following is the recommended procedure when using autodetect. Please see Figure 7, 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.



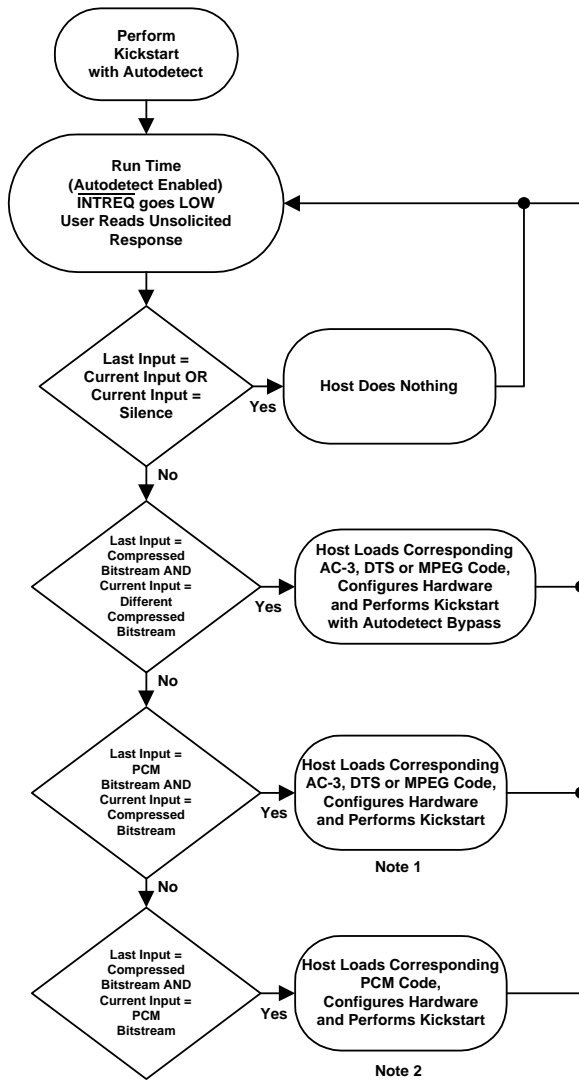


Figure 6. Autodetect Flow Chart

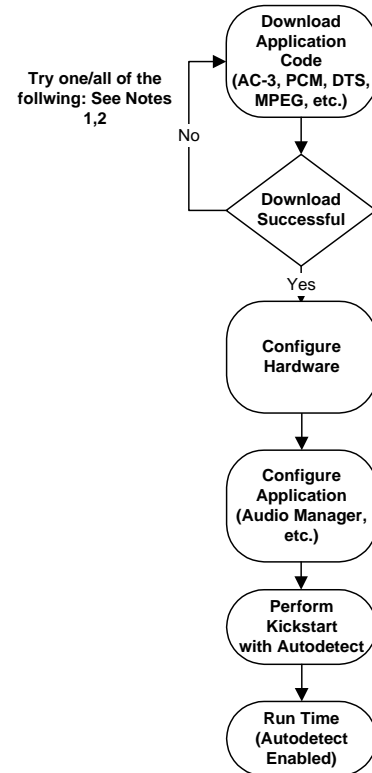


Figure 7. Typical Download and Configuration

- 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.

- 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.

## 5.0 APPLICATION CONFIGURATION EXAMPLES

This section covers various application modes available with the CS4923/4/5/6. Although many application modes are possible, only the most common modes have been presented here as examples. This should not be considered the complete list of modes in which the part will operate, as the system designer can configure the application parameters of the CS4923/4/5/6 as desired in order to differentiate the system.

Tables 21-27 provide the message to be sent and a description of what the message is doing. Care should be taken to guarantee that the correct value is sent directly after the associated opcode and index word. The command to kickstart the application should always be sent last.

When switching between applications (i.e. from AC-3 to PCM) the host should perform either a soft

reset (See the CS4923/4/5/6/7/8/9 Hardware User's Guide) or application restart (see Section 4.1, *Audio Manager*). This will reset all application configuration.

### 5.1 AC-3 decode with Autodetect

In this mode the input should be compressed AC-3. The data will be taken from the compressed input port as designated by the hardware configuration message. Table 21 gives a description of AC-3 decode with Autodetect. If the input on this port is not AC-3 the Autodetect function will notify the host as described in Section 4.6, *Unsolicited Messages (Read-Only)*.

Figure 7 gives an example of pseudocode to configure the part AC-3 decode with autodetect. 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    |
|---------------|-----------------------|--|----------------|----------|
| AC-3          | AC-3 Control          | AC-3 Enabled<br>LFE Enabled                                | 0x8A0000       | 0x000011 |
| AC-3          | Compression_Control   | Set to Line Out Mode                                       | 0x8A0002       | 0x000002 |
| Audio Manager | Audio Manager Control | Autodetect Enabled<br>PLL Enabled<br>Kickstart Application | 0x880000       | 0x001011 |

**Table 21. Enabling AC-3 Decode with Autodetect**

```
void AC-3_AD_Config()
{
    unsigned char AC-3_AD_config_message[] =
        {0x8A, 0x00, 0x00,
         0x00, 0x00, 0x11,

         0x8A, 0x00, 0x02
         0x00, 0x00, 0x02

         0x88, 0x00, 0x00,
         0x00, 0x10, 0x11};

    Write_*(AC-3_AD_config_message,18); /* Replace * with I2C or SPI depending on protocol */
}
```

**Figure 7. Pseudocode Example to Configure Part AC-3 Decode with Autodetect**

## 5.2 AC-3 with Pro Logic Decode and Autodetect

In this mode the input should be two-channel compressed AC-3 (acmod = 2) with Pro Logic encoded data (dsurmod = 2). The data will be taken from the compressed input port as designated by the hardware configuration message. Table 22 gives a description of AC-3 Pro Logic and Autodetect. If the input on this port is not AC-3 the Autodetect function will notify the host as described in section , *Unsolicited Messages (Read-Only)*.

## 5.3 Pro Logic decode of 48 kHz input PCM

In this mode the input should be 48 kHz Pro Logic encoded PCM data. The data will be taken from the PCM input port as designated by the hardware configuration message. Table 23 gives a description of Pro Logic decode of 48 MHz input PCM.

## 5.4 PCM Pass-through of 44.1 kHz input PCM

In this mode the input should be 44.1 kHz PCM data. The data will be taken from the PCM input port as designated by the hardware configuration message. Table 25 gives a description of PCM Pass-through of 44.1 kHz input PCM.

| Module        | Index                 | Description  | Opcode & Index | Value    |
|---------------|-----------------------|--|----------------|----------|
| Pro Logic     | ProLogic Control      | Enabled<br>Autobalance Enabled<br>B Type NR Enabled        | 0x8e0000       | 0x000111 |
| AC-3          | AC-3 Control          | AC-3 Enabled<br>LFE Enabled                                | 0x8A0000       | 0x000011 |
| AC-3          | Compression_Control   | Set to Line Out Mode                                       | 0x8A0002       | 0x000002 |
| Audio Manager | Audio Manager Control | Autodetect Enabled<br>PLL Enabled<br>Kickstart Application | 0x880000       | 0x001011 |

**Table 22. Enabling AC-3 with Pro Logic Decode with Autodetect**

| Module        | Index                 | Description  | Opcode & Index | Value    |
|---------------|-----------------------|--|----------------|----------|
| PCM           | PCM Control           | Enabled  | 0x9c0000       | 0x000001 |
| Pro Logic     | ProLogic Control      | Enabled<br>Autobalance Enabled<br>B-Type NR Enabled        | 0x8e0000       | 0x000111 |
| Audio Manager | Sampling Frequency    | 48 kHz   | 0x880014       | 0x000000 |
| Audio Manager | Audio Manager Control | Autodetect Enabled<br>PLL Enabled<br>Kickstart Application | 0x880000       | 0x001011 |

**Table 23. Enabling Pro Logic Decode of 48 kHz PCM**

## 5.5 Pink/White Noise Generation

To cycle through the channels (after sending the above messages) the host would send the following sequence of commands pausing between each command to hold the pink noise on a particular channel.

White noise can be generated by additionally setting Bit 4 of Noise\_Control in the above messages. In the same message, if Bit 5 is set, a 10.5dB boosted noise is received. Tables 25 and 26 give descriptions of Pink/White Noise Generation.

### 5.5.1 Special Considerations For Pink/White Noise

Pink/White Noise is controlled by three parameters. These parameters are the currently configured application (i.e. PCM), the channel selected by the Noise\_Control variable, and the state of the autodetect enable. In order to use Pink/White Noise, the code should be configured strictly for PCM before kickstart. If Pro Logic is enabled, the host loses independent control of Pink/White Noise on the Ls and Rs channels -- the surround channels operate in unison.

| Module        | Index                 | Description  | Opcode & Index | Value    |
|---------------|-----------------------|--|----------------|----------|
| PCM           | PCM Control           | Enabled  | 0x9c0000       | 0x000001 |
| Audio Manager | Sampling Frequency    | 44.1 kHz   | 0x880014       | 0x000001 |
| Audio Manager | Audio Manager Control | Autodetect Enabled<br>PLL Enabled<br>Kickstart Application | 0x880000       | 0x001011 |

**Table 24. Enabling Pro Logic Decode of 44.1 kHz PCM**

| Module        | Index                 | Description                                  | Opcode & Index | Value    |
|---------------|-----------------------|--|----------------|----------|
| PCM           | PCM Control           | Enabled                                      | 0x9c0000       | 0x000001 |
| Audio Manager | Noise Control         | Pink Noise On Left Channel                   | 0x880001       | 0x000000 |
| Audio Manager | Audio Manager Control | Kickstart Application<br>Autodetect Disabled | 0x880000       | 0x000011 |

**Table 25. Enabling Pink/White Noise Generation**

| Module        | Index         | Description                          | Opcode & Index | Value    |
|---------------|---------------|--------------------------------------|----------------|----------|
| Audio Manager | Noise Control | Pink Noise On Left Channel           | 0x880001       | 0x000000 |
| Audio Manager | Noise Control | Pink Noise On Center Channel         | 0x880001       | 0x000001 |
| Audio Manager | Noise Control | Pink Noise On Right Channel          | 0x880001       | 0x000002 |
| Audio Manager | Noise Control | Pink Noise On Left Surround Channel  | 0x880001       | 0x000003 |
| Audio Manager | Noise Control | Pink Noise On Right Surround Channel | 0x880001       | 0x000004 |
| Audio Manager | Noise Control | Pink Noise On LFE Channel            | 0x880001       | 0x000005 |

**Table 26. Pink/White Noise Generation for All 5.1 Channel**

## 5.6 Using the PLL

All the application modes described use the PLL to generate DSP clock from an external oscillator (CLKIN) assuming a 12.288 MHz clock. Thus, if using a 12.288 MHz oscillator, one could just skip the first and second messages in Table 27, and simply send in the last message (0x880000, 0x001011) with Bits 23:8 set according to the desired application mode.

However, if CLKIN is a 11.2896 MHz oscillator, simply substitute the Kickstart Application message in any of the previous examples (Opcode & Index 0x880000, typically the last message) with the messages specified above. The value 0xmmm11 in the bottom row and last column denotes that Bits 7:0 should be forced to 11, and the remaining bits be maintained as specified in the original application mode above.

| Module        | Index                 | Description                             | Opcode & Index | Value    |
|---------------|-----------------------|---|----------------|----------|
| Audio Manager | PLL_Register_1        | PLL Setup for 11.2896 MHz CLKIN         | 0x880017       | 0x24BC34 |
| Audio Manager | PLL_Register_2        | PLL Setup for 11.2896 MHz CLKIN         | 0x880018       | 0x0001fe |
| Audio Manager | Audio Manager Control | Kickstart Application with PLL enabled. | 0x880000       | 0xHHHH11 |

**Table 27. Configuring the PLL for 11.2896 MHz CLKIN**

SMART  
Analog™