AN122

Application Note

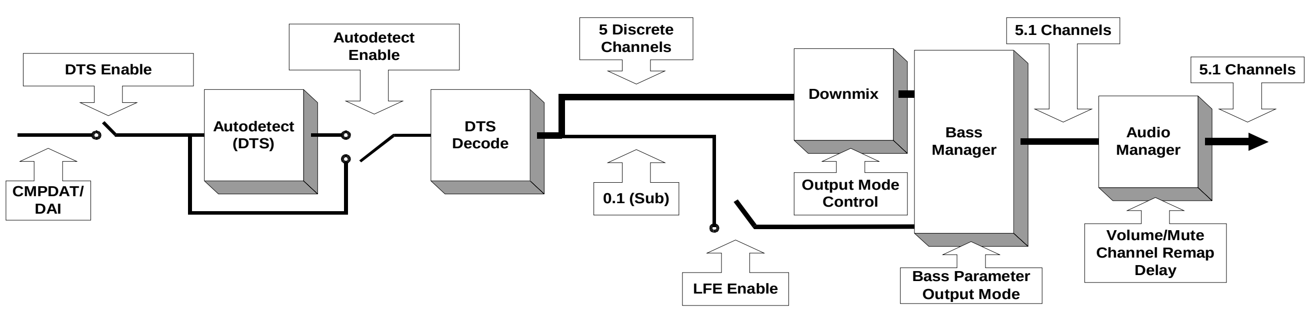
DTS® USER'S GUIDE FOR THE CS4926 AND CS4928

Contents

Description

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

This document covers code supported by CS492X rev D silicon.



CIRRUS LOGIC®
P.O. Box 17847, Austin, Texas 78760
(512) 445 7222 FAX: (512) 445 7581
http://www.cirrus.com

TABLE OF CONTENTS

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/

LIST OF FIGURES

LIST OF TABLES

1. OVERVIEW

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

These parts are generally targeted at two different market segments. The broadcast market where audio/video (A/V) synchronization is required, and the outboard decoder markets where audio/video synchronization is not required. The important differentiation is the format in which the data will be received by the CS4923/4/5/6/7/8/9. In systems where A/V synchronization is required from the CS4923/4/5/6/7/8/9, the incoming data is typically PES encoded. In an outboard decoder application the data typically comes in the IEC61937 format (as specified by the DVD consortium). An important point to remember is that the CS4923/4/5/6/7/8/9 will support both environments, but different downloads are required depending on the input data type.

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

This user's guide covers code that provides DTS decoding for IEC61937 compliant bitstreams and elementary DTS bitstreams.

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

1.1 Multi-Channel Decoder Family of Parts

CS4923 - Dolby Digital™ Audio Decoder

The CS4923 is the original member of the family and is intended to be used if only Dolby Digital decoding is required. For Dolby Digital, post processing includes bass management, delays and Dolby Pro Logic® decoding. Separate downloads can also be used to support stereo to 5.1 channel effects processing and stereo MPEG decoding.

CS4924 - Dolby Digital Source Product Decoder

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

CS4925 - International Multi-Channel DVD Audio Decoder

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

CS4926 - DTS/Dolby Multi-Channel Audio Decoder

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

CS4927 - MPEG-2 Multi-Channel Decoder

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

CS4928 - DTS Multi-Channel Decoder

The CS4928 supports DTS multi-channel decoding and should be used in applications where Dolby Digital decoding is not necessary. For DTS multi-channel decoding, post processing includes bass management. Separate downloads can also be used to support stereo to 5.1 channel effects processing and stereo MPEG decoding.

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

The CS4929 is capable of decoding both 2-channel AAC 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

This 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 referred to, it should be assumed that one or more of the below documents are being referenced. The following list covers currently released application notes. This list will grow with each new application released. For a current list of released user's guides please see *www.cirrus.com* and search for the part number.

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

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

AN121 - MPEG User's Guide for the CS4925

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

AN122 - DTS User's Guide for the CS4926, CS4928

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

AN123 - Surround User's Guide for the CS4923/4/5/6/7/8

This code covers the different Stereo PCM to surround effects processing code. Optional appendices are available that document Crystal Original Surround, SRS Circle Surround™ and Logic 7™.

AN140 - Broadcast Systems Guide for the CS4923/4/5/6/7/8/9

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

1.3 Using the CS4923/4/5/6/7/8/9

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

1) Reset and/or Download Code - Detailed information can be found in AN115.

2) Hardware Configuration - Detailed information can be found in AN115.

3) Application configuration - Detailed information can be found in the appropriate Application Code User's Guide.
4) Kickstart - This is the "Go" command to the CS492X once the system is properly configured. Detailed information can be found in the appropriate Application Code User's Guide.

1.4 Software Naming Convention

To keep up with the different applications, parts, hardware revisions and software revisions the following naming convention has been adopted for the CS4923/4/5/6/7/8/9 software:

AAAACCRV.LD

where:

AAAA = three or four letter Application description (e.g. AC3_)

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

R = ROM ID specification

V = actual version release of particular code

1.5 DTS Digital Surround Description

DTS Digital Surround is a digital audio compression algorithm developed by Digital Theater Systems for up to 5.1 channels of audio. This users guide covers the software designed to run on the CS4926/8 that decodes DTS. This users guide covers all code that is named with AAAA == DTS_, and CC==26 and CC==28 as described in the previous section.

It is assumed that the reader is familiar with the requirements and features of DTS as specified in:

- **DTS Specification:** DTS Coherent Acoustics Decoder, DTS Technology, Version 1.0, January 20, 1998
- **Licensee Manual:** DTS Master Quality Multichannel Digital Audio Decoding System for Consumer Products Licensee Manual, Version 1.0, July 1998

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

In this document CS4926/8 has been used interchangeably with CS492X. Unless otherwise specified, CS492X should be interpreted as applying to the CS4926 and CS4928.

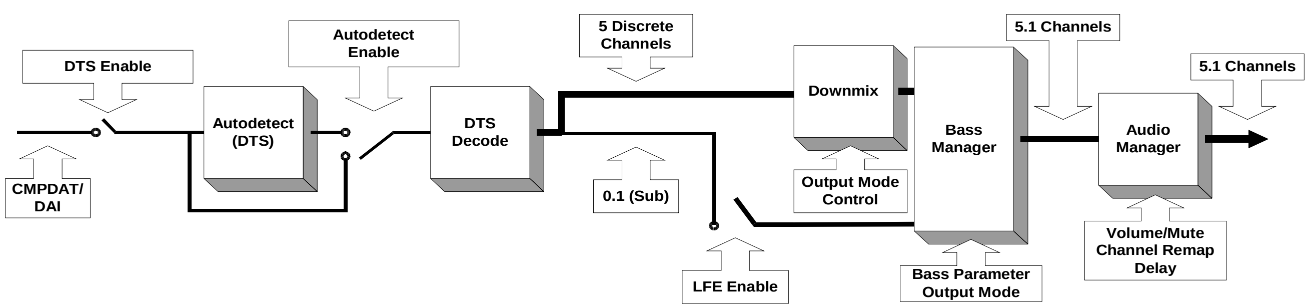


Figure 1. DTS Code Functional Diagram

2. HARDWARE CONFIGURATION

After download or soft reset, and before kickstarting the application (please see Section 4.1 "Audio Manager", for more information on kickstarting), the host has the option of changing the default hardware configuration. Address checking for serial communication and data type and format for digital data input and output can be changed through hardware configuration settings. The entire hardware configuration can only be changed immediately after download or soft reset.

2.1 Supported Input/Output Modes

The CS492X has two input ports and one output port. AN115 describes the digital audio formats supported by the ports and gives a description of the ports themselves. The capabilities of each port are presented in AN115 although all configurations are not supported by all applications. Each Software Application User's Guide specifies the exact input/output modes supported by the application.

Referring to AN115, Table 1 shows the input/output modes that are supported by the DTS 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 DTS

3. APPLICATION MESSAGING

While using the CS492X it may be necessary to control or monitor the application to take full advantage of the rich feature set employed by the CS492X and its software. Application messaging allows the user to do this. Whether it be configuring the part after download, e.g. enabling DTS decode, or changing run-time parameters, e.g. controlling digital volume, the host will use application messaging to communicate with the CS492X.

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

3.1 Indexed Module Communication Protocol

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

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

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

3.1.1 Write Session

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

Table 2 shows the format of a write message.

3.1.2 Solicited Read Message Format

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

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

After the host sends the read command word it should wait for the $\overline{\text{INTREQ}}$ line to fall. See the CS4923/4/5/6/7/8/9 Hardware User's Guide for more on hardware communication with the CS492X.

Write Command Word:

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

Write Data Word:

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

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

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

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

Table 2. Write Message Format

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

3.2 Unsolicited Read Message Format

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

The 6 byte unsolicited read messages from the CS492X consist of a 3 byte read command word which defines the type of unsolicited message (as described in Section 4.4 “Unsolicited Messages (Read-Only)”) and a 3 byte associated data word that contains more information describing a system condition. Every time the existence of a message is detected (by sensing that $\overline{\text{INTREQ}}$ is low), the host should read out the 6 byte read unsolicited message.

Table 5 shows the format of an unsolicited read message.

4. APPLICATION MODULES

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

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

Read Command Word:

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

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

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

Table 3. Read Command Message Format

Read Response Command Word:

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

Read Response Data Word:

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

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

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

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

Table 4. Read Response Message Format

Unsolicited Read Command Word:

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

Unsolicited Read Data Word:

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

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

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

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

Table 5. Unsolicited Read Message Format

- 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 0x000000-0x7FFFFF. This range is linear, i.e. 1/4 volume, or -12 dB, is represented as 0x1FFFFF 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 (variable setting/max value for variable)

  e.g. 20 * log (0x1FFFFF/0x7FFFFF) == -12 dB

  Numbers preceded by 0x should be interpreted as hex, and numbers followed by ‘b’ should be interpreted as binary. All values sent to the CS4923/4/5/6/7/8/9 should be converted to hexadecimal. Likewise all values read from the part are in hexadecimal.

4.1 Audio Manager

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

Write Message = 0x8800HH 0xhhhhhh

Read Request Message = 0x0900HH;

Read Response Message = 0x8900HH 0xhhhhhh

0xHH = index

0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	AUDIO_MGR_CONTROL	**Bit 15** = Audio_Config_Change_Notification_Enable = 0/1= Disable/Enable unsolicited message notification of any change in audio configuration of input bitstream (Valid only for AC-3 application). See “Unsolicited Messages (Read-Only)” on page 23 for more details **Bit 14** = Startup_Autodetect_Bypass_Enable= 0/1= Disable/Enable bypassing of autodetection at start-up, i.e. for the first time after download/reset. (Valid only with Autodetect_Enable==1). Note: If bit 14 is enabled, during readback this bit is reserved and may read 0 or 1. **Bit 12** = Autodetect_Enable = 0/1= Disable/Enable autodetect functionality. **Bit 4** = PLL_Enable = 0/1 = Disable/Enable Phase Locked Loop generation of DSP clock. It is mandatory that the PLL be enabled for Rev D CS492X. **Bit 0** = Kickstart_Enable = 0/1 = Disable/Enable Kickstart of application. *Default = 0x000000*
0x06	MASTER_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*
0x07	L_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*
0x08	C_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*
0x09	R_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*
0x0a	LS_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*
0x0b	RS_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*
0x0c	LFE_VOLUME	0x800001-0x7FFFFF (-1.0 to 1.0). *Default* = 0x7FFFFF*

Table 6. Audio Manager

Index	Variable	Dataword Content
0x0d	MUTE	0/1 = Unmute/mute audio. This is a soft mute. *Default = 0*
0x0e	DAO0_CHANNEL	0...5 = Channel type for Digital Audio Output 0. Each channel must be mapped to one and only one unique output. *Default* = 0(L)*
0x0f	DAO1_CHANNEL	0...5 = Channel type for Digital Audio Output 1. Each channel must be mapped to one and only one unique output. *Default* = 2(R)*
0x10	DAO2_CHANNEL	0...5 = Channel type for Digital Audio Output 2. Each channel must be mapped to one and only one unique output. *Default* = 3(Ls)*
0x11	DAO3_CHANNEL	0...5 = Channel type for Digital Audio Output 3. Each channel must be mapped to one and only one unique output. *Default* = 4(Rs)*
0x12	DAO4_CHANNEL	0...5 = Channel type for Digital Audio Output 4. Each channel must be mapped to one and only one unique output. *Default* = 1(C)*
0x13	DAO5_CHANNEL	0...5 = Channel type for Digital Audio Output 5. Each channel must be mapped to one and only one unique output. *Default* = 5(LFE)*
0x16	LAST_UNSOLICITED_MESSAGE	Last Unsolicited Message Value (see Section 4.4) ***READ ONLY*** Default=*0x000000*
0x17	PLL_REGISTER_1	Register 1 value for Phase Locked Loop (no Fs generation). *Default* = 0x07f270 (CLKIN=12.288MHz, DSPCLK=60MHz)*
0x18	PLL_REGISTER_2	Register 2 value for Phase Locked Loop (no Fs generation). *Default* = 0x0001fe (CLKIN=12.288MHz, DSPCLK=60MHz)*

Table 6. Audio Manager (Continued)

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

4.1.1 Audio_Control: (Index 0x00)

Bit_15:
Audio_Config_Change_Notification_Enable
Makes the CS492X sensitive to changes in the Audio configuration reported in the input bitstream. If any of AMODE or LFF variables change, the CS492X will issue an unsolicited message (see Section 4.4 “Unsolicited Messages (Read-Only)”).

0 - Disables Audio_Config_Change_Notification
1 - Enables Audio_Config_Change_Notification

Bit 14: *Startup_Autodetect_Bypass_Enable*
Maintains the CS492X in autodetect mode, but bypasses autodetection (directly enters input data processing) at start-up, i.e. for the first time after download/reset. This bit should only be changed after download, soft reset, or Application Restart. The state of autodetect should not be modified during run-time.

0 - Disables Startup_Autodetect_Bypass
1 - Enables Startup_Autodetect_Bypass

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

The above Startup_Autodetect_Bypass mode is useful in saving the autodetect latency when the controller already knows that the current input data format is decodable by the downloaded application. The controller is usually aware of this due to the autodetection message from the previous (most recent) downloaded application. NOTE: If this mode is enabled, Bit 14 readback value is reserved (either 0 or 1 may be received and should be ignored).

Bit 12: *Autodetect_Enable*
Puts the CS492X into autodetect mode (see Section 4.4 “Unsolicited Messages (Read-Only)”). This bit should only be changed after download, soft reset, or Application Restart. The state of autodetect should not be modified during run-time.

0 - Disables Autodetect
1 - Enables Autodetect

Bit 4: *PLL_Enable*
Enables the use of the Phase Locked Loop (PLL) to generate internal DSPCLK from the desired external CLKIN.

0 - Disables PLL
1 - Enables PLL

The register values in PLL_Register_1 (0x17) and PLL_Register_2 (0x18) define the desired DSP clock and reference CLKIN used. The default values at download setup a 60 MHz DSPCLK from a 12.288 MHz external CLKIN. For a CLKIN of 11.2896 MHz, PLL_Register_1=0x24BC34 and PLL_Register_2 = 0x0001fe should be used. These register values should be downloaded prior to Kickstart with PLL_Enable.

Bit 0: *Kickstart Enable*
Puts CS492X into run-time mode. Indicates that all hardware and software configuration has been completed and CS492X can start the application.

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

4.2 DTS Manager

Write Opcode = 0x90; Read Opcode = 0x11; Read Response Opcode = 0x91

Write Message = 0x9000HH 0xhhhhhh

Read Request Message = 0x1100HH;

Read Response Message = 0x9100HH 0xhhhhhh

0xHH = index

0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	DTS_CONTROL	**Bit 12 =** LFE_Summer_Enable = 0/1 = Disable/Enable summing of all channels into LFE output. **Bits 7:4** = Output_Mode = 0...7 0 = 2/0 Lt, Rt Surround Encoded. 1 = Reserved. 2 = 2/0 L, R 3 = 3/0 L, C, R 4 = Reserved. 5 = Reserved. 6 = 2/2 L, R, Ls, Rs 7 = 3/2 L, C, R, Ls, Rs **Bit 0 =** DTS_Enable = 0/1 = Disable/Enable DTS decoding. Note: Undesired Channel outputs should be explicitly muted by setting the corresponding Ch_Vol (see Audio Manager) to 0.0 *Default* = 0x000070*
0x01	FTYPE	0..1 = Frame Type Identifier. ***READ-ONLY***
0x02	SHORT	0..31 = Deficit Sample Count. ***READ-ONLY***
0x03	CPF	0..1 = CRC Present Flag. ***READ-ONLY***
0x04	NBLKS	5..127 = Number of PCM Sample Blocks. ***READ-ONLY***
0x05	FSIZE	96..8192 = Primary Frame Byte Size. ***READ-ONLY***
0x06	AMODE	0..63 = Audio Channel Arrangement. ***READ-ONLY***
0x07	SFREQ	0..15 = Source Sampling Frequency. ***READ-ONLY***
0x08	RATE	0..31 = Transmission Bit Rate. ***READ-ONLY***

Table 7. DTS Manager

Index	Variable	Dataword Content
0x09	MIX	0..1 = Embedded Down Mix Enabled. ***READ-ONLY***
0x0a	DYNF	0..1 = Embedded Dynamic Range Flag. ***READ-ONLY***
0x0b	TIMEF	0..1 = Embedded Time Stamp Flag. ***READ-ONLY***
0x0c	AUXF	0..1 = Auxiliary Data Flag. ***READ-ONLY***
0x0d	EXT_AUDIO	0..1 = Extended Coding Flag. ***READ-ONLY***
0x0e	ASPF	0..1 = Audio Sync Word Insertion Flag. ***READ-ONLY***
0x0f	LFF	0..3 = Low Frequency Effects Flag. ***READ-ONLY***
0x10	HFLAG	0..1 = Predictor History Flag Switch
0x11	FILTS	0..1 = Multi-rate Interpolator Switch. ***READ-ONLY***
0x12	PCMR	0..7 = Source PCM coding Resolution. ***READ-ONLY***
0x13	SUMF	0..1 = Front Sum / Difference Flag. ***READ-ONLY***
0x14	SUMS	0..1 = Surround Sum / Difference Flag. ***READ-ONLY***
0x15	SUBFS	0..15 = Number of Subframes. ***READ-ONLY***
0x16	PCHS	0..7 = Number of Primary Audio Channels. ***READ-ONLY***
0x17	SSC	0..3 = Subsubframe Count. ***READ-ONLY***
0x18	PSC	0..7 = Partial Subsubframe Sample Count. ***READ-ONLY***
0x19	RANGE	Dynamic Range Coefficient. ***READ-ONLY***
0x1a..0x1e	SUBS[0...4]	0..31 = Subband Activity Count. ***READ-ONLY***
0x1f..0x23	VQSUB[0...4]	0..31 = High Frequency VQ Start Subband. ***READ-ONLY***
0x24..0x28	JOINX[0...4]	0..7 = Joint Intensity Coding Index. ***READ-ONLY***
0x29..0x32	DOWN[0...4][0...1]	0..127 = Scale Factors for downmixing. ***READ-ONLY***

Table 7. DTS Manager (Continued)

4.2.1 DTS_Control

Bit 12: *LFE_Summer_Enable*
This bit will enable summing of all channels into the LFE output. No low pass filtering is performed on the LFE output so external low pass filtering should be performed if it is desired.

1 - Enable summing of all channels into LFE output
0 - Disable summing of all channels into LFE output

Bits 7:4: *Output_Mode*
These bits set up the output speaker downmix configuration. When configuring the Output _Mode, first the individual volumes of each undesired channel should be set to 0. Next, the desired output mode should be specified. This should be done before kickstart of the application. For example, if output mode 3/0 (L, C, R) is chosen, Ls and Rs volume must explicitly be set to 0 in the Audio Manager module to guarantee no output on those channels.

0x0 = 2/0 L, R Dolby Surround compatible
0x2 = 2/0 L, R
0x3 = 3/0 L, C, R
0x6 = 2/2 L, R, Ls, Rs
0x7 = 3/2 L, C, R, Ls, Rs

4.2.2 DTS Stream Information

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

For example, to read the AMODE value, host should issue the 3 byte Read Request 0x110006 and then read the 6 byte Read Response 0x910006, 0x00000h, where the dataword (latter 3 bytes) indicates the reported value of amode. Please refer to the DTS Specification for more information on these stream variables.

4.3 Generalized Bass Manager

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

Write = 0x9400HH 0xhhhhhh

Read Request = 0x1500HH;

Read Response = 0x9500HH 0xhhhhhh

0xHH = index

0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	BASS_MGR_CONTROL	**Bit 16** = LFE_Only_Sub = 0/1 = Disable/Enable LFE Only to subwoofer. **Bit 14** = 4th_Order_LPF = 0/1 = Disable/Enable 4th order LPF at output of summer. Bit 12 must be set in addition to Bit 14 to enable the 4th order LPF. **Bit 12** = LPF = 0/1 = Disable/Enable LPF at output of summer. **Bit 8** = HPF_L = 0/1 = Disable/Enable L channel HPF. **Bit 7** = HPF_C = 0/1 = Disable/Enable C channel HPF. **Bit 6** = HPF_R = 0/1 = Disable/Enable R channel HPF. **Bit 5** = HPF_Ls = 0/1 = Disable/Enable Ls channel HPF. **Bit 4** = HPF_Rs = 0/1 = Disable/Enable Rs channel HPF. **Bit 0** = Bass_Mgr = 0/1 = Disable/Enable Bass Manager post-processing. Default = 0x000000
0x01	INPUT_L_LEVEL	0x000000-0x7FFFFF = Level adjustment for input L channel pass-through. *Default* = 0x7FFFFF (0 dB)*
0x02	INPUT_C_LEVEL	0x000000-0x7FFFFF = Level adjustment for input C channel input pass-through. *Default* = 0x7FFFFF (0 dB)*
0x03	INPUT_R_LEVEL	0x000000-0x7FFFFF = Level adjustment for input R channel input pass-through. *Default* = 0x7FFFFF (0 dB)*
0x04	INPUT_LS_LEVEL	0x000000-0x7FFFFF = Level adjustment for input Ls channel pass-through. *Default* = 0x7FFFFF (0 dB)*
0x05	INPUT_RS_LEVEL	0x000000-0x7FFFFF = Level adjustment for input Rs channel pass-through. *Default* = 0x7FFFFF (0 dB)*
0x06	INPUT_LFE_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE channel pass-through. *Default* = 0x7FFFFF (0 dB)*
0x07	SUM_OUTPUT_L_LEVEL	0x000000-0x7FFFFF = Level adjustment for bass summer output contribution to L channel output. *Default* = 0x000000 (-∞ dB)*
0x08	SUM_OUTPUT_C_LEVEL	0x000000-0x7FFFFF = Level adjustment for bass summer output contribution to C channel output. *Default* = 0x000000 (-∞ dB)*
0x09	SUM_OUTPUT_R_LEVEL	0x000000-0x7FFFFF = Level adjustment for bass summer output contribution to R channel output. *Default* = 0x000000 (-∞ dB)*

Table 8. Bass Manager

Index	Variable	Dataword Content
0x0a	SUM_OUTPUT_LS_LEVEL	0x000000-0x7FFFFF = Level adjustment for bass summer output contribution to Ls channel output. *Default* = 0x000000 (-∞* dB)
0x0b	SUM_OUTPUT_RS_LEVEL	0x000000-0x7FFFFF = Level adjustment for bass summer output contribution to Rs channel output. *Default* = 0x000000 (-∞* dB)
0x0c	LFE_L_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE contribution to L channel output. *Default* = 0x000000 (-∞* dB)
0x0d	LFE_C_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE contribution to C channel output. *Default* = 0x000000 (-∞* dB)
0x0e	LFE_R_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE contribution to R channel output. *Default* = 0x000000 (-∞* dB)
0x0f	LFE_LS_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE contribution to Ls channel output. *Default* = 0x000000 (-∞* dB)
0x10	LFE_RS_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE contribution to Rs channel output. *Default* = 0x000000 (-∞* dB)
0x11	SUM_INPUT_L_LEVEL	0x000000-0x7FFFFF = Level adjustment for input L channel contribution to bass summer input. *Default* = 0x16C311 (-15 dB)*
0x12	SUM_INPUT_C_LEVEL	0x000000-0x7FFFFF = Level adjustment for input C channel contribution to bass summer input. *Default* = 0x16C311 (-15 dB)*
0x13	SUM_INPUT_R_LEVEL	0x000000-0x7FFFFF = Level adjustment for input R channel contribution to bass summer input. *Default* = 0x16C311 (-15 dB)*
0x14	SUM_INPUT_LS_LEVEL	0x000000-0x7FFFFF = Level adjustment for input Ls channel contribution to bass summer input. *Default* = 0x16C311 (-15 dB)*
0x15	SUM_INPUT_RS_LEVEL	0x000000-0x7FFFFF = Level adjustment for input Rs channel contribution to bass summer input. *Default* = 0x16C311 (-15 dB)*
0x16	SUM_INPUT_LFE_LEVEL	0x000000-0x7FFFFF = Level adjustment for input LFE channel contribution to bass summer. *Default* = 0x47FACD (-5 dB)*
0x17	LPF_48_B0	0x000000-0x7FFFFF = b0 coefficient for Low Pass Filter, Fs=48 KHz. *Default* = 0x014D4C*
0x18	LPF_48_B1	0x000000-0x7FFFFF = b1 coefficient for Low Pass Filter, Fs=48 KHz. *Default* = 0x014D4C*
0x19	LPF_48_A1	0x000000-0x7FFFFF = a1 coefficient for Low Pass Filter, Fs=48 KHz. *Default* = 0x7D6567*

Table 8. Bass Manager (Continued)

Index	Variable	Dataword Content
0x1a	HPF_48_B0	0x000000-0x7FFFFF = b0 coefficient for High Pass Filter, Fs=48 KHz. *Default* = 0x7F7750*
0x1b	HPF_48_B1	0x000000-0x7FFFFF = b1 coefficient for High Pass Filter, Fs=48 KHz. *Default* = 0x8088B0*
0x1c	HPF_48_A1	0x000000-0x7FFFFF = a1 coefficient for High Pass Filter, Fs=48 KHz. *Default* = 0x7EEEA0*
0x1d	LPF_44_B0	0x000000-0x7FFFFF = b0 coefficient for Low Pass Filter, Fs=44 KHz. *Default* = 0x016A73*
0x1e	LPF_44_B1	0x000000-0x7FFFFF = b1 coefficient for Low Pass Filter, Fs=44 KHz. *Default* = 0x016A73*
0x1f	LPF_44_A1	0x000000-0x7FFFFF = a1 coefficient for Low Pass Filter, Fs=44 KHz. *Default* = 0x7D2B19*
0x20	HPF_44_B0	0x000000-0x7FFFFF = b0 coefficient for High Pass Filter, Fs=44 KHz. *Default* = 0x7F6B48*
0x21	HPF_44_B1	0x000000-0x7FFFFF = b1 coefficient for High Pass Filter, Fs=44 KHz. *Default* = 0x8094B8*
0x22	HPF_44_A1	0x000000-0x7FFFFF = a1 coefficient for High Pass Filter, Fs=44 KHz. *Default* = 0x7ED68F*
0x23	LPF_32_B0	0x000000-0x7FFFFF = b0 coefficient for Low Pass Filter, Fs=32 KHz. *Default* = 0x01F171*
0x24	LPF_32_B1	0x000000-0x7FFFFF = b1 coefficient for Low Pass Filter, Fs=32 KHz. *Default* = 0x01F171*
0x25	LPF_32_A1	0x000000-0x7FFFFF = a1 coefficient for Low Pass Filter, Fs=32 KHz. *Default* = 0x7C1D1F*
0x26	HPF_32_B0	0x000000-0x7FFFFF = b0 coefficient for High Pass Filter, Fs=32 KHz. *Default* = 0x7F3365*
0x27	HPF_32_B1	0x000000-0x7FFFFF = b1 coefficient for High Pass Filter, Fs=32 KHz. *Default* = 0x80CC9B*
0x28	HPF_32_A1	0x000000-0x7FFFFF = a1 coefficient for High Pass Filter, Fs=32 KHz. *Default* = 0x7E66CA*

Table 8. Bass Manager (Continued)

Notes: "*Default*" vs. "*Default**" See definition on page 11.

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

AN120 has examples of many typical bass management configurations.

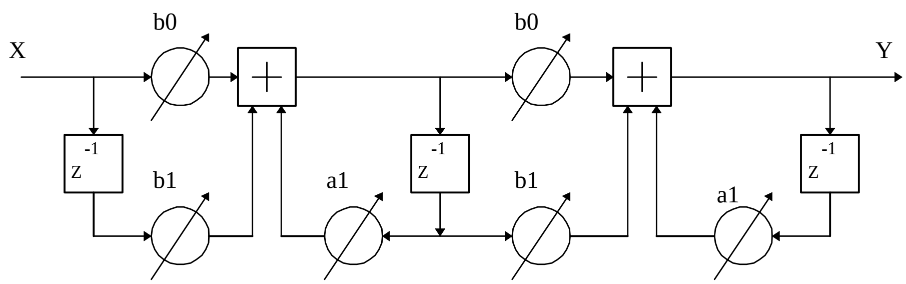


Figure 2. Filter Topology

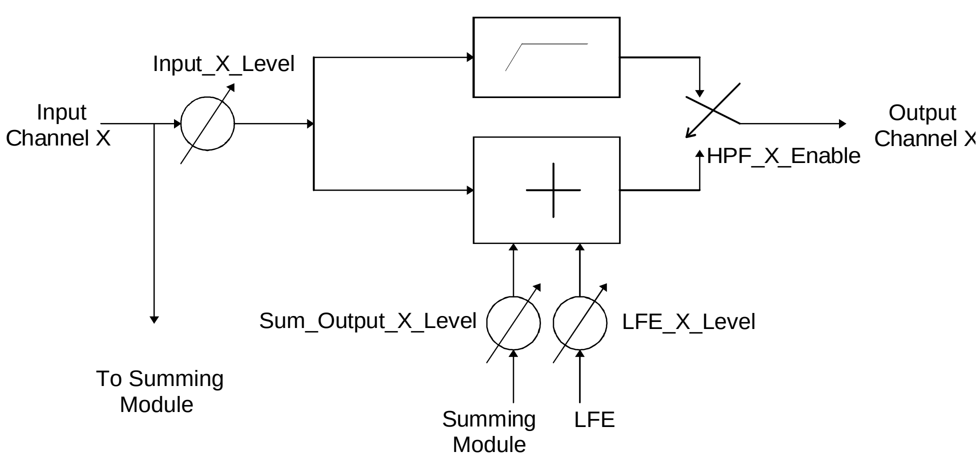


Figure 3. Bass Manager Processing Unit

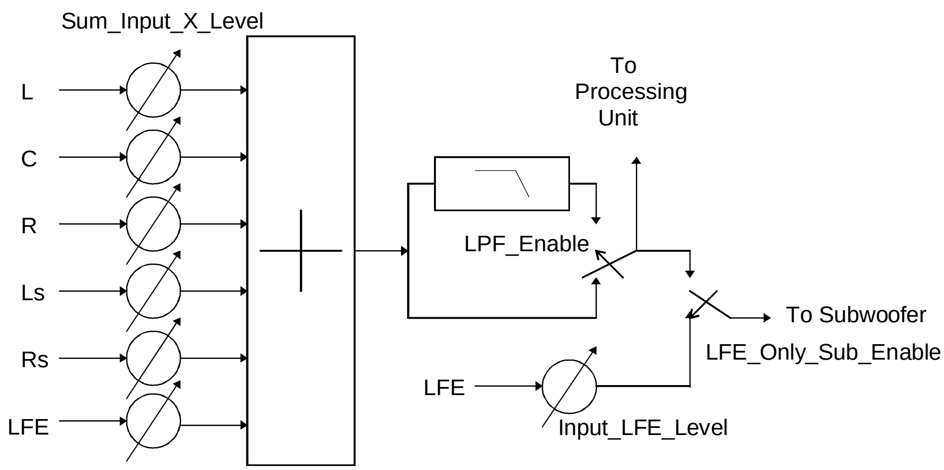


Figure 4. Subwoofer Summing Module

4.4 Unsolicited Messages (Read-Only)

No Write Message. No Read Request.

Unsolicited Read Response = 0x8700HH 0xhhhhhh

0xHH = index,

0xhhhhhh = data value

Index	Variable	Dataword Content
0x00	AUTODETECT_RESPONSE	**Bit 23** = Decodable_Stream_Flag= 0/1 = This stream is not/is decodable by the application (no need for new download if 1). **Bits 22:6** = Reserved. **Bit 5** = Non_IEC61937_Stream_Flag= 1/0 = This stream is not/is IEC61937 compressed data. **If Non_IEC61937_Stream_Flag=1** **Bits 4:0** = Non_IEC61937 Stream Descriptor. 0x00 = Silent Input Data (Out of Application Sync) 0x01 = DTS Format-16 elementary stream 0x02 = DTS Format-14 elementary stream 0x03 = Linear PCM stream 0x04 = HDCD PCM stream (only available in HDCD application) 0x05-0x1F = Reserved **If Non_IEC61937_Stream_Flag=0** **Bits 4:0** = 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): 0x00 = Never Reported. (Null data is ignored by CS492X) 0x01 = AC-3 data 0x02 = Reserved. 0x03 = Never Reported. (Pause is ignored by CS492X) 0x04 = MPEG-1 Layer 1 data 0x05 = MPEG-1 Layer 2 or 3 data or MPEG-2 without extension. 0x06 = MPEG-2 data with extension 0x07 = Reserved. 0x08 = MPEG-2 Layer 1 Low sampling frequency 0x09 = MPEG-2 Layer 2 or 3 Low sampling frequency 0x0A = Reserved 0x0B = DTS-1 data (512-sample bursts). 0x0C = DTS-2 data (1024-sample bursts). 0x0D = DTS-3 data (2048-sample bursts). 0x0E - 0x1F = Reserved
0x03	AUDIO_CONFIGURATION_ CHANGED	**Bits 23:0 =** Reserved. The DSP will send out this message to indicate that one of the following has changed in the DTS stream: AMODE, LFF. It is the hosts responsibility to check exactly what has changed.
0x10	PLL_OUT_OF_LOCK	**Bit 23 =** 1. **Bits 22:0** = Reserved.

Table 9. Unsolicited Messages

4.4.1 Autodetect Operation

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

1) Host downloads CS492X with a tentative application code, for this example we will use AC3_****.LD.

2) Host then configures the CS492X hardware appropriately and sets up application parameters as desired - including enable of the desired application. For this example we will say the code is configured for AC-3® decode.

3) Host then kickstarts CS492X with Autodetect enabled (see Section 4.1 "Audio Manager" for details).

4) The autodetect module of the enabled application of the CS492X analyzes the input for a maximum of 500 ms of non-silent/non-pause data and determines the content of the input bitstream.

5) (a) If the enabled application can play the detected input (i.e. if AC-3 was detected in this case), then the CS492X issues an Unsolicited Message to the host indicating the datatype with Decodable_Bitstream_Flag=1. In our example of the AC-3 stream, the message would be 0x870000 0x800001. CS492X then goes ahead and processes it according to the application parameters as setup in Step 2 above.

   (b) If the enabled application cannot play the detected input (say Non-IEC61937 LD DTS was detected), then the CS492X soft mutes the outputs, and issues an Unsolicited Message to the host indicating the datatype with Decodable_Bitstream_Flag=0. In our example, the message would be 0x870000 0x000021.

   On receiving this message, host repeats Steps 1 onwards but this time downloads the DTS code to the CS492X (along with correct hardware configuration and application configuration for DTS with autodetect enabled). Subsequently, DTS will be detected within 500 mS and successfully played by the new DTS code, after sending the corresponding unsolicited message (0x870000 0x800021).

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

6) After the above steps and while CS492X is successfully playing the input bitstream (still AC-3 in our example), if the host receives external information that the input has been changed (e.g. the user selects a new source using the front panel buttons), then before switching the input data to the CS492X, the host should send an Application Restart message (see Section 4.1 "Audio Manager") or a Soft Reset. If application restart is used, this effectively puts CS492X in Step 2, without changing the output hardware configuration (i.e. output clocks are not interrupted). The input configuration would need to be changed if it is different than the default. If Soft Reset is issued then the entire hardware configuration needs to be resent.

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

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

   If the input content is detected as different (non-AC-3 in our example), the CS492X responds as in Step 5b and continues monitoring the input stream for change in content.

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

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

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

If the input content is detected as unchanged (still AC-3 in our example), CS492X continues processing it like in Step 5a, without requiring any further action from the host. This situation could arise due to a pause or track change upstream in the source, like from a player. In the case of compressed data being played currently (like AC-3 in our example), the host could see an out-of-sync Unsolicited Message followed by an Unsolicited Message indicating AC-3 data. The presence of out-of-sync messages due to 'special' functions like pause or track change is completely dependent upon the audio source (such as the DVD player).

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

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

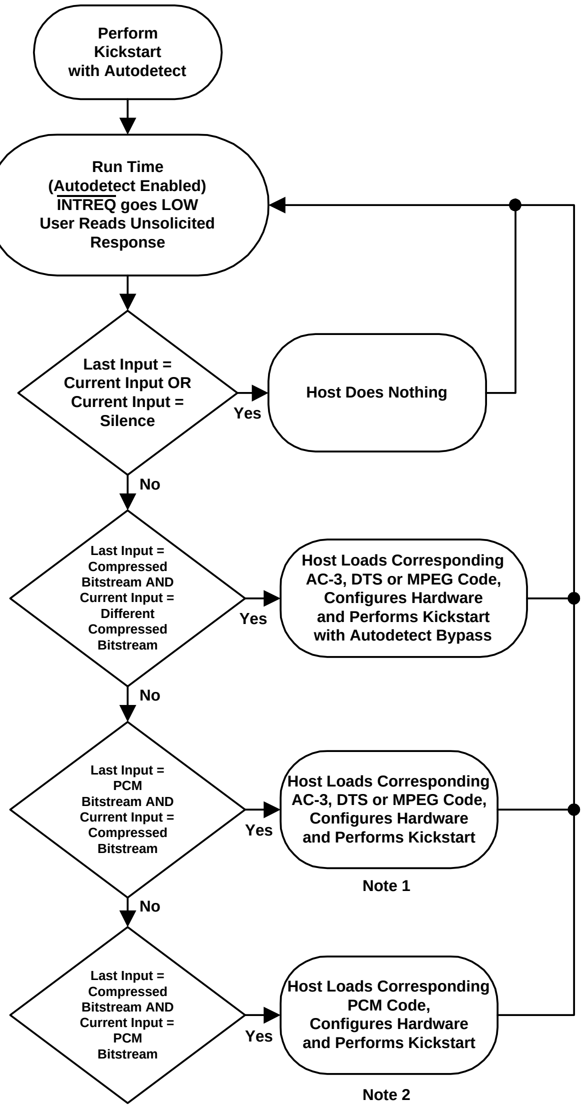


Figure 5. Generic Autodetection Flow Chart

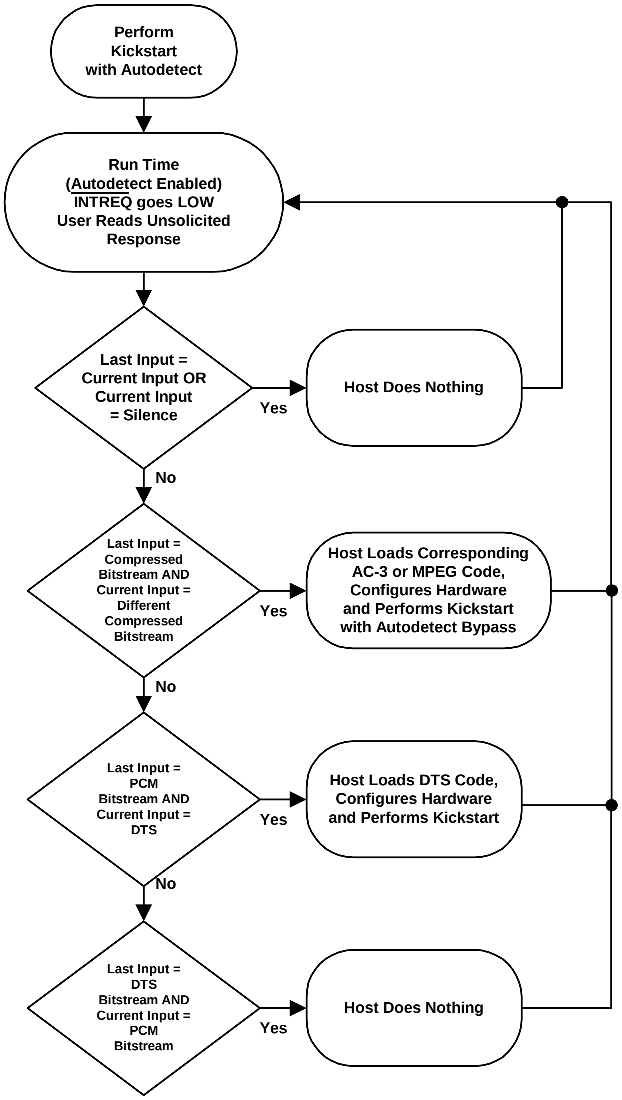


Figure 6. DTS Specific Autodetect Flow Chart

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

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

4.4.2 Special Considerations For DTS Autodetect

A DTS CD is treated like a linear PCM CD by a CD/DVD player. During FF/REW ("trick" modes), the CD/DVD player drops audio samples to create an effect audible to the listener. Dropping DTS data, however, causes corruption of the bit-stream which makes the data indistinguishable from linear PCM. As a result, the DTS decoder will generate an autodetect message indicating PCM was detected. In normal usage, the time required to change from a DTS CD to a linear PCM CD should trigger an autodetect message of silence. An abrupt transition from DTS to PCM should indicate to the system controller that a "trick" mode is being used and PCM code should not be loaded, unless the system controller has special knowledge of perfect DTS/PCM splice (e.g. a button on the device front panel). Please see Figure 6 for an overview of the previously described flow.

4.4.3 Typical Download and Configuration

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.

For the DTS code, DTS must be enabled *before* the autodetect enable is sent with the kickstart. If the Autodetect function senses DTS data, then the appropriate Unsolicited Message is sent and decode will start automatically without any direction from the host.

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

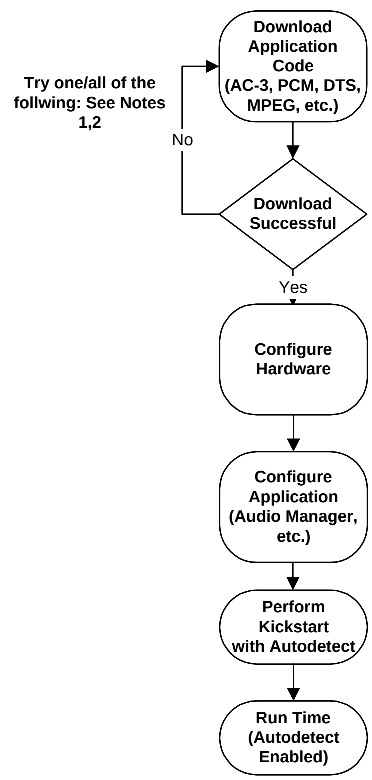


Figure 7. Typical Download and Configuration

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

5. APPLICATION CONFIGURATION EXAMPLES

This section covers various application modes available with the CS4926/8. Although many application modes are possible, only the most common modes have been presented here as examples. This should not be considered the limit of what modes to run the part in as the system designer can configure the CS4926/8 to fit the needs of the system.

Table 10 in the following section provides the message to be sent and what the message is doing. Care should be taken to guarantee that the correct value is sent directly after the associated opcode and index word. The command to kickstart the application should always be sent last.

5.1 DTS decode with Autodetect

In this mode the input should be compressed DTS. The data will be taken from the compressed input port as designated by the hardware configuration message. Table 10 gives a description of DTS decode with Autodetect. If the input on this port is not DTS the Autodetect function will notify the host as described in Section 4.4 “Unsolicited Messages (Read-Only)”.

Figure 8 gives an example of pseudocode to configure either part for basic decode of DTS. This pseudocode can be used as a template for other configurations by swapping out the message array and the size.

6. SPECIAL CONSIDERATIONS FOR DTS CERTIFICATION

Systems using the autodetection feature of the DTS Application Code may require the ability to bypass autodetection in order to obtain DTS certification.

The autodetection engine does not look for certain streams that are considered atypical for real world scenarios. Specifically low bitrate streams that do not employ IEC61937 packing are the streams in question.

It should be noted that the decoder does not have a problem decoding these streams. The autodetection mechanism may suppress these streams and thus the DSP will never decode them if autodetection is enabled.

Because some of these atypical streams are part of the DTS test suite it is suggested that all systems looking for DTS certification employ a method to enable DTS decoding with autodetection bypassed. If a button is not realistic, then other alternatives might include a special keystroke or a menu choice.

Once again please understand that this is not a problem with the decoders ability to be 100% compliant with DTS test streams, since the CS4926 is a DTS certified IC. In addition please be aware that the streams in question are not commercially available. The autodetection will successfully detect all DTS streams that are commercially available.

Module	Index	Description	Opcode and Index	Value
DTS.	DTS Control.	Output Mode 3/2 DTS Enabled	0x900000.	0x000071.
Audio Manager.	Audio Manager Control.	Autodetect Enabled PLL Enabled Kickstart Application.	0x880000.	0x001011.

Table 10. Enabling DTS Decode with Autodetect

```
void DTS_Config()
{
      unsigned char DTS_config_message[] =
                                          {0x90,  0x00,  0x00,
                                          0x00,  0x00,  0x71,

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

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

Figure 8. Pseudocode Example to Configure Part DTS Decode with Autodetect

• Notes •

SMART Analog™