



**MPEG-2**

# **IBM MPEG-2 Audio/Video Decoders User Application Guide**

SE09-4001-00

**Note:** Before using this information and the product it supports, be sure to read the information under "Notices:" on page xi

This document, SE09-4001-00, applies to the IBM MPEG-2 Digital Audio/Video Decoder chip. Changes are made periodically to the information herein.

Reader's comments may be addressed to: Digital Video Products, Dept. V60, 1701 North Street, Endicott, NY, 13760-5553 USA. Reader's comments may also be e-mailed to: [techpubs@vnet.ibm.com](mailto:techpubs@vnet.ibm.com).

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

**(c) COPYRIGHT: International Business Machine Corporation 1997. All rights reserved.**

Note to U.S. Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Table of Contents

<b>Chapter 1 MPEG Overview</b>	<b>1-1</b>
1.1 MPEG Data Stream	1-1
1.1.1 MPEG Data Structure (Video)	1-1
1.1.2 MPEG Data Structure (Audio)	1-6
<b>Chapter 2 Product Description</b>	<b>2-1</b>
2.1 MPEGCD20	2-1
2.2 MPEGCD21	2-1
2.3 MPEGCD20/MPEGCD21 Features	2-1
<b>Chapter 3 Signal Descriptions</b>	<b>3-1</b>
3.1 Host Interface Signals	3-3
3.2 Global Control Signals	3-5
3.3 DRAM Interface Signals	3-5
3.4 Display Interface Signals	3-7
3.5 Audio Interface Signals	3-7
3.6 Phase Lock Loop Signals	3-8
<b>Chapter 4 Functional Description</b>	<b>4-1</b>
4.1 MPEGCD20/MPEGCD21 Block Diagram	4-1
4.2 Host Interface	4-2
4.2.1 Interface Control Signals Settings	4-3
4.3 DRAM Controller	4-4
4.3.1 DRAM Requirements	4-4
4.3.2 Memory Allocations	4-4
4.3.3 DRAM Configuration/Usage	4-5
4.4 Video Decoder	4-9
4.4.1 Internal processor (controller)	4-9
4.4.2 Huffman decoder (Variable Length Code Decoder) (VLCD)	4-9
4.4.3 Inverse Quantizer	4-9
4.4.4 IDCT	4-11
4.4.4.1 Motion Compensation Unit	4-11
4.4.5 Video Rate Buffer	4-11
4.5 On-Screen Display (OSD)	4-12
4.5.1 OSD Region	4-12
4.5.2 Colors and Resolutions	4-13
4.5.3 Blending	4-13
4.5.4 Shading	4-13
4.5.5 Animation Support	4-13
4.5.6 Header Definition	4-13
4.5.6.1 Control Fields	4-14
4.5.6.2 Color Table	4-16
4.5.7 OSD Operations	4-18
4.5.7.1 OSD Control Registers:	4-18
4.5.7.2 OSD Data Input	4-18
4.5.7.3 OSD Data	4-19

4.5.7.4 OSD Data Management	4-19
4.5.7.5 OSD Chroma Decimation	4-19
4.6 Video Display Interface	4-19
4.6.1 3:2 Pulldown	4-20
4.6.2 Single Field Progressive Scan	4-20
4.6.3 Image Re-sizing	4-20
4.6.3.1 Resolution/Aspect Ratio	4-21
4.6.3.2 MPEG-2 Pan/Scan Support	4-21
4.6.4 Picture in Picture Support	4-23
4.6.5 Display Interface Controls and Timings	4-23
4.6.6 Teletext And Vertical Blanking Interval Data Output	4-25
4.7 Audio Decoder	4-26
4.7.1 Audio Processor	4-27
4.7.2 Controlling the Audio Decoder	4-27
4.7.3 Tone Generation	4-27
4.7.4 Playing PCM Data	4-29
4.7.5 Audio Decode Status	4-29
4.7.5.1 Reset Done	4-29
4.7.5.2 Empty	4-30
4.7.5.3 Play	4-30
4.7.6 Audio Clocks Generation	4-30
4.8 Audio Interface	4-30
4.9 PLL External Components And Connections	4-31
4.9.1 27 MHz Clock Input Requirements	4-32
4.10 Audio/Video Synchronization	4-33
4.10.1 STC Master Synchronization	4-33
4.10.2 Video Master Synchronization Mode	4-33
4.10.3 Audio Master Synchronization Mode	4-33
4.10.4 Synchronization Disabled Mode	4-34
4.10.5 Broadcast Environment	4-34
4.10.6 Synchronization With Clock Recovery	4-34
4.10.7 Synchronization Without Clock Recovery	4-35
4.10.8 PC System Environment	4-35
4.11 Error Detection and Concealment	4-35
4.11.1 Video Error Concealment	4-35
4.11.2 Audio Error Concealment	4-36
4.11.3 Transport Error Handling	4-37
4.12 4:2:2 Profile Requirements	4-37
<b>Chapter 5 Programming Reference</b>	<b>5-1</b>
5.1 Host Accessible Registers	5-1
5.1.1 Register Definitions	5-4
5.2 Host Driven Operations	5-30
5.2.1 Initialization	5-30
5.2.2 Resets	5-32
5.2.3 Video FIFO Buffer Access	5-32
5.2.4 Audio FIFO Buffer Access	5-33

5.2.5 Maskable Interrupts to Host	5-33
5.2.6 User Data Access	5-35
5.3 Audio Ancillary Data	5-35
5.4 External Memory (DRAM) Access From Host	5-36
5.4.1 2-Byte DRAM Read/Write	5-37
5.4.2 Block Copy within DRAM	5-38
5.4.3 Block Transfer to DRAM	5-38
5.4.4 Block Transfer from DRAM	5-39
5.4.5 Block Fill in DRAM	5-39
5.4.6 OSD Write to DRAM	5-39
5.4.7 VBI Write to DRAM	5-40
5.5 User Commands	5-40
5.5.1 Command Registers	5-41
5.5.2 List of Commands	5-42
5.5.3 Command Descriptions	5-45
5.6 Command Interface Details	5-46
5.6.1 CD 20 Host Command Execution	5-46
5.6.2 CD20 Configuration	5-47
5.6.2.1 Description of Configure-Chip Command	5-47
5.6.2.2 Execution of Configure-Chip Command	5-48
5.6.2.3 Freeze Frame Support	5-49
5.6.2.4 Invoking Freeze Frame	5-49
5.6.3 Normal Play Support	5-49
5.6.3.1 Immediate Normal Play	5-50
5.6.3.2 Reset Video Rate Buffer Support	5-50
5.6.4 Still Picture (or Sample Play) Support	5-50
5.6.4.1 Invoking Sample Play	5-50
5.6.5 Channel Switching	5-51
5.6.6 VBI Teletext Data	5-52
<b>Chapter 6 Electrical and Physical Specifications</b>	<b>6-1</b>
6.1 Environmental and Electrical Specifications	6-1
6.2 Package Specification	6-2
6.3 Pin Assignments	6-4
6.4 Timing Diagrams	6-13
6.4.1 Host Interface Timing Diagrams	6-13
6.4.2 DRAM Interface Timing Diagrams	6-26
6.4.3 Video Display Interface Timing Diagrams	6-31
6.4.4 Audio Interface Timings	6-48



---

# Figures

- Figure 1-1. Typical Picture Sequence Showing Picture Types 1-3
- Figure 1-2. Section of Picture (frame) Showing Slice and Macroblock(s) 1-4
- Figure 1-3. Macroblock Composition 1-6
- Figure 3-1. Interface Signals 3-2
- Figure 4-1. Functional Block Diagram 4-1
- Figure 4-2. 1MB External DRAM Configuration 4-6
- Figure 4-3. 2MB External DRAM Configuration 4-7
- Figure 4-4. 4MB External DRAM Configuration 4-8
- Figure 4-5. Zigzag Scanning Order of DCT Coefficients 4-10
- Figure 4-6. Alternate Scanning Order of DCT Coefficients 4-10
- Figure 4-7. Display (Image Size) Scaling for Pan Scan 4-22
- Figure 4-8. Ferrite bead detail 4-31
- Figure 4-9. Pin description 4-32
- Figure 5-1. Command Buffer 5-42
- Figure 5-2. Sequence for loading VBI or Teletext Data 5-53
- Figure 6-1. Package Specification 6-3
- Figure 6-2. Host Interface W/R Timings (16/8-bit SRAM Mode) 6-13
- Figure 6-3. Host Interface W/R Timings (8-bit Acknowledge Mode) 6-15
- Figure 6-4. Video Compressed Data And Host Write Timings (8-bit Acknowledge Mode) 6-17
- Figure 6-5. Video Compressed Data Write Timings (8-bit Acknowledge Mode) 6-18
- Figure 6-6. Audio Compressed Data And Host Write Timings (8-bit Acknowledge Mode) 6-19
- Figure 6-7. Audio Compressed Data Write Timings (8-bit Acknowledge Mode) 6-20
- Figure 6-8. Serial Video Compressed Data Write Timings With Byte Sync 6-21
- Figure 6-9. Serial Video Compressed Data Write Timings Without Byte Sync 6-22
- Figure 6-10. Serial Audio Compressed Data Write Timings 6-23
- Figure 6-11. ERROR# signal and parallel compressed data timing. 6-24
- Figure 6-12. ERROR# signal and serial compressed data timing. 6-25
- Figure 6-13. DRAM Interface Timing (Fast Page Mode Read Cycle) 6-26
- Figure 6-14. DRAM Interface Timing (Fast Page Mode Early Write Cycle) 6-28
- Figure 6-15. DRAM Interface Timing: CAS Before RAS (CBR) Refresh Cycle 6-30
- Figure 6-16. Horizontal Positioning Using Hdelay and DISP\_LBOR 6-31
- Figure 6-17. Vertical Positioning Using Vdelay, VBI and DISP\_TBOR 6-31
- Figure 6-18. NTSC Vertical Timing using Composite Blanking and Field ID. V-Delay is set to 0 in this example 6-32
- Figure 6-19. NTSC Vertical Timing using HSYNC and VSYNC (trailing edge) V-Delay is set to 15 in this example. 6-34
- Figure 6-20. Detail of timing window. T= 1 cycle @ 27 MHz 6-35
- Figure 6-21. NTSC Vertical Timing using HSYNC and VSYNC (leading edge). V-Delay is set to 18 in this example. 6-36
- Figure 6-22. Detail of leading edge vsync. T= 1 cycle @ 27 MHz 6-37
- Figure 6-23. NTSC Vertical Timing using H and V signals. V-Delay is set to 21 in this ex-

ample. ....	6-38
Figure 6-24. PAL Vertical Timing using Composite Blanking and Field ID. V-Delay is set to 0 in this example. ....	6-39
Figure 6-25. PAL Vertical Timing using HSYNC and VSYNC (trailing edge). V-Delay is set to 20 in this example. For field 2, an internal adjustment is made to 21. ....	6-41
Figure 6-26. PAL Vertical Timing using HSYNC and VSYNC (leading edge). V-Delay is set to 23 in this example. ....	6-43
Figure 6-27. PAL Vertical Timing using H and V signals. V-Delay is set to 25 in this example. For field 2, an internal adjustment is made to 26. ....	6-45
Figure 6-28. Horizontal Timing. This shows the horizontal timing starting with the activation of HORZ_SYNC_CNTL both for 16 and 8 bit pixel modes ....	6-46
Figure 6-29. Timing Specification. Timing Specifications for the Display Interface. ...	6-47
Figure 6-30. Audio Output Interface in Default Mode ....	6-48
Figure 6-31. Audio Output Interface in I <sup>2</sup> S Mode ....	6-48



---

## Tables

Table 1-1. IBM MPEG-2 Audio/Video Decoder part numbers	x-xi
Table 4-1. Interface control signal settings	4-3
Table 4-2. DRAM Requirements	4-4
Table 4-3. Example of Memory Allocations	4-4
Table 4-4. Control Fields definition	4-14
Table 4-5. Color Table entry fields definition	4-16
Table 4-6. Input Resolutions	4-21
Table 4-7. Attenuation below maximum possible volume.	4-28
Table 4-8. Tone Indices and Frequencies	4-28
Table 4-9. 27 MHz Clock Input requirements	4-32
Table 5-1. Video Stream Data Stored on Interrupt	5-34
Table 5-2. Audio Ancillary Data Store in DRAM	5-35
Table 5-3. Configuration and Control Commands	5-43
Table 5-4. Command Descriptions	5-45
Table 6-1. Maximum Ratings/Operating Range - Environment	6-1
Table 6-2. Current Specifications	6-1
Table 6-3. DC Electrical Characteristics	6-2
Table 6-4. Pin assignments	6-4
Table 6-5. Host Interface W/R Timings (16/8-bit SRAM Mode)	6-14
Table 6-6. Host Interface W/R Timings (8-bit Acknowledge Mode)	6-16
Table 6-7. Video Compressed Data And Host Write Timings (8-bit Acknowledge Mode)	6-17
Table 6-8. Video Compressed Data Write Timings (8-bit Acknowledge Mode)	6-18
Table 6-9. Audio Compressed Data And Host Write Timings (8-bit Acknowledge Mode)	6-19
Table 6-10. Audio Compressed Data Write Timings (8-bit Acknowledge Mode)	6-20
Table 6-11. Serial Video Compressed Data Write Timings With Byte Sync	6-21
Table 6-12. Serial Video Compressed Data Write Timings Without Byte Sync	6-22
Table 6-13. Serial Audio Compressed Data Write Timings	6-23
Table 6-14. Error Signal Input to Compressed Data Strobe Timings	6-24
Table 6-15. Error Signal Input to Serial Compressed Data Clock Timings	6-25



---

## Notices:

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, USA.

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM

IBM Logo

Other company, product, and service names, which may be denoted by a double asterisk (\*\*), may be trademarks or service marks of others.

---

# Preface

The *IBM MPEG-2 Audio/Video Decoder User Application Guide* is intended for engineers and designers who are familiar with MPEG audio and video compression concepts and are considering the **IBM MPEG-2 Audio/Video Decoder (MPEGCD20)**, or the **IBM MPEG-2 Enhanced Audio/Video Decoder (MPEGCD21)** for an application. The following table lists the part numbers:

**Table 1-1. IBM MPEG-2 Audio/Video Decoder part numbers**

OEM Part Number	Module Part Number	Descriptive Name
IBM39 MPEGCD20 PFD 22C	75H1315	AUDIO/VIDEO DECODER
IBM39 MPEGCD21 PFE 18C	75H1316	ENHANCED AUDIO/VIDEO DECODER

## Where to Find More Information

You can find more information on MPEG-1 and MPEG-2 coded representation of moving pictures and associated audio in the “Bibliography”.

## Applicable Standards

The following standards are used in this document:

### MPEG-2 Standard

ITU-T Rec. H.222 | ISO/IEC 13818-1: 1994 Standard  
ITU-T Rec. H.262 | ISO/IEC 13818-2: 1994 Standard  
ISO/IEC 13818-2 Amendment 3  
ISO/IEC 13818-3: 1994 Standard

### CCIR 601

International Radio Consultative Committee

### MPEG-1 Standard

ISO/IEC 11172-1: 1991 Standard  
ISO/IEC 11172-2: 1991 Standard

## Sending Comments for This Publication

You can send comments by e-mailing [techpubs@ibm.com](mailto:techpubs@ibm.com) or by FAX at (607)755-9069. Please include:

- Publication title
- Order number (including suffix)
- Page, section title, or topic

Your comments will help improve our documentation. Thank you.



---

# Chapter 1. MPEG Overview

The MPEG (Moving Pictures Experts Group) committee was formed in 1988 to establish standards for coding moving pictures and associated audio information on digital media. The first phase of their work was completed in 1991, namely ISO standard 11172, which defines MPEG-1. MPEG-1 was defined to cover a continuous bitrate of about 1.5 Mbit/second.

In 1990, the MPEG committee began developing extensions to MPEG-1 to allow more flexibility in input formats, handle higher data rates, and improve error handling. This second phase has defined ISO standard 13818-2, called MPEG-2. The MPEG-2 standard defines several syntactic subsets at which compliant encoders and decoders may operate. Each subset is defined in terms of a **profile** and **level**. The profile designation represents the syntactic subset of the MPEG-2 standard supported by a given encoder or decoder implementation. The level designation provides constraints for various bitstream parameters, for example maximum picture resolution and frame rate.

In MPEG-2 compression, full resolution CCIR601 NTSC (720 X 480 pixels/frame, 30 frames/sec.) and PAL (720 X 576 pixels/frame, 25 frames/sec.) video can be compressed. The syntax for MPEG-2 is very robust and permits various types of compression options, such as the handling of interlaced video source, and frame or field-based coding. The extensions in MPEG-2 allow increased picture quality and bitrate flexibility, enabling applications from the broadcast market to the consumer market segments. Some example applications are DVD (Digital Versatile Disc) and DBS (Direct Broadcast Satellite) systems.

## 1.1 MPEG Data Stream

The MPEG data stream consists of a number of structured layers that are defined in the MPEG-2 standard. The **system layer** contains timing and other supporting data used to separate (demultiplex) the audio and video data, enable the decoding of each, then combine (synchronize) the audio and video during playback. The system level encoding operation involves combining the audio and video compressed data in a way that is consistent with the standard so that the system level decoding operation can separate the audio and video compressed data.

The **audio layer** contains the coded information required to represent the audio portion of the data stream, and the **video layer** contains the coded information required to represent the video portion of the data stream.

### 1.1.1 MPEG Data Structure (Video)

The MPEG data structure defines a layered structured coding sequence. All elements (layers) except the macroblock layer are identified by unique start codes. Each element contains header information followed by data for one or more of the next lower elements. The elements (layers) in this structure are:

- The **Video Sequence Layer (SEQ)** is the top level of coding. This layer provides data relative to picture sizes, rates, and quantization matrixes.

## MPEG Overview

- The **Group of Pictures (GOP) layer** contains one or more GOPs, and follows the video sequence layer. The organization of Groups of Pictures enables random access to a picture within a sequence.
- The **Picture (or Frame) layer** follows the GOP layer. The picture layer consists of the individual pictures (or frames) in the video sequence. There are several types of pictures in the video sequence. These are:

### **I-Picture (I-frame)**

Intra-coded pictures (I-pictures) are coded using only information contained in that picture.

### **P-Picture (P-frame)**

Predictive-coded pictures (P-pictures) are coded using motion compensated prediction from a past I-picture or a past P-picture.

### **B-Picture (B-frame)**

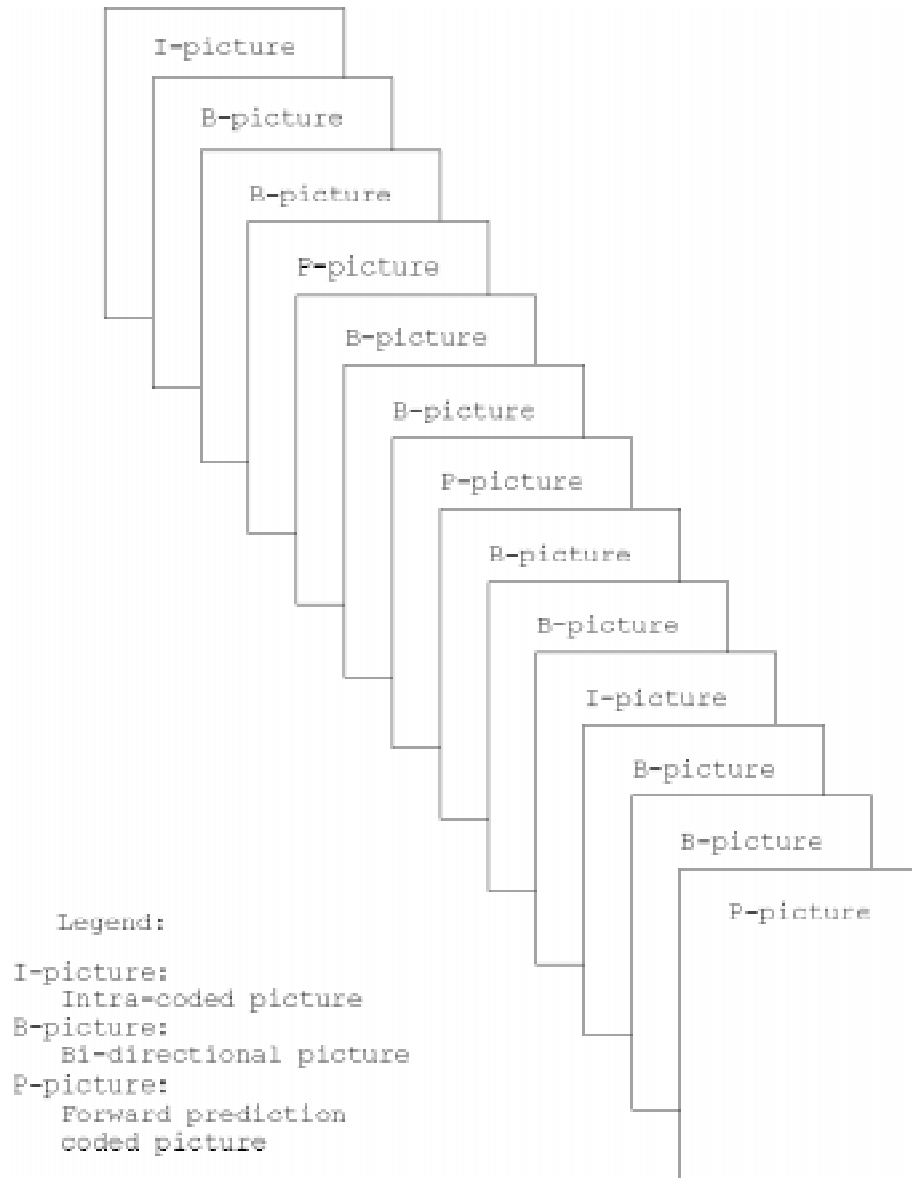
Bidirectionally-coded pictures (B-pictures) are coded using motion compensated information from both a past and a future I-picture or P-picture.

### **D-Picture (D-frame)**

DC-coded pictures (D-pictures) are coded using only DC values of DCT data. D-pictures are not supported.

A typical sequence of the I, P, and B pictures is shown in Figure 1-1.





**Figure 1-1. Typical Picture Sequence Showing Picture Types**

The sequence of pictures shown in Figure 1-1 is the sequence in which the decoder plays back the pictures. The sequence (by picture type) is shown below:

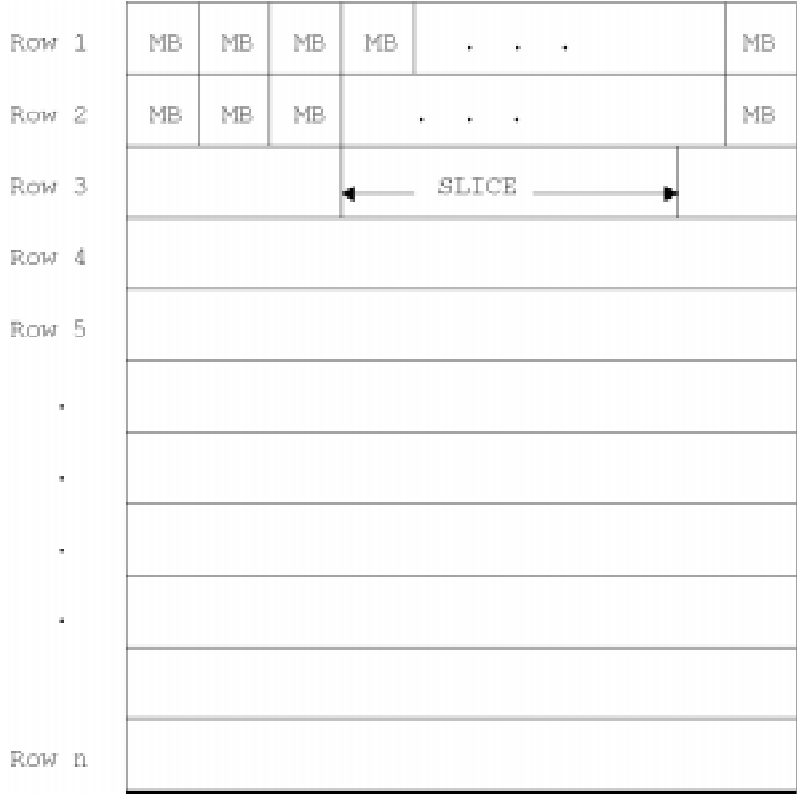
**I - B - B - P - B - B - P - B - B - I - B - B - P**

# MPEG Overview

Because the decoder actually requires a P-picture (or another I-picture) before it can process a B-picture, the actual sequence of pictures in the input encoded data stream would be as shown below:

**I - P - B - B - P - B - B - I - B - B - P - B - B**

- The picture layer is followed by the **slice layer**. The individual pictures are subdivided into slices that each contain an integral number of macroblocks in raster scan order. Slices can start at any macroblock position within a row. Slices can involve multiple rows in MPEG-1, but are defined to start and stop on the same row in the MPEG-2 Main Profile/Main Level description.
- The **macroblock layer** is the last layer of the structure. A macroblock is composed of blocks of 8 x 8 pixels of video data. Each macroblock contains a section of the luminance data and the spatially corresponding chrominance data. The relationship between a slice and a macroblock within a picture is shown in Figure 1-2.



**Figure 1-2. Section of Picture (frame) Showing Slice and Macroblock(s)**

The MPEG-2 standard defines several types of macroblocks based on the composition of the macroblock structure in terms of luminance and chrominance blocks:

The 4:2:0 macroblock format is composed of 6 blocks:

- 4 luminance (Y) blocks
- 1 chrominance (Cb) block
- 1 chrominance (Cr) block

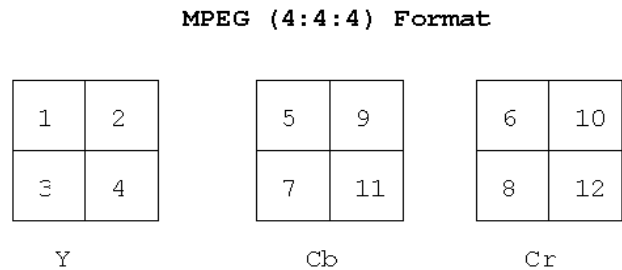
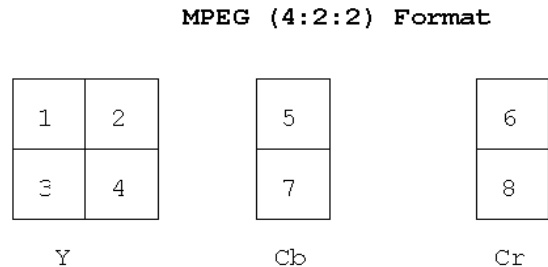
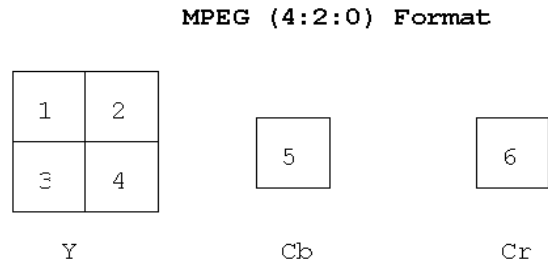
The 4:2:2 macroblock format is composed of 8 blocks:

- 4 luminance blocks
- 2 Cb chrominance blocks
- 2 Cr chrominance blocks

Finally, the 4:4:4 format is composed of 12 blocks:

- 4 luminance blocks
- 4 Cb chrominance blocks
- 4 Cr chrominance blocks

The MPEG-2 Digital Audio/Video Decoder Chip processes the 4:2:0 macroblock that is the basis for the Main Profile/Main Level of the MPEG-2 standard or the 4:2:2 macroblock format as input. The configuration of the blocks in the three macroblock formats is shown in Figure 1-3.



**Figure 1-3. Macroblock Composition**

### 1.1.2 MPEG Data Structure (Audio)

MPEG encoding works by removing redundancy from the data, and by concentrating on accurately representing the information from the original signal that is most important to human audio perception. This is predicted by a psycho-acoustic model that is part of the encoder. There are different models that may be used. MPEG is a 'lossy' compression method, so the decoded output may not precisely match the input, but the psycho-acoustic model will ensure that the difference is as small as possible to human perception.

MPEG audio compression starts by taking groups of digital audio samples called frames. The number of samples in a frame, and the complexity of the compression methods used are determined by the Layer of the MPEG Encoder. Layer II Encoders use more samples per frame and more complex encoding methods than Layer I. Layer II will generally perform a better job of compression than Layer I, but requires more computational effort. This trade-off is decided at encode time.

The compressed audio frames make up an Elementary Stream. This contains all of the information necessary for the MPEG decoder to produce the audio output. Each frame may also include other information, such as copyright, error detection information, or user-defined data. To allow the Audio Elementary Stream to be combined with associated Video data, the frames can be put into Packets. Both MPEG-1 and MPEG-2 have the concept of Packets, but the details are slightly different.

Each Packet starts off with a header which identifies the data that the Packet contains. This allows the Packets to be directed to the proper decoder. The Packet Header can also contain a Presentation Time Stamp (PTS), which indicates when the data should be played. This allows the Audio and Video to be synchronized during playback.



---

## Chapter 2. Product Description

### 2.1 MPEGCD20

The MPEGCD20 is a single-chip decoder solution that accepts video Packetized Elementary Streams (PES) compliant with Main Profile / Main Level (MP@ML), and audio PES compliant with Layer I and Layer II. The decoder is also capable of handling MPEG-1 system packets and elementary streams.

The MPEGCD20 requires just a single 27MHz clock input and includes two on-chip phase-locked-loops (PLLs) that generate the necessary internal clocks as well as all of the required audio sampling frequencies and clocks. This eliminates the need for multiple crystals/oscillators and/or clock generator components, and greatly simplifies the task of achieving tight audio/video synchronization.

Audio and video data can be delivered to the MPEGCD20 at up to 15Mb/s sustained (for I/P/B frames), and up to 50Mb/s sustained for I-frames only in either 4:2:0 or 4:2:2 chroma format.

The MPEGCD20 product is well suited for most MPEG-2 applications including Direct Broadcast Satellite receivers, cable television digital receivers, or other types of high-speed digital audio/video receivers. The MPEGCD20 can also be used in PC multimedia applications that require MPEG-1 and MPEG-2 decompression.

### 2.2 MPEGCD21

The MPEGCD21 has all of the same functions & features of the MPEGCD20 decoder, plus support for the 4:2:2 Profile which includes the following:

- 4:2:2 video chroma format
- up to 50Mb/s IPB sustained data rate
- additional 32 lines for both NTSC and PAL
- unconstrained bits in macroblock
- 4 quantization tables

The MPEGCD21 product is ideal for those applications requiring the highest levels of video quality such as studio-to-studio broadcast, professional editing and video post-processing.

### 2.3 MPEGCD20/MPEGCD21 Features

- Video Decoder - ISO/IEC 13818-2 Main Profile at Main Level
- Audio Decoder - ISO/IEC 13818-3 MPEG-2 Layers I & II, 2 Channel
- Supports European DVB standard

## Product Description

- On-chip PES layer decoding for both audio and video to extract the Presentation Time Stamp (PTS)
- Decodes elementary or MPEG-1 system packet layer streams for both audio and video
- MPEG-2 MP@ML compliance with 2MBytes DRAM. Only 2MBytes of DRAM (70ns) are needed to decode full CCIR601 resolution NTSC and PAL encoded MPEG-2 bitstreams with sustained data rates up to 15Mbits/sec
- Support for sustained data rates up to 50Mb/s for I-frame only with 4:2:0 or 4:2:2 chroma format
- Full 4:2:2 profile compliant (MPEGCD21 only)
  - 4:2:2 chroma format
  - up to 50Mb/s IPB data rate
  - additional 32 lines for NTSC and PAL
  - unconstrained bits in macroblock
  - 4 quantization tables
- User data supported by DRAM access from Host
- Horizontal and vertical filters for output of high quality video. Pan and scan support (1/16 pel accuracy) for display of 16:9 aspect ratio images on a 4:3 aspect ratio display.

The following are the supported image sizes for automatic expansion (4:3 image aspect ratio):

- 352 x 240 NTSC/288 PAL (1:2 horizontal expansion, 1:2 vertical expansion)
- 352 x 480 NTSC/576 PAL (1:2 horizontal expansion)
- 480 x 480 NTSC/576 PAL (2:3 horizontal expansion)
- 544 x 480 NTSC/576 PAL (3:4 horizontal expansion)

The following image sizes are also supported for automatic expansion using Pan & Scan (16:9 image aspect ratio):

- 720 x 480 NTSC/576 PAL (3:4 horizontal expansion)
- 544 x 480 NTSC/576 PAL (9:16 horizontal expansion)
- 480 x 480 NTSC/576 PAL (1:2 horizontal expansion)
- 352 x 240 NTSC/288 PAL (3:8 horizontal expansion, 1:2 vertical expansion)
- 352 x 480 NTSC/576 PAL (3:8 horizontal expansion)

- On-Screen Display (OSD)
  - Multi-region link list OSD with a color table for each region
  - block copy capability
  - Overlay and video blending
  - Video shading in OSD area
  - Animation support
  - Programmable bitmap resolution on a region by region basis



2 bits/pixel pair  
2 bits/pixel  
4 bits/pixel pair  
4 bits/pixel

- Input interface flexibility
  - 8-bit audio/video or 16-bit video compressed data and host data interface
  - 8-bit with acknowledge for easy connection to transport chips
  - Serial audio/video compressed data input for connection to transport chip, and simultaneous parallel host data interface
- Display interface flexibility (programmable controls)
  - Composite blanking and Field ID signals
  - V-sync and H-sync signals
  - V-ref and H-ref signals
  - Programmable signal polarity
- Flexible interface to DRAM
  - 1MB DRAM, 32-bit interface for MPEG-1 and some MPEG-2 streams
  - 2MB DRAM, 64-bit interface for MPEG-2 NTSC/PAL up to 15Mb/s
  - 4MB DRAM addressing capability for extended applications (MPEGCD21 requires 4MB DRAM for decompression using 4:2:2 chroma format)
- Error concealment including transport error support
- 3:2 pull-down support
- VBI output support
  - Teletext mode
  - VBI data mode
- Picture-in-picture support
- Built-in phase-locked-loop to provide the required audio clocks and sampling frequencies for simplified audio/video synchronization
- Common 16-bit serial audio output interface to support most D/A converters
- Audio tone generation
- Audio attenuation and de-emphasis controls
- Single clock input (27MHz)
- Plays Audio PCM data



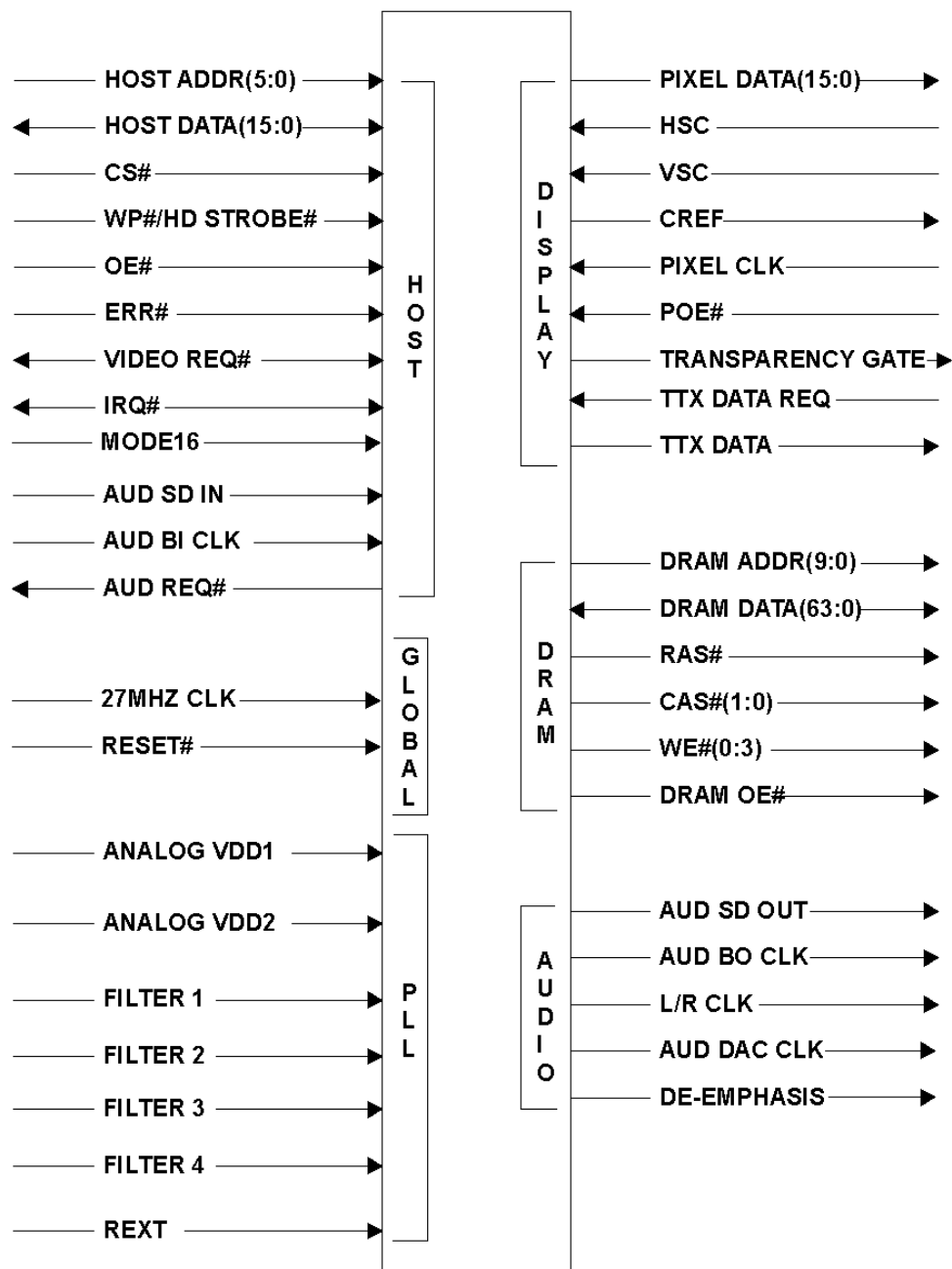
---

## Chapter 3. Signal Descriptions

This section provides a description of the MPEGCD20/MPEGCD21 external signals. The signals are grouped as follows:

- Host Interface Signals
- Global Control Signals
- DRAM Interface Signals
- Display Interface Signals
- Audio Interface Signals
- PLL Interface Signals

## Signal Descriptions



Note: # indicates active low

Figure 3-1. Interface Signals

## 3.1 Host Interface Signals

### HOST ADDR (5:0) - Host Address Bus

**Input**

The Host Address bus is used to select any one of the internal registers or storage units in the decoder. The bits are numbered from 5 to 0, and bit 5 is the most significant bit. In 8-bit mode, the LOW BYTE signal is used as the least significant host address bit.

### HOST DATA (15:0) - Host Data Bus

**Bidirectional**

The Host Data Bus is used for the data associated with host control operations, and can also be used for the delivery of compressed audio or video data. In 16-bit mode, both bytes of this bus, HOST DATA (15:8) and HOST DATA (7:0), are used for data transfers. In 8-bit mode, HOST DATA (15:8) is used for data transfers, and HOST DATA (7:0) is redefined as various control signals. The bits are numbered from 15 to 0, and bit 15 is the most significant bit.

### MODE16 - 16-bit/8-bit Mode Select:

**Input**

The MODE16 input specifies whether the host interface will operate in 16-bit mode or 8-bit mode. Mode16 should be tied to Vdd for 16-bit mode, or tied to Ground for 8-bit mode.

**Note:** When the Host Interface is set to 8-bit mode, the **HOST DATA (7:0)** signals are redefined as follows:

#### HOST DATA (7) -> SER DATA - Video Serial Data In

**Input**

In 8-bit mode, and when the decoder is configured to the video serial mode (CHIP\_MODE bit 13 = '1'), SER DATA is the serial data input for video compressed data. The input sequence of the serial data is the high order bit first.

#### HOST DATA (6) -> SD CLK - Video Serial Data Clock

**Input**

In 8-bit mode, and when the decoder is configured to the video serial mode (CHIP\_MODE bit 13 = '1'), SD CLK is the serial data clock input for video compressed data. This signal must be asserted or de-asserted at the byte boundary. The polarity of this signal is programmable.

#### HOST DATA (4) -> CD STROBE# - Video Compressed Data Strobe

**Input**

In 8-bit mode, CD STROBE# is used to load the compressed video data into the FIFO (parallel 8-bit input). The polarity of the strobe signal is programmable. If video serial mode is being used (CHIP\_MODE bit 13 = '1'), this signal can be programmed to be the byte alignment signal which will signal the low order bit of a data byte.

#### HOST DATA (3) -> RD/WR# - Read/Write

**Input**

In 8-bit mode, RD/WR# determines whether a host access is a read or a write operation. When this signal is high, it indicates a read operation. When this signal is low, it indicates a write operation.

#### HOST DATA (2) -> LOW BYTE - Low Byte

**Input**

In 8-bit mode, LOW BYTE is used as the least significant host address bit. The internal register format is always 16-bit.

## Signal Descriptions

### HOST DATA (1) -> ACK MODE - Acknowledge Mode

Input

In 8-bit mode, this signal selects between 8-bit SRAM Mode and 8-bit Acknowledge Mode. When this signal is a '0', 8-bit SRAM Mode is used. When it is a '1', 8-bit Acknowledge Mode is used.

### CS# - Chip Select

Input

The Chip Select signal indicates that the decoder chip is being selected for register access. The specific register to be selected is determined by the Host Address Bus.

### WP# - Write Pulse

Input

WP# is used to write data to the decoder registers in SRAM mode.

**Note:** In 8-bit Acknowledge mode **WP#** is redefined as follows:

### WP# -> HD STROBE# - Host Data Strobe

Input

In 8-bit Acknowledge mode, HD STROBE# is used for read and write operations to host registers; RD/WR# determines whether it is a read operation or a write operation, and HOST ADDR(5:0) and LOW BYTE specifies the host register. HD STROBE# must be held inactive when CD STROBE# is used to write video compressed data into the FIFO.

### OE# - Output Enable

Input

In SRAM mode, OE# provides the tri-state control for the Host Data Bus. When OE# is asserted, the decoder can drive the Host Data Bus for read operations. In Acknowledge Mode, this signal is not used and must be tied to Vdd.

### ERR# - Transport Error

Input

ERR# indicates a data error during a compressed data write operation. The decoder will not allow data to be written into the compressed data FIFO while ERR# is asserted. The decoder assumes valid compressed data when ERR# is inactive. The polarity of this signal is programmable.

### VIDEO REQ# - Video Request

Output

VIDEO REQ# is asserted by the decoder when there is room in the compressed data FIFO to receive more data. When the FIFO fills to the programmed threshold, VIDEO REQ is de-asserted. The polarity of this signal is programmable.

### AUD REQ# - Audio Request

Output

This signal requests for audio data when it is low. The polarity of this signal is programmable.

### AUD SD IN - Audio Serial Data In

Input

This is the serial input for the audio data.

**AUD BI CLK - Audio Serial Data Clock****Input**

When audio compressed data is written serially into the decoder, AUD BI CLK is used to latch the data on AUD SD IN on the rising edge. This signal must be asserted or de-asserted at the byte boundary. The polarity of this signal is programmable.

**Note:** When the HOST DATA bus is used to write audio compressed data to the decoder, the **AUD BI CLK** signal is redefined as follows:

**AUD BI CLK -> AUD STROBE# - Audio Compressed Data Strobe    Input**

AUD STROBE# is used to latch the audio compressed data from the HOST DATA bus on the rising edge. The polarity of this signal is programmable.

**IRQ# - Interrupt Request****Output**

IRQ# is provided as a means for the decoder chip to signal the host processor that attention is required. IRQ# is active low and stays low until the interrupt register is read by the host. This signal pin requires a pull-up resistor of 2K Ohms.

## 3.2 Global Control Signals

**27MHZ CLK - 27MHz Clock****Input**

The 27MHZ CLK input is the only required clock signal for the MPEGCD20 decoder. It is used to control the System Time Clock (STC) register, and to feed two on-chip Phased-Locked-Loops (PLLs) that generate the internal system clock, and all of the required audio sampling frequencies. It is highly recommended that this same clock signal be used for the PIXEL CLK input to the Video Display Interface to maintain smooth audio/video playback and precise synchronization.

**RESET# - Reset****Input**

The Reset signal is the Power-On Reset (POR) signal. This signal resets all the registers on the decoder chip. Instruction Storage must be loaded (or reloaded) after this POR is performed. This reset does not include performing the initialization sequence for the internal processor or execution unit. The host processor must perform the initialization sequence after the reset has been completed.

## 3.3 DRAM Interface Signals

**DRAM ADDR(9:0) - Memory Address****Output**

The Memory Address pins connect directly to the address pins of the memory modules. The 256K x 16 DRAM modules support two types of addressing schemes, which distribute the row and column addresses:

- Symmetrically, with nine bits for Row Address Strobe (RAS) and nine bits for Column Address Strobe (CAS).
- Asymmetrically, with ten bits for RAS and eight bits for CAS.

## Signal Descriptions

The IBM MPEG-2 decoder chip DRAM address interface supports either configuration by repeating bit 9 of the RAS address on the CAS address range. Both types (symmetric and asymmetric) can be supported with the same board layout design. Route the board using the asymmetric pin layout, connecting lines A9 through A0 of the decoder to pins A9 through A0 of the asymmetric DRAM. On a symmetric DRAM, the A9 pin position is a No Connect, so the same layout supports both types. These signals are positive active. The address bits are numbered from 9 to 0, and bit 9 is the most significant.

### **DRAM DATA(63:0) - Memory Data**

### **Bidirectional**

Sixty four (64) data bits are used in the 2.0MB and 4.0MB configurations to support the required bandwidth. This enables a DRAM configuration to support various resolutions. The configurations and wiring are shown in the preceding "DRAM Configuration" section. These signals are positive active. The data bits are numbered from 63 to 0 with the most significant bit being bit 63.

**Note:** In the 1MB DRAM configuration, the memory data bus is thirty two (32) bits numbered from 31 to 0 with the most significant bit being bit 31. The decoder chip is wired so that these bits are the high order bits of the 64-bit data bus (MD 63 - 32).

### **RAS# - Row Address Select**

### **Output**

The Row Address Select signal is tied directly to the RAS pin of the DRAM chips. The RAS signal selects the row addresses in the DRAM array. This signal is negative active.

### **CAS#(0) - Column Address Select 0**

### **Output**

This signal is used to differentiate between Banks 0 and 1 of the DRAM configuration. It is tied to every CAS pin of each DRAM module in Bank 0 and is used in all DRAM configurations. This signal is negative active.

### **CAS#(1) - Column Address Select 1**

### **Output**

This signal is used to differentiate between Banks 0 and 1 of the DRAM configuration. It is tied to every CAS pin of each DRAM module in Bank 1 and is used only in the 4.0MB DRAM configuration. This signal is negative active.

### **WE#(0:3) - Write Enable**

### **Output**

The Write Enable signals select any one of the possible DRAM modules for writing the 16 bits associated with that DRAM. This allows a minimum of 16-bit writes for partial data updates, and also allows all of the available 64 bits to be written simultaneously. Bit 0 controls the high order 16 bits. Bit 3 controls the low order 16 bits. Each of the Write Enable signals can drive up to four typical loads. This supports DRAM modules with two byte write lines, both of which are tied together to a single WE line. The decoder chip uses the 'Early Write' mode of the DRAM. These signals are negative active.

### **DRAM OE# - DRAM Output Enable**

### **Output**

The DRAM Output Enable signal enables the output buffers of each of the DRAM chips. It is attached directly to the OE pin of each DRAM module. This signal is negative active.



### 3.4 Display Interface Signals

#### PIXEL DATA(15:0) - Pixel Data Out

Output

YCbCr (4:2:2) format - Output pixel data bus byte that contains both the luminance (Y) and the chrominance (Cb and Cr) components for the display. The bits are numbered from 15 to 0 with bit 15 being the most significant bit. When the pixel data bus is programmed to an 8-bit bus, the high order 8 bits, bits 15:8, are used.

#### HSC - Horizontal Sync Control

Input

This signal is programmed by the sync-mode field of the Display Mode register. It can be the Composite Blanking, H-ref or H-sync signal. The polarity of this signal is programmable.

#### VSC - Vertical Sync Control

Input

This signal is programmed by the sync-mode field of the Display Mode register. It can be the Field ID, V-ref or V-sync signal. The polarity of this signal is programmable.

#### CREF - Clock Reference

Output

This signal may be used to qualify which edge of the pixel clock to latch up data when operating in 16-bit pixel mode. In 8-bit mode, the signal may be used to indicate chrominance vs. luminance phases on the bus.

#### PIXEL CLK - Pixel Clock

Input

This clock is used to control the pixel data transfer. Normally this pin is connected to the same 27 MHz clock connected to the decoder.

#### POE# - Pixel Output Enable

Input

The Pixel Output Enable is generated from the display controller and is used by the MPEGCD20 to control the drivers for the output. This signal is negative active.

#### TRANSPARENCY GATE - Transparency Gate

Output

This output is controlled by the On Screen Display bitmap. It can be used to control the muxing with another pixel bus. The polarity of this signal is programmable.

#### TTX DATA REQ - Teletext Data Request

Input

This signal indicates that the digital encoder is ready to receive teletext data.

#### TTX DATA - Teletext Data

Output

This is the teletext data stream output to the digital encoder.

### 3.5 Audio Interface Signals

#### AUD SD OUT - Audio Serial Data Out

Output

This is the serial output for the audio data.

## Signal Descriptions

### **AUD BO CLK - Audio Bit Out Clock**

**Output**

This clock is used by the D-to-A converter to latch the audio data. The frequency is 32 times the sampling rate.

### **L/R CLK - Left/Right Channel Clock**

**Output**

This clock is used to indicate the left or right channel audio data. The frequency is the sample rate.

### **AUD DAC CLK - Audio Over-Sampling Clock**

**Output**

This clock is used by some D-to-A converters. The frequency is 256 times the sampling rate.

### **DE-EMPHASIS - De-Emphasis**

**Output**

This signal is used to control the external audio emphasis circuit. It is controlled by the emphasis bits in the audio bit stream when bit 4 of the Aud\_Ctl register is set to 1.

## **3.6 Phase Lock Loop Signals**

### **ANALOG VDD1 - Analog Voltage**

**Input**

Analog voltage for the PLL circuitry.

### **ANALOG VDD2 - Analog Voltage**

**Input**

Analog voltage for the PLL circuitry.

### **FILTER 1 - PLL Filter**

**Input**

External filter required by internal PLL to ensure proper lockup to supplied clock reference.

### **FILTER 2 - PLL Filter**

**Input**

External filter required by internal PLL to ensure proper lockup to supplied clock reference.

### **FILTER 3 - PLL Filter**

**Input**

External filter required by internal PLL to ensure proper lockup to supplied clock reference.

### **FILTER 4 - PLL Filter**

**Input**

External filter required by internal PLL to ensure proper lockup to supplied clock reference.

### **REXT - PLL External Resistor**

**Input**

External 4.99K resistor is used by the PLL circuitry.

## Chapter 4. Functional Description

### 4.1 MPEGCD20/MPEGCD21 Block Diagram

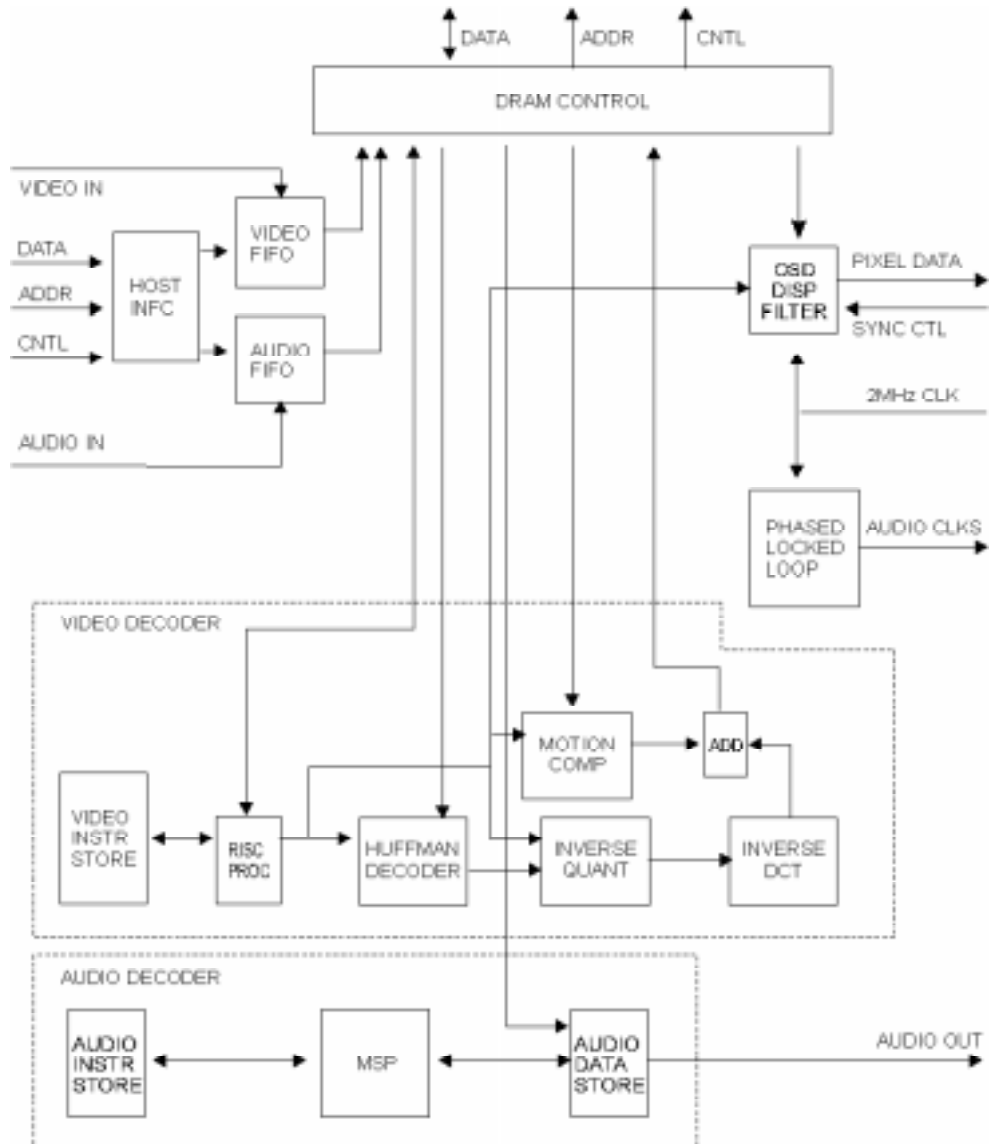


Figure 4-1. Functional Block Diagram

### 4.2 Host Interface

This interface functions as the primary control and compressed data input interface. This is a 16-bit interface that provides host processor access to the internal registers, local storage, and functional units. High level system commands and status are passed between the decoder and the host processor through this interface. The host interface is also used to load the microcode for the on-chip internal processor and to supply the compressed audio and video data. The interface can be set to operate in either 8 or 16 bits SRAM mode or 8-bit Acknowledge mode.

- 16/8-bit SRAM mode

In the 16-bit mode, the 16-bit data is controlled by the write pulse and the output enable signals to perform the write and read operations. The chip select is used to select the decoder. In the 8-bit mode, the low order byte of the 2-byte data bus is used for the control signals. The operation is the same as the 16-bit SRAM mode except that the Low Byte signal is used to indicate which byte is being accessed. In a write operation, the high order byte must be written first, immediately followed by the low order byte. Because the internal registers are 16 bits wide, the data does not get written until the low byte is received. If a write to another register occurs between the two writes, the data will be corrupted. This restriction does not apply to the 8-bit compressed data write. The Low Byte signal is ignored and the data is immediately written to the compressed data FIFO. The read operation can read the two bytes in any order.

- 8-Bit Acknowledge mode

This mode requires the same restriction for the read/write operations as described in the 8-bit SRAM mode. A few more signals are needed for this mode. The Read/Write signal specifies either a read or a write operation. A Wait signal is generated as a response to the Write Pulse signal for control purposes. The Output Enable signal is not required.

- Parallel Compressed data inputs

The audio and video compressed data can be inputted through the same 8-bit host data port. Separate audio and video strobe signals are provided to strobe the compressed data to the decoder. The host or transport must ensure that only one type of data transfer can occur at a time. This input mode can be used in either the 8-bit SRAM or the 8-bit Acknowledge mode. The Low Byte signal is not used for the compressed data write operation. The compressed data input can interleave between the host high byte and low byte writes.

- Serial Compressed data inputs

The audio and video compressed data can be inputted through the serial ports simultaneously with the host data in either the 8-bit SRAM or 8-bit Acknowledge mode. For the video, there is an option to use the external byte sync signal to do the byte alignment. The CD Strobe must be active when the eighth bit of the serial data is sent. When the internal byte sync option is chosen, the internal counter must be reset to ensure the correct alignment. These modes are selected by setting the Internal Byte Sync Mode bit of the Chip Mode register. For the audio, the serial data is always aligned by the internal counter. The internal counter can be reset by issuing a Reset Rate Buffer command or through a chip reset.

## 4.2.1 Interface Control Signals Settings

Table 4-1. Interface control signal settings

Signal Name	16-Bit SRAM Mode	8-Bit SRAM Mode	8-Bit Ack Mode
CS#	0	0	0
WP#/HD STROBE#	0/1	0/1	0/1
OE#	0/1	0/1	
AUD STROBE# (AUD BI CLK)	1	0/1	0/1
SER DATA (DATA(7))	data	data	data
SD CLK (DATA(6))	data	0/1	0/1
DTACK MODE (DATA(5))	data	0	0/1
CD STROBE# (DATA(4))	data	0/1	0/1
RD/WR# (DATA(3))	data	0	0/1
LOW BYTE (DATA(2))	data	0/1	0/1
RD/WR# (DATA(3))	data	0	1
WAIT# (DATA(0))	data	X	output

### 4.3 DRAM Controller

The MPEGCD20 decoder requires external memory (DRAM) to hold compressed audio and video data as well as the decompressed images during the decompression sequence. The decoder is designed to support only the 256Kx16 DRAM modules in DRAM configurations of 1MB, 2MB, and 4MB.

The user of the decoder chip can select the desired DRAM configuration based on the required operational environment. This selection is typically a trade-off between cost and flexibility. The MPEGCD20 decoder chip is designed to provide the required support for a variety of applications using this variable memory configuration capability.

#### 4.3.1 DRAM Requirements

The required DRAM size to support the decoder can be selected from the following table:

**Table 4-2. DRAM Requirements**

DRAM Size	Format	NTSC	PAL
1MB	4:2:0	up to 352x480	up to 352x576
2MB	4:2:0	up to 720x480	up to 720x576
4MB	4:2:2	up to 720x512	up to 720x608

#### 4.3.2 Memory Allocations

**Table 4-3. Example of Memory Allocations**

Function	NTSC-1MB	NTSC-2MB	PAL-1MB	PAL-2MB
User/Parameter data	512	512	512	512
Audio	16384	16384	16384	16384
Frame Buffers	777600	1555200	836352	1710720
OSD Buffer	n/a	170056	n/a	10464
VBI Buffer(1)	n/a	0	n/a	0
Compressed Video Buffer(2)	254080	355000	195328	380000
Total Bytes	1048576	2097152	1048576	2097152

**Note:** (1) VBI buffers can be calculated by 2 times number of lines x 1440 bytes. Buffer calculated with 0 bytes for VBI. (2) Compressed video buffer calculation for 2MB is based on VBV size +  $2R/P$ , where R is the maximum bit rate and P is the frame rate.

### 4.3.3 DRAM Configuration/Usage

The 1MB DRAM configuration provides a 4 byte wide data bus (32 bits) to decode SIF and half-D1 sizes of pictures.

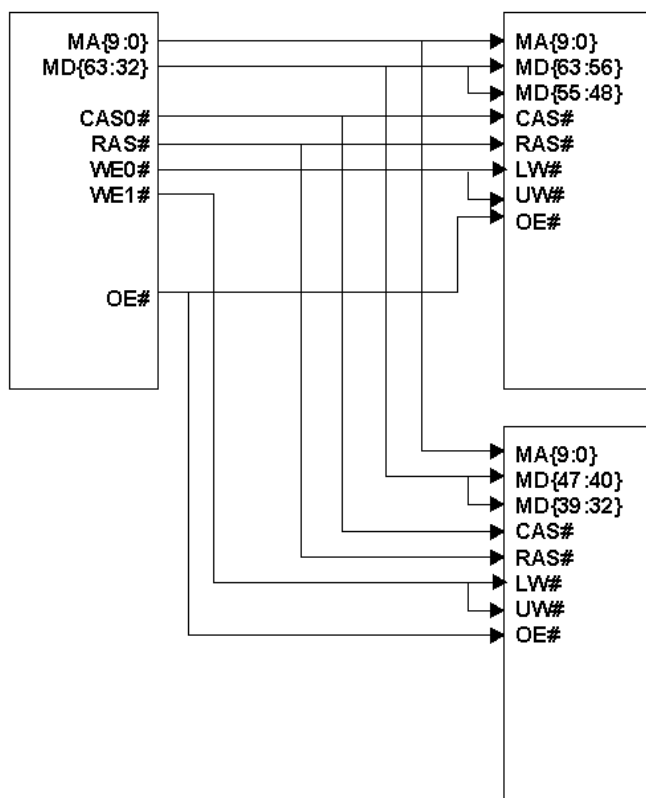
The data interface between the decoder and the 2MB or 4MB DRAM configurations is 64 bits. This interface will drive up to two (2) banks of DRAM modules. The timings for the DRAM are directly generated from the decoder chip system clock. The memory configurations shown in the figures are wired using 256Kx16 DRAM modules. As shown, the WRITE ENABLE lines (WE0,1,2,3) write to both the upper and lower write connections (UW and LW) of the DRAM modules. The DRAM interface supports 70ns or faster DRAM. The 4:2:2 Profile requires 60ns DRAM. 1Mx16 DRAM can also be used, however, the two high order address bits must be tied down.

In configurations supporting B-frames, the DRAM configuration will hold up to three (3) decompressed frames of information. These three frames consist of two reference frames and one B-frame that is displayed as it is being decompressed.

In the case where no B-frames are supported, the DRAM configuration is required to hold two frames of data. One of these is the frame that is currently being decompressed and the other is the single reference frame. In each of these cases, the reference and display frames may be overlapped.

In addition to storing the decompressed frames, the DRAM also serves as a temporary holding place for the incoming compressed audio and video stream. The amount of storage required to hold the compressed data stream is a function of the compressed data rates since higher incoming data rates require more storage. The amount of storage available for this temporary storage of compressed video is configuration dependent

### 1MB External DRAM Configuration



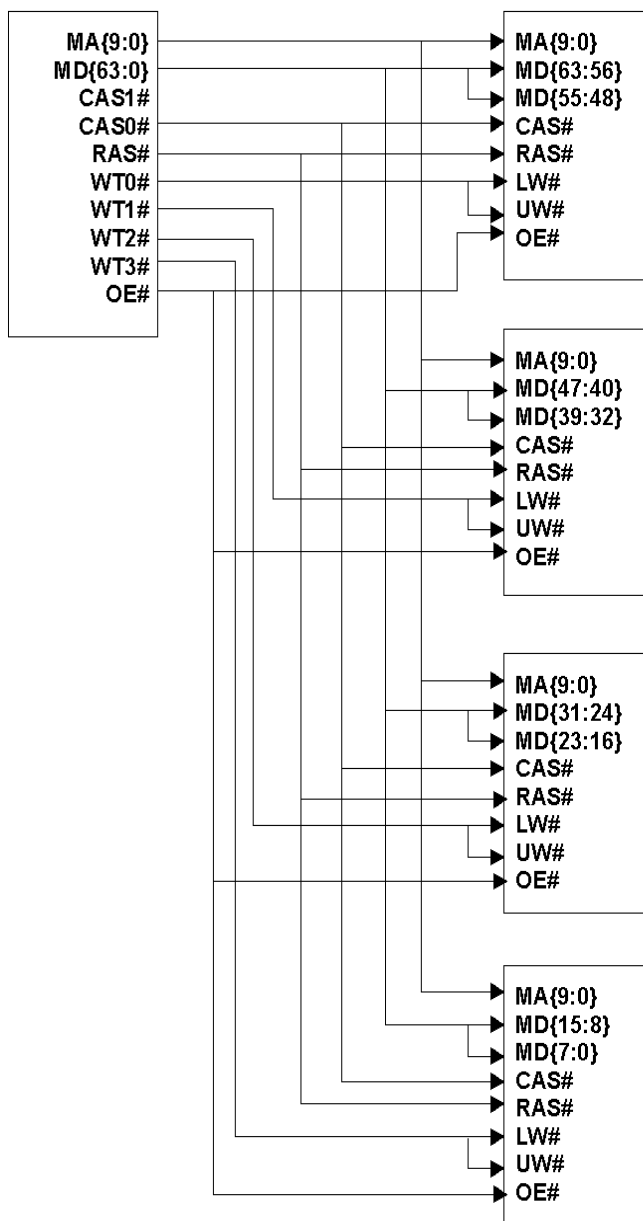
NOTE 1: A negative active signal is indicated by a '#' symbol.

NOTE 2: For DRAM using high and low CAS write enables, the CAS lines are tied together.

Figure 4-2. 1MB External DRAM Configuration



### 2MB External DRAM Configuration

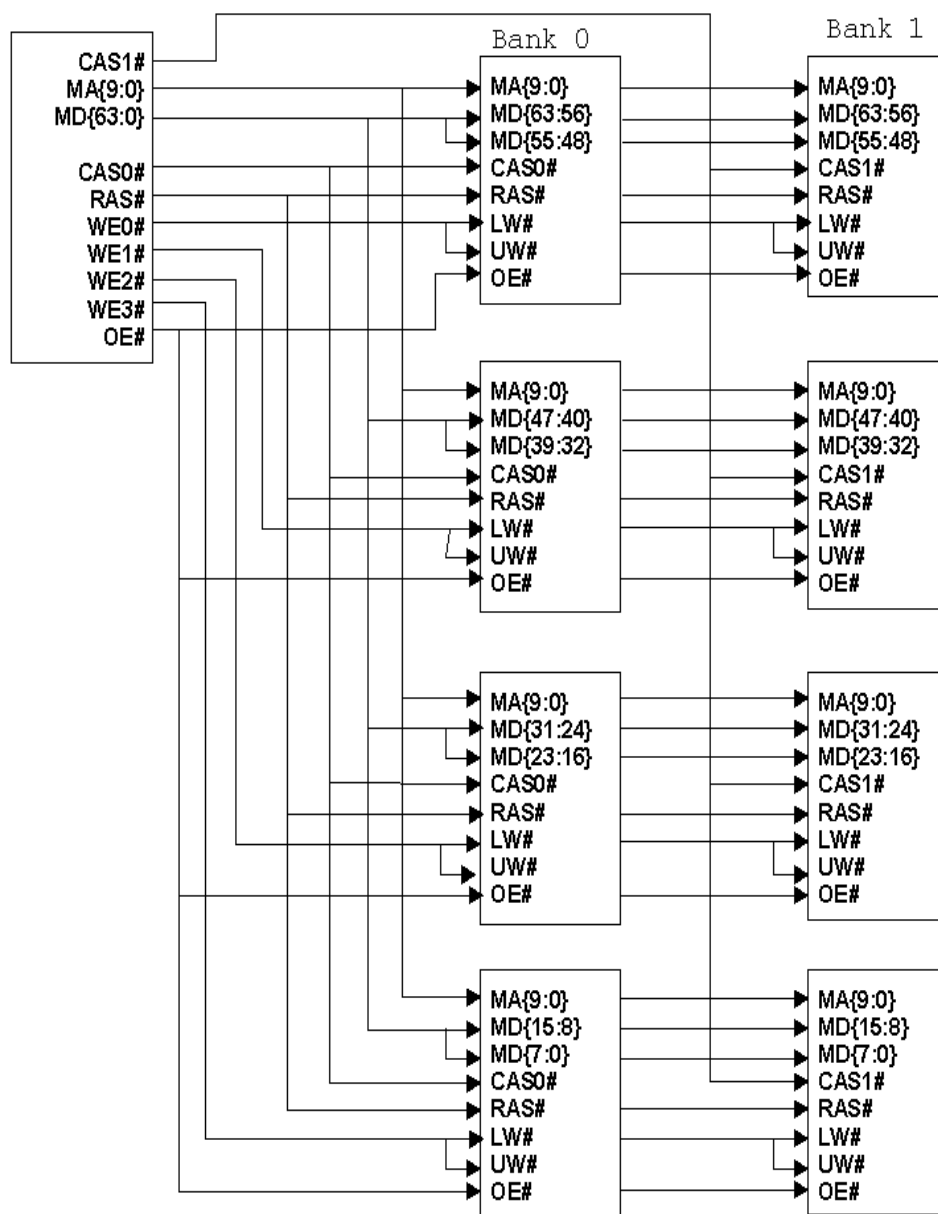


NOTE 1: A negative active signal is indicated by a '#' symbol.

NOTE 2: For DRAM using high and low CAS write enables, the CAS lines are tied together.

Figure 4-3. 2MB External DRAM Configuration

## 4MB External DRAM Configuration



NOTE 1: A negative active signal is indicated by a '#' symbol.

NOTE 2: For DRAM using high and low CAS write enables, the CAS lines are tied together.

**Figure 4-4. 4MB External DRAM Configuration**

## 4.4 Video Decoder

The video decoder includes the following functional units:

- Internal RISC processor/controller
- Huffman decoder (variable length code decoder(VLCD))
- Inverse quantizer (IQ)
- Inverse discrete cosine transform (IDCT)
- Motion compensation (MC)
- Video rate buffer

### 4.4.1 Internal processor (controller)

The internal processor is the central point of control for the video decoder. The processor microcode is contained in a local Instruction Store. The processor interacts with the host processor through the Host interface for high level commands and status. The internal processor is responsible for the control/command and the global synchronization of the other video decoder functional units.

The internal processor is closely coupled to the VLCD (variable length code decoder) unit. Parsing of the code stream and processing of all the header information is done by this processor interacting with the VLCD.

### 4.4.2 Huffman decoder (Variable Length Code Decoder) (VLCD)

The VLCD unit contains tables for decoding the VLC symbols from the coded data stream and a local state machine that controls the decoding of run/level data for macroblocks. The VLCD is controlled by the internal processor while header information is being decoded. After the header information for a macroblock is processed, the local state machine decodes the run/level symbols and interacts with the inverse quantizer to process the decoded symbols.

Variable Length Coding (Huffman coding) is a statistical coding technique that assigns codewords to symbols. Symbols with high probability are assigned shorter code words and symbols with low probability are assigned longer code words. The codes form a class of codes known as prefix codes. No valid code is a prefix of another code. The number of bits assigned to each codeword is variable, from a minimum of 1 to a maximum of 16. The coded bits used to represent a sequence of symbols are a variable length string of bits. This bit string must be decoded sequentially in symbol order to reconstruct the original sequence of symbols. This coding procedure is "lossless coding" since the exact sequence of symbols encoded is recovered by the decoding process.

### 4.4.3 Inverse Quantizer

The inverse quantizer receives run/length coded symbols from the VLCD unit and outputs a block of 64 coefficients that are sent to the IDCT unit. The inverse quantizer:

## Functional Description

- Converts the run/length coded symbols to zeroes and symbols
- Unzigs the data
- Handles Differential Pulse Code Modulation (DPCM) decoding for the DC coefficients
- Dequantizes the data

The first figure shows the zigzag scanning order of reading the DCT coefficients. The top left coefficient is the DC coefficient. All others are considered as AC terms. The numbers indicate the order in which coefficients are read for run length coding. The second figure shows an alternate scan order that can also be used.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

**Figure 4-5. Zigzag Scanning Order of DCT Coefficients**

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

**Figure 4-6. Alternate Scanning Order of DCT Coefficients**

The reason for using one of these scan orders, rather than a raster scan, is that it increases the coding efficiency. The process after run length coding is to reorder the data. This means placing the "level" data in the correct positions relative to an 8x8 block. The "runs" identify the number of skipped positions (zeroes) in this 8x8 array. The "levels" are also modified using a process called quantization. This quantization step introduces some degree of "loss" into the coded data. The level is divided by a number called the quantization factor to become a quantized coefficient.

In the decoding process, the unzipped quantized coefficients (levels) are multiplied by the quantization factor to produce dequantized coefficients. These coefficients (frequency domain) are sent to the IDCT to be transformed back into time domain based signals. The output of the inverse quantizer are the 8x8 blocks of frequency domain dequantized coefficients that are sent to the inverse DCT (IDCT).

Intra blocks contain DC coefficients that are DPCM coded. This means that the DC value of the previous value is used as a predictor for the current 8x8 block DC value. The quantized difference is then coded and transmitted. Note that the DPCM coding for the DC coefficient is applicable for all three components (Y, Cb, and Cr). After DPCM decoding, the DC coefficients (Y, Cb, Cr) of the 8x8 block will go through an inverse quantization process before being sent to the IDCT unit.

## **4.4.4 IDCT**

After a block of symbols has been dequantized, the IDCT unit performs the 2D IDCT on the 8x8 block to form a reconstructed image block. The reconstructed block is added to the predicted block from the motion compensation unit, if required, and then stored in external memory for display and reference. The IDCT and motion compensation units synchronize their outputs through this summation step. The summation result is the output to external memory. The summation step also "clips" the output values to stay within a valid range.

### **4.4.4.1 Motion Compensation Unit**

The Motion Compensation Unit (MCU) receives commands and addresses from the internal processor and performs the required interpolations to form predicted image blocks. The MCU supports all MPEG-1 and MPEG-2 motion compensation modes. The predicted output block is synchronized to the output of the IDCT unit and added to the IDCT reconstructed block by the summation step.

MPEG-2 motion compensation includes: (1) prediction from past, future, or both types of pictures; (2) motion vectors of full pel or half pel accuracy; and (3) capability of handling both interlaced and non-interlaced video data streams. The MPEGCD20 decoder chip MCU supports both field and frame motion prediction as well as the special motion prediction modes for 16x8 and dual prime motion compensation.

## **4.4.5 Video Rate Buffer**

Contained within MPEGCD20's DRAM is the rate buffer. This buffer is used for compressed data storage. The rate buffer is filled either in a constant bit rate or in response to the data request lines generated by the MPEGCD20. The constant bit rate filling is typical in a live feed

## Functional Description

environment. Using the data request lines is typical for a PC environment. This buffer that contains the compressed data must never overrun or underrun. If an overrun condition happens, compressed data will be lost. Unpredictable results will happen due to the unpredictability of the exact data that is lost. If an underrun conditions happens, then the decode process will be suspended until more data enters the rate buffer.

The rate at which data is removed from the rate buffer is one picture's worth of data every frame time. The video decoder processes one picture every frame time. The data delivery path to the decoder must insure that one picture's worth of data arrives every frame time. If this is not guaranteed, then an underrun condition may develop.

## 4.5 On-Screen Display (OSD)

The OSD function displays a frame structured bitmap on top of a decoded video image. The size of a bitmap region can be from 8x2 to 720x480 NTSC or 8x2 to 720x576 PAL and is not affected by the video bitstream parameters. The OSD bitmap data contains a header which specifies the coordinates of the region, size of the region, address of the next region, color resolutions, color table and other controls. The OSD function has the following features:

- Multiple rectangular regions of bitmaps linked by addresses
- Each line pair can be a region and up to one region per horizontal line pair
- 16 or 4 colors per pixel or pixel pair by region
- Each region has its own color table
- 16 levels of blending between video and OSD
- 16 levels of shading of video
- Easy removal of OSD region from screen
- Animation Support

### 4.5.1 OSD Region

The OSD region is defined as a rectangular area on the screen. The screen can have one or up to 288 regions in PAL, one region per line pair. Only one region is allowed on a line pair. The OSD data of a region contains the header and the bitmap. Each region has its color look-up-table defined in the header and the bitmap data. The starting address of a region is stored in the header of the previous region. The topmost region on the screen has the starting address stored in the OSD LINK ADDR register. These link addresses can be easily modified to display different data in the DRAM.

Do not permit the last region to exceed the size of the screen. If the right side of the last region must exceed the right screen boundary, ensure that the region also extends beyond the bottom of the screen, or place a small transparent region below the last region. The screen coordinates for this transparent region must terminate well within the right side of the screen.

## 4.5.2 Colors and Resolutions

- 2 bits per pixel pair
- 2 bits per pixel
- 4 bits per pixel pair
- 4 bits per pixel

The number of colors and pixel resolutions are selectable on per region basis. Each region has its own color table and the number of colors can be changed on a region by region basis. The color table header may have four colors or sixteen colors. The corresponding bitmap is expected to adjust accordingly.

## 4.5.3 Blending

The blending feature can be enabled by setting the Blend Enable bit in each color entry. There are two blending modes. One has 16 levels but a fixed level across the whole region. The other has 4 levels with the ability to change the level color by color. The blending mode is set in the OSD Mode register. The level controls the amount of luma and chroma used in mixing the OSD and the video.

When the OSD operates with single pixel resolution, OSD/Video color blending can be used throughout a region, but it should be avoided at the video-OSD horizontal boundary. Use an unblended OSD color or 100% video transparency at that boundary. This restriction does not apply if pixel pair OSD resolution is used.

## 4.5.4 Shading

The shading feature is enabled the same way as the blending feature. When the Blend Enable bit is set to 1 on the transparent color,  $Y=Cb=Cr=0$ , the shading level is used to reduce the luminance of the video. The level is multiplied by 8 and subtracted from the luminance of the video. The resulted image gives a defined area for the OSD but still shows the video behind it.

## 4.5.5 Animation Support

The animation is achieved by linking different copies of regions at the synchronized timing. At the end of the screen display time, the OSD Data interrupt is raised to notify the host to modify the link addresses. The interrupt is controlled by the Animation Mode bit and the Animation Rate in the OSD Mode register. While the OSD is using one copy of the region, the other copy of region can be updated by the host.

## 4.5.6 Header Definition

The header contains 8 bytes of control information, followed by 8 bytes or 32 bytes of color table depending on 4-color or 16-color region.

# Functional Description

## 4.5.6.1 Control Fields

The following table defines the header control fields.

Table 4-4. Control Fields definition

Field Description	Bit #	Length
Color Table Update	63	1 bit
Region H-Size	62:56	7 bits
Shade Level	55:52	4 bits
Reserved	51:50	2 bits
Start Row	49:41	9 bits
Start Column	40:32	9 bits
Link Address	31:16	16 bits
Color Resolution	15	1 bit
Region V-Size	14:06	9 bits
Pixel Resolution	05	1 bit
Blend Level	04:01	4 bits
Force Transparency	00	1 bit

The following is a description of the Control Fields:

<b>Color Table Update</b>	When this bit is 1, 256 or 64 bits will be appended to the end of the control header, and will contain the descriptions for 16 or 4 colors respectively, depending on the Color Resolution setting. These colors will be in effect until the next Color Table Update is encountered. If the Color Table Update bit is zero, the Color Table header is not present.
---------------------------	--



<b>Region H-size</b>	<p>This value is in multiples of 4 bytes that describes the horizontal size of the OSD region. The effective horizontal size of the region is defined as follows:</p> <ul style="list-style-type: none"> <li>4 bits/pixel - multiply by 4</li> <li>4 bits/pixel pair - multiply by 8</li> <li>2 bits/pixel - multiply by 8</li> <li>2 bits/pixel pair - multiply by 16</li> </ul> <p>This field cannot be zero. For the 4 bits/pixel pair and 2 bits/pixel cases, the value cannot be greater than 63. For the 2 bits/pixel pair case, the value cannot be greater than 31.</p>
<b>Shade Level</b>	<p>This value represents the amount of Luminance in units of 8 that is subtracted off from the video of any pixel that corresponds to a transparent OSD bitmapped pixel "color" that has the Blend Enable bit set to 1. This will create a "shading" effect. If the shaded luminance value results in a value of zero or less, the final value is clipped to 1.</p>
<b>Start Row</b>	<p>This is the screen's vertical position where the first line-pair of bitmap data in this OSD region is to be displayed on a frame based screen. Start Row = 0 refers to the topmost screen line when the top field is being displayed, and refers to the 2nd topmost screen line when the bottom field is being displayed.</p>
<b>Start Column</b>	<p>This is the screen's horizontal position where the first pixel pair of bit-map data in this OSD region is to be displayed on the screen. Start Column = 0 refers to the 2 leftmost screen pixels.</p>
<b>Link Address</b>	<p>This address is added to OSD Base Address to find the location of the next OSD region. This value is in units of 32 bytes. The last OSD region to be displayed on the screen will have the link address equal to the link address of the previous OSD region. (The last OSD region points to itself.) If there is only one OSD region, then the link address will be equal to the value in the OSD_LINK_ADDR register.</p>
<b>Color Resolution</b>	<p>0 = 16 colors possible per Region. 1 = 4 colors possible per Region.</p>
<b>Region V-Size</b>	<p>This value specifies the number of line-pairs contained in this OSD region. Note that this is the total number of line-pairs in a frame based OSD region.</p>

## Functional Description

<b>Pixel Resolution</b>	0 = A color per pixel 1 = A color per pixel pair
<b>Blend Level</b>	<p>This value represents the ratio of Blend Level / 16, the amount of video that is allowed to show through the OSD. Resulting pixels are weighted such that</p> $(\text{Video} \times \text{Blend Level} + \text{OSD} \times (16 - \text{Blend Level})) / 16.$ <p>This blending is only processed for selected OSD colors that have the Blend Enable bit set to 1.</p>
<b>Force Transparency</b>	<p>Transparency is forced for all pixels in this OSD region. The Shade Level is assumed to be in effect for all pixels in this OSD region.</p> <p>OSD with video transparency can be enabled after a chip reset, but before video decoding begins. If a black video transparency is required during this interval, set the display control for the Left Border to 720, then reset it to 0 at the appropriate time to display the decoded video. Video will not be visible until the Left Border is 0.</p>

### 4.5.6.2 Color Table

The color table can have either 4 or 16 entries. Each entry is 2 bytes and is defined in the following table.

**Table 4-5. Color Table entry fields definition**

Field description	Bit #	Length
Y	Bits 15:10	6 bits
Cb	Bits 09:06	4 bits
Cr	Bits 05:02	4 bits
Blend Enable - when Blend mode=0 Blend Factor(1) - when Blend mode=1	Bit 01	1 bit
Fill Enable - when Blend mode=0 Blend Factor(0) - when Blend mode=1	Bit 00	1 bit

The following describe the Color Table Fields in further detail

<b>Y</b>	This 6-bit value is the Y component of the color in multiples of 4. Transparency is represented by Y=0, Cb=0 and Cr=0.
<b>Cb</b>	This 4-bit value is the Cb component of the color in multiples of 16. Transparency is represented by Y=0, Cb=0 and Cr=0.
<b>Cr</b>	This 4-bit value is the Cr component of the color in multiples of 16. Transparency is represented by Y=0, Cb=0 and Cr=0.
<b>Blend Enable</b>	When Blend Mode is 0 and this bit is set to 1 on the transparent "color", the Shade Level is used to reduce the luminance of the video. When Blend Mode is 0 and this bit is set to 1 on the non-transparent color, the Blend Level is used for blending.
<b>Fill Enable</b>	When Blend Mode is 0 and this bit is set to 1, the color is used to fill to the end of the region. In this case, the remaining bitmap is not needed and will be ignored.
<b>Blend Factor(1:0)</b>	<p>When Blend Mode is 1, these two bits control the Blend Factor on this color and the corresponding video pixel.</p> <p>00 = Video pixel x 0% + OSD pixel x 100%</p> <p>01 = Video pixel x 25% + OSD pixel x 75%</p> <p>10 = Video pixel x 50% + OSD pixel x 50%</p> <p>11 = Video pixel x 75% + OSD pixel x 25%</p>

### 4.5.7 OSD Operations

The MPEGCD20 provides several host accessible registers to support the OSD function

#### 4.5.7.1 OSD Control Registers:

<b>OSD_ADDR</b>	This register is used to set the starting address for writing OSD data to the OSD buffer in DRAM. The address has a maximum range of 512k bytes and is on 8-byte boundary.
<b>OSD_DATA</b>	This register is used to write the OSD data to the OSD buffer. The data must be maintained at the rate of no more than 8 bytes per 1.6us.
<b>OSD_MODE</b>	This register is used to set the Blending and Animation mode.
<b>OSD_LINK_ADDR</b>	This register is used to set the starting address of the first region to be displayed on the screen. This address must be on a 32-byte boundary.
<b>OSD_SIZE</b>	This register is used to set the OSD Buffer size which is in units of 32 bytes.
<b>DISP_MODE</b>	Bit 0 of this register, Enable OSD, is used to activate or de-activate the display of the OSD.

#### 4.5.7.2 OSD Data Input

Before any OSD data is input to the decoder, the OSD buffer must be allocated by setting the OSD Size register before the decoder initialization. The OSD size is units of 32 bytes. The OSD buffer is structured in frame format such that the odd line and the even line of a region are stored consecutively. When the decoder is initialized, an OSD base address which is available to the host is generated. If a different buffer size is needed, the decoder must be re-initialized by the Configure command.

Although the OSD address updates automatically for every 8 bytes, it should be set for a new starting address. This address which has a 512k-byte range is added to a base address to form an absolute address. The base address can be read by using the DRAM\_CONTROL register. Once the base address is known, the absolute address can be determined to use the other DRAM access commands to manipulate the OSD data. The Block Copy, Block Fill and Block Transfer commands are available to move the data around. Also the OSD data can be input by using the Block Transfer command. It extends the addressing range to 2MB.

The OSD data input is capable of handling data up to 5 MB/s. The data rate must be maintained at no more than 8 bytes per 1.6us. The data must be in multiples of 8 bytes in order to write into DRAM. Although the OSD address in units of 8 bytes, the link address is expected

to be on 32-byte boundary. The smaller unit in the OSD address is to allow partial bitmap modification.

#### 4.5.7.3 OSD Data

The OSD data in a region is stored sequentially in DRAM. Each region must have an 8-byte control header and an optional color table. The regions cannot overlap or be on the same horizontal line. Although the regions can be placed anywhere in the DRAM, the link address must link the regions in the display order from top to bottom. The first region which is linked by the OSD Link Address register must have a color table. Multiple copies of a region can be placed in DRAM but only one can be linked at a time. A region may be specified with part of the region hanging outside of the screen. In that case, the outside portion will not be displayed.

The header controls the size of a bitmap through the Region H-Size, Region V-Size, Color and Pixel Resolutions. The Region H-Size represents the number of bytes used per line and the Region V-Size represents the number of line pairs. The Color and Pixel Resolutions can be changed region by region to save memory space.

#### 4.5.7.4 OSD Data Management

To enhance the OSD operation, several DRAM commands are included to move the OSD data around. The Block Copy command can copy a block of data from one location of DRAM to another location of DRAM without the host's intervention. The Block Fill command can help to build a simple OSD bitmap easily. The Block Transfer command can be used to input the OSD data in 2MB range to make use of the 2MB DRAM when the video decoding is not active.

#### 4.5.7.5 OSD Chroma Decimation

The OSD pixel output is decimated from 4:4:4 to 4:2:2 in the following pattern.

OSD Bitmap (4:4:4) Y0 Y1 Y2 Y3 ...  
                           U0 U1 U2 U3  
                           V0 V1 V2 V3

OSD Pixel (4:2:2) Y0 Y1 Y2 Y3 ...  
                           U0 U0 U2 U2  
                           V0 V0 V2 V2

### 4.6 Video Display Interface

The display interface bus is designed to interface either through a digital video encoder, or to support a graphics environment that contains a frame buffer. The interface behaves as a slave unit to the digital video encoder, and is synchronized to it by means of the horizontal and vertical synchronization signals. The synchronization signals are programmable to fit various digital encoders. The timings and the polarity can be changed as needed.

The pixel clock to the display interface bus is provided by the external controller, and regulates the rate at which pixel data will be extracted. The output pixel data is represented in either 16 bit or 8 bit YCbCr format.

## Functional Description

Interlaced displays at full CCIR resolution for NTSC and PAL are supported. Also various other display resolutions and formats are supported through a re-sizing unit that performs vertical interpolation and horizontal filtering. In addition, single field progressive scan mode is supported.

For source material that comes from film, frame rate conversion using 3:2 pulldown is supported. The MPEG-2 Pan Scan feature is supported which allows conversion of 16:9 aspect ratio images to conventional 4:3 aspect ratio displays.

### 4.6.1 3:2 Pulldown

The decoder chip supports the automatic conversion of MPEG-1 video sequences that are compressed in 24 frames/second progressive mode to be displayed in a 60 Hz interlaced display environment. This is accomplished through the use of 3:2 pulldown methodology that involves the repetition of every alternate field. For MPEG-2 sequences, the frame header carries the information as to how many times each field is to be displayed. The decoder chip conforms to this standard for MPEG-2 sequences.

### 4.6.2 Single Field Progressive Scan

This mode of display is activated by setting bit 5 of the DISP\_MODE register through the host interface. When this bit is set to '1', the first field of a frame decoded is displayed for both field 1 and field 2 times. This feature is useful when displaying interlaced pictures in freeze frame and single frame modes to eliminate motion between two fields.

### 4.6.3 Image Re-sizing

The MPEGCD20 supports a variety of resolutions of images that are determined from the incoming video stream. The resolutions listed below are supported directly or by using the re-sizing unit which allows the source data to be expanded to 720 x 480 (720 x 576 for PAL). The on-chip microcode automatically controls the expansion modes in the re-sizing unit.

Images can be expanded both vertically and horizontally. Vertical expansion is done by linear interpolation, using a weighted average between two lines. Horizontal expansion uses a digital filter. Vertical interpolation is performed first, if necessary, forming a single set of line data independently for Y, Cb and Cr. This is then followed by horizontal filtering of the line data prior to transmitting the data on the pixel bus.

A multi-tap Finite Impulse Response (FIR) polyphase digital filter provides the horizontal expansion. The digital filter is a hardware unit which employs four multipliers.

One vertical expansion mode exists for 1:2 expansion. This can be used with the horizontal 1:2 expansion mode for displaying SIF pictures to full CCIR resolution. In addition, chroma upsampling from 4:2:0 to 4:2:2 is also performed.

### 4.6.3.1 Resolution/Aspect Ratio

The following table shows the input resolutions to be expanded automatically by the re-sizing unit. The host has the ability to override the expansion by setting the expansion control bits in the DISP\_MODE register.

**Table 4-6. Input Resolutions**

Input Image NTSC (PAL)	Aspect Ratio	Horizontal Expansion	Vertical Expansion
352 x 240 (288)	4:3	1 -> 2	1 -> 2
352 x 480 (576)	4:3	1 -> 2	-
480 x 480 (576)	4:3	2 -> 3	-
544 x 480 (576)	4:3	3 -> 4	-
720 x 480 (576)	16:9	3 -> 4	-
544 x 480 (576)	16:9	9 -> 16	-
480 x 480 (576)	16:9	1 -> 2	-
352 x 240 (288)	16:9	3 -> 8	1 -> 2
352 x 480 (576)	16:9	3 -> 8	-

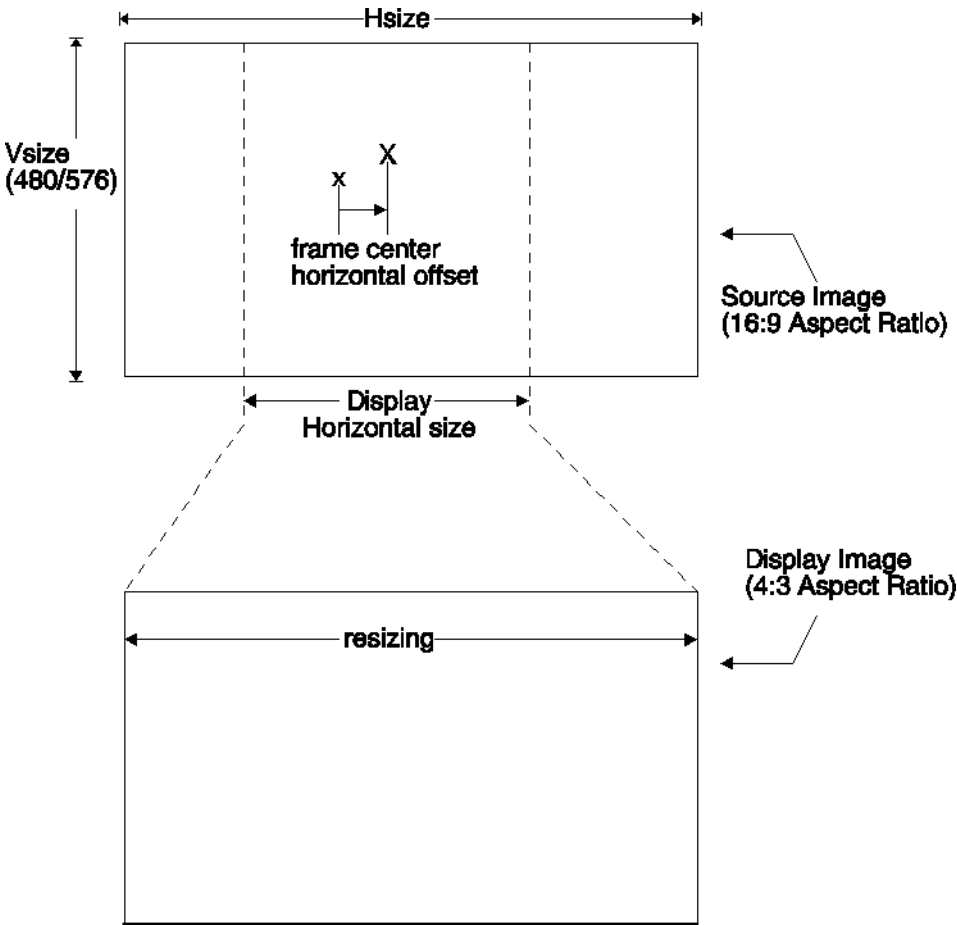
### 4.6.3.2 MPEG-2 Pan/Scan Support

The MPEGCD20 supports MPEG-2 Pan/Scan, enabling the display of 16:9 aspect ratio video sequences on a 4:3 aspect ratio monitor. The original 16 x 9 image must be expanded properly after decompression to avoid distortion. A section of the source decompressed image is resized to full horizontal width. A variety of resolutions are supported with this feature. The section position can be placed to an accuracy of 1/16 of a pel, and can be specified for individual fields.

Information pertaining to the aspect ratio, section size and the section position are contained in the MPEG-2 compressed bitstream. Figure 4-7 shows how the section is positioned within the source image, and is resized for display. Hsize and Vsize represent the horizontal and vertical sizes of the source image, and capital "X" represents the center of the source image. The Section is defined by the display\_horizontal\_size and its center is marked by the lower case "x".

**Functional Description**

The horizontal expansion for the resizing is automatically determined based upon the size of the section.



**Figure 4-7. Display (Image Size) Scaling for Pan Scan**



#### 4.6.4 Picture in Picture Support

This feature allows a system to merge an image produced by the MPEGCD20 chip, to another image or graphic that is produced somewhere else in the system.

The transparency gate signal can be used to control the multiplexing of video data produced by the MPEGCD20 chip and another source. When the Transparency gate signal is active, it indicates that image data is being transmitted, when it is inactive, OSD information is being transmitted. Transparency gate is enabled through the CHIP\_MODE register bit 12 (Enable PIP mode).

The default state of Transparency Gate is low (inactive) following a chip reset. The polarity of the signal may be changed using DISP\_MODE register bit 9.

Transparency is controlled by the On Screen Display function. The bitmap can be coded to control which section of the display area is transparent, or driven from the internal color lookup tables.

#### 4.6.5 Display Interface Controls and Timings

The operation of the display interface is controlled through the following registers:

- (20) DISP\_MODE
- (21) DISP\_DLY
- (22) VBI\_CTL
- (23) DISP\_LBOR
- (24) DISP\_TBOR

Three synchronization modes are allowed. Each mode determines how the MPEGCD20 synchronizes to the VERT\_SYNC\_CNTL and HORZ\_SYNC\_CNTL inputs. Bits 15-13 of the DISP\_MODE register define the Sync Mode. After a chip reset, there is no synchronization established. Synchronization is established when the Sync Mode bits are written with a valid mode. The modes are defined as:

<b>000</b>	Initial State; No Sync established
<b>001</b>	HORZ_SYNC_CNTL = Composite Blank; VERT_SYNC_CNTL = Field ID. The transition of Field id causes the MPEGCD20 to change fields. Composite Blank is held in the blank state during the Vertical Blanking Interval where Field ID changes state.
<b>010</b>	HORZ_SYNC_CNTL = H; VERT_SYNC_CNTL = V. The leading edge of the V signal causes the MPEGCD20 to change fields. H is active during every line interval including the vertical blanking interval.
<b>011</b>	HORZ_SYNC_CNTL = Hsync; VERT_SYNC_CNTL = Vsync. The trailing edge of Vsync causes the MPEGCD20 to change fields. Hsync is active during every line interval including the vertical blanking interval.

## Functional Description

111        `HORZ_SYNC_CNTL = Hsync`; `VERT_SYNC_CNTL = Vsync`. The leading edge of `Vsync` causes the MPEGCD20 to change fields. `Hsync` is active during every line interval including the vertical blanking interval.

**Note:** When setting the Sync Mode bits to 111, the user must wait at least 40 milliseconds before setting the Display Mode to PAL (`DISP_MODE` register, PAL Mode = 1).

When a field change is detected by the MPEGCD20 chip, it begins to count line intervals with an internal line counter. The line counter decrements at the start of an active line. The line counter will process in order, three different values before it loads the image vertical size, these are:

1. Vertical Delay (from `DISP_DLY` register bits 7-0)
2. VBI Count (from `VBI_CTL` register bits 3-0)
3. `DISP_TBOR` register bits 9-0

If any of these values are zero, then they will not be processed, and the line counter will go on to process the next value or the image vertical size. This is illustrated in Figure 6-17.

Examples are shown starting with Figure 6-18 for cases using the four types of sync modes for NTSC and PAL operation. These examples show cases in which the VBI count and `DISP_TBOR` are equal to 0.

At the start of an active line, a similar count system is used. An active line is detected when the `HORZ_SYNC_CNTL` switches to the active state. This triggers the start of an internal pixel counter which will also process two different counts before loading the image horizontal size. These are:

1. Horizontal Delay (from `DISP_DLY` register bits 15-8)
2. `DISP_LBOR` register bits 9-0

This is illustrated in Figure 6-16.

If any of these values are zero, then they will not be processed, and the pixel counter will go on to process the next value or the image horizontal size. Figure 6-28 shows an example of the relationships of when the `HORZ_SYNC_CNTL` changes to when the pixel counter starts and pixel data emerges. If horizontal delay and `DISP_LBOR` are zero, then the horizontal size would be loaded at the start, and valid pixel data will emerge 74ns after the detection of active video. If either the horizontal delay or `DISP_LBOR` is non-zero, then N will be loaded first with the horizontal delay, which will count down to 1, then it will be loaded with `DISP_LBOR`, which will count down to 1 before the horizontal size is loaded.

Following a chip reset and prior to having a picture to display, the MPEGCD20 chip will output a black color. This will apply to both 16 bit and 8 bit modes of operation. A value of `X'80'` will be transmitted on pixel data bits 7-0. On pixel data bits 15-8, the value will alternate between a value of `X'80'` when `CREF` is high and `X'01'` when `CREF` is low until the `DISP_MODE` register is set with the appropriate sync mode, polarity information, and whether 8 or 16 bit mode is chosen. If 16 bit mode is selected, then pixel data bits 15-8 will transmit a value of `X'01'`. If 8 bit mode is selected, then the alternating `X'80'` and `X'01'` pattern will continue.

When a valid image is decoded, and/or an OSD image is loaded and ready to display, then the display will switch from black to the image. After an image is being displayed, a black screen may be forced by setting CHIP\_CONTROL bit 7 (Black Out Display).

#### 4.6.6 Teletext And Vertical Blanking Interval Data Output

The MPEGCD20 supports teletext and data transmission on the pixel data bus during the vertical blanking interval (VBI). This feature is controlled by the VBI\_CTL register and two bits in the DRAM\_CONTROL register. The VBI data is transmitted just ahead of the active video data. The number of VBI lines transmitted is controlled by the VBI Line Count bits in the VBI\_CTL register. The value in the VBI Line Count indicates the number of VBI lines that will precede the active video lines. All the VBI lines can be used to deliver data. If the VBI Line Count is zero, the VBI function is disabled. The VBI data is written by the host into the VBI0 and the VBI1 buffers in the DRAM. The VBI Size must be set into the VBI\_CTL register before the internal processor is started because the microcode uses the values to set the DRAM base addresses. VBI0 and VBI1 are allocated to the same size.

The VBI buffer is structured with alternating chrominance and luminance data. The 720 bytes of chrominance data are followed by 720 bytes of luminance data. Each unit of the VBI Size represents a pair of chrominance and luminance lines.

The VBI buffer selection is controlled by the VBI Pointer bit in the DRAM\_CONTROL register. The double buffering allows the data to be output from one buffer while the other buffer is being loaded. The host must set the VBI Access bit in the DRAM\_CONTROL register before the DRAM\_ADDRLO register can be used as the offset address for writing the VBI data into DRAM.

For display, the selection of the two VBI buffers is accomplished in hardware when the VBI interrupt is signalled. At this time, the hardware will use the opposite buffer to the one pointed to by the VBI Pointer Bit in the DRAM\_CONTROL register. This feature allows the host to write in the VBI data for the first field to be displayed, then switch the VBI Pointer bit, and write in the second buffer to be displayed. By monitoring the VBI interrupt, the host can time when it is appropriate to update VBI data for successive fields.

Note that even if VBI data is to be displayed on only one field and not the other, the VBI Pointer bit must still be switched so that the hardware displays the correct buffer. The pixel data bus is used to output the VBI data. For teletext data, a separate output (TTX DATA) is used to output the data stream.

DRAM space does not need to be allocated to process all the VBI lines that were specified in the VBI Line Count. For example, if only line 18 were to be used for passing VBI data, then the VBI0 and VBI1 only need to be large enough to hold one line of data. The VBI Line Count would still be set to 4 in order to position line 18 at the correct time, assuming line 22 is the first active video line. The output data for lines 19, 20 and 21 would be meaningless.

Use of VBI may require some adjustment to the vertical delay in the DISP\_DLY register. If the HSC input is programmed to use Hsync or H as input, then the vertical delay setting will have to be reduced by VBI Line Count. If HSC input is programmed as composite blanking, then the composite blanking pulses must allow additional lines before the active video, and if VBI is

## Functional Description

turned off (ie. setting the VBI Line Count to zero), then vertical delay equal to the additional lines would have to be added.

For teletext operation, a single line of VBI space (1440 bytes) is used to buffer all the necessary lines of teletext data for one field. VBI0 and VBI1 buffers are accessed via the host in the same manner that was described previously for luminance and chrominance VBI data. The first line of teletext begins at offset 0 in the buffer. Subsequent lines of teletext begin at intervals of 48 bytes in the buffer.

Field 1 and Field 2 are allowed to have different numbers of teletext lines. The first line always starts at offset 0 in the buffer. The lines must be contiguous for each field.

For example, Field 1 may require 16 lines of teletext to be output on video lines 5 through 20, and Field 2 may require 10 lines of teletext on video lines 9 through 18. VBI0 can be used for field 1 and VBI1 can be used for field 2. The first 48 bytes of VBI0 would contain the teletext data for field 1, line 5, and the first 48 bytes of VBI1 would contain the teletext for field 2, line 9.

The MPEGCD20 chip will take the bytes of teletext information and transmit it on the TTX DATA output. The transmitted bits will be output using a protocol that clocks out one bit every three or four periods of the pixel clocks. The protocol transmits every 10th, 19th, 28th and 37th bit using three clock periods and all others using four clock periods. The pattern repeats after each cycle of 37 teletext bits are transmitted. VBI\_CTL register bit 6 (Teletext Bit Reverse) allows for the capability to transmit the bits from least significant to most significant order for each byte. Otherwise the bits will be transmitted from most significant to least significant order.

Teletext will be transmitted following the rise of TTX DATA REQ as shown in Figure 6-28.

## 4.7 Audio Decoder

The MPEGCD20's integrated audio decoder is capable of decoding a MPEG-2 compatible Layer I and Layer II audio stream to produce a two-channel (no downmixing) output. It supports single channel, dual channel, joint stereo and stereo. It supports all bit rates except free format. The supported output sample rates are 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1 kHz and 48 kHz. These sampling frequencies are generated by a built-in phase-locked-loop, controlled by the audio input stream, from the same 27 MHz clock which governs the display output. This will help to maintain the audio and video synchronization once they are automatically adjusted at the start-up after a channel switch.

A separate audio rate buffer shares the same memory with the video decoder to eliminate any extra memory usually needed by an audio decoder. The audio PTS is extracted from the PES input before the compressed data is sent to the rate buffer in DRAM. The PTS is then used to synchronize the audio data.

The audio decoder can play a PCM data file or elementary audio stream if desired. In addition, it has the ability to generate different tones with different durations at the command of the host for user feedback.

### 4.7.1 Audio Processor

The audio processor is the MWave Signal Processor (MSP). The MSP is a 16-bit processor, optimized for high speed logical and mathematical operations.

The rest of the decode process is performed by code running on the MSP. This includes items such as MPEG decode, audio synchronization, and error detection and concealment.

### 4.7.2 Controlling the Audio Decoder

Before starting the decoder, the host must set the AUD\_CTL, INFC\_CTL and the BEEP\_CTL registers to the desired modes. The audio instruction store must be loaded also. After the instructions are loaded, the audio processor is started by setting the Start DSP bit to 1.

After completing Reset processing, the Audio Processor will check to see if the Start Audio Decode bit is set to b'1'. If not, the processor will wait for the bit to be set. In this state, only a small amount of audio input data (about 48 bytes) will be accepted, since the processor will not know whether to scan for a PES header, or to place the data into the Audio FIFO. Tones can be played in this state.

Once the Start Audio Decode bit has been set, the Audio Processor will read the mode bits and determine the format of the incoming data. This mode will not change until the next Audio reset. Any changes to the mode bits, except for Synchronization, will have no effect.

Once the Start Audio Decode bit is set to 1, it will start parsing the audio compressed data from the audio FIFO if there is data. If the FIFO is empty, then the DSP will wait. If the Start Audio Decode bit is set to 0, it will stop at the audio frame boundary until the Start Audio Decode bit is turned on. The decoder can be muted or stopped by the host by controlling the Mute Audio and Start Audio Decode bits. The audio output can be attenuated by the host. Each channel has its own attenuation control in 63 steps at 2 dB each.

### 4.7.3 Tone Generation

The MPEGCD20 can generate a tone or substitute a tone for the program currently being played. The host controls the tone by writing to the BEEP\_CTL register. When the Beep Duration bits of the BEEP\_CTL register are not zero, the tone samples will replace the decoded audio samples at the audio output. The user may select from 128 possible tones, 31 durations, and 8 attenuations.

The tones are over ten octaves of American Standard pitch ("A" = 440Hz), but tones on both the high and low end (high index and low index) may not be reproducible or may not be audible in some environments.

The lowest tone, index 0, is "C" 6 octaves under middle C, or about 8.175 Hz. The highest tone is index X'7F', a "G" at 12543.7 Hz.

The highest tones may not be audible, or if played at a low sampling rate may not be correctly reproduced. To guarantee correct reproduction, the highest tone playable at the lowest sampling rate (16K Samples/Second) should be under 7272 Hz. That is, the index should be less than 75 hexadecimal. The sampling rate is determined by the audio program being played.

## Functional Description

If no program has been started since the last reset, the sampling rate will be 48K samples/second.

The duration is in units of .1 seconds. A tone will be played for a minimum of .1 seconds to a maximum of 3.1 seconds, depending upon the selected duration value.

The maximum volume tone is approximately 70% of full scale, or 3 dB lower than the maximum possible volume. Attenuation index 7 selects the loudest tone. Each attenuation lower index step attenuates the tone by approximately 3 dB. Thus, attenuation index 0 selects a volume that is approximately 24 dB lower than full volume.

**Table 4-7. Attenuation below maximum possible volume.**

Index	Attenuation	Index	Attenuation
0	-24 dB	4	-12 dB
1	-21 dB	5	-9 dB
2	-18 dB	6	-6 dB
3	-15 dB	7	-3 dB

The following table describes the index-tone relationship. The table contains some representative values. Frequencies for other indices in the range 0 to 0x07F may be calculated from the values given.

**Table 4-8. Tone Indices and Frequencies**

Note	Index	Freq.	Index	Freq.	Index	Freq.	Index	Freq.
A	2D	110	39	220	45	440	51	880
A#	2E	116.54	3A	233.08	46	466.16	52	932.33
B	2F	123.47	3B	246.94	47	493.88	53	987.77
C	30	130.81	3C	261.63	48	523.25	54	1046.50
C#	31	138.59	3D	277.18	49	554.37	55	1108.73
D	32	146.83	3E	293.66	4A	587.33	56	1174.66
D#	33	155.56	3F	311.13	4B	622.25	57	1244.51
E	34	164.81	40	329.63	4C	659.26	58	1318.51

Table 4-8. Tone Indices and Frequencies

Note	Index	Freq.	Index	Freq.	Index	Freq.	Index	Freq.
F	35	174.61	41	349.23	4D	698.46	59	1396.91
F#	36	185.00	42	369.99	4E	739.99	5A	1479.98
G	37	196.00	43	392.00	4F	783.99	5B	1567.98
G#	38	207.65	44	415.30	50	830.61	5C	1661.22

#### 4.7.4 Playing PCM Data

The audio decoder is capable of playing PCM data from the host. The audio rate buffer must be purged as if it is a channel switch condition. This is done by issuing a Reset Audio Rate Buffer command from the host, or by using the Reset Audio bit in the BEEP\_CTL register. The PCM data is fed through the host data bus using the AUD\_FIFO register. The PCM Enable bit of the AUD\_CTL register must be set to '1' to play the PCM data. PCM play will start when the Start Audio Decode bit of the AUD\_CTL register is set to 1. Once PCM play starts, resetting the Start Audio Decode bit will not stop PCM play if PCM data is available. It can be stopped two ways:

1. PCM play will stop if the PCM data transmission is stopped.
2. PCM play will stop if the audio is muted for 100 sample times, using the Mute Audio bit, then the DSP is turned off using the Start DSP bit.

The PCM data may be mono or stereo. The 'Mono PCM' bit in the BEEP\_CTL Register must be set to indicate whether the data is mono or stereo. If mono, the PCM data will be played on both Audio Channels. The format of the Audio data is Most Significant Byte followed by Least Significant Byte.

The stereo samples will be played first sample to left Audio Channel, second sample to right Audio Channel. The format of the Audio data is Left Channel (Most Significant Byte, Least Significant Byte) then Right Channel (Most Significant Byte, Least Significant Byte)

#### 4.7.5 Audio Decode Status

An Audio Status region is defined in the DRAM. It will start at DRAM address 0x2000 and extends for 0x20 bytes. The first word (two bytes) contains Output State information, defining the availability of data at the output. 'Mute' and 'Beep' requests will have no effect on this state variable. Note that time for decoding data will introduce a lag between data arriving in or being consumed from the DRAM FIFO and changes in this state variable. The following states are defined.

##### 4.7.5.1 Reset Done

(Value x'0000') This Output State indicates that the Audio sub-system has completed its reset processing, and it is now ready to begin processing data. It will remain in this state until the

## Functional Description

'Start Audio Decode' bit is set to 1. If the 'Start Audio Decode' bit is set when the reset completes, this state may not persist long enough to be seen by the host.

### 4.7.5.2 Empty

(Value x'4000') This Output State indicates that there is no output data for the Audio Processor to send out. It will enter this state from the Reset Done State until data is ready to send out, and return to this state from the Play State any time that data is not available.

### 4.7.5.3 Play

(Value x'8000') This Output State indicates that there is output data for the Audio Processor to send out. It will enter this state anytime that there is data available to play.

The sequence that the Host should follow to play an Audio clip with reset, without affecting video is:

1. Set 'Start Audio Decode' bit to b'0'
2. Set the 'Audio Reset' bit to b'1'
3. Wait for 'Output State' in DRAM to go to 'Reset Done'
4. Set the 'Audio Reset' bit to b'0'
5. Set desired Audio Mode
6. Set 'Start Audio Decode' bit to b'1'
7. Send audio data
8. Wait for 'Output State' in DRAM to go to 'EMPTY'

The Audio clip has now been completely played.

**Note:** (1) The Audio-only reset will only be performed when the Start Audio Decode bit in the Aud\_Control register is b'0'. (2) This reset should only be issued when serial data transfer has been stopped on a byte boundary. (3) The Reset bit should not be set to b'1' until after the Start Decode bit has been set to b'0'.

## 4.7.6 Audio Clocks Generation

The phase-locked-loop plays an important role in maintaining the audio and video synchronization after the initial adjustment. Using the same 27MHz clock input, it precisely generates the audio bit out clock, L/R channel clock (sampling clock) and the oversampling clock for the DAC. The oversampling clock is 256 times of the sampling rate. These clocks are generated based on the sample rate of the audio input stream.

## 4.8 Audio Interface

The audio compressed data can be inputted via the serial interface or the parallel interface. The serial interface is controlled by the AUD\_REQ and the AUD\_BI\_CLK signals. The serial data input is on the AUD\_SD\_IN signal. When the AUD\_REQ is asserted by the decoder, it is ready to receive the audio compressed data. This signal is also applied to the parallel interface to input the audio compressed data. Bit 2 of the Chip Mode register, when set to 1, changes the

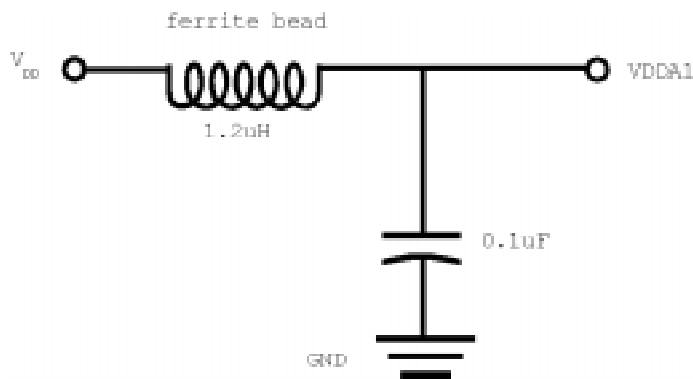


AUD\_BI\_CLK signal to a data strobe signal to clock the parallel data into the decoder. Also, the host has ability to send audio data into the decoder by using register address X'1C'.

The audio output is a serial PCM data output with three signals, AUD\_SD\_OUT, AUD\_BO\_CLK and L/R\_CLK. The interface can be the I<sup>2</sup>S mode or the default mode. The AUD\_SD\_OUT is the serial data output. The rising edge of the AUD\_BO\_CLK signal should be used to latch the serial data. The L/R\_CLK signal indicates the left or right channel sample. The output sample is 16 bits in two's complement with MSB first.

## 4.9 PLL External Components And Connections

The MPEGCD20 has two on-chip PLL's to generate the required audio and operating frequencies. To assure proper operation of the PLL's, care must be taken when the circuit board is designed. The analog power pins, VDDA1 and VDDA2, should be connected with ferrite beads and capacitors to minimize the digital noise which can cause phase jitter at the PLL output. All wire lengths should be kept as short as possible to minimize inductive coupling from other signals.



**Figure 4-8. Ferrite bead detail**

**Note:** The impedance of the ferrite bead should be much greater than that of the capacitors at frequency where noise is expected.

The REXT pin is sensitive to parasitic capacitance which should be no more than 10pf to ensure stability. This resistor should be surrounded by the ground leads connecting to ground pins, 131 and 133. The wiring should be kept as short as possible.

The filter pins, FLT1, FLT2, FLT3 and FLT4, must be connected to the filter components for the PLL to lock. The connections must be as short as possible. Nodes M1, M2, M3 and M4 are sensitive to parasitic capacitance. The filter pairs, FLT1/2 and FLT3/4 should not be coupled together.

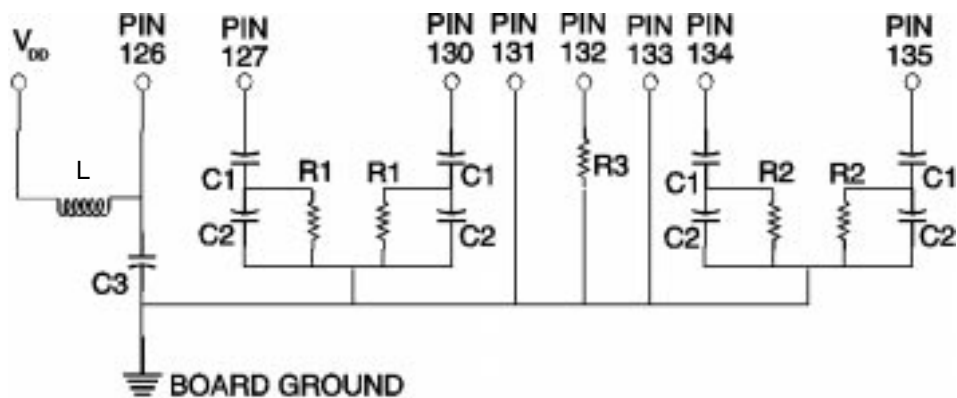


Figure 4-9. Pin description

C1	470pF +/-5%	R1	15kΩ +/-5% 0.1w
C2	22pF +/-5%	R2	10kΩ +/-5% 0.1w
C3	0.1uF +/-10%	R3	4.99kΩ +/-1% 0.1w
L	1.2uH		

4.9.1 27 MHz Clock Input Requirements

Table 4-9. 27 MHz Clock Input requirements

Voltage Levels	TTL, MPDL=0.8V. LPUL=2.0V
Duty Cycle	40% - 60%
Rise Time	1 - 4 V/ns.
Phase Jitter	+/- 150ps.

## 4.10 Audio/Video Synchronization

Through the use of an on-chip PLL, the MPEGCD20 locks the audio clocks to the 27 MHz clock together. To insure that the audio and video decoding processes do not drift apart, the 27 MHz clock input should be used to feed the pixel clock and to derive the video timing signal. This insures that the audio and video decoding processes do not drift apart.

### 4.10.1 STC Master Synchronization

Synchronization of both the audio and video streams to the STC is simplified by the fact that MPEGCD20 accepts Packet layer audio and video streams. Both the audio and video processors compare the PTS's received in the associated stream to the STC. If the comparison of the PTS to the STC is out of range, the audio and video processors will adjust the presentation of the output material to correct for this out of sync condition.

The adjustment method used by the video processor to achieve synchronization to the STC is by skipping or repeating frames. The decoder will repeat as many frames as necessary to achieve synchronization. When the PTS is behind the STC and it is necessary to skip frames of video, the decoder will skip a single B-frame at a time to achieve synchronization. If the input stream does not contain any B-frames then P frames are skipped or repeated. The limit used to determine if the video is synchronized to the STC is  $\pm 22.5$  milliseconds.

The adjustment mechanism used by the audio decoder to achieve synchronization is by skipping or repeating audio samples. This is done in a manner to insure that no audio artifacts are produced. The audio processor synchronizes to the STC with a tolerance of  $\pm 5$  milliseconds.

### 4.10.2 Video Master Synchronization Mode

When host register 00 (Chip\_Control) bit 5 is set, this places the chip in video master synchronization mode. Within this mode, any PTS's that arrive in the video stream and are outside the  $\pm 22.5$  milliseconds tolerance, are loaded into the STC register automatically. If the PTS is within the  $\pm 22.5$  millisecond tolerance, then the STC is not updated. Within this mode no video frames are skipped or repeated.

When running in video master synchronization mode, the audio processor's synchronization method is the same as that of STC master synchronization mode.

### 4.10.3 Audio Master Synchronization Mode

When host register 00 (Chip\_Control) bit 6 is set, this places the chip in audio master synchronization mode. Within this mode, any PTS's that arrive in the audio stream are loaded into the STC register automatically. The audio decoder will search for the first PTS in the stream. Once it finds the PTS, the PTS will load that value into the STC register. The audio decoder will then begin to decode and play the data. Within this mode all audio samples are decoded and played. No samples will be skipped or repeated.

## Functional Description

When running in audio master synchronization mode, the video processor synchronizes to the STC in the same manner as when it is in STC master synchronization mode. The tolerances that the video uses to synchronize to the STC are the same.

### 4.10.4 Synchronization Disabled Mode

When host register 00 (Chip\_Control) bit 4 is set, this disables any synchronization within MPEGCD20. Within this mode, all PTS's in both the audio and video streams are ignored. The video processor will decode and display all video frames without any regard to the PTS's contained in the stream. No frames of video will be skipped or repeated.

The audio decoder will search for the first PTS in the stream, then begin decoding and playing the data. The audio processor will decode all audio samples and play all audio samples without any regard to the PTS's contained in the stream. No audio samples will be skipped or repeated.

### 4.10.5 Broadcast Environment

One possible application would be in the broadcast environment, where the decoder has no control over the input data rate. As defined in the MPEG-2 standard, there are several time references that can be effectively utilized for synchronization purposes. These are the System Clock Reference (SCR), Program Clock Reference (PCR), and the PTS. The PCR is used in transport streams, the SCR is in program streams.

In a broadcast environment, the PCR is commonly used for synchronization. Its arrival time is controlled by the encoding, and the broadcast path. The PCR synchronizes the STC. If synchronization starts to drift, the decoder's rate buffer may either overflow or run out of compressed data. Two possible methods of preventing these conditions are described.

### 4.10.6 Synchronization With Clock Recovery

Fine synchronization can be achieved by locking the decoder's 27 MHz clock to the encoder clock PCR using an external clock recovery circuitry. This 27 MHz clock can then be used to drive the pixel output and the audio output from the MPEGCD20. The display controls, VSC and HSC, will also be synchronized with this 27 MHz clock.

The MPEGCD20 has an internal counter that plays the role of STC. This counter is driven by the 27 MHz clock and has a selectable time base resolution of either 90 kHz or 27 MHz. The counter will run whenever the Start Video Decode bit is set to 1. The counter is accessible to the host processor through the decoder chip's host interface. This host interface can also be used to load video PTS values into the decoder chip if the input stream is an elementary stream. The PTS must be sent in before the picture header which the PTS belongs to.

The STC can be initialized with a value derived from the PCR at the beginning of the decoding process. This value contains the PCR and the decoding delay which needs to be determined from the system design. Because the 27 MHz clock is synchronized to the encoder, the STC will track PCR changes accurately, except when the PCR changes dramatically such as when changing a channel. The audio is also locked to the STC because the audio sampling clock

from the PLL locks to the same 27 MHz clock. During normal operation effective audio/video synchronization is achieved without host intervention.

### **4.10.7 Synchronization Without Clock Recovery**

Coarse synchronization can be achieved without using an external clock recovery circuitry. In this implementation, the STC can be made to stay approximately in synchronization with the encoder clock by continuously refreshing its contents based on the value of arriving PCRs. A host may also choose to selectively refresh the STC whenever it determines that the STC is drifting too far from the PCR values.

For synchronization without clock recovery, the 27 MHz clock is not synchronized with the encoder clock, the decoder STC will drift with respect to the received PTS's. If the PTS is not within the tolerance of the decoder's STC, the audio and video decoders will be forced to make the appropriate adjustments to recover synchronization.

### **4.10.8 PC System Environment**

In the application environment where the compressed data is input from a computer (PC) disk drive, there are several ways to synchronize video and audio. Here are two possible implementations:

- The audio decoder is the master. The video decoder STC is updated with the audio PTS by the audio decoder subsystem. The video STC is taken and compared with the associated video PTS. If the compare is outside the tolerance, a video frame is either skipped or repeated.
- The video decoder is the master and will present pictures in sequential order governed by the display clock. The audio decoder will compare its PTS with the STC and adjust its audio presentation accordingly.

## **4.11 Error Detection and Concealment**

The MPEGCD20 is capable of detecting certain types of errors in the incoming bit stream and concealing them.

### **4.11.1 Video Error Concealment**

Error detection is accomplished by the internal processor through testing and comparison with expected conditions. When detected, these error conditions initiate error recovery actions in the on-chip code. The specific error recovery activities initiated depend on which phase of the bit stream processing is active at the time of the error detection.

A number of error conditions can be detected by the on-chip microcode. These are shown below along with the action taken to correct or conceal the error condition.

## Functional Description

- Action: Skip to next Sequence header in encoded bit stream.
  - marker\_bit(s) = "0" in the Sequence header
  - invalid aspect\_ratio\_information value ("0")
  - invalid frame\_rate\_code value ("0")
  - invalid chroma\_format value ("0")
- Action: Search for next Picture header
  - invalid picture\_type value ("0")
  - invalid picture\_structure value ("0")
- Action: Skip to the next Slice header and then rebuild the skipped macroblocks
  - marker\_bit = "0" in the macroblock data
  - invalid frame\_motion\_type value ("0")
  - invalid field\_motion\_type value ("0")

Error conditions detected at the slice or macroblock levels can be detected by either hardware or code. The hardware will detect any invalid bit stream data resulting in invalid VLC values. Errors detected at the slice or macroblock levels initiate error recovery by setting an error flag. This flag is monitored by the on-chip microcode every time a new macroblock is about to be processed. When the error flag is ON, the basic recovery actions include skipping ahead to the next Slice Header and then rebuilding the skipped macroblocks. The rebuilding activity depends on the picture type being processed.

The macroblock rebuilding is always done with simple forward motion compensation from the past reference frame. Frame or field based motion compensation is invoked based on the current picture structure. The motion vectors used to rebuild the *failing* macroblocks also depend on the current picture types:

**I-frames**      If the error condition is detected during processing of an Intra-coded macroblock and concealment motion vectors have been sent, the concealment motion vectors will be used to rebuild the contents of the skipped macroblocks. In the case where no concealment motion vectors have been sent, the intervening macroblocks are processed as *skipped* macroblocks (as if it were a P-frame).

**Note:** There is no requirement in the *MPEG-2 standard* for concealment motion vectors to be sent in the encoded bit stream.

**P-frames or B-frames:**      If the error conditions are detected during the processing of a P-frame or a B-frame, the forward motion vectors for Field 1 are saved for each macroblock in a row, that is, one pixel line across the picture. These motion vectors are then used to rebuild the *failing* macroblocks.

### 4.11.2 Audio Error Concealment

The MPEGCD20 audio decoder detects the CRC errors and certain syntax errors. When an error is detected, the decoder will perform concealment and skip over the bad data. If a new

frame cannot be found, it will mute the output until good data is found. The error concealment modes are defined by bits 6-5 of the AUD\_CTL register which can be set by the host.

### **4.11.3 Transport Error Handling**

The MPEGCD20 also can be alerted of an error detected externally via the Error input. When the Error input is asserted, the video compressed data will be ignored until the Error input is de-asserted. During the time that the Error is asserted, the video decoding is stopped and the last decoded picture is displayed repeatedly. When the Error is de-asserted, the decoder will start searching for a sequence start code to resume decoding.

There is another option to notify the MPEGCD20 on error by inserting a sequence-error-code in the video bit stream. When the video decoder detects the sequence-error-code, it will search for a sequence start code to continue decoding.

## **4.12 4:2:2 Profile Requirements**

The MPEGCD21 is designed to support the 4:2:2 profile at main level.

- Chroma format - 4:2:2 or 4:2:0
- Picture types - I, P, B
- Picture size - 720x512 @ 30Hz and 720x608 @ 25Hz
- Intra dc precision - 8, 9, 10, 11
- Max. number of bits in a macroblock - unconstrained
- Max. bit rate - 50 Mbits/s
- Separate luminance and chrominance quantization tables

The following items are needed to support the 4:2:2 profile.

- Tie up on pin 136 to generate a higher internal clock frequency.
- Use 60ns DRAM.





## Chapter 5. Programming Reference

### 5.1 Host Accessible Registers

The host processor communicates with the MPEGCD20 using the set of registers defined in the register address space. These registers were defined to allow the Host to perform specific functions. The registers are grouped by function in the following tables and each of the registers is explained in more detail in the following register definitions.

Chip Control and Status				
Function	Address	Register	Usage	R/W
Chip Control	00	CHIP_CONTROL	To control chip operation	R/W
	01	CHIP_MODE	To set chip mode of operation	R/W

Timer Control and Status				
Function	Address	Register	Usage	R/W
Timer Control	02	SYNC_STC2	STC high order bits 32-30	R/W
	03	SYNC_STC1	STC middle order bits 29-15	R/W
	04	SYNC_STC0	STC low order bits 14-0	R/W
	05	SYNC_PTS2	PTS high order bits 32-30	W
	06	SYNC_PTS1	PTS middle order bits 29-15	W
	07	SYNC_PTS0	PTS low order bits 14-0	W

Video FIFO Control				
Function	Address	Register	Usage	R/W
FIFO Control	08	FIFO	FIFO data port	R/W
	09	FIFO_STAT	FIFO status	R/W
	2B	RB_THRESHOLD	Rate buffer threshold	R/W

## Programming Reference

Command Buffer				
Function	Address	Register	Usage	R/W
Command I/F	0A	COMMAND	To send a command to decoder	W
	0B	CDM_DATA	Data port for command parameters	R/W
	0C	CMD_STAT	Command status	R/W
	0D	CMD_ADDR	Command address	R/W

Internal Processor Control and Status				
Function	Address	Register	Usage	R/W
I-Store Access	0E	PROC_IADDR	I-Store (Instruction Store) address	R/W
	0F	PROC_IDATA	Data port for I-Store	R/W
	35	WR_PROT	Prevent accidental I-Store changes	R/W

DRAM Access				
Function	Address	Register	Usage	R/W
DRAM Access	10	DRAM_ADDRHI	DRAM high order address bits 21-16	R/W
	11	DRAM_ADDRLO	DRAM low order address bits 15-0	R/W
	12	DRAM_DATA	DRAM access data port	R/W
	13	DRAM_CMD/STAT	DRAM command and status	R/W
	3B	BLOCK_SIZE	DRAM access block size	R/W
	3C	SRC_ADDRHI	DRAM source address bits 21-16	R/W
	3D	SRC_ADDRLO	DRAM source address bits 15-0	R/W

Onscreen Display Control				
Function	Address	Register	Usage	R/W
OSD Control	14	OSD_ADDR	Address for OSD data	R/W
	15	OSD_DATA	Data port for OSD	W
	17	OSD_MODE	OSD mode control	R/W
	29	OSD_LINK_ADDR	OSD first link address	R/W
	2A	OSD_SIZE	OSD buffer size	R/W

Interrupt Control				
Function	Address	Register	Usage	R/W
Interrupt Control	18	HOST_INT	Interrupts	R
	19	MASK	Mask for interrupts	R/W
	3E	HOST_INT1	Interrupts	R
	3F	MASK1	Mask for interrupts	R/W

Audio Control				
Function	Address	Register	Usage	R/W
Audio Control	1A	AUD_IADDR	Audio instruction address	R/W
	1B	AUD_IDATA	Data port for audio instruction store	R/W
	1C	AUD_FIFO	Audio FIFO data port and status	R/W
	1D	AUD_CTL	Audio control	R/W
	1E	BEEP_CTL	Beep control	R/W
	2D	FRNT_ATTEN	Audio attenuation	R/W

Display Control				
Function	Address	Register	Usage	R/W
Display Control	20	DISP_MODE	To set display mode of operation	R/W
	21	DISP_DLY	To set sync signal delay	R/W
	22	VBI_CTL	To set VBI control	R/W
	23	DISP_LBOR	To set left border of display	R/W
	24	DISP_TBOR	To set top border of display	R/W

Polarity Control				
Function	Address	Register	Usage	R/W
Polarity Control	2C	INFC_CTL	Program signal polarity	R/W

### 5.1.1 Register Definitions

Reserved bits must be written to zero.

CHIP_CONTROL	
Address = 00	Type = RW      Reset = 3-00
Bits	Function
15 - 12	<b>Chip Version</b> These 4 bits are read only to identify the chip version. CD21 is identified by 0011.
11 - 8	Reserved
7	<b>Black Out Display</b> When this bit is set to 1, the display output will produce a black screen. This bit must be set to 0 to output video.
6	<b>Audio Master Synchronization Mode</b> When this bit is set to 1, the audio will become the master for synchronization. The audio PTS is used to set the STC. When this bit is 0, the audio will synchronize to the STC by skipping or repeating frames. The adjustment can be disabled by setting bit 4 of this register to 1.

CHIP_CONTROL		
Address = 00	Type = RW	Reset = 3-00
Bits	Function	
5	<b>Video Master Synchronization Mode</b> This bit, when set to 1, causes the video PTS to be set into the STC whenever the PTS/STC comparison is out of range. The picture is not skipped or repeated. When this bit is 0 the STC is set by the host. When bit 4 of this register is set to 1, this bit is meaningless.	
4	<b>Disable Synchronization</b> This bit, when set to 1, causes the PTS and STC compare to be disabled.	
3	Reserved	
2	<b>Chip Reset</b> This bit, when set to 1, causes the chip to reset all registers. Instruction Storage is not affected by this reset. The host processor is responsible for setting this bit back to 0.	
1	<b>Start Video Processor</b> This bit, when set to 1, causes the internal processor to start the instruction execution. When Instruction Storage is being loaded or the instruction address register is to be written, this bit must be 0.	
0	<b>Start Video Decode</b> This bit, when set to 1, causes the internal processor to start the decoding process. When it is set to 0, the decoding process will stop and the STC will not increment. When it is set to 1, the STC starts to update.	

CHIP_MODE		
Address = 01	Type = RW	Reset = 0000
Bits	Function	
15	<b>2MB PAL Mode</b> When this bit is set to 1, the 2MB PAL mode is activated. To save DRAM, only 2.75 frame buffers are used in this mode. When this bit is set to 0, three frame buffers are used.	
14	<b>Internal Byte Sync Mode</b> When this bit is set to 1, the internal byte sync generation is activated. It eliminates the byte sync input for the video serial interface. When it is 0, the byte sync input is required for the video serial interface. When the parallel input is used for the video compressed data, this bit must be 0.	

## Programming Reference

CHIP_MODE		
Address = 01	Type = RW	Reset = 0000
Bits	Function	
13	<b>Serial Input Mode</b> When this bit is set to 1, the video compressed data is expected to be inputted through the serial input. When it is 0, the parallel input is used for the video compressed data.	
12	<b>Enable PIP Mode</b> This bit is used to enable the picture-in-picture feature support. When this bit is set to 1, the Transparency Gate is enabled. The Transparency Gate is controlled by the transparent pixel of an OSD bitmap.	
11	Reserved	
10	<b>Load STC Mode</b> When this bit is set to 1, the System Time Clock (STC) value written by the host is not loaded to the STC counter until the V-sync of field one. When the bit is 0, the STC value is loaded to the STC counter immediately.	
9-6	Reserved	
5	<b>Serial Input Bit Counter Reset Mode</b> When this bit is set to 1, the bit counter which is used for byte alignment will not be reset when the rate buffer is reset. When this bit is 0, the bit counter is reset as the rate buffer is reset. The bit counter is reset when the decoder is reset.	
4	<b>STC Time Base</b> When this bit is set to 1, the System Time Clock is updated every 27MHz cycle. When this bit is set to 0, the STC is updated every 90KHz cycle.	
3	<b>Enable DAC Clock</b> When this bit is set to 1, the driver for the DAC clock is enabled. When the bit is 0, the driver is tri-stated.	
2	<b>Audio Parallel Data Strobe Mode</b> When this bit is set to 1, the 8-bit data bus is used for the audio compressed data input. The audio serial data clock becomes the audio data strobe and the polarity is programmable. The audio serial data input is disabled.	
1	<b>Byte Swap</b> This bit, when set to 1, causes the data bytes into the video Rate Buffer to be swapped to handle the Intel notation. The bit only affects the data into the DRAM.	
0	<b>No B Frame Mode</b> When this bit is set to 1, B-pictures are not allowed in the bit stream.	

SYNC_STC2		
Address = 02	Type = RW	Reset = ---0
Bits	Function	
15 - 4	Reserved	
3 - 1	STC 32-30 These are the high order bits of the System Time Clock (STC). See the following description of SYNC_STC0 for additional detail.	
0	Reserved	

SYNC_STC1		
Address = 03	Type = RW	Reset = 0000
Bits	Function	
15 - 1	STC 29-15 These are bits 29 to 15 of the System Time Clock (STC). See the following description of SYNC_STC0 for additional detail.	
0	Reserved	

SYNC_STC0		
Address = 04	Type = RW	Reset = 0000
Bits	Function	
15 - 1	STC 14-0 These are bits 14 to 0 of the System Time Clock (STC). The SYNC_STC2 register is written first, then the SYNC_STC1 and followed by the SYNC_STC0. After SYNC_STC0 is written, all 33 bits are loaded into the STC at the same time. When the host processor reads the STC, it must read SYNC_STC2 first, then followed by SYNC_STC1 finally SYNC_STC0. The STC will continue to count while the host processor reads the SYNC_STC2,1,0 registers	
0	Reserved	

SYNC_PTS2		
Address = 05	Type = W	Reset = ---0
Bits	Function	
15 - 4	Reserved	
3 - 1	STC 32-30 These are the high order bits of the Presentation Time Stamp (PTS).	
0	Reserved	

SYNC_PTS1		
Address = 06	Type = W	Reset = 0000
Bits	Function	
15 - 1	STC 29-15 These are bits 29 to 15 of the Presentation Time Stamp (PTS).	
0	Reserved	

SYNC_PTS0		
Address = 07	Type = W	Reset = 0000
Bits	Function	
15 - 1	STC 14-0 These are bits 14 to 0 of the Presentation Time Stamp (PTS).	
0	Reserved	



FIFO		
Address = 08	Type = RW	Reset = 0000
Bits	Function	
15 - 0	<b>FIFO Data</b> This register is used to write the incoming video coded data into the video FIFO. The video FIFO is a temporary storage area used to buffer the incoming data between the Host and the rate buffer area in DRAM. When this register is read, the value indicates the number 32-byte blocks of compressed data in the DRAM.	

FIFO_STAT		
Address = 09	Type = R	Reset = 0000
Bits	Function	
15- 8	Reserved	
7 - 0	<b>Available Byte Count</b> When this register is read, the value represents the number of bytes that can be accepted in the input video FIFO (temporary storage area). A '0000 0000' represents FULL and '1000 0000' represents EMPTY.	

FIFO_STAT		
Address = 09	Type = W	Reset = 0000
Bits	Function	
15- 2	Reserved	
1 - 0	<b>FIFO Margin Control</b> These 2 bits are used to program the available space in the video FIFO when the video request signal is de-asserted. <div style="margin-left: 40px;">             00 - 32 bytes              01 - 16 bytes              10 - 8 bytes              11 - 4 bytes           </div>	

## Programming Reference

COMMAND		
Address = 0A		Type = W      Reset = --00
Bits	Function	
15- 5	Reserved	
4 -1	Command Code These 4 bits represent one of the sixteen possible commands to be executed by the internal processor. The commands are defined in the Host Command section.	
0	Command Chaining This bit, when set to 1, indicates that another command will be sent after this command is completed. The internal processor will continue to test the command pending bit for the next command.	

CMD_DATA		
Address = 0B		Type = RW      Reset = 0000
Bits	Function	
15 - 0	Command Data This register is used to write data to or read data from the parameter stack. The parameter stack has four 16-bit registers which are addressed by the command address register.	

CMD_STAT		
Address = 0C		Type = RW      Reset = ---0
Bits	Function	
15- 1	Reserved	
0	Command Pending This bit, when set to 1, indicates that a command is active. After the host processor writes the command and all the parameters, it sets this bit to 1 to indicate that a command is pending. The internal processor will then execute this command and reset the command pending bit.	

CMD_ADDR		
Address = 0D	Type = RW	Reset = ---0
Bits	Function	
15- 2	Reserved	
1 - 0	Parameter Stack Address This 2-bit register is used to address the parameter stack. It must be set before the desired parameter can be addressed.	

PROC_IADDR		
Address = 0E	Type = RW	Reset = -000
Bits	Function	
15- 12	Reserved	
11 - 0	I-Store Address These 12 bits are used to address the video processor's Instruction Storage that contains the microcode in a set of 16-bit words. After the address is set, the data is read or written using the PROC_IDATA register. The address is automatically incremented by one after an Instruction Store WRITE operation. For Instruction Store READS, this address register must be set each time. This register is also used to set the starting address for instruction execution. The processor must be stopped before accessing bits 11-0 of the PROC_IADDR register.	

PROC_IDATA		
Address = 0F	Type = RW	Reset = 0000
Bits	Function	
15- 0	I-Store Data This register is used to read data from or write data to the video Instruction Store. The PROC_IADDR register provides the address to Instruction Store. The processor must be stopped before accessing the PROC_IDATA register. <b>Note:</b> Bit 0 of the WR_PROT register must be set to "1" before writing to Instruction Store.	

DRAM_ADDRHI		
Address = 10	Type = RW	Reset = --00
Bits	Function	
15- 6	Reserved	
5 - 0	DRAM Address 21-16 These are the high order bits of the DRAM address and the destination address for a block copy, block transfer to DRAM or DRAM fill operation.	

DRAM_ADDRLO		
Address = 11	Type = RW	Reset = 0000
Bits	Function	
15- 0	DRAM Address 15-0 These are the low order bits of the DRAM address. The address is used for accessing the DRAM when the DRAM_CONTROL register is written. After each access, the address is automatically incremented by two (2-byte word). The low order bit must be zero. These bits are also the low order bits of the destination address for a block copy, block transfer to DRAM or DRAM fill operation. This register is updated by hardware to indicate the next byte of data to be destined. If bit 1, VBI Access bit, of the DRAM_CONTROL register is set to 1, this register is used as the offset address to access the Vertical Blanking Interval (VBI) buffers.	

DRAM_DATA		
Address = 12	Type = RW	Reset = 0000
Bits	Function	
15- 0	DRAM Data This register is used to read data from or write data to the DRAM. When the DRAM_CONTROL register is written with bit 15 set to 0 to access the base address, this register is used to hold the base address.	

DRAM_CONTROL		
Address = 13	Type = RW	Reset = 0--0
Bits	Function	
15	<b>DRAM Busy</b> This bit must be set to 1 by the host processor to initiate a DRAM operation. It indicates that the DRAM is busy. When the DRAM operation is completed, this bit is reset by the DRAM control hardware. This bit should be checked by the host processor before a new DRAM access command is issued.	
14	<b>Data Ready</b> This bit is used in the Block Transfer From DRAM operation. When this bit is 1, it indicates that two bytes of data can be read from the DRAM_DATA register. This bit must be checked before the DRAM_DATA register is read.	
13 - 12	Reserved	
11-8	<b>Base Address Index</b> These four bits are used to read the base addresses allocated for the buffers in the DRAM. When this register is written with bit 15, DRAM Busy, set to 0, the base address indicated by these four bits is loaded into the DRAM_DATA register. At least one dummy cycle (80ns) must be inserted before the DRAM_DATA register can be read. If the DRAM_DATA register is not read immediately, the content can be overwritten by other operations. The address is in units of 128 bytes. <div style="margin-left: 40px;">             0000 - Base address of buffer 0 for luminance              0001 - Base address of buffer 0 for chrominance              0010 - Base address of buffer 1 for luminance              0011 - Base address of buffer 1 for chrominance              0100 - Base address of buffer 2 for luminance              0101 - Base address of buffer 2 for chrominance              1000 - Base address of rate buffer              1001 - Base address of OSD buffer              1010 - Base address of user data buffer              1011 - Base address of audio buffer              1101 - Base address of VBI buffer 0              1110 - Base address of VBI buffer 1           </div>	
7-6	Reserved	

DRAM_CONTROL		
Address = 13	Type = RW	Reset = 0–0
Bits	Function	
5	<p><b>Block Fill</b></p> <p>This bit is used to indicate that a block fill operation is to be performed. When this bit and bit 15, DRAM Busy, are both set to 1, the two-byte data pattern in the DRAM_DATA register is written to DRAM starting from the address indicated by the DRAM_ADDRHI and DRAM_ADDRLO registers for a length specified by the BLOCK_SIZE register. The address and length are on 2-byte boundary. When the BLOCK_SIZE register decrements to 0, the Block Move Complete interrupt is raised. While the Block Fill is in progress, the Block Copy, Block Transfer, DRAM access, OSD data write or the VBI data write is not allowed.</p>	
4	<p><b>Block Transfer</b></p> <p>This bit is used to indicate that a block transfer operation is to be performed. When this bit and bit 15, DRAM Busy, are both set to 1, the number of bytes indicated by the BLOCK_SIZE register are transferred to/from DRAM. The address must be on an 8-byte boundary and the length must be a multiple of 8. The direction is indicated by the DRAM Write bit. When the DRAM Write bit is 1, the host can continue to write data to the DRAM_DATA register. But the time period cannot be less than 1.6us per 8 bytes. When the BLOCK_SIZE register decrements to 0, the Block Move Complete interrupt is raised. If the DRAM Write bit is 0, the host must check the Data Ready bit before it can read data from the DRAM_DATA register. 8 bytes of data can be read before the Data Ready bit needs to be checked again. While the Block Transfer is in progress, the Block Copy, Block Fill, DRAM access, OSD data write or the VBI data write is not allowed.</p>	
3	<p><b>Block Copy</b></p> <p>This bit is used to indicate that a block copy operation is to be performed. When this bit and bit 15, DRAM Busy, are both set to 1, the number of 2-byte words indicated by the BLOCK_SIZE register are copied from the Source Address registers to the location indicated by the DRAM Address registers. When the operation is complete, the Block Move Complete interrupt is raised and the DRAM Busy bit is set to 0. While this operation is in progress, the Block Transfer, Block Fill, DRAM access, OSD data write or the VBI data write is not allowed.</p>	
2	<p><b>VBI Pointer</b></p> <p>This bit is used to indicate which VBI buffer is to be written. There are two VBI buffers, VBI0 and VBI1, which can be set up for the VBI application. When the bit is set to 0, it indicates VBI0 is to be written. If the bit is 1, VBI1 is used. If only one VBI buffer is used, the VBI Pointer bit must be flipped to indicate that the VBI buffer just written is ready for display.</p>	
1	<p><b>VBI Access</b></p> <p>When this bit is set to 1, the DRAM_ADDRLO register value is used as the offset address to access the VBI buffer in DRAM. This bit must be set to 0 to perform the normal DRAM accesses.</p>	

DRAM_CONTROL		
Address = 13	Type = RW	Reset = 0--0
Bits	Function	
0	<b>DRAM Write</b> When this bit is set to 1, a write operation is performed. If the bit is 0, a read operation is performed.	

OSD_ADDR		
Address = 14	Type = RW	Reset = 0000
Bits	Function	
15- 0	<b>OSD Data Address</b> This address specifies the location within the OSD space in DRAM. On-chip FIFO accumulates 8 bytes of data before transferring to DRAM. The address which is in units of 8 is automatically incremented by 1.	

OSD_DATA		
Address = 15	Type = W	Reset = 0000
Bits	Function	
15- 0	<b>OSD Data</b> This register is used to write data to the OSD space in DRAM. Data can be written successively without overflowing the FIFO if the rate of less than 8 bytes per 1.6us is maintained.	

OSD_MODE		
Address = 17	Type = RW	Reset =0000
Bits	Function	
15 - 5	Reserved	

## Programming Reference

OSD_MODE		
Address = 17	Type = RW	Reset = 0000
Bits	Function	
4	<b>Blending Mode</b> When this bit is set to 0, each region can only have one of the fifteen OSD Blending levels. When this bit is set to 1, only three blending levels are possible but each color can have a different blending level.	
3	<b>Animation</b> When this bit is set to 1, the OSD operates in animation mode.	
2 - 0	<b>Animation Rate</b> These three bits define the rate of animation. (shown in NTSC) For PAL, multiply the NTSC rate by 25/30. <div style="margin-left: 40px;">             000 - 30 frames per second              001 - 15 frames per second              010 - 10 frames per second              011 - 7.5 frames per second              100 - 6 frames per second              101 - 5 frames per second              110 - 4.3 frames per second              111 - 3.75 frames per second           </div>	

HOST_INT		
Address = 18	Type = RW	Reset = 0000
Bits	Function	
15	<b>Sequence Start</b> This bit, when set to 1, indicates a sequence start code has been detected. This bit is reset to 0 when the high byte of this register is read.	
14	<b>Sequence Error</b> This bit, when set to 1, indicates a sequence error has been detected. This bit is reset to 0 when the high byte of this register is read.	
13	<b>Sequence End</b> This bit, when set to 1, indicates an end of sequence start code has been detected. This bit is reset to 0 when the high byte of this register is read.	



HOST_INT		
Address = 18	Type = RW	Reset = 0000
Bits	Function	
12	<b>GOP/SMPTE</b> This bit, when set to 1, indicates that the SMPTE time code has been received. This bit is reset to 0 when the high byte of this register is read.	
11	<b>Picture Skipped</b> This bit, when set to 1, indicates a picture has been skipped to speed up the decoding process. This bit is reset to 0 when the high byte of this register is read.	
10	<b>Picture Start</b> This bit, when set to 1, indicates that the decoder is ready to start decoding a new picture. This bit is reset to 0 when the high byte of this register is read.	
9	<b>Picture Resolution Changed</b> This bit, when set to 1, indicates that the picture resolution has been changed.	
8	<b>User Data</b> This bit, when 1, indicates that the user data has been received and stored. This bit is reset to 0 when the high byte of this register is read.	
7	<b>VBI Start</b> The bit is set to 1 at the V-sync time to indicate that the VBI region of a field has started. This interrupt bit is set only when the VBI Line Count field is non-zero. This bit is reset to 0 when the low byte of this register is read.	
6	<b>Video Start</b> This bit, when 1, indicates that the active video of a field has started. This bit is reset to 0 when the low byte of this register is read.	
5	<b>First Field Video Start</b> The bit is set to 1 at V-sync time to indicate the start of the first field display. This bit is reset to 0 when the low byte of this register is read.	
4	<b>Block Move Complete</b> This bit, when 1, indicates that the Block Copy or Block Transfer operation is complete. This bit is reset to 0 when the low byte of this register is read.	
3	<b>Audio Rate Buffer Threshold</b> When the audio compressed data falls below a programmable threshold, this bit is set to 1. This bit is reset to 0 when the low byte of this register is read.	
2	<b>Video Rate Buffer Threshold</b> When the video compressed data falls below a programmable threshold, this bit is set to 1. This bit is reset to 0 when the low byte of this register is read.	

## Programming Reference

HOST_INT		
Address = 18	Type = RW	Reset = 0000
Bits	Function	
1	<b>Video Rate Buffer Overflow</b> When the video rate buffer is overflow, this bit is set to 1. This bit is reset to 0 when the low byte of this register is read.	
0	<b>OSD Data</b> When OSD is enabled and this bit is set to 1 by hardware, it indicates that the OSD can receive additional OSD data. This bit is reset to 0 when the low byte of this register is read.	

MASK		
Address = 19	Type = RW	Reset = 0000
Bits	Function	
15	<b>Sequence Start</b> This bit, when set to 1, allows the sequence start code interrupt.	
14	<b>Sequence Error</b> This bit, when set to 1, allows the sequence error interrupt.	
13	<b>Sequence End</b> This bit, when set to 1, allows the end of sequence start code interrupt.	
12	<b>GOP/SMPTE</b> This bit, when set to 1, allows the SMPTE time code interrupt.	
11	<b>Picture Skipped</b> This bit, when set to 1, allows the skip-picture interrupt.	
10	<b>Picture Start</b> This bit, when set to 1, allows the picture start interrupt.	
9	<b>Picture Resolution Changed</b> This bit, when set to 1, allows the picture resolution change interrupt.	
8	<b>User Data</b> This bit, when 1, allows the user data interrupt.	
7	<b>VBI Start</b> This bit, when 1, allows the VBI Start interrupt.	

MASK		
Address = 19		Reset = 0000
Type = RW		
Bits	Function	
6	Video Start This bit, when 1, allows the video start interrupt.	
5	First Field Video Start This bit, when 1, allows the first field video start interrupt.	
4	Block Move Complete This bit, when 1, allows the Block Move Complete interrupt.	
3	Audio Rate Buffer Threshold This bit, when set to 1, allows the Audio Rate Buffer Threshold interrupt.	
2	Video Rate Buffer Threshold This bit, when set to 1, allows the Video Rate Buffer Threshold interrupt.	
1	Video Rate Buffer Overflow This bit, when set to 1, allows the Video Rate Buffer Overflow interrupt.	
0	OSD Data This bit, when set to 1, allows the OSD data interrupt.	

AUD_IADDR		
Address = 1A		Reset = 0000
Type = RW		
Bits	Function	
15	Reserved	
14 - 0	Audio I-Store Address  These 15 bits are used to address the audio instruction store. After the address is set, the instruction data is read or written using the AUD_IDATA register. The address is automatically incremented by one after the data is written. Since the audio instruction is three bytes wide, the low order two bits are used to address the byte within the instruction. The high order 13 bits are used to address the 3-byte instruction. The low order two bits will generate a carry out in mod-3 arithmetic and must start from zero. For the audio I-Store read, this address must be set for each read. The audio processor must be stopped and the I-Store write protect bit is 1 before the AUD_IADDR register can be set.	

## Programming Reference

AUD_IDATA		
Address = 1B	Type = RW	Reset = --00
Bits	Function	
15 - 8	Reserved	
7 - 0	Audio I-Store Data  This register is used to read data from or write data to the audio instruction store. The AUD_IADDR register provides the address to the audio instruction store. Since this register is defined in eight bits, it must be written three times for each instruction. The audio processor must be stopped before this register can be accessed.	

AUD_FIFO		
Address = 1C	Type = RW	Reset = -000
Bits	Function	
15 - 9	Reserved	
8	Audio Buffer Full  This bit, when set to 1, indicates the audio buffer is full. There is still room for 4 more bytes of data coming in when the bit is set to 1. When this bit is 0, the audio data request signal is asserted.	
7 - 0	Audio FIFO Data  This register is used to write data to the audio compressed data buffer in the DRAM. When this register is read, this field will provide the amount of audio compressed data in DRAM in units of 64 bytes.	

AUD_CONTROL		
Address = 1D	Type = RW	Reset = -000
Bits	Function	
15	Reserved.	
14 - 8	Tone Control  These 7 bits are used to control the tone of the beep.	
7	Reserved	

AUD_CONTROL		
Address = 1D	Type = RW	Reset = -000
Bits	Function	
6 - 5	<p>Audio Error Concealment Mode</p> <p>These two bits select the error concealment mode for audio.</p> <p>00 - Repeat last frame once. Mute if error persists.</p> <p>01 - Reserved.</p> <p>10 - Mute on error.</p> <p>11 - Play as is. No error concealment.</p>	
4	<p>De-Emphasis Enable</p> <p>When this bit is set to 1, it enables the de-emphasis output.</p>	
3	<p>PCM Enable</p> <p>When this bit is set to 1, it enables the audio decoder to play the PCM samples. The PCM data is expected to come in through the audio compressed data port. The rate buffer must be purged and reset before the data can be sent. This bit must be 0 to decode the compressed data.</p>	
2	<p>Mute Audio</p> <p>This bit is used to control the audio output. When the bit is set to 1, the audio decoder will output zeros. When the bit reset to 0, the audio output will resume. While the output is in mute, the audio decoding continues.</p>	
1	<p>Start DSP</p> <p>This bit controls the DSP execution. When the bit is set to 1, the DSP will start execution. When it is set to 0, the DSP execution will stop.</p>	
0	<p>Start Audio Decode</p> <p>This bit controls the audio decoding process. When it is set to 1, the audio decoding process will start. When it is set to 0, the decoding process will stop at the frame boundary.</p>	

BEEP_CTL		
Address = 1E	Type = RW	Reset = 0000
Bits	Function	
15 - 13	<p><b>Sample Rate</b></p> <p>These three bits are set to the sampling rate of the PCM data to be played.</p> <p>000 - 22.05 KHz  001 - 24 KHz  010 - 16 KHz  100 - 44.1 KHz  101 - 48 KHz  110 - 32 KHz</p>	
12 - 8	<p><b>Beep Duration</b></p> <p>These five bits specify the length of time that the pre-generated tone in units of 1/10 second is to be played. The maximum duration is 3.1 seconds. A zero means no tone to be played. A non-zero value indicates a tone to be played with that duration. The value is automatically reset to zero after the tone has begun.</p>	
7	<p><b>Mono PCM</b></p> <p>When this bit is set to 1, the mono PCM audio data is played. When it is 0, the stereo PCM audio data is played</p>	
6	<p><b>Reserved</b></p>	
5	<p><b>Suppress STC Load</b></p> <p>When this bit is set to 1, the audio decoder will not load the audio PTS into the STC. This bit is effective only in either Audio Master Sync or Disable Sync mode.</p>	
4	<p><b>Reset Audio Decoder</b></p> <p>When this bit is set to 1, the audio decoder will be reset. The audio decoder must be stopped first by setting the Start Audio Decode bit to 0 before this bit is set to 1.</p>	
3	<p><b>Audio Elementary Stream</b></p> <p>When this bit is set to 1, the audio elementary stream is expected. When it is 0, the audio PES input is expected. When the audio elementary stream is the input, the audio/video synchronization is not available.</p>	
2 - 0	<p><b>Beep Attenuation</b></p> <p>These three bits are used to attenuate the beep tone. A value of 0 means no attenuation.</p>	

DISP_MODE		
Address = 20	Type = RW	Reset = 0000
Bits	Function	
15 - 13	<p>Sync Mode</p> <p>These three bits are used to set the display sync control mode. The display will not acquire sync until the host sets a valid sync mode.</p> <p>000 - no sync (reset state)</p> <p>001 - HSC = Composite Blanking; VSC = Field ID</p> <p>010 - HSC = H; VSC = V</p> <p>011 - HSC = Hsync; VSC = Vsync, sync on trailing edge.</p> <p>111 - HSC = Hsync; VSC = Vsync, sync on leading edge.</p>	
12	<p>HSC Polarity</p> <p>This bit defines the polarity of HSC. When it is set to 0, HSC is active low. When it is set to 1, HSC is active high.</p>	
11	<p>VSC Polarity</p> <p>This bit defines the polarity of VSC. When it is set to 0, VSC is active low. When it is set to 1, VSC is active high.</p>	
10	<p>PAL Mode</p> <p>This bit must be set to 1 for PAL operation. For NTSC, this bit is set to 0.</p>	
9	<p>Transparency Gate Polarity</p> <p>This bit defines the polarity of the Transparency Gate output. When it is set to 0, the output is active high. When it is set to 1, the output is active low.</p>	
8	<p>Enable 8-bit Display Output</p> <p>When this bit is set to 1 the display output is enabled for 8-bit mode.</p>	
7	<p>Disable YCbCr Clipping</p> <p>When this bit is set to 1, the YCbCr clipping is disabled. When the bit is 0, the pixel value is clipped to be within the range of 1 to 254.</p>	
6	<p>Expansion Override</p> <p>When this bit is set to 1, the expansion control bits, bits 4-1 of this register are used to control the image expansion. If it is 0, the expansion control bits are ignored. The bit stream will control the image expansion.</p>	
5	<p>Single Field Mode</p> <p>When this bit is set to 1, it enables the single field mode which outputs the same field for both field 1 and field 2 times.</p>	

## Programming Reference

DISP_MODE		
Address = 20	Type = RW	Reset = 0000
Bits	Function	
4-1	Expansion Control Bits 0000 - no expansion 0001 - 2 to 3 horizontal expansion 0100 - 3 to 4 horizontal expansion 0111 - 9 to 16 horizontal expansion 1000 - 3 to 8 horizontal expansion 1001 - 3 to 8 horizontal and 1 to 2 vertical expansion 1010 - 1 to 2 horizontal expansion 1011 - 1 to 2 horizontal and 1 to 2 vertical expansion	
0	Enable OSD When this bit is set to 1 the OSD function is enabled.	

DISP_DLY		
Address = 21	Type = RW	Reset = 0000
Bits	Function	
15 - 9	Horizontal Delay These 8 bits are used to define the delay from the time that the H-sync signal becomes active to the time of outputting pixels. The count is in units of 13.5MHz clock cycles.	
7 - 0	Vertical Delay These 8 bits are used to define the delay from the time that the V-sync signal becomes active to the first line to be displayed. The count is in units of lines.	

VBI_CTL		
Address = 22	Type = RW	Reset = 00-0
Bits	Function	
15 - 13	Reserved	



VBI_CTL		
Address = 22	Type = RW	Reset = 00-0
Bits	Function	
12 - 8	<b>VBI Size</b> These 5 bits specify the number of VBI lines in each of the two VBI buffers. When the buffer is allocated for the VBI data, each line will reserve 720 bytes of chrominance data and 720 bytes of luminance data. If the buffer is set up for teletext, one line of space, 1440 bytes, is sufficient. This value is used during initialization.	
7	Reserved	
6	<b>Teletext Bit Reverse</b> When this bit is set to 1, the bit order of each teletext byte is reversed. When the bit is 0, no reversal is performed.	
5	<b>VBI Select</b> This bit selects the VBI operation. 0 - VBI Teletext 1 - VBI Data	
4 - 0	<b>VBI Line Count</b> These 5 bits are used to define the number of VBI lines to be outputted on the display output bus. It must be defined such that the active video immediately follows the VBI output, i.e. no gap between the VBI and the active video. When the line count is 0, no VBI data is transmitted. If the teletext has different number of lines for the two fields, the field having the earlier line number determines the VBI line count.	

DISP_LBOR		
Address = 23	Type = RW	Reset = -000
Bits	Function	
15 - 10	Reserved.	
9 - 0	<b>Left Border</b> These 10 bits determine the left border of the display. The count is in number of pixels. This register is used to move a small decoded picture to the right of the screen.	

DISP_TBOR		
Address = 24	Type = RW	Reset = -000
Bits	Function	
15 - 10	Reserved.	
9 - 0	<p>Top Border</p> <p>These 10 bits determine the top border of the display. The count is in number of lines. This register is used to move a small decoded picture to the bottom of the screen. The line count cannot exceed the difference of the vertical dimension of the screen and the vertical dimension of the picture.</p>	

OSD_LINK_ADDR		
Address = 29	Type = RW	Reset = 0000
Bits	Function	
15 - 0	<p>OSD Link Start Address</p> <p>This register specifies the starting address of a linked list for OSD. It is in units of 32 bytes.</p>	

OSD_SIZE		
Address = 29	Type = RW	Reset = 0000
Bits	Function	
15 - 0	<p>OSD Buffer Size</p> <p>This register is used to specify the OSD buffer size in the DRAM. The value is in units of 32 bytes.</p>	

RB_THRESHOLD		
Address = 2B	Type = RW	Reset = 0000
Bits	Function	
15 - 8	<b>Video Rate Buffer Threshold</b> These 8 bits specify the video rate buffer threshold such that when the compressed data falls below this value, an interrupt is generated. The value is in 2KB units. A value of zero will not generate an interrupt.	
7 - 0	<b>Audio Rate Buffer Threshold</b> These 8 bits specify the audio rate buffer threshold such that when the compressed data falls below this value, an interrupt is generated. The value is in 64B units. A value of zero will not generate an interrupt.	

INFC_CTL		
Address = 2C	Type = RW	Reset = --00
Bits	Function	
15 - 7	Reserved	
6	<b>Audio Output Mode</b> When this bit is set to 0, the audio output is in default format. When this bit is set to 1, the audio output is in I <sup>2</sup> S format.	
5	<b>Error Signal Polarity</b> When this bit is set to 0, the error signal is active low. When this bit is set to 1, the error signal is active high.	
4	<b>Video Serial Clock Polarity</b> When this bit is set to 0, the rising edge is used to latch data. When this bit is set to 1, the falling edge is used to latch data.	
3	<b>Audio Serial Clock/Audio Strobe Polarity</b> When this bit is set to 0, the rising edge is used to latch data. When this bit is set to 1, the falling edge is used to latch data.	
2	<b>Compressed Data Strobe Polarity</b> When this bit is set to 0, the rising edge is used to latch data. When this bit is set to 1, the falling edge is used to latch data.	

## Programming Reference

INFC_CTL		
Address = 2C	Type = RW	Reset = --00
Bits	Function	
1	Video Request Polarity When this bit is set to 0, the video request signal is active low. When this bit is set to 1, the video request signal is active high.	
0	Audio Request Polarity When this bit is set to 0, the audio request signal is active low. When this bit is set to 1, the audio request signal is active high.	

FRNT_ATTEN		
Address = 2D	Type = RW	Reset = 0000
Bits	Function	
15 - 14	Reserved	
13 - 8	Front Left Channel Attenuation These 6 bits are used to attenuate the front left channel volume. The maximum attenuation is all 1's and the minimum attenuation is all 0's.	
7-6	Reserved	
5 - 0	Front Right Channel Attenuation These 6 bits are used to attenuate the front right channel volume. The maximum attenuation is all 1's and the minimum attenuation is all 0's.	

WR_PROT		
Address = 35	Type = RW	Reset = --00
Bits	Function	
15 - 1	Reserved	
0	I-Store Write Protect This bit is used to protect the I-Store from accidental changes. This bit must be set to 1 before the I-store can be loaded.	

BLOCK_SIZE		
Address = 3B	Type = RW	Reset =0000
Bits	Function	
15 - 0	<p><b>Block Size</b></p> <p>This register is used to indicate the size of the block to be accessed in units of two bytes. The value is initially set by the host and is updated while the block copy, block fill or block transfer is in progress. When it reaches to 0, an interrupt is raised. If the block size is set to 0, a size of 128K bytes is assumed.</p>	

SRC_ADDRHI		
Address = 3C	Type = RW	Reset =--00
Bits	Function	
15 - 6	Reserved	
5 - 0	<p><b>DRAM Address 21-16</b></p> <p>These are the high order bits of the DRAM address for the source data of a Block Copy or Block Transfer From DRAM operation.</p>	

SRC_ADDRLO		
Address = 3D	Type = RW	Reset =0000
Bits	Function	
15 - 0	<p><b>DRAM Address 15-0</b></p> <p>These are the low order bits of the DRAM address for the source data of a Block Copy or Block Transfer From DRAM operation. The address is incremented by hardware to indicate the next byte of data to be sourced. The low order bit must be zero.</p>	

HOST_INT1		
Address = 3E		Type = R      Reset = 0000
Bits	Function	
15	Reserved	
14	Freeze Frame Status Flag This bit, when set to 1, indicates that Freeze Frame mode is active. It is reset to 0 when Freeze Frame mode is reset in the decoder. This flag is updated with each Picture Start Interrupt, as posted in HOST_INT (10).	
13 - 7	Reserved	
6	Audio Ancillary Data This bit, when 1, indicates that the audio ancillary data is in DRAM. This bit is reset to 0 when the low byte of this register is read.	
5 - 0	Reserved	

MASK1		
Address = 3F		Type = RW      Reset = 0000
Bits	Function	
15	Reserved	
14	Freeze Frame Status This bit, when 1, allows the Freeze Frame Status interrupt.	
13 - 7	Reserved	
6	Audio Ancillary Data This bit, when 1, allows the Audio Ancillary Data interrupt.	
5 - 0	Reserved	

## 5.2 Host Driven Operations

The host operations provides communications to the decoder.

### 5.2.1 Initialization

Initialization of the MPEGCD20 requires a specific sequence of steps, as detailed in this section. The steps must be performed in the sequence shown:

## 1. Reset Decoder

1. Set Chip Reset in CHIP\_CONTROL (Bit 2 Register 00)
2. Wait 1 msec
3. CHIP\_CONTROL = 0 (Register 00)

## 2. Initialize Host Registers

These are hardware configuration dependant and must be initialized to match your system requirements.

1. CHIP\_MODE =

**Note:** Use the following decoder initialization steps for PAL operation when the synchronization mode is set to Hsync/Vsync with leading edge sync. Set the synchronization mode by setting DISP\_MODE register, Sync Mode bits to 111, then wait at least 40 milliseconds before setting the Display Mode to PAL (DISP\_MODE register, PAL Mode = 1).

2. DISP\_MODE =
3. AUD\_CTL =
4. OSD\_MODE =
5. OSD\_SIZE =
6. VBI\_CTL =

## 3. Load Video Control Store

1. Set I-Store Write Protect bit in WR\_PROT (Bit 0 Register 35)
2. PROC\_IADDR = 0
3. Put the Video Control Store file data into a buffer.
  - Read 16-bit word from the buffer
  - PROC\_IDATA = 16-bit word read from buffer
    - First byte in Video Control file goes to PROC\_IDATA <15-8>
    - Second byte in Video Control file goes to PROC\_IDATA <7-0>
    - Depending on Host Processor byte swapping may have to occur
  - Do the two sub-steps above for a total size of the Video Control Store file
4. PROC\_IADDR = 0
5. Reset I-Store Write Protect bit in WR\_PROT (Bit 0 Register 35)

## 4. Load Audio Control Store

1. Set I-Store Write Protect bit in WR\_PROT (Bit 0 Register 35)
2. AUD\_IADDR = 0
3. Put the Audio Control Store file data into a buffer.
  - Read a single byte from the buffer
  - AUD\_IDATA = 8-bit byte
  - Do the two sub-steps above for a total of size of the Audio Control Store file
4. AUD\_IADDR = 0
5. Reset I-Store Write Protect bit in WR\_PROT (Bit 0 Register 35)

## 5. Set Start Video Processor bit in CHIP\_CONTROL (Bit 2 Register 01)

## 6. Issue Host Command

## Programming Reference

This configuration setting is system dependent.

1. CONFIGURE DECODER =
7. Set Start DSP bit in AUD\_CTL (Bit 2 Register 1D)
8. Initialize Rate Buffer with at least 32 bytes of data
  1. FIFO = data
  2. AUD\_FIFO = data
9. Initialize the STC

This may or may not be required dependent on which synchronization mode you use.

1. SYNC\_STC2 = STC <32-30>
2. SYNC\_STC1 = STC <29-15>
3. SYNC\_STC0 = STC <14-00>
10. Set START\_DECODE bit in CHIP\_CONTROL (Bit 0 register 00)
11. Set START\_AUD\_DECODE bit in AUD\_CTL (Bit 0 register 1D)

### 5.2.2 Resets

There are two types of resets defined for the MPEGCD20. These are the *POR* (Power On Reset) and the *Chip Reset*. The main difference between these two resets is that the chip reset can be initiated by the host processor through bit 2, chip reset bit, of the Chip Control register. This chip reset does not affect Instruction Storage and will not reset the chip reset bit. The POR is initiated through the RESET pin when power is turned on. This resets all of the latches. Instruction Storage must be reloaded after a POR. The resets must be active for at least 5 microseconds.

### 5.2.3 Video FIFO Buffer Access

The compressed video data comes into the MPEGCD20 through a data FIFO that is addressed directly through the host address and data busses, or through the serial Video Data port. This data is in MPEG format either packetized or as an elementary stream. The video data FIFO is a temporary storage area between the host and the rate buffer area in DRAM. The incoming data is stored in the video data FIFO, then sent to the rate buffer area in DRAM in 32 byte blocks.

The video data FIFO interface is *write only*. A data write operation to the video data FIFO is completed in one cycle, with no wait states inserted. Data at the end of the incoming bit stream must be padded with zeros to ensure that the last 32 bytes of data are processed through the video data FIFO and sent to the rate buffer area in DRAM.

The host must ensure that the video data FIFO is not overrun. This can be accomplished by monitoring the status of the video data FIFO through the Video\_FIFO\_Status register or the external Video Request signal. The Video Request signal is de-asserted when the video data FIFO is filled to the programmed margin. The fullness of the rate buffer in DRAM can be accessed by reading the video FIFO register. The value indicates the number of 32-byte blocks of compressed data in the DRAM.



There are two interrupt bits to indicate the condition of the video rate buffer, Video Rate Buffer Threshold interrupt and Video Rate Buffer Overflow interrupt. A value can be set into the RB Threshold register. If the rate buffer falls below this value, the Video Rate Buffer Threshold interrupt is asserted. If the rate buffer is full, the Video Rate Buffer Overflow interrupt is asserted.

## 5.2.4 Audio FIFO Buffer Access

The Audio data comes into the MPEGCD20 through a data FIFO that can be addressed directly through the host address and data busses, or through the serial Audio Data port. This data can be in PCM format or it can be MPEG compressed Audio, either packetized or as an elementary stream. The data is placed in the on-chip data FIFO to allow the MSP to strip off packet level information before transferring the audio data to the audio data FIFO located in DRAM.

The data is moved to DRAM in 8 byte blocks. Data at the end of the transfer must be padded with zeros to ensure that the last 8 bytes of data are transferred to DRAM.

The host must ensure that the audio data FIFO is not overrun. This can be accomplished by monitoring the status of the audio data FIFO using the Audio Buffer Full bit in the AUD\_FIFO register, or by monitoring the Audio Data Request signal. The amount of data in the audio data buffer in DRAM can be determined by reading the AUD\_FIFO register. The low order byte of this register indicates the number of 64-byte blocks of data in the buffer.

The Audio Rate Buffer Threshold interrupt may be used to indicate a buffer underflow condition. The threshold value can be set in the RB Threshold register.

## 5.2.5 Maskable Interrupts to Host

The host can set up the decoder to allow the interrupts to occur if needed. There are eight interrupt bits which can be micro-programmed to interrupt the host. These eight bits have eight corresponding mask bits to control the individual interrupts. Currently there are eight interrupts defined in the Register Definition section of this document. The other eight interrupts in the low order byte are hard wired interrupts.

When the interrupt signal is active low, it will stay active until the host reads the interrupt register. Once the register is read, the interrupt bits will be reset and the interrupt signal will be de-activated.

The interrupts currently set via the on-chip microcode are:

- HOST\_INT register
  - Bit 15 - Sequence Start detected
  - Bit 14 - Sequence Error detected
  - Bit 13 - Sequence End detected
  - Bit 12 - GOP/SMPTE detected
  - Bit 11 - Picture Skipped
  - Bit 10 - Picture Synchronization completed
  - Bit 09 - Picture Resolution changed

## Programming Reference

- Bit 08 - User Data received
- HOST\_INT1 register
  - Bit 14 - Freeze Frame Status Flag
  - Bit 06 - Audio Ancillary Data

The hard wired interrupts are listed below.

- HOST\_INT register
  - Bit 07 - VBI Start
  - Bit 06 - Video Start
  - Bit 05 - First Field Video Start
  - Bit 04 - Block Move Complete
  - Bit 03 - Audio Rate Buffer Threshold
  - Bit 02 - Video Rate Buffer Threshold
  - Bit 01 - Video Rate Buffer Overflow
  - Bit 00 - OSD Data

Some of the above interrupt bits also have corresponding data fields that are written into the video DRAM for use by the host processor. The following table details these bits and their associated data fields.

**Table 5-1. Video Stream Data Stored on Interrupt**

Interrupt Flag	Data Stored	Data Size (note-1)	DRAM Address in Hex
Sequence Start	h_size	12 bits	0001F4
	pel_aspect_ratio	4 bits	0001F5 (lo-order bits)
	v_size	12 bits	0001F6
	frame_rate	4 bits	0001F7 (lo-order bits)
GOP Start	SMPTE time code	25 bits	0001FC
Picture Synchronization	temporal_reference	10 bits	0001F8
	picture_type	3 bits	0001FA
User Data	flags	8 bits	000000
	user data byte count	8 bits	000001
	user data (maximum of 254 bytes)	254 bytes	000002

**Note:** Each data field is stored on a word boundary (2-bytes). Bit fields less than 16-bits in length will be stored left-justified on a word boundary.

## 5.2.6 User Data Access

User data (private data) from the incoming bit stream can be handled by the MPEGCD20. The user data is read from the video bit stream and written into DRAM by the internal processor. The host processor can then access this data using the DRAM access technique described later in this section. The user data is written to the DRAM at address '000002'. The maximum size of the user data storage is 254 bytes.

## 5.3 Audio Ancillary Data

Ancillary data is the bits remaining in an audio frame after the data used in the audio decode have been extracted. The audio code maintains the audio ancillary data from the last sixteen frames. The header contains the index of the buffer last used. The index is incremented mod 16 for each frame processed. The audio code sends an interrupt to the host each time a buffer and the header were updated.

Each ancillary data buffer contains a one-byte bit count and up to thirty-one bytes of ancillary data. The data is right-justified and padded on the left with zero bits. If there were more than thirty-one bytes of ancillary data in a frame, the bit count is set to 249.

The header is one byte and is at DRAM address 001DFC. The ancillary data buffers start at DRAM address 001E00 and extend to DRAM address 001FFF.

**Table 5-2. Audio Ancillary Data Store in DRAM**

Interrupt Flag.	Data Stored	Data Size	DRAM Address in Hex
Audio Ancillary Data	header	8 bits	001DFC
	bit count buffer 0	8 bits	001E00
	ancillary buffer 0	31 bytes	001E01
	bit count buffer 1	8 bits	001E20
	ancillary buffer 1	31 bytes	001E21
	bit count buffer 2	8 bits	001E40
	ancillary buffer 2	31 bytes	001E41
	bit count buffer 3	8 bits	001E60
	ancillary buffer 3	31 bytes	001E61
	bit count buffer 4	8 bits	001E80
	ancillary buffer 4	31 bytes	001E81
	bit count buffer 5	8 bits	001EA0
	ancillary buffer 5	31 bytes	001EA1

**Table 5-2. Audio Ancillary Data Store in DRAM**

Interrupt Flag.	Data Stored	Data Size	DRAM Address in Hex
	bit count buffer 6 ancillary buffer 6	8 bits 31 bytes	001EC0 001EC1
	bit count buffer 7 ancillary buffer 7	8 bits 31 bytes	001EE0 001EE1
	bit count buffer 8 ancillary buffer 8	8 bits 31 bytes	001F00 001F01
	bit count buffer 9 ancillary buffer 9	8 bits 31 bytes	001F20 001F21
	bit count buffer A ancillary buffer A	8 bits 31 bytes	001F40 001F41
	bit count buffer B ancillary buffer B	8 bits 31 bytes	001F60 001F61
	bit count buffer C ancillary buffer C	8 bits 31 bytes	001F80 001F81
	bit count buffer D ancillary buffer D	8 bits 31 bytes	001FA0 001FA1
	bit count buffer E ancillary buffer E	8 bits 31 bytes	001FC0 001FC1
	bit count buffer F ancillary buffer F	8 bits 31 bytes	001FE0 001FE1

## 5.4 External Memory (DRAM) Access From Host

The external memory (DRAM) is accessed by the host processor indirectly through the register interface. This requires several register ports. There are two registers that set up the DRAM access. One register for the read/write commands and status, and the other register to hold the data being processed to or from the DRAM. Because the DRAM access may require a variable number of operations, the host should monitor the contents of the DRAM Busy bit in the DRAM\_CONTROL register to determine whether the operation is busy or has completed.

All host accesses to DRAM are 16 bits wide. After one data operation has been performed the address register is automatically incremented to the next 16-bit data word. But in 8-bit mode, the 16-bit access are made up of two 8-bit accesses.

The registers and the steps necessary for the host processor to access the IBM MPEG-2 DRAM external memory are shown in the following lists.

<b>DRAM_ADDRHI</b>	This register sets the high order bits 21-16 of the DRAM address to be accessed.
<b>DRAM_ADDRLO</b>	This register sets the low order bits 15-0 of the DRAM address to be accessed.
<b>DRAM_DATA</b>	This register contains the data to be written to or read from DRAM.
<b>DRAM_CONTROL</b>	This register detects DRAM busy status, and enables the DRAM commands.

The host can initiate the following DRAM operations.

- 2-byte DRAM read/write
- Block Copy within DRAM
- Block Transfer to DRAM
- Block Transfer from DRAM
- Block Fill in DRAM
- OSD Write to DRAM
- VBI Write to DRAM

Only one of these DRAM access operations can occur at a time. One must complete before the other can start.

### 5.4.1 2-Byte DRAM Read/Write

This function allows the host to read/write two bytes of data from/to DRAM in 4MB addressing range.

Steps to read DRAM:

1. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
2. Write DRAM\_ADDRHI for high order address bits.
3. Write DRAM\_ADDRLO for low order address bits.
4. Write DRAM\_CONTROL with DRAM Busy bit set to 1 and DRAM Write set to 0.
5. Read and test DRAM Busy bit for 0.
6. Read two bytes from DRAM\_DATA.

Steps to write DRAM:

1. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
2. Write DRAM\_ADDRHI for high order address bits.
3. Write DRAM\_ADDRLO for low order address bits.

4. Write two bytes of data to DRAM\_DATA.
5. Write DRAM\_CONTROL with both DRAM Busy and DRAM Write bits set to 1.

The hardware will reset the DRAM Busy bit when the DRAM operation completes.

### 5.4.2 Block Copy within DRAM

This function allows the host to copy data from one location to another location in DRAM with a specified length in the BLOCK\_SIZE register. The address and length are on 2-byte boundary. The Block Move Complete interrupt will be raised when the copy is complete.

Steps to initiate a Block Copy:

1. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
2. Write DRAM\_ADDRHI for high order bits of destine address.
3. Write DRAM\_ADDRLO for low order bits of destine address.
4. Write SRC\_ADDRHI for high order bits of source address.
5. Write SRC\_ADDRLO for low order bits of source address.
6. Write length to BLOCK\_SIZE.
7. Write DRAM\_CONTROL with both DRAM Busy and Block Copy bits set to 1.

The Block Move Complete interrupt bit will be set and the DRAM Busy bit will be reset when the block copy is completed.

### 5.4.3 Block Transfer to DRAM

This function allows the host to write data to DRAM with a specified length. It eliminates the overhead of testing and setting of the DRAM Busy bit. The data can be written successively at no less than 1.6us per 8 bytes. The DRAM address must be on 8-byte boundary and the length must be a multiple of 8 bytes. If more data than the block size are written, the extra data will be ignored.

Steps to initiate a Block Transfer to DRAM:

1. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
2. Write DRAM\_ADDRHI for high order bits of destine address.
3. Write DRAM\_ADDRLO for low order bits of destine address.
4. Write length to BLOCK\_SIZE.
5. Write DRAM\_CONTROL with the DRAM Busy, Block Transfer and DRAM Write bits set to 1.
6. Write two bytes of data to DRAM\_DATA. Repeat until done.

The Block Move Complete interrupt bit will be set and the DRAM Busy bit will be reset when the block transfer is completed.

### 5.4.4 Block Transfer from DRAM

This function allows the host to read data from DRAM with a specified length. It minimizes the overhead of testing and setting of the DRAM Busy bit. The DRAM address must be on 8-byte boundary and the length must be a multiple of 8 bytes.

Steps to initiate a Block Transfer from DRAM:

1. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
2. Write SRC\_ADDRHI for high order bits of source address.
3. Write SRC\_ADDRLO for low order bits of source address.
4. Write length to BLOCK\_SIZE.
5. Write DRAM\_CONTROL with the DRAM Busy and Block Transfer bits set to 1 and the DRAM Write bit set to 0.
6. Read and test Data Ready bit in DRAM\_CONTROL for 1.
7. Read two bytes from DRAM\_DATA. 8 bytes can be read without retesting the Data Ready bit.

The Block Move Complete interrupt bit will be set and the DRAM Busy bit will be reset when the block transfer is completed.

### 5.4.5 Block Fill in DRAM

This function allows the host to fill DRAM with a data pattern specified by the DRAM\_DATA register and the length by the BLOCK\_SIZE register. The DRAM address and the length are on 2-byte boundary.

Steps to initiate a Block Fill in DRAM:

1. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
2. Write DRAM\_ADDRHI for high order bits of destine address.
3. Write DRAM\_ADDRLO for low order bits of destine address.
4. Write length to BLOCK\_SIZE.
5. Write two bytes of pattern data to DRAM\_DATA.
6. Write DRAM\_CONTROL with the DRAM Busy, and Block Fill bits set to 1.

The Block Move Complete interrupt bit will be set and the DRAM Busy bit will be reset when the Block Fill operation is completed.

### 5.4.6 OSD Write to DRAM

This function allows the host to write OSD data to DRAM in 8-byte blocks. The OSD address which is the offset address of the OSD base address must be on 8-byte boundary. The offset address has a range of 512k bytes. Data must be written at a data rate of no more than 8 bytes per 1.6 $\mu$ s. Partial block must be padded to 8 bytes to be transferred to DRAM.

Steps to write OSD data to DRAM:

1. Assume OSD buffer is allocated by OSD\_SIZE during initialization.

## Programming Reference

1. Write offset address to OSD\_ADDR. Address is automatically incremented when data is written to DRAM.
3. Write two bytes of data to OSD\_DATA. Repeat until done.

Writing OSD data to DRAM can be done by using the Block Transfer to DRAM function. It requires more setup but it can write to anywhere in the 2MB DRAM. The OSD base address must be read from the decoder to generate the absolute address.

### 5.4.7 VBI Write to DRAM

This function allows the host to write data to the VBI buffers. The buffer pointer must be set to the buffer to be written. The DRAM address must be on 2-byte boundary.

Steps to write VBI data to DRAM:

1. Assume VBI buffer is allocated by VBI\_SIZE during initialization.
2. Read and test DRAM Busy bit in DRAM\_CONTROL for 0.
3. Write offset address to DRAM\_ADDRLO. Address is automatically incremented when data is written to DRAM.
4. Write two bytes of data to DRAM\_DATA.
5. Write DRAM\_CONTROL with the DRAM Busy, VBI Access and DRAM Write bits set to 1 and the VBI Pointer set to the desired buffer.

The above procedure does not need to know the base address but it requires to interlock with DRAM on every two bytes of data. The Block Transfer to DRAM function may be used to write data to the VBI buffers. When this method is used, the VBI Point bit in the DRAM\_CONTROL register still needs to be flipped for every field.

Alternate method to write VBI data to DRAM:

1. Assume VBI buffer is allocated by VBI\_SIZE during initialization.
2. Read base address of VBI buffers 0 and 1.
3. Generate the absolute DRAM address.
4. Follow steps for Block Transfer to DRAM.

## 5.5 User Commands

The command buffer passes commands from the host to the internal processor on the MPEGCD20. This operation is accomplished by the Host writing both the command code and the associated parameters into the command buffer. The parameters associated with a command vary from command to command. The command buffer supports a maximum of four parameters for each command. These parameters are handled as a push down stack that is written by the host and read by the internal processor.



## 5.5.1 Command Registers

The following four registers are used by the host processor to communicate with the MPEGCD20.

<b>COMMAND</b>	Contains the command sent from the Host to the decoder.
<b>CMD_ADDR</b>	Addresses the command parameter stack.
<b>CMD_DATA</b>	Writes data to, or reads data from, the parameter stack.
<b>CMD_STAT</b>	Indicates a pending command.

Figure 5-1 shows the bit assignments for the four command registers. The CC and CP bits are defined as:

### CC (Command Chaining)

When this bit is set to 1, it indicates that another command will be sent after this command is completed. The internal processor will continue testing the command pending bit for the next command.

### CP (Command Pending)

When this bit is set to 1, it indicates that a command is currently being processed. After the host processor writes the command and the required parameters, it sets this bit to 1. This bit is reset when the internal processor completes the command.

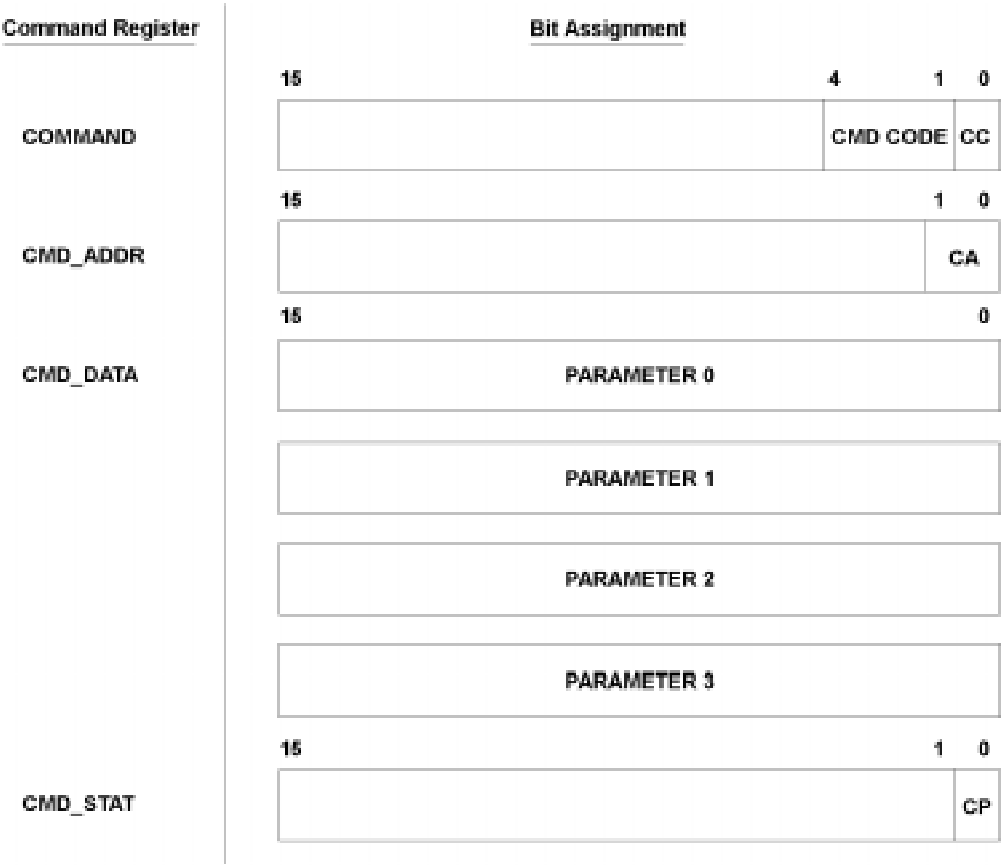


Figure 5-1. Command Buffer

The Host performs the following sequence of operations whenever it needs to send a command to the internal processor:

1. Check the flag (CP) in CMD\_STAT
2. Write command to COMMAND
3. Write address to CMD\_ADDR
4. Write parameters in CMD\_DATA
5. Write CMD\_STAT
6. Test CMD\_STAT
7. Repeat the sequence for the next command

### 5.5.2 List of Commands

The commands used by the host to interact with the MPEGCD20, and the command definitions, are shown in the following tables.

The existing command set controls only video-related functions, except for the *Channel Switch* and *Reset Audio Rate Buffer* commands, which will reset the input buffers for the audio decoder. For all of the commands that affect video operations (Pause, Single Frame, Fast Forward, Slow Motion, and Freeze Frame) it is recommended that the audio be muted during their operation. It is also important to note that the audio and video must be brought back into synchronization when normal play is resumed. Simply executing the *Normal Play* command will not accomplish this. The *Channel Switch* command should be issued before resuming normal play. See Table 5-4 for details regarding the commands.

**Table 5-3. Configuration and Control Commands**

Command code	Function	Parameters
0 0 0 0	Normal Play	
0 0 0 1	Pause	
0 0 1 0	Single Frame	
0 0 1 1	Fast Forward	
0 1 0 0	Slow Motion	Speed (0-7) - 0=normal, 7=slowest
0 1 0 1	Immediate Normal Play	
0 1 1 0	Reset Window	
0 1 1 1	Freeze Frame	
1 0 0 0	Reset Video Rate Buffer	
1 0 0 1	Configure Decoder	Configuration Flags Bit(15) - 0=NTSC, 1=PAL Bit(14) - 0=4:2:0, 1=4:2:2 Bit(13) - 0=2MB, 1=4MB-DRAM In the <i>Low Memory</i> coreloads this flag selects 1MB vs. 2MB of DRAM. Bit(12) - 0=4:3 Monitor, 1=16:9 Monitor Bits(11-0) - reserved (set to '0')
1 0 1 0	Channel Switch	
1 0 1 1	Reset Audio Rate Buffer	
1 1 0 0	Channel Switch Pre-Op	No parameter is required, however, Command Chaining MUST be set ON when this command is issued (see 5.6.1 CD 20 Host Command Execution)
1 1 0 1	Reserved	

Table 5-3. Configuration and Control Commands

Command code	Function	Parameters
1 1 1 0	Set Still Picture Mode	
1 1 1 1	Reserved	

### 5.5.3 Command Descriptions

**Table 5-4. Command Descriptions**

Command	Command Description
Normal Play	Process frames in normal playback mode and speed.
Pause	This command stops the decoding process and freezes the currently displayed frame. The user uses the Normal Play command to return to normal playback of the decoded video stream.
Single frame	Process the decoded video stream one frame at a time. This command sets the Start Decode bit of the CHIP_CONTROL register to 0 to stop the decoding process. The user sets the Start Decode bit to 1 to process the next sequential frame.
Fast forward	This command processes only the I-frames to achieve the fast forward effect assuming the host can deliver the compressed fast enough.
Slow Motion	Process frames at slower than normal playback speed. The desired speed is selected based on a range where '0' is normal speed and '7' is the slowest speed. Slow motion within the decoder is accomplished by repeating frames.
Immediate Normal Play	This command is available for those cases when it is necessary to exit a "non-normal" mode of operation without waiting for any further picture processing (i.e., when it is necessary to exit Freeze Frame mode without waiting for 2 reference frames, as would the normal procedure).
Freeze Frame	Freeze the currently displayed frame, but the bit stream continues to be processed at the normal pace. The user uses the Normal Play command to return to normal playback of the decoded video stream.
Reset Video Rate Buffer	This command resets the on-chip video FIFO and the video rate buffer in DRAM and resumes video decoding on the next sequence start code. Video B-frames will be skipped until 2 reference frames are decoded to prevent video disturbances due to the absence of proper reference frames. Audio decoding is not affected.
Configure Decoder	This command determines buffer size definition based on the decoder system configuration parameter. For more information on bits, See "List of Commands" on page 42.. This command must be issued after a chip reset (and BEFORE loading the rate buffer).

Table 5-4. Command Descriptions

Command	Command Description
Channel Switch	This command resets the audio and video FIFOs and the rate buffers in DRAM and resumes decoding on the next set of reference frames. Video B-frames will be skipped until 2 reference frames are decoded to prevent video disturbances due to the absence of proper reference frames. Audio decoding is not affected.
Reset Audio Rate Buffer	This command resets the audio FIFO and the audio rate buffer in DRAM. Video decoding is not affected.
Host Command Pre-op	<p>This is a "NOP" that is used to proceed forced commands and should always be used with Command Chaining set to ON. This design takes advantage of the Error_Start_Code Interface (described in 5.6.1 CD 20 Host Command Execution) where this Pre-Op command is used to differentiate the Host command request operations from the Error Start Code Recovery cases.</p> <p><b>Note:</b> If Command Chaining is NOT set, the command will initiate an Error Start Code Recovery procedure. <i>Always use with Command chaining on.</i></p>
Set Still Picture Mode	This command is used to set up for still picture display. It must be issued for the MPEGCD20 to properly decode and display a "Still Picture" (i.e., a single I-frame immediately preceded by a Start-Sequence header and immediately followed by an End-of-Sequence header). This mode will remain in effect until a "Normal Play" command is issued.

**Note:** During PAL single frame operation, the B-picture could exhibit a horizontal split unless CHIP\_MODE bit 15 is set to 0.

## 5.6 Command Interface Details

### 5.6.1 CD 20 Host Command Execution

The host command interface has been expanded on the CD20 decoder. Previous implementations allowed host command processing only at picture boundaries, i.e., the host command registers (specifically, the Command-Pending flag) were only inspected at picture-synchronization time. Now, host commands are processed after the following:

1. Chip Reconfiguration
2. Reset Video Rate Buffer
3. Error Start Code Recovery

4. any other chip reset operation that causes the CD20 to search for a new Sequence Start Code header

Under normal CD20 operation, host commands will be executed at these boundary conditions. However, if the data input to CD20 is halted, the video microcode will not be able to proceed and the decoder will not be able to get to one of these boundary conditions. In those cases, there is an alternate interface available that will "force" CD20 to a state where host commands can be accepted and executed. This process takes advantage of the the Sequence-Error-Start-Code interface of CD20 to provide a "clean" hook into the video decode/display process.

**Note:** It is important that the user is aware that if this interface is used, CD20 will resume decoding operations with the next Sequence Start Code header. Also, as part of this "restart", the current PTS value will be discarded and synchronization testing will resume with the *next* PTS value received from the input data stream.

This special host command interface can be implemented with the following steps:

1. load the COMMAND register with x'0007' ("Host Command Pre-op" with Command-Chaining flag set ON) and CMD\_STAT register x'0001' (Command-Pending flag set ON)
2. load Sequence\_Error\_Start\_Code header into input Rate Buffer
3. pad with 128-bytes of zeros to insure the start code is read into the CD20 pipeline
4. wait for Command-Pending flag to be reset (by video microcode)
5. load CMD\_ADDR register with x'0000' and CMD\_DATA with the following values to set Command Parameter 0. This value will direct the video microcode processing as described:
  - x'8000' ==> Go search for the next Sequence Start Code WITHOUT resetting the input Rate Buffers
  - x'4000' ==> Reset the input Video Rate Buffer, then search for the next Sequence Start Code
  - any other value ==> reset BOTH Audio and Video input Rate Buffers, then search for the next Sequence Start Code
6. issue the next required Host command

Commands requiring Parameter values CANNOT be issued at this time, since the parameter register is used to indicate processing as described above. Therefore, the following host commands are NOT allowed at this point:

- Slow Motion
- Configure Decoder
- a 2nd Host Command Pre-Op

## 5.6.2 CD20 Configuration

### 5.6.2.1 Description of Configure-Chip Command

The "Configure Decoder" command is used to reserve the proper buffer sizes in memory based on the configuration selections. See Table 5-4 for a list of the possible options specified in the parameters of this command.

## Programming Reference

**Note:** This command **MUST** be executed when initializing the decoder from a chip-reset state. When the video microcode begins executing from its chip-reset address (i.e. address x'0000'), it will wait for the Configure Command to be issued - with Command Pending set ON - before initializing hardware and proceeding with any decoding functions. Once up and running, CD20 may be Re-Configured, via the Host Command interface, whenever the Command-Pending bit is interrogated.

As part of the configuration procedure, memory allocations are reserved for VBI and OSD buffers. The host registers that define the size of these buffers must be set **BEFORE** the Configure-Decoder command is executed (i.e., before Command-Pending is set ON). The host registers in question are:

- VBI\_CTL (address x'22')
- OSD\_SIZE (address x'2A')

The following buffers are reserved in memory. They are listed in the same order in which they are placed in memory (low to high).

- Pico-DRAM Data Buffer

This is where the video microcode writes video decoding data that can be read by the host processor.

- Audio Buffer

CD20 reserves x'4000' bytes of data for the on-chip Audio processor.

- OSD Buffer

- VBI Buffer

- Frame Buffers

Luminance and Chrominance data is stored for each of 3 Frame Buffers. The size of the Frame Buffers is determined by NTSC vs. PAL and 4:2:0 vs 4:2:2 mode settings in the configuration parameters. The 3rd Frame Buffer is reserved for B-Frame processing and it will NOT be a full-sized Frame Buffer when CD20 is configured for 2MB-PAL-Mode (specified in the CHIP\_MODE host register, address x'01').

- Input Rate Buffer

Whatever space is left in memory after the Frame Buffers will be designated as the Rate Buffer. The "end of memory" is determined by the setting of the 2MB vs. 4MB configuration parameter.

### 5.6.2.2 Execution of Configure-Chip Command

As noted in Table 5-4, Command Chaining is not allowed with this command. However, if the Start-Decode flag is set OFF (bit 0 of the CHIP\_CONTROL register - host register address x'00') before executing this command, the video microcode will loop on this flag upon completion of the Configuration set-up. Part of this loop also checks for new host commands, so this interface can be used to Configure (or Re-configure) CD20 and set it up for another host command before the decode process is resumed.



**Note:** The user should also be aware that after the Configure-Chip command is completed, CD20 will search for the next Sequence\_Start\_Code header before resuming normal decode operations.

### 5.6.2.3 Freeze Frame Support

The "Freeze-Frame" command allows the CD20 to freeze an image on the display while continuing to parse/decode MPEG video and audio data while the STC continues to run. This allows CD20 to 'stop' on an image while remaining 'in sync' with the incoming bit stream. There is a "Freeze-Frame status" flag in bit-14 of the host register HOST\_INT1 (address x'3E') that indicates the current Freeze-Frame-mode status of CD20. This flag is set at the same time the Picture-Start interrupt is set (bit 10 of the HOST\_INT register, address x'18'). When the flag is ON, it indicates that Freeze-Frame-mode is active in the CD20 hardware and when it is OFF it indicates Freeze-Frame mode is NOT active.

### 5.6.2.4 Invoking Freeze Frame

**Freeze-Frame Entry** Since B-frames are not stored in their entirety in 2MB-PAL-Mode, CD20 will only freeze on reference frames (i.e., I or P pictures). This means that Freeze-Frame mode may not take effect immediately when the Command-Pending flag is reset. The host processor must interrogate the Freeze-Frame status flag described above to determine the current state of the CD20 hardware.

**Freeze-Frame Exit** In order for CD20 to exit Freeze-Frame and resume normal decoding and displaying operations without disturbing the frozen video image, the decoder must find 2 reference frames. This insures that all subsequent frames will be decoded and displayed correctly without artifacts. As a result, the Freeze-Frame status may not change immediately after the Command-Pending flag is reset following a Normal-Play command. Once again, the host processor must interrogate the Freeze-Frame status flag described above to determine the current state of the CD20 hardware.

### 5.6.3 Normal Play Support

The Normal-Play command will reset any special modes that have been programmed into the CD20. The following modes will be reset at the completion of the Normal-Play command (i.e., when the Command Pending bit is reset):

- Pause
- Single Frame
- Fast Forward
- Slow Motion
- No Pan-Scan (may also be called Centered-Pan-Scan)
- Freeze Frame
- Still Picture (may also called Sample Play)

Resetting of Freeze-Frame mode does not occur immediately, as described in 5.6.2.3 Freeze Frame Support.

**Note:** There is no need to execute the Normal-Play command after initial power-up. The chip comes up from the reset state in Normal-Play mode.

### 5.6.3.1 Immediate Normal Play

This command is available to force an immediate reset of all special modes, including Freeze-Frame.

**Note:** Use of this command should be reserved for special cases, since there may be some disruption in the video display due to the immediate nature of the mode-reset action.

### 5.6.3.2 Reset Video Rate Buffer Support

There are 2 commands available to clear the input Rate Buffer in order to prepare CD20 for a new input stream.

1. Reset Video Rate Buffer (command code x'08')
2. Reset Audio Rate Buffer (command code x'0B')
3. Channel Switch (command code x'0A') - which resets BOTH video and audio input Rate Buffers.

**Note:** There is a separate hardware interface available to control the Audio Rate Buffer operations without involving the video processor at all. See the description host register BEEP\_CONTROL(4).

It is very important that data is no longer being sent to CD20 when the Rate Buffer Reset command is issued. After completion of this command, CD20 will search for a new Sequence Start Code header to resume normal decoding operations.

### 5.6.4 Still Picture (or Sample Play) Support

This mode of operation requires execution of the Still Picture host command. When CD20 is in this mode (after the command has completed), an End\_of\_Sequence\_Start\_Code header will trigger an internal frame-switch which will display the LAST decoded picture stored in the frame buffers (i.e., the picture immediately preceding the End\_of\_Sequence\_Start\_Code header).

#### 5.6.4.1 Invoking Sample Play

##### Sample-Play Entry

It is important the the video input to CD20 is not active when entering into this mode of operation. To start Sample Play:

1. Reset the Video Rate Buffer (see 5.6.3.2 Reset Video Rate Buffer Support)
2. issue Still Picture command
3. send Sample-play video data sequence, followed by 128 bytes of zeros, into the input Rate Buffer
4. wait for End\_of\_Sequence interrupt
5. repeat last 2 steps for as many Sample-play sequences as desired

##### Sample-Play Exit

When the last Sample-play sequence has been played, exit from this mode and returning to normal decoding is a simple 2-step process:

1. Issue the Normal Play command

2. Resume sending normal video (and audio) data into CD20

## Feeding Sample-Play Data to CD20

A "Sample-play sequence" is any valid MPEG stream that starts with a Sequence\_Start\_Code header and ends with a Sequence\_End\_Start\_Code header. It may contain as little as a single I-frame or any valid combination of I, P, and B pictures, as long as the Sequence and End-Sequence headers are in place. Each Sample-play sequence must be entered into the Rate Buffer independently, since CD20 will resume decoding at the next available Sequence\_Start\_Code. In addition, each sequence must be padded with 128-bytes of zeros to insure the entire stream, including the End\_of\_Sequence\_Start\_Code header, is loaded into the CD20 pipeline.

## 5.6.5 Channel Switching

The MPEGCD20 is capable of seamlessly switching from one input stream to the another, as might be required in a settop implementation when selecting a new program channel. This action does not require the MPEGCD20 to be reset, but some of the late *initialization* steps must be repeated (See "Initialization" on page 30.) in order to re-synchronize the audio and video decoding. The following is a sample of the sequence of steps that could be used to switch programs without any disturbance to the output of the decoder.

1. Issue *Freeze Frame* command (wait for Command Pending to be reset)
2. Reset "Start Video Decoder" (CHIP\_CONTROL(0)) to '0'
3. Halt delivery of input audio and video data from the Transport chip (insure Transport buffers are emptied)
4. Issue *Channel Switch* command to clear input Rate Buffers (wait for Command Pending to be reset)
5. Re-enable the Transport chip to deliver the new Transport Stream
6. Set "Start Video Decoder" (CHIP\_CONTROL(0)) to '1'
7. Set the new System Time Clock (from the new Transport Stream)
8. Load compressed data into the audio and video rate buffers
9. Issue *Normal Play* command (wait for Command Pending to be reset)

After performing the above actions the MPEGCD20 will resume video decoding following the next Video Sequence Start header in the new Transport stream. Audio decoding will resume with the next audio frame containing a PTS value greater than or equal to the current STC value.

Any new programs that require configuration changes (ex. switching from NTSC to PAL or changes to OSD and/or the VBI buffer sizes) will require a new Configure Decoder command before restarting the decoder. To eliminate the disturbance on the screen, the Black Out Display bit of the Chip Control register should be set to 1 before issuing the Configure Decoder command. After the decoding has started, the Black Out Display bit must be set to 0 to show the video.

### 5.6.6 VBI Teletext Data

The MPEGCD20 will handle VBI or teletext data for all VBI lines per field. This procedure explains the sequence of execution for initializing and sending the data to the MPEGCD20.

At chip initialization, the VBI Size, bits 12-8, of the VBI\_CTL register should be set to the number of VBI lines to be supported for allocating enough memory for the VBI data buffering. Each increment of VBI Size will allocate 1440 bytes for each of the two buffers. For teletext, one line allocation is sufficient.

Before setting the the other bits in VBI\_CTL, the VBI0 buffer should be loaded with the appropriate data. The address in DRAM\_ADDRLO should be set to 0 for writing the first line of teletext data. After each 2 bytes written to the DRAM\_DATA register, the DRAM\_CONTROL register must be written to start the DRAM write. The VBI Pointer bit should be set to 0 in order to write the VBI0 buffer. The DRAM Busy bit, VBI Access bit, and DRAM Write bit of the DRAM\_CONTROL register are set to 1. Then check for a zero in the DRAM Busy bit to indicate the write is complete. This should be done after each 2 bytes written. The address will be auto-incremented. If the data being loaded is teletext, the 2 byte run-in code should be written first followed by the teletext data (43 bytes). Subsequent teletext lines should be written into memory at 48 byte offsets from the beginning of the buffer.

It should be noted that the Block Transfer to DRAM function can be used to load data into the VBI buffers as an alternative to the method described here. Whichever method is used, when the loading for that buffer is complete, the VBI Pointer bit of the DRAM\_CONTROL register should be set to 1 to prepare for loading the next buffer. This also signals the VBI logic that the loading of VBI0 buffer is complete and may now be displayed. Note that the loading of a VBI buffer may take longer than one field-display period.

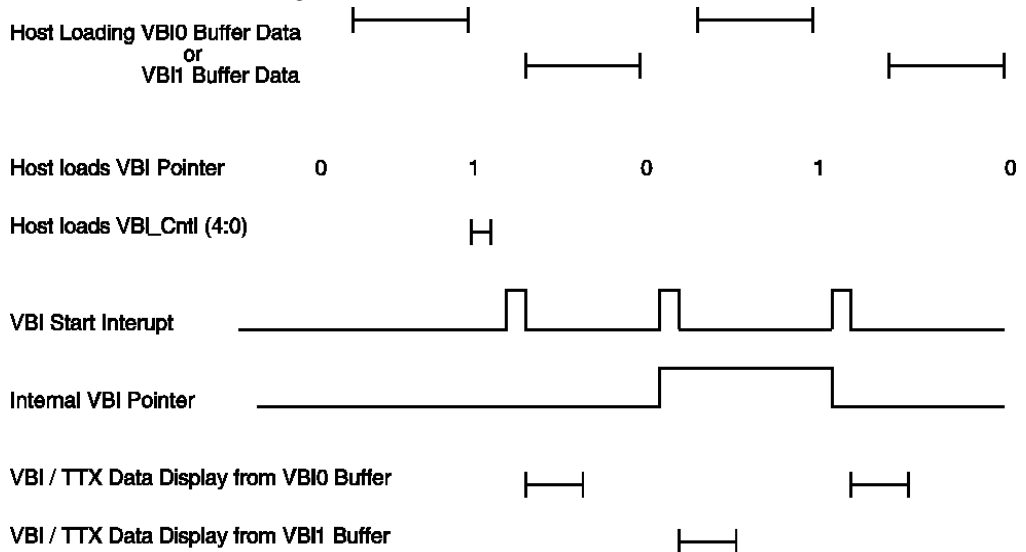
The VBI Start interrupt mask, bit 7, of the MASK register must be enabled in order to get the VBI Start interrupt, bit 7 of the HOST\_INT register.

It is then necessary to set the VBI\_CTL register. The VBI Select Bit of this register should be set to 0 for teletext and the VBI Line Count should contain the number of lines between the first teletext line and the first active video line.

After the receipt of a VBI start interrupt after the completion of the loading of a VBI buffer, the data for the next field can be loaded into VBI1 Buffer as was the first. Note that the VBI Pointer bit of the DRAM\_CONTROL register is already set up for this. After the VBI or teletext data is loaded, as before, the VBI Pointer bit of the DRAM\_CONTROL register should be flipped to prepare for loading the next buffer. This process should be continued as long as there is VBI or teletext data to send in a field. This is shown in Figure 5-2 on page 5-53

Turning on the VBI Line Count of the VBI\_CTL register allows the MPEGCD20 to send out the data when the request signal from the DENC is active. If the data is teletext, it will be expanded bit by bit accordingly on the teletext output pin (TTX DATA).

The DENC should be programmed to handle the requested VBI or teletext data lines per field. When no data is to be merged, the DENC and the VBI Line Count should be set to 0.



**Figure 5-2. Sequence for loading VBI or Teletext Data**



## Chapter 6. Electrical and Physical Specifications

This chapter describes the environmental and electrical specifications.

### 6.1 Environmental and Electrical Specifications

The MPEGCD20 is designed for use in equipment installed in a non-air conditioned building with normal winter heating and sufficient ventilation.

**Table 6-1. Maximum Ratings/Operating Range - Environment**

Parameter	Value
Vdd (supply voltage)(maximum)	3.80 Volts
Vdd (Nominal supply voltage)	3.30 Volts +/- 5%
Power dissipation (range)	1.4W(-CD20), 1.5W(-CD21)
Ambient temperature range (operating conditions)	0 - 70° C
Operating temperature range (chip)	0 - 125° C
Storage temperature range	-65° C to 150° C
<b>Note:</b> (1) The storage temperature range is guaranteed for up to 100 worst case cycles. (2) The operating temperature range is based on ambient air temperature with natural convection.	

**Table 6-2. Current Specifications**

Usage	I <sub>sink</sub> (DC)	I <sub>source</sub> (DC)
RAS/CAS/Host Data	12.0ma	12.0ma
All other drivers	8.0ma	8.0ma
<b>Note:</b> (1) I <sub>source</sub> is measured at 2.4 V. (2) I <sub>sink</sub> is measured at 0.5 V.		

Table 6-3. DC Electrical Characteristics

Usage	MAUL	MPUL	LPUL	MPDL	LPDL	LADL
5 Volt TTL Receiver	5.50V	5.50V	2.00V	0.80V	0.00V	-0.50V
5 Volt TTL Driver	5.50V	3.80V	2.40V	0.50V	0.00V	-0.50V

**Definitions**

**MAUL** - Maximum Allowable Up Level

**MPUL** - Maximum Positive Up Level

**LPUL** - Least Positive Up Level

**MPDL** - Maximum Positive Down Level

**LPDL** - Least Positive Down Level

**LADL** - Least Allowable Down Level

**Note:** (1) The maximum input capacitance is estimated as 6pF. (2) The MAUL of 5.50V for the driver is also valid if the power supply voltage is zero. A maximum of 150uA per driver will sink into the bus. (3) The maximum output capacitance is estimated as 8pF for the driver and 10pF for any bi-directional I/O.

## 6.2 Package Specification

The MPEGCD20/21 is a 28 mm x 28 mm quad flat pack module with 208 pins. The module can be mounted directly on a card, or can be installed using a pluggable socket.



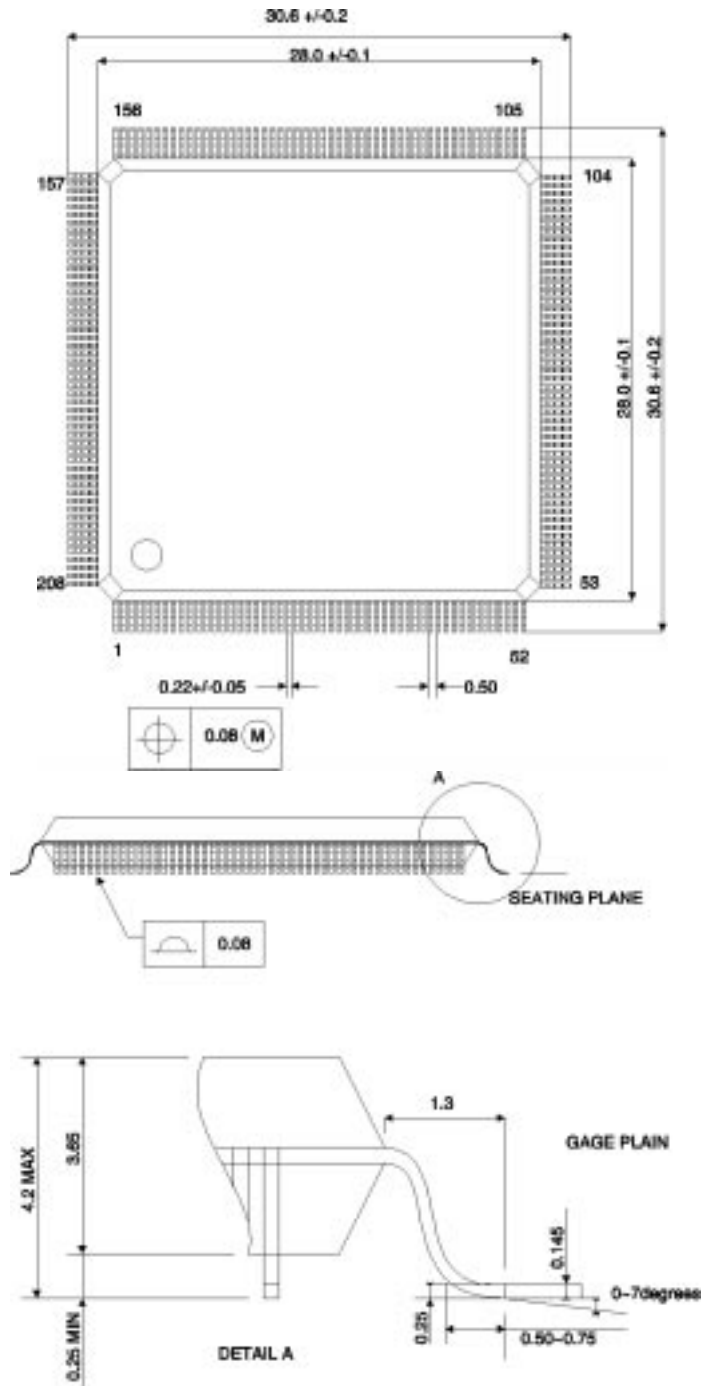


Figure 6-1. Package Specification

## 6.3 Pin Assignments

The 28 mm plastic quad flat pack has 208 pins. The following table shows the signal/pin assignment by pin number sequence.

**Table 6-4. Pin assignments**

Signal name	Mnemonic	Module Pin	Pin Type
GROUND	GND	1	PWR
DRAM_DATA_31	MD31	2	BI-DI
DRAM_DATA_30	MD30	3	BI-DI
DRAM_DATA_29	MD29	4	BI-DI
DRAM_DATA_28	MD28	5	BI-DI
DRAM_DATA_27	MD27	6	BI-DI
DRAM_DATA_26	MD26	7	BI-DI
DRAM_DATA_25	MD25	8	BI-DI
DRAM_DATA_24	MD24	9	BI-DI
DRAM_DATA_23	MD23	10	BI-DI
DRAM_DATA_22	MD22	11	BI-DI
DRAM_DATA_21	MD21	12	BI-DI
VOLTAGE	VDD	13	PWR
DRAM_DATA_20	MD20	14	BI-DI
DRAM_DATA_19	MD19	15	BI-DI
DRAM_DATA_18	MD18	16	BI-DI
DRAM_DATA_17	MD17	17	BI-DI
DRAM_DATA_16	MD16	18	BI-DI
NO CONNECT	N/C	19	N/C
DRAM_DATA_15	MD15	20	BI-DI
DRAM_DATA_14	MD14	21	BI-DI
DRAM_DATA_13	MD13	22	BI-DI
DRAM_DATA_12	MD12	23	BI-DI

Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
NO CONNECT	N/C	24	N/C
GROUND	GND	25	PWR
ANALOG GND	GNDA1	26	PWR
TIE UP	VDD	27	IN
DRAM_DATA_11	MD11	28	BI-DI
TIE DOWN	GND	29	IN
ANALOG VDD	VDDA1	30	PWR
DRAM_DATA_10	MD10	31	BI-DI
DRAM_DATA_9	MD9	32	BI-DI
VOLTAGE	VDD	33	PWR
NO CONNECT	N/C	34	N/C
DRAM_DATA_8	MD8	35	BI-DI
DRAM_DATA_7	MD7	36	BI-DI
DRAM_DATA_6	MD6	37	BI-DI
DRAM_DATA_5	MD5	38	BI-DI
GROUND	GND	39	PWR
DRAM_DATA_4	MD4	40	BI-DI
DRAM_DATA_3	MD3	41	BI-DI
TIE UP	VDD	42	IN
DRAM_DATA_2	MD2	43	BI-DI
DRAM_DATA_1	MD1	44	BI-DI
TIE UP	VDD	45	IN
DRAM_DATA_0	MD0	46	BI-DI
NO CONNECT	N/C	47	N/C
L/R_CLK	LRCLK	48	OUT
AUD_SD_OUT	AUDOUT	49	OUT

Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
NO CONNECT	N/C	50	N/C
AUD_BO_CLK	BOCLK	51	OUT
VOLTAGE	VDD	52	PWR
GROUND	GND	53	PWR
PIXEL_DATA_15	PD15	54	OUT
PIXEL_DATA_14	PD14	55	OUT
PIXEL_DATA_13	PD13	56	OUT
PIXEL_DATA_12	PD12	57	OUT
PIXEL_DATA_11	PD11	58	OUT
PIXEL_DATA_10	PD10	59	OUT
PIXEL_DATA_9	PD9	60	OUT
PIXEL_DATA_8	PD8	61	OUT
PIXEL_DATA_7	PD7	62	OUT
PIXEL_DATA_6	PD6	63	OUT
PIXEL_DATA_5	PD5	64	OUT
VOLTAGE	VDD	65	PWR
PIXEL_DATA_4	PD4	66	OUT
PIXEL_DATA_3	PD3	67	OUT
PIXEL_DATA_2	PD2	68	OUT
PIXEL_DATA_1	PD1	69	OUT
PIXEL_DATA_0	PD0	70	OUT
TTX_DATA_REQ	TTXREQ	71	IN
CREF	CREF	72	OUT
TRANSPARENCY_GATE	TRANGT	73	OUT
POE#	POE	74	IN
PIXEL_CLK	PCLK	75	IN

Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
TTX_DATA	TTX	76	OUT
GROUND	GND	77	PWR
HOST_DATA_15	D15	78	BI-DI
HOST_DATA_14	D14	79	BI-DI
HOST_DATA_13	D13	80	BI-DI
VERT_SYNC_CNTL	VCNTL	81	IN
HORZ_SYNC_CNTL	HCNTL	82	IN
HOST_DATA_12	D12	83	BI-DI
HOST_DATA_11	D11	84	BI-DI
VOLTAGE	VDD	85	PWR
NO CONNECT	N/C	86	N/C
27MHZ_CLK	CLK27	87	BI-DI
NO CONNECT	N/C	88	N/C
HOST_DATA_10	D10	89	BI-DI
HOST_DATA_9	D9	90	BI-DI
GROUND	GND	91	PWR
HOST_DATA_8	D8	92	BI-DI
NO CONNECT	N/C	93	N/C
NO CONNECT	N/C	94	N/C
HOST_DATA_7/SER DATA	D7	95	BI-DI/IN
HOST_DATA_6/SD CLK	D6	96	BI-DI/IN
NO CONNECT	N/C	97	N/C
VIDEO_REQ#	VDREQ	98	OUT
NO CONNECT	N/C	99	N/C
HOST_DATA_5/DTACK MODE	D5	100	BI-DI/IN
HOST_DATA_4/CD STROBE#	D4	101	BI-DI/IN

Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
HOST_DATA_3/RD/WR#	D3	102	BI-DI/IN
HOST_DATA_2/LOW BYTE	D2	103	BI-DI/IN
VOLTAGE	VDD	104	PWR
GROUND	GND	105	PWR
HOST_DATA_1/ACK MODE	D1	106	BI-DI/IN
HOST_DATA_0/WAIT#	D0	107	BI-DI/OUT O.D.
HOST_ADDR_5	A5	108	IN
HOST_ADDR_4	A4	109	IN
OE#	OE	110	IN
TIE DOWN	GND	111	IN
HOST_ADDR_3	A3	112	IN
HOST_ADDR_2	A2	113	IN
TIE UP	VDD	114	IN
RESET#	RESET	115	IN
HOST_ADDR_1	A1	116	IN
VOLTAGE	VDD	117	PWR
HOST_ADDR_0	A0	118	IN
DEEMPHASIS	DEEMP	119	OUT
CHIP_SELECT#	CS	120	IN
WRITE_PULSE#	WP	121	IN
IRQ#	IRQ	122	OUT O.D.
NO CONNECT	N/C	123	N/C
ERROR#	ERR	124	IN
TIE DOWN	GND	125	IN
ANALOG VDD	VDDA2	126	PWR
FILTER_1	FLT1	127	ANALOG

Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
NO CONNECT	N/C	128	N/C
GROUND	GND	129	PWR
FILTER_2	FLT2	130	ANALOG
ANALOG GROUND	GND A2	131	PWR
REXT	REXT	132	ANALOG
ANALOG GROUND	GND A3	133	PWR
FILTER_3	FLT3	134	ANALOG
FILTER_4	FLT4	135	ANALOG
CONFIG (See Note)	GND/VDD	136	IN
VOLTAGE	VDD	137	PWR
NO CONNECT	N/C	138	N/C
TIE UP	VDD	139	IN
MODE16	MODE16	140	IN
AUD_DAC_CLK	DACCLK	141	OUT
AUD_REQ#	ADREQ	142	OUT
GROUND	GND	143	PWR
AUD_BI_CLK	BICLK	144	IN
AUD_SD_IN	AUDIN	145	IN
DRAM_DATA_63	MD63	146	BI-DI
DRAM_DATA_62	MD62	147	BI-DI
DRAM_DATA_61	MD61	148	BI-DI
NO CONNECT	N/C	149	N/C
DRAM_DATA_60	MD60	150	BI-DI
NO CONNECT	N/C	151	N/C
DRAM_DATA_59	MD59	152	BI-DI
DRAM_DATA_58	MD58	153	BI-DI

Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
DRAM_DATA_57	MD57	154	BI-DI
DRAM_DATA_56	MD56	155	BI-DI
VOLTAGE	VDD	156	PWR
GROUND	GND	157	PWR
DRAM_DATA_55	MD55	158	BI-DI
DRAM_DATA_54	MD54	159	BI-DI
DRAM_DATA_53	MD53	160	BI-DI
DRAM_DATA_52	MD52	161	BI-DI
DRAM_ADDR_0	MA0	162	OUT
DRAM_ADDR_1	MA1	163	OUT
DRAM_DATA_51	MD51	164	BI-DI
DRAM_DATA_50	MD50	165	BI-DI
DRAM_ADDR_2	MA2	166	OUT
DRAM_ADDR_3	MA3	167	OUT
DRAM_DATA_49	MD49	168	BI-DI
VOLTAGE	VDD	169	PWR
DRAM_DATA_48	MD48	170	BI-DI
NO CONNECT	N/C	171	N/C
DRAM_ADDR_4	MA4	172	OUT
DRAM_ADDR_5	MA5	173	OUT
RAS#	RAS	174	OUT
NO CONNECT	N/C	175	N/C
CAS0#	CAS0	176	OUT
CAS1#	CAS1	177	OUT
DRAM_ADDR_6	MA6	178	OUT
DRAM_ADDR_7	MA7	179	OUT



Table 6-4. Pin assignments

Signal name	Mnemonic	Module Pin	Pin Type
NO CONNECT	N/C	180	N/C
GROUND	GND	181	PWR
WE0#	WE0	182	OUT
WE1#	WE1	183	OUT
WE2#	WE2	184	OUT
DRAM_ADDR_8	MA8	185	OUT
DRAM_ADDR_9	MA9	186	OUT
WE3#	WE3	187	OUT
DRAM_OE#	MOE	188	OUT
VOLTAGE	VDD	189	PWR
NO CONNECT	N/C	190	N/C
DRAM_DATA_47	MD47	191	BI-DI
DRAM_DATA_46	MD46	192	BI-DI
DRAM_DATA_45	MD45	193	BI-DI
DRAM_DATA_44	MD44	194	BI-DI
GROUND	GND	195	PWR
DRAM_DATA_43	MD43	196	BI-DI
DRAM_DATA_42	MD42	197	BI-DI
DRAM_DATA_41	MD41	198	BI-DI
DRAM_DATA_40	MD40	199	BI-DI
DRAM_DATA_39	MD39	200	BI-DI
DRAM_DATA_38	MD38	201	BI-DI
DRAM_DATA_37	MD37	202	BI-DI
DRAM_DATA_36	MD36	203	BI-DI
DRAM_DATA_35	MD35	204	BI-DI
DRAM_DATA_34	MD34	205	BI-DI

Table 6-4. Pin assignments

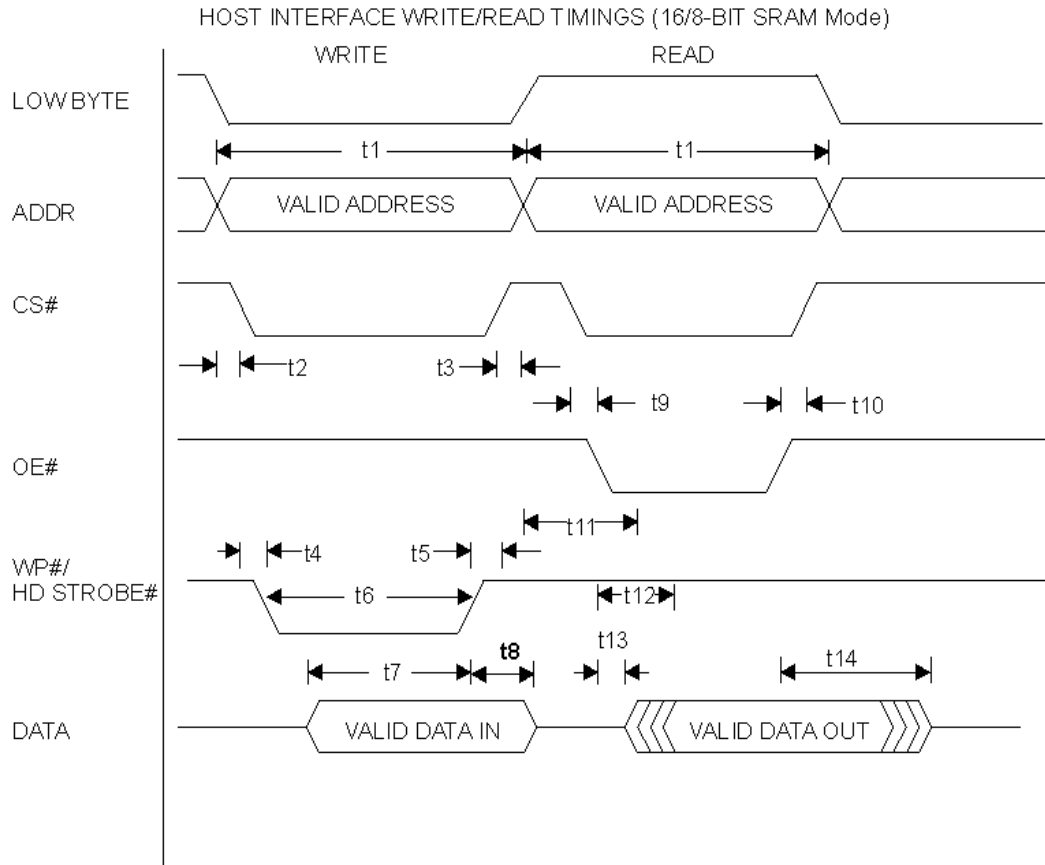
Signal name	Mnemonic	Module Pin	Pin Type
DRAM_DATA_33	MD33	206	BI-DI
DRAM_DATA_32	MD32	207	BI-DI
VOLTAGE	VDD	208	PWR

Note:

Name	Description
BI-DI	Bi Directional
PWR	Power
N/C	No Connect
IN	Input
OUT	Output
O.D.	Open Drain
CONFIG	For MPEGCD20 tie to GND For MPEGCD21 tie to VDD

## 6.4 Timing Diagrams

### 6.4.1 Host Interface Timing Diagrams



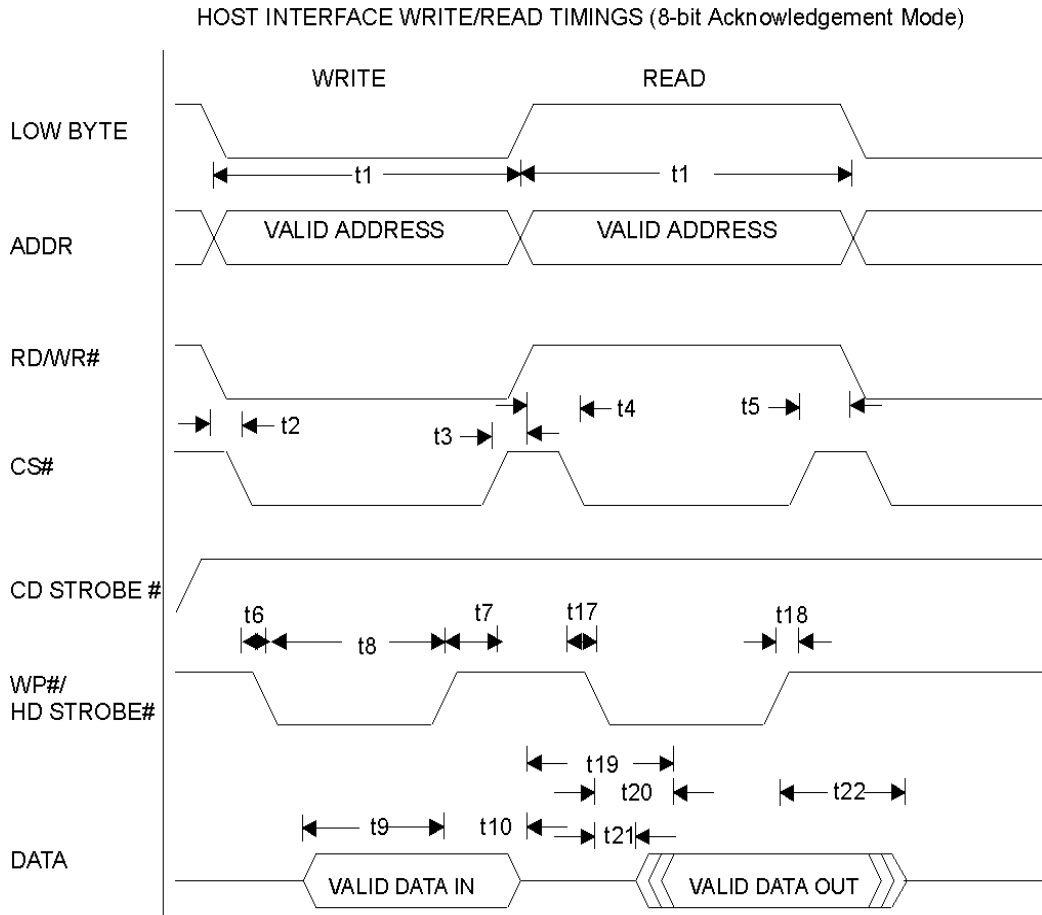
Note: Minimum 74ns must be maintained between rising edges of WP#/HD STROBE#.

Either CS# or ADDR must be changed at the end of the Read cycle.

**Figure 6-2. Host Interface W/R Timings (16/8-bit SRAM Mode)**

**Table 6-5. Host Interface W/R Timings (16/8-bit SRAM Mode)**

Parameter	Symbol	Minimum	Maximum
Read/write cycle	t1	74 ns	-
Address valid to CS# falling edge	t2	0	-
CS# rising edge to address change	t3	0	-
CS# falling edge to WP# falling edge	t4	5 ns	-
WP# rising edge to CS# rising edge	t5	10 ns	-
WP# pulse width	t6	15 ns	-
Write data setup to WP# rising edge	t7	10 ns	-
Write data hold to WP# rising edge	t8	5 ns	-
CS# falling edge to OE# falling edge	t9	0	-
OE# rising edge to CS# rising edge	t10	0	-
Address valid to data valid (access time)	t11	-	25 ns
OE# falling edge to data valid	t12	0	10 ns
OE# falling edge to data bus enabled	t13	0	10 ns
OE# rising edge to data bus tri-stated	t14	0	10 ns

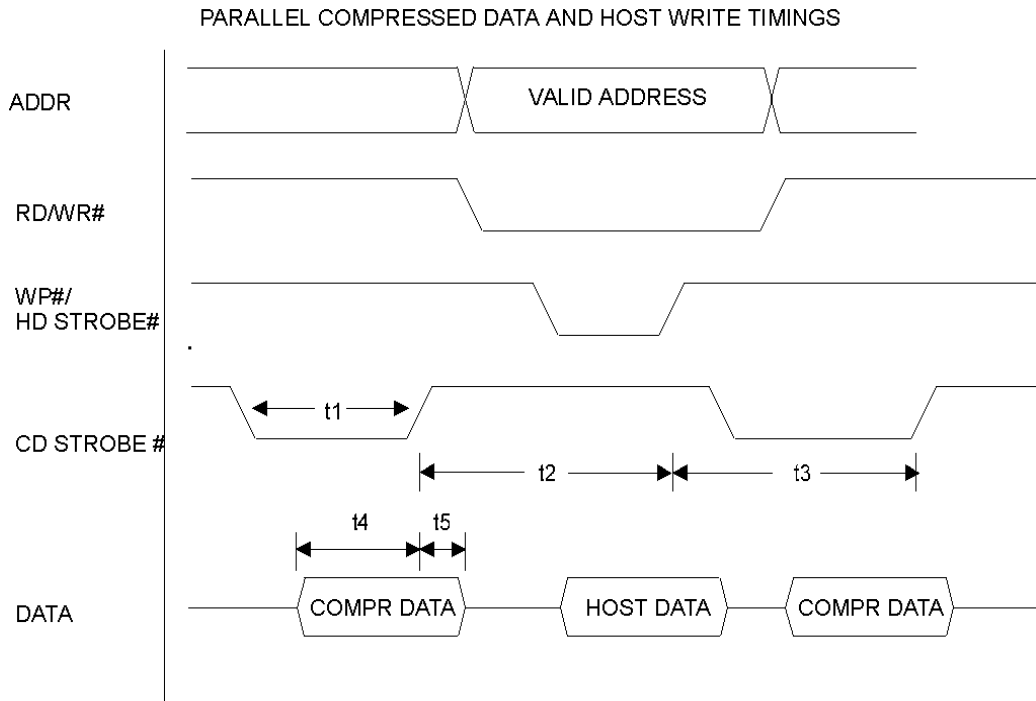


**Note:** (1) WP#/HD STROBE# is used to enable the data bus drivers. (2) Minimum host access cycle time can be 74ns. (3) Minimum 74ns must be maintained between rising edges of WP#/HD STROBE#. (4) CS# can be tied low at all time.

**Figure 6-3. Host Interface W/R Timings (8-bit Acknowledge Mode)**

**Table 6-6. Host Interface W/R Timings (8-bit Acknowledge Mode)**

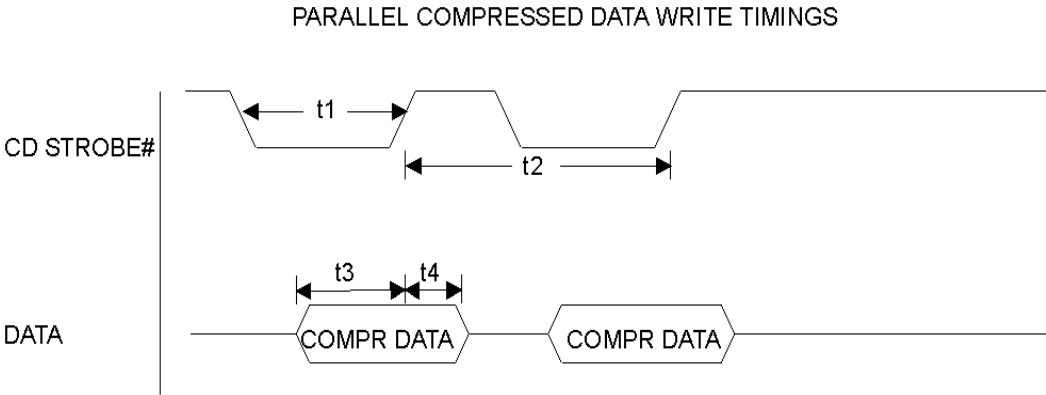
Parameter	Symbol	Minimum	Maximum
Read/write cycle	t1	74 ns	-
Address valid and RD/WR# falling edge to CS# falling edge	t2	0	-
CS# rising edge to address change and RD/WR# rising edge	t3	0	-
Address valid and RD/WR# rising edge to CS# falling edge	t4	0	-
CS# rising edge to address change and RD/WR# falling edge	t5	0	-
CS# falling edge to WP# falling edge	t6	5 ns	-
WP# rising edge to CS# rising edge	t7	10 ns	-
WP# pulse width	t8	15 ns	-
Write data setup to WP# rising edge	t9	10 ns	-
Write data hold to WP# rising edge	t10	5 ns	-
CS# falling edge to WP# falling edge	t17	5 ns	-
WP# rising edge to CS# rising edge	t18	10 ns	-
Address valid to data valid (access time)	t19	-	25 ns
WP# falling edge to data valid	t20	0	10 ns
WP# falling edge to data bus enabled	t21	0	10 ns
WP# rising edge to data bus tri-stated	t22	0	10 ns



**Figure 6-4. Video Compressed Data And Host Write Timings (8-bit Acknowledge Mode)**

**Table 6-7. Video Compressed Data And Host Write Timings (8-bit Acknowledge Mode)**

Parameter	Symbol	Minimum	Maximum
CD STROBE# pulse width	t1	15 ns	-
CD STROBE# rising edge to WP# rising edge	t2	74 ns	-
WP# rising edge to CD STROBE# rising edge	t3	74 ns	-
Compressed data setup to CD STROBE# rising edge	t4	10 ns	-
Compressed data hold to CD STROBE# rising edge	t5	5 ns	-

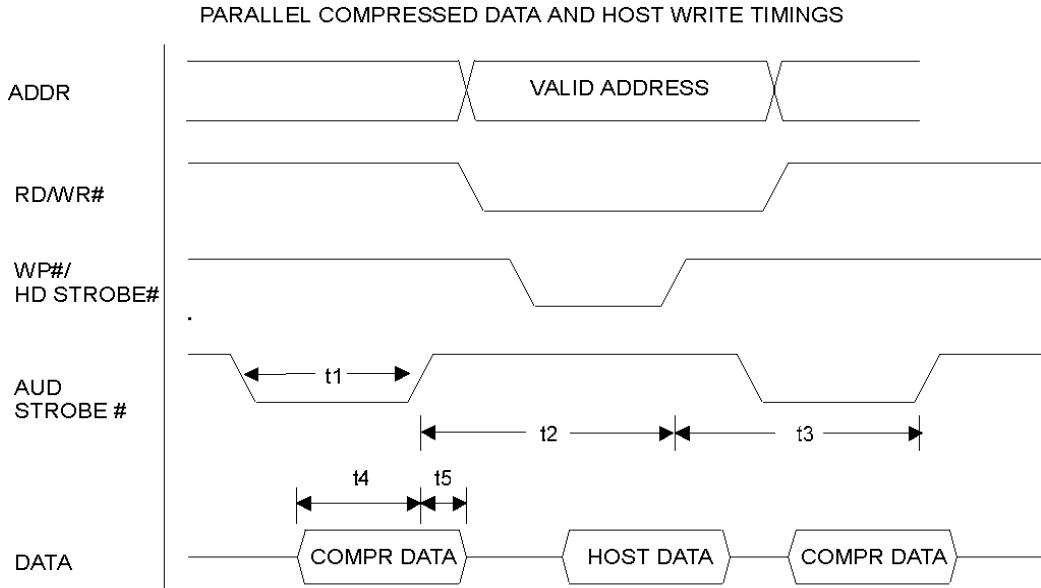


**Figure 6-5. Video Compressed Data Write Timings (8-bit Acknowledge Mode)**

**Table 6-8. Video Compressed Data Write Timings (8-bit Acknowledge Mode)**

Parameter	Symbol	Minimum	Maximum
CD STROBE# pulse width	t1	15 ns	-
CD STROBE# rising edge to CD STROBE# rising edge	t2	74 ns	-
Compressed data setup to CD STROBE# rising edge	t3	10 ns	-
Compressed data hold to CD STROBE# rising edge	t4	5 ns	-

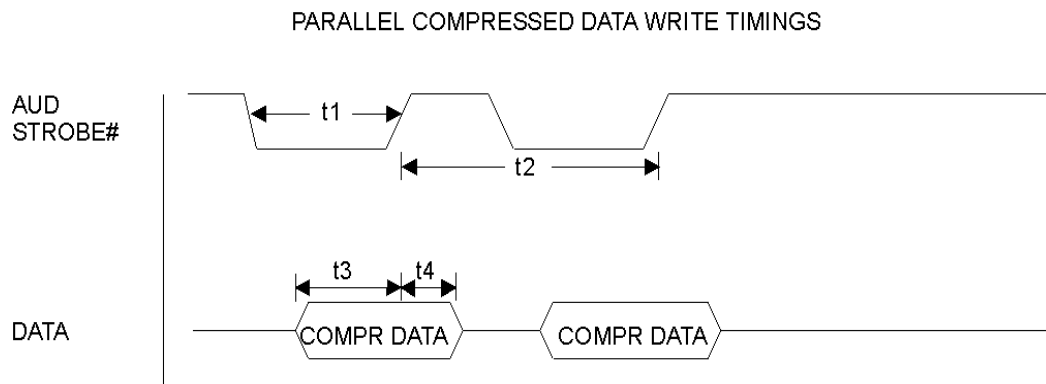




**Figure 6-6. Audio Compressed Data And Host Write Timings (8-bit Acknowledge Mode)**

**Table 6-9. Audio Compressed Data And Host Write Timings (8-bit Acknowledge Mode)**

Parameter	Symbol	Minimum	Maximum
AUD STROBE# pulse width	t1	15 ns	-
AUD STROBE# rising edge to WP# rising edge	t2	74 ns	-
WP# rising edge to AUD STROBE# rising edge	t3	74 ns	-
Compressed data setup to AUD STROBE# rising edge	t4	10 ns	-
Compressed data hold to AUD STROBE# rising edge	t5	5 ns	-



**Figure 6-7. Audio Compressed Data Write Timings (8-bit Acknowledge Mode)**

**Table 6-10. Audio Compressed Data Write Timings (8-bit Acknowledge Mode)**

Parameter	Symbol	Minimum	Maximum
AUD STROBE# pulse width	t1	15 ns	-
AUD STROBE# rising edge to AUD STROBE# rising edge	t2	74 ns	-
Compressed data setup to AUD STROBE# rising edge	t3	10 ns	-
Compressed data hold to AUD STROBE# rising edge	t4	5 ns	-

## SERIAL VIDEO COMPRESSED DATA WRITE TIMINGS WITH BYTE SYNC

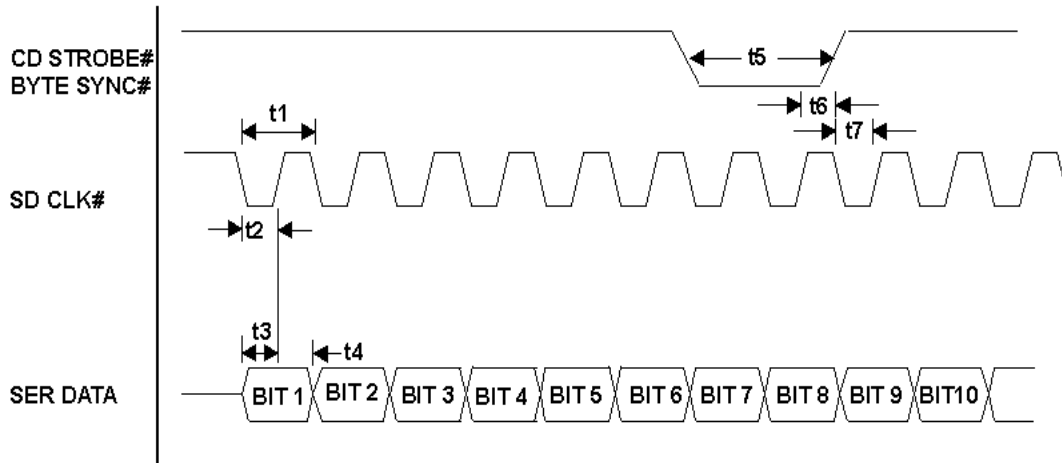


Figure 6-8. Serial Video Compressed Data Write Timings With Byte Sync

Table 6-11. Serial Video Compressed Data Write Timings With Byte Sync

Parameter	Symbol	Minimum	Maximum
SD CLK# cycle	$t_1$	30 ns	-
SD CLK# pulse width	$t_2$	15 ns	-
Serial data setup to SD CLK# rising edge	$t_3$	10 ns	-
Serial data hold to SD CLK# rising edge	$t_4$	5 ns	-
Byte Sync# pulse width	$t_5$	15 ns	100 ns
SD CLK# rising edge to BYTE SYNC# rising edge	$t_6$	10 ns	-
BYTE SYNC# rising edge to SD CLK# rising edge	$t_7$	10 ns	-

SERIAL VIDEO COMPRESSED DATA WRITE TIMINGS WITHOUT BYTE SYNC

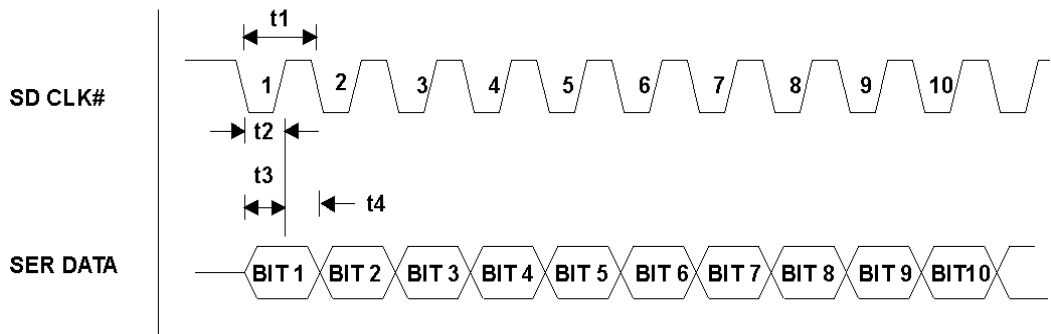


Figure 6-9. Serial Video Compressed Data Write Timings Without Byte Sync

Table 6-12. Serial Video Compressed Data Write Timings Without Byte Sync

Parameter	Symbol	Minimum	Maximum
SD CLK# cycle	t1	30 ns	-
SD CLK# pulse width	t2	15 ns	-
Serial data setup to SD CLK# rising edge	t3	10 ns	-
Serial data hold to SD CLK# rising edge	t4	5 ns	-

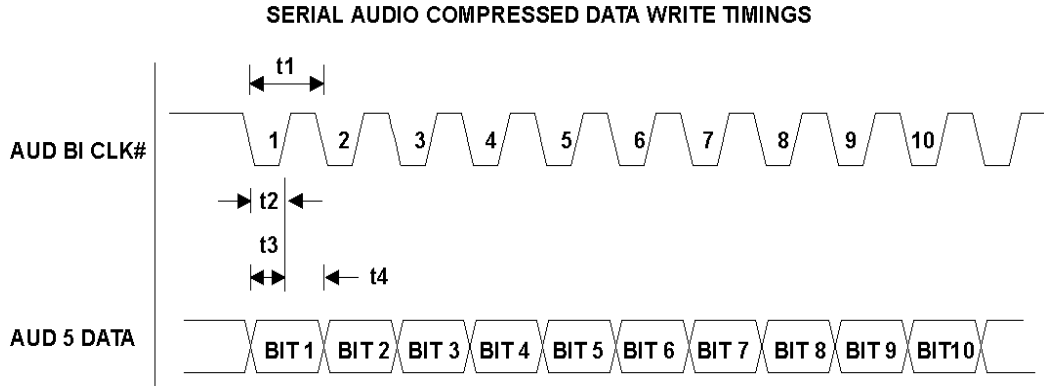


Figure 6-10. Serial Audio Compressed Data Write Timings

Table 6-13. Serial Audio Compressed Data Write Timings

Parameter	Symbol	Minimum	Maximum
AUD BI CLK# cycle	t1	30 ns	-
AUD BI CLK# pulse width	t2	15 ns	-
Serial data setup to AUD BI CLK# rising edge	t3	10 ns	-
Serial data hold to AUD BI CLK# rising edge	t4	5 ns	-

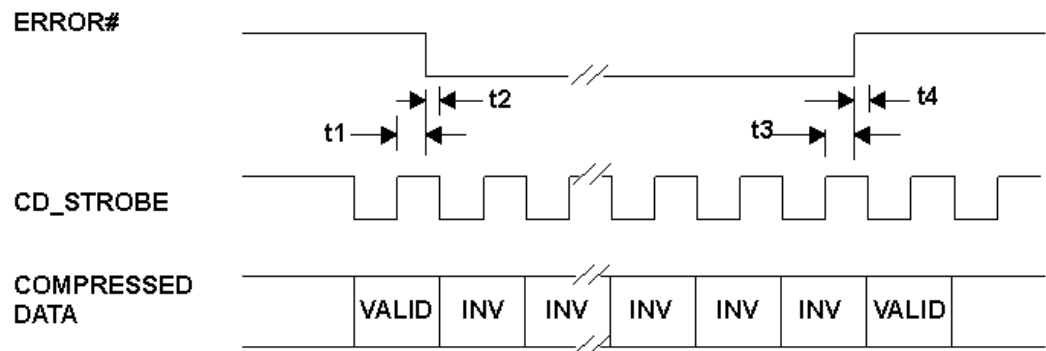


Figure 6-11. ERROR# signal and parallel compressed data timing.

Table 6-14. Error Signal Input to Compressed Data Strobe Timings

Parameter	Symbol	Minimum	Maximum
CD STROBE# rising edge to ERROR# falling edge	t1	5 ns	-
ERROR# falling edge to CD STROBE# falling edge	t2	5 ns	-
CD STROBE# rising edge to ERROR# rising edge	t3	5 ns	-
ERROR# rising edge to CD STROBE# falling edge	t4	5 ns	-

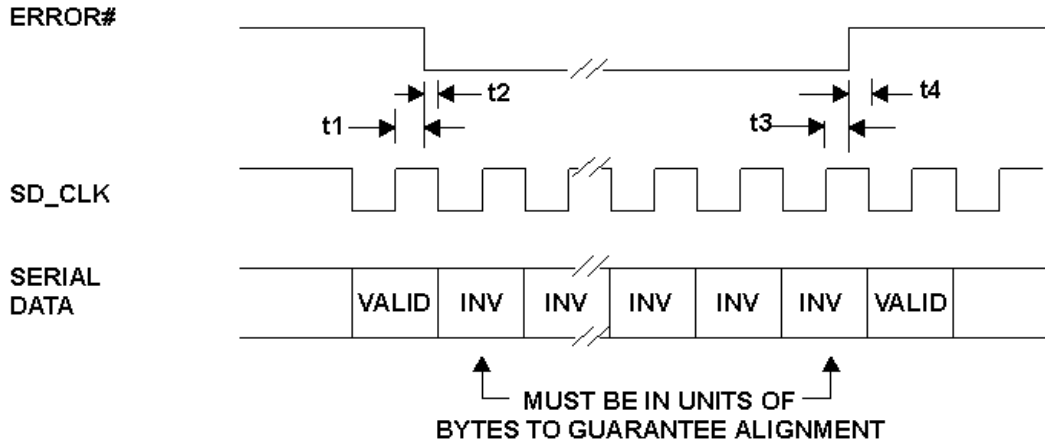


Figure 6-12. ERROR# signal and serial compressed data timing.

Table 6-15. Error Signal Input to Serial Compressed Data Clock Timings

Parameter	Symbol	Minimum	Maximum
SD CLK rising edge to ERROR# falling edge	t1	5 ns	-
ERROR# falling edge to SD CLK falling edge	t2	5 ns	-
SD CLK rising edge to ERROR# rising edge	t3	5 ns	-
ERROR# rising edge to SD CLK falling edge	t4	5 ns	-

## 6.4.2 DRAM Interface Timing Diagrams

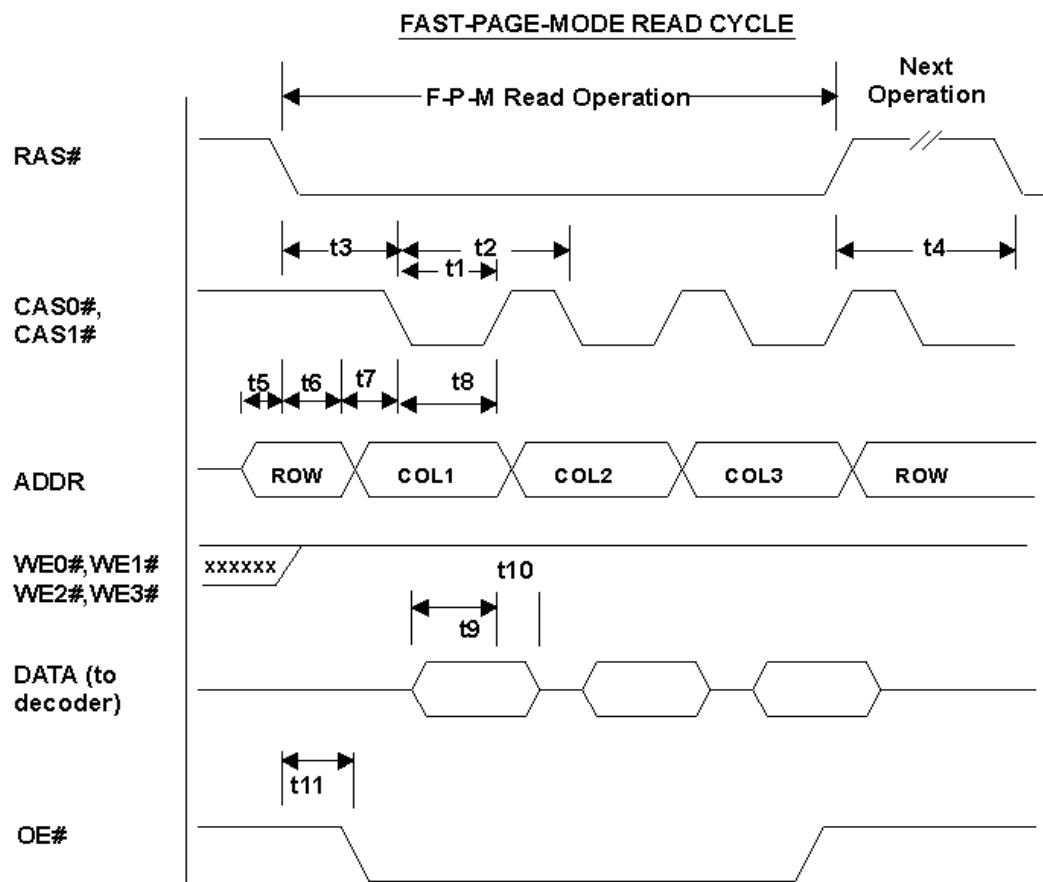


Figure 6-13. DRAM Interface Timing (Fast Page Mode Read Cycle)



	<b>MPEGCD20</b>	<b>MPEGCD21</b>
t1 = t(CAS) =	30 ns Min/33ns Max	25ns Min/27ns Max
t2 = t(PC) =	44ns	37ns
t3 = t(RCD) =	50ns Min	42ns Min
t4 = t(RP) =	53ns Min	46ns Min
t5 = t(ASR) =	4ns Min	4ns Min
t6 = t(RAH) =	22ns Min	18ns Min
t7 = t(ASC) =	15ns Min	12ns Min
t8 = t(CAH) =	20ns Min	18ns Min
t9 = t(DS) =	6ns Min	6ns Min
t10 = t(DH) =	0ns Min	0ns Min
t11 = t(OE) =	25ns Max	25ns Max

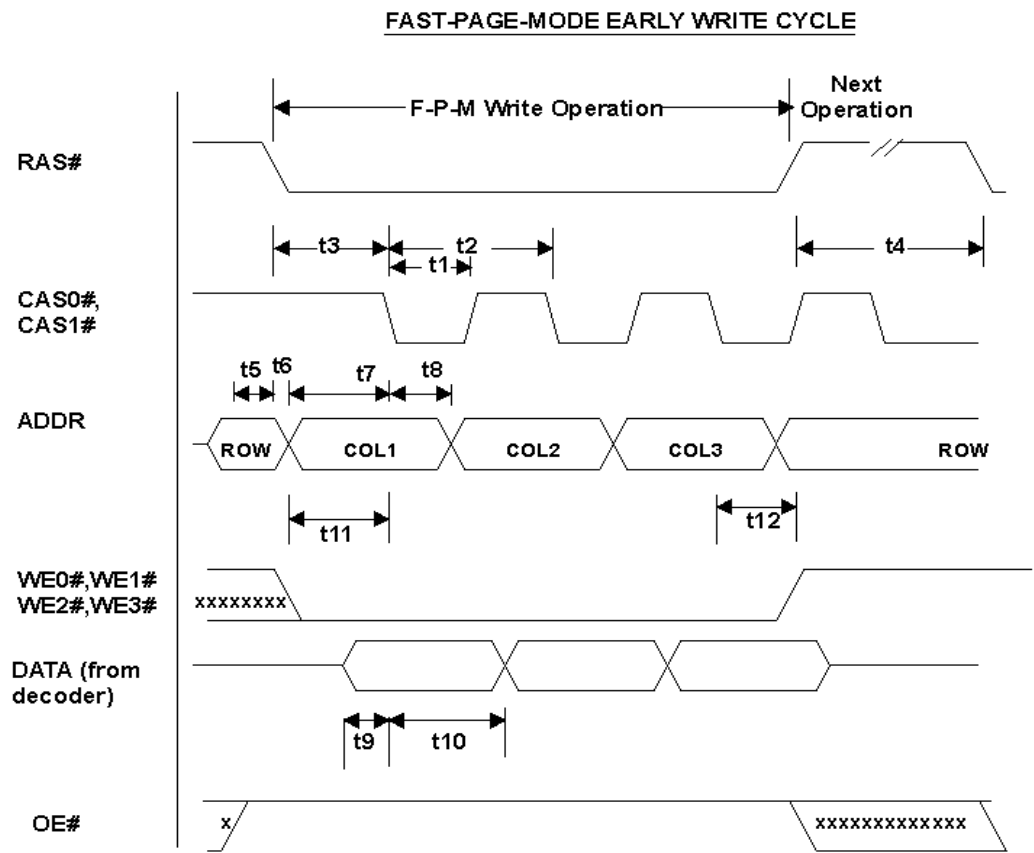


Figure 6-14. DRAM Interface Timing (Fast Page Mode Early Write Cycle)

	<b>MPEGCD20</b>	<b>MPEGCD21</b>
t1 = t(CAS) =	22ns Min	18ns Min
t2 = t(PC) =	44ns	37ns
t3 = t(RCD) =	53ns Min	46ns Min
t4 = t(RP) =	53ns Min	46ns Min
t5 = t(ASR) =	4ns Min	4ns Min
t6 = t(RAH) =	22ns Min	18ns Min
t7 = t(ASC) =	15ns Min	12ns Min
t8 = t(CAH) =	20ns Min	17ns Min
t9 = t(DS) =	5ns Min	5ns Min
t10 = t(DH) =	17ns Min	14ns Min
t11 = t(WCS) =	10ns Min	9ns Min
t12 = t(WCH) =	19ns Min	15ns Min

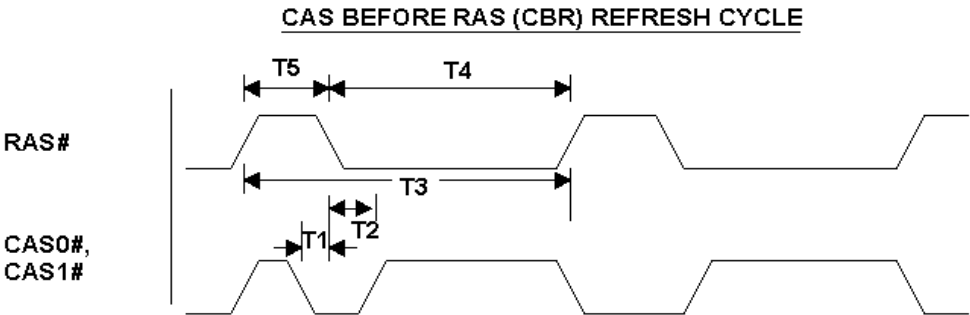


Figure 6-15. DRAM Interface Timing: CAS Before RAS (CBR) Refresh Cycle

	MPEGCD20	MPEGCD21
$t1 = t(CSR) =$	15ns Min	13ns Min
$t2 = t(CHR) =$	15ns	13ns
$t3 = t(RC) =$	133ns Min	111ns Min
$t4 = t(RAS) =$	85ns Min	70ns Min
$t5 = t(RP) =$	45ns Min	40ns Min

### 6.4.3 Video Display Interface Timing Diagrams

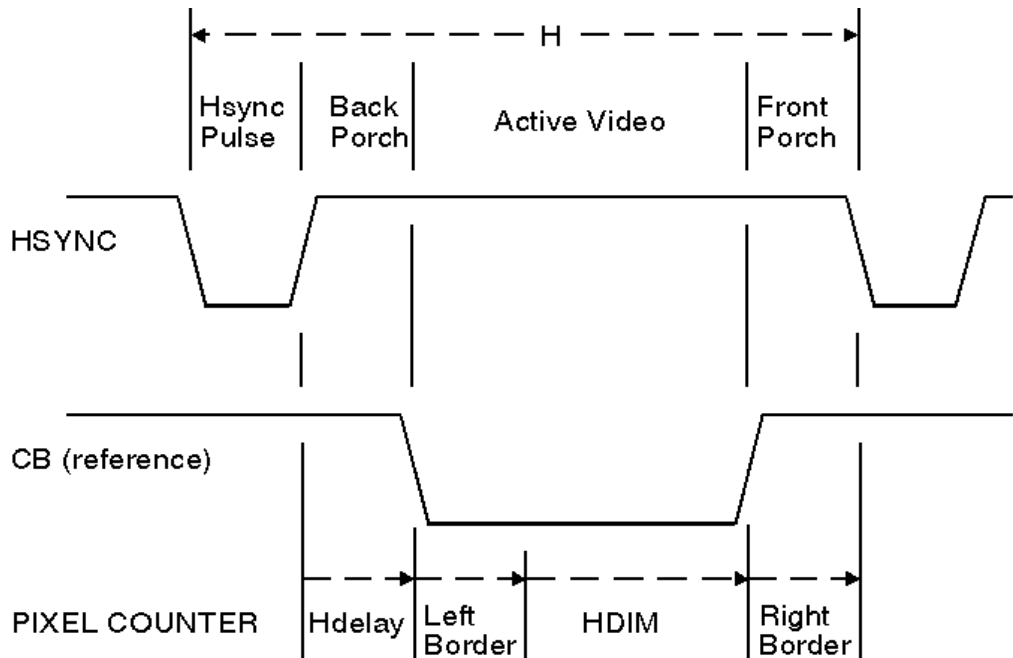


Figure 6-16. Horizontal Positioning Using Hdelay and DISP\_LBOR

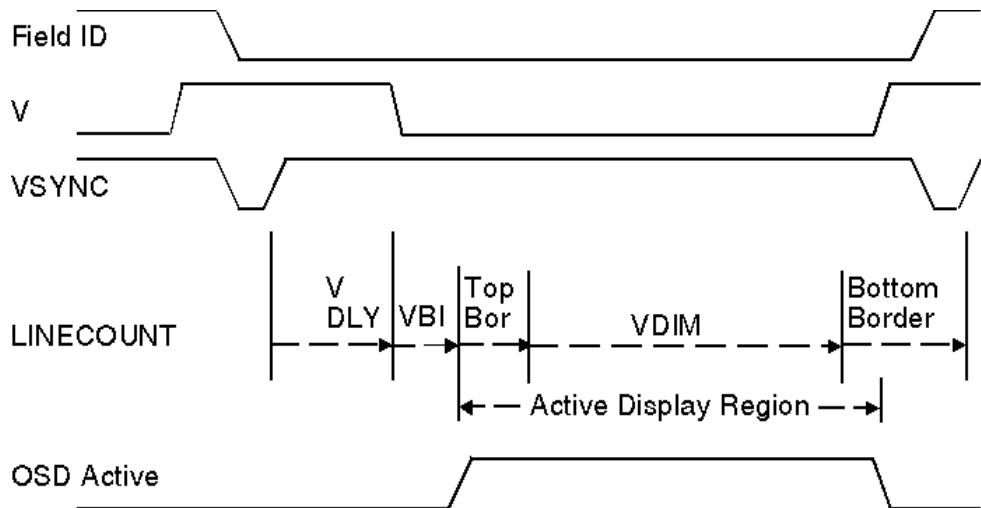
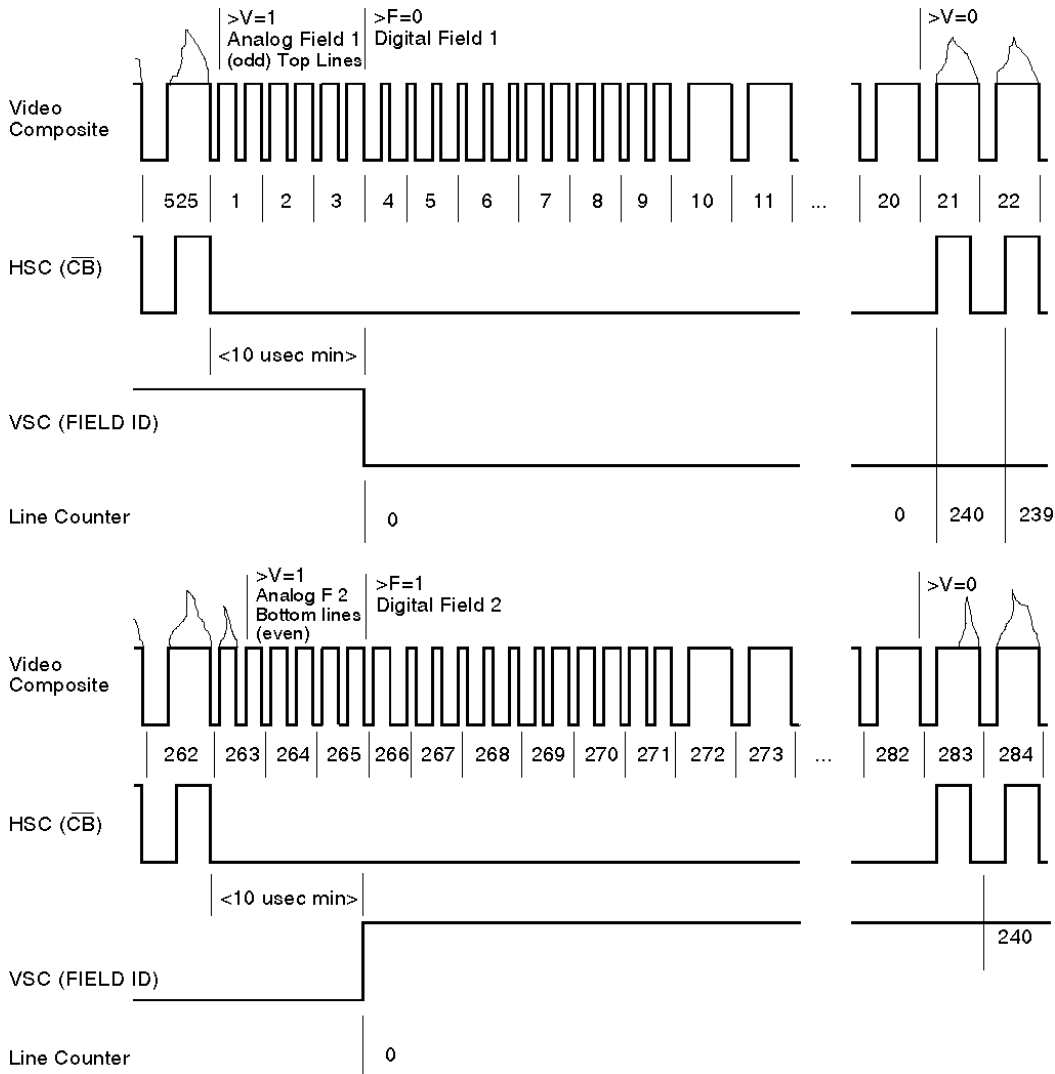


Figure 6-17. Vertical Positioning Using Vdelay, VBI and DISP\_TBOR

# Electrical and Physical Specifications



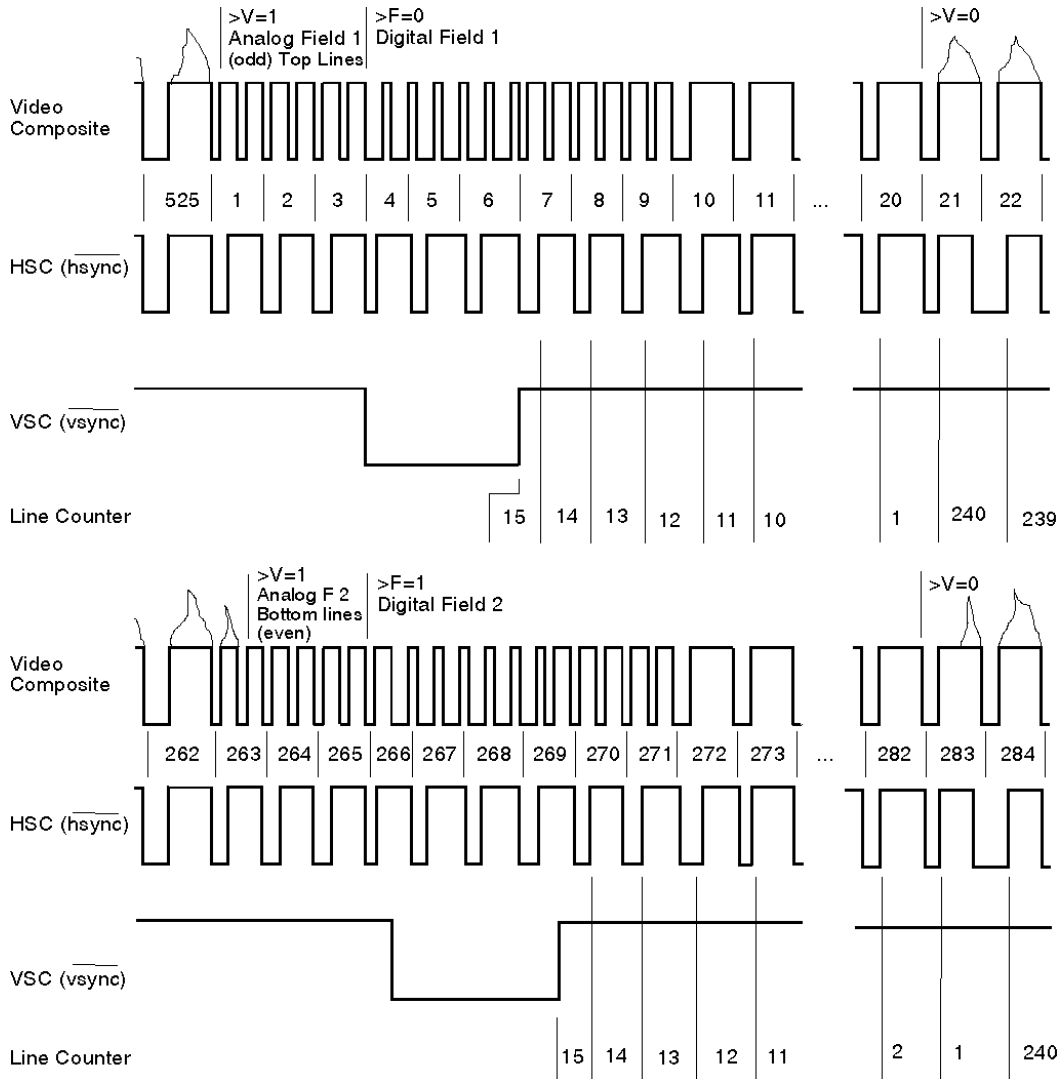
**Figure 6-18. NTSC Vertical Timing using Composite Blanking and Field ID. V-Delay is set to 0 in this example**

**SYNC MODE= 001**  
**HSC Polarity= 0 (Active Low)**  
**VSC Polarity= 1 (active High)**

In this mode of operation VSC may change at any time during the vertical blanking interval subject to the following limits. On the front end, VSC may not change earlier than 10 microseconds following the last falling edge (activation) of HSC. At the back end, VSC may change any time prior to, although not simultaneous with, the first rising edge (deactivation) of HSC. If VBI data output is required, VSC must change prior to that line time as well. As shown

in the upper set of waveforms in Figure 6-18, when VSC deactivates within the allowed window of the vertical blanking interval, the decoder will recognize this as the start of field-1. The lower set of waveforms show an active VSC transition during the vertical blanking interval, causing the decoder to recognize this as the start of field-2.

## Electrical and Physical Specifications



**Figure 6-19. NTSC Vertical Timing using HSYNC and VSYNC (trailing edge) V-Delay is set to 15 in this example.**

**SYNC MODE= 011**

**HSC Polarity= 0 (Active Low)**

**VSC Polarity= 0 (Active Low)**

The activation (falling edge) of HSC is used as the timing point in this operational mode. The deactivation (rising edge) of VSC must occur within a time window about this point to be recognized. The duration of the time window is plus or minus 64 cycles of the 27 MHz clock as shown in Figure 6-20. When VSC deactivates within the 64 cycle window of HSC activating,



this causes the decoder to recognize this as the start of field-1. The lower set of waveforms show an inactive VSC transition outside the 64 cycle window of HSC activating, causing the decoder to recognize this as the start of field-2.

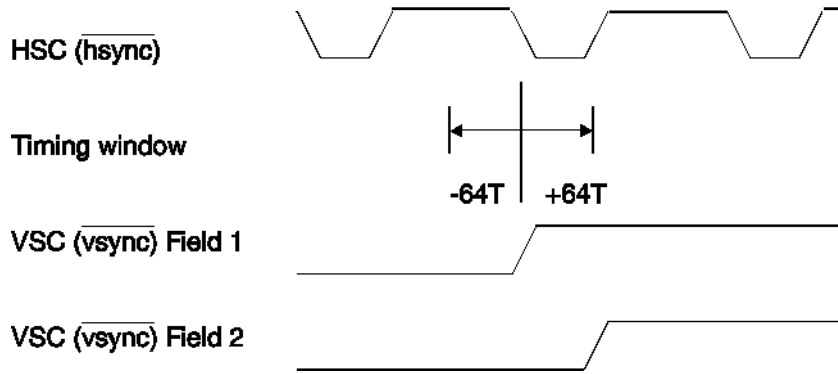
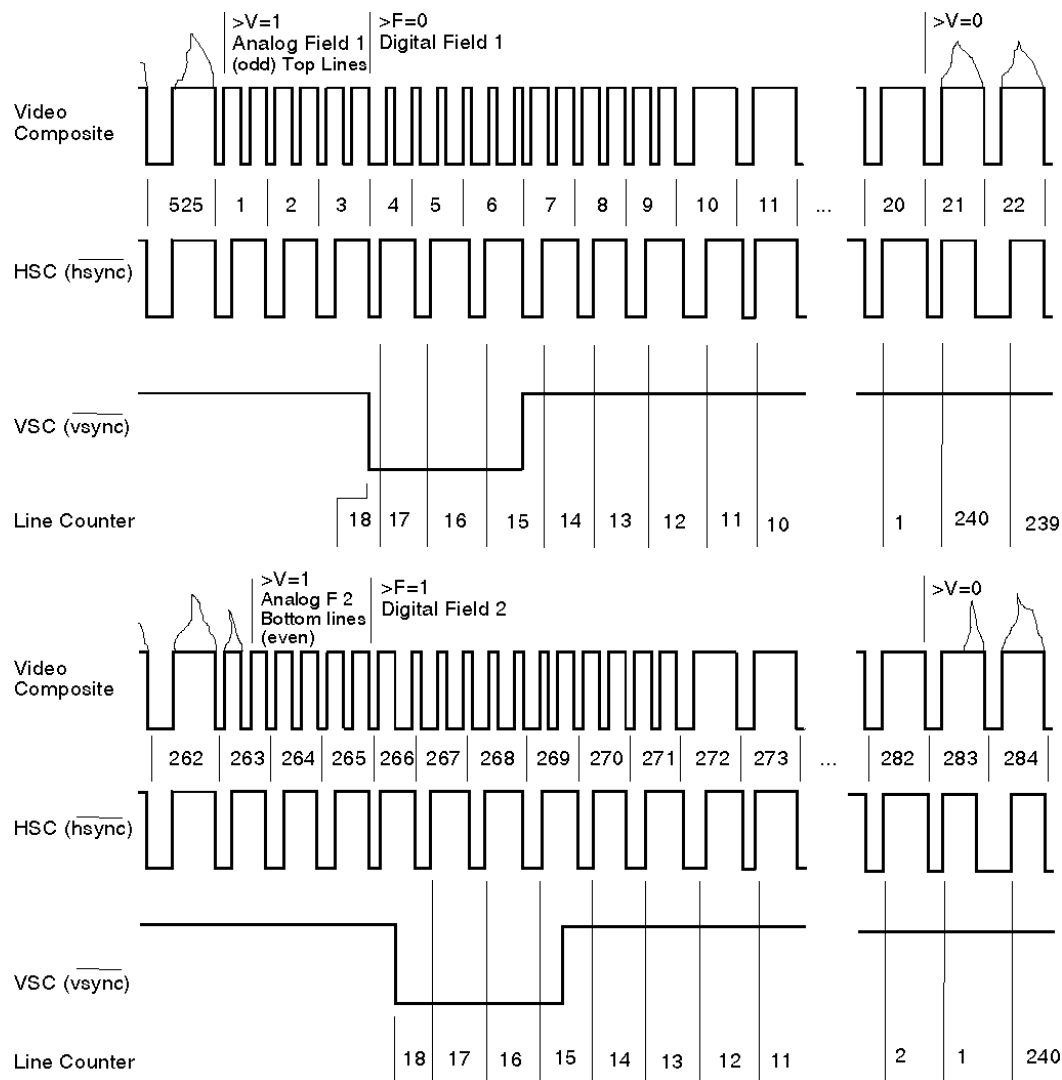


Figure 6-20. Detail of timing window. T= 1 cycle @ 27 MHz

## Electrical and Physical Specifications



**Figure 6-21. NTSC Vertical Timing using HSYNC and VSYNC (leading edge). V-Delay is set to 18 in this example.**

**SYNC MODE= 111**

**HSC Polarity= 0 (Active Low)**

**VSC Polarity= 0 (Active Low)**

The activation (falling edge) of HSC is used as the timing point in this operational mode. Activation (falling edge) of VSC must occur within a time window about this point to be recognized. The duration of the time window is plus or minus 64 cycles of the 27 MHz clock as shown in Figure 6-22. When VSC becomes active within the 64 cycle window of HSC

activating, this causes the decoder to recognize this as the start of field-1. The lower set of waveforms show an active VSC transition outside the 64 cycle window of HSC activating, causing the decoder to recognize this as the start of field-2.

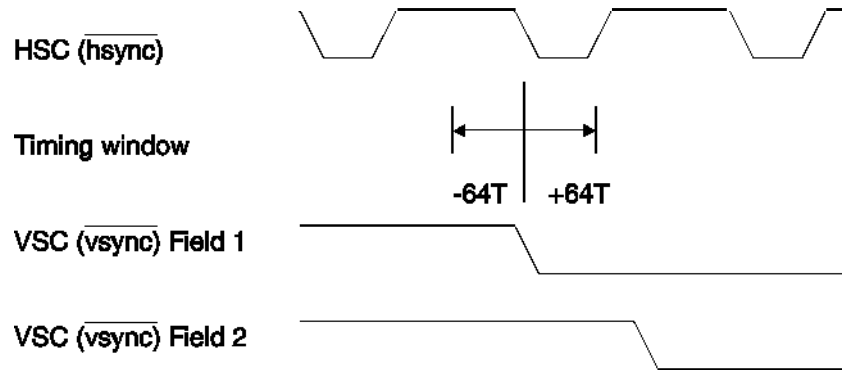
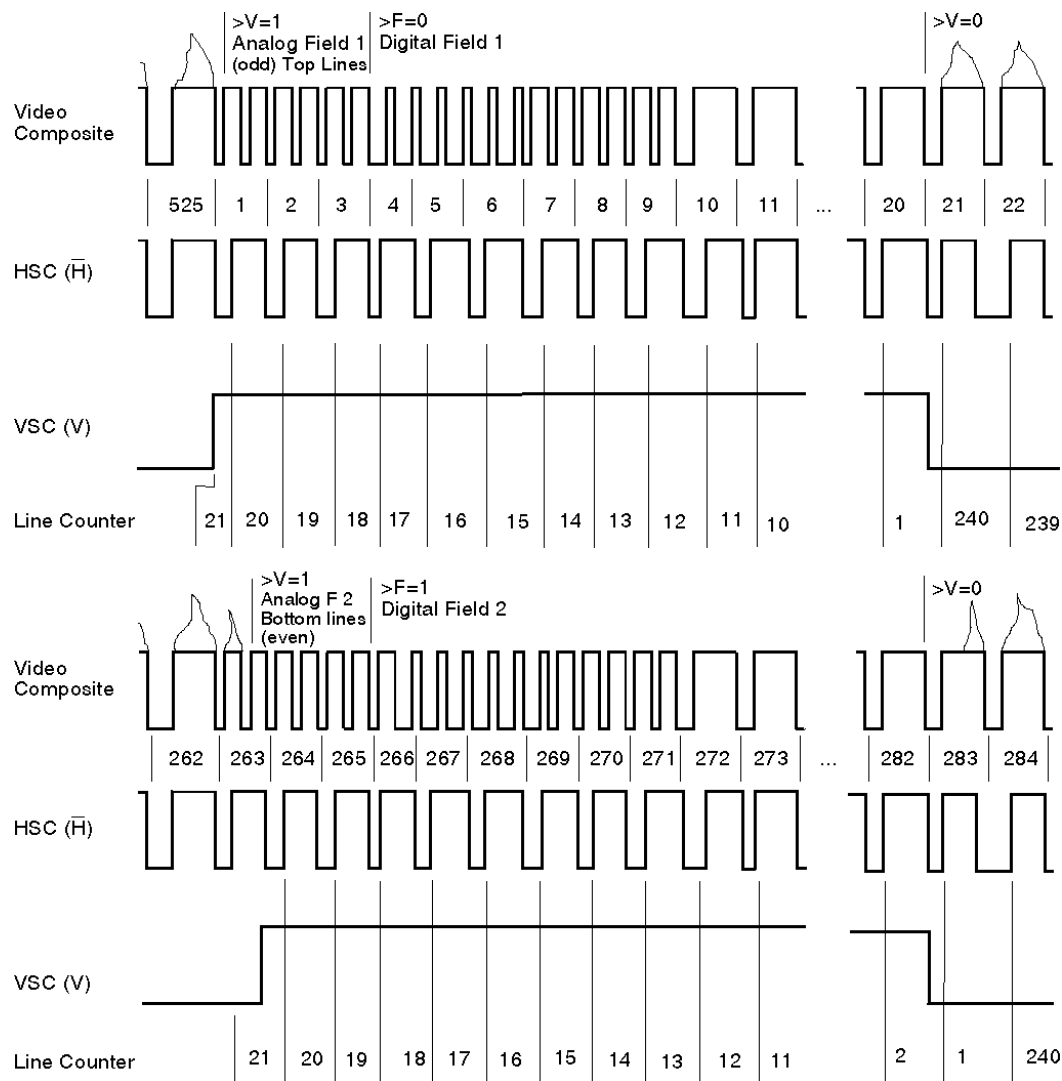


Figure 6-22. Detail of leading edge vsync. T= 1 cycle @ 27 MHz

## Electrical and Physical Specifications



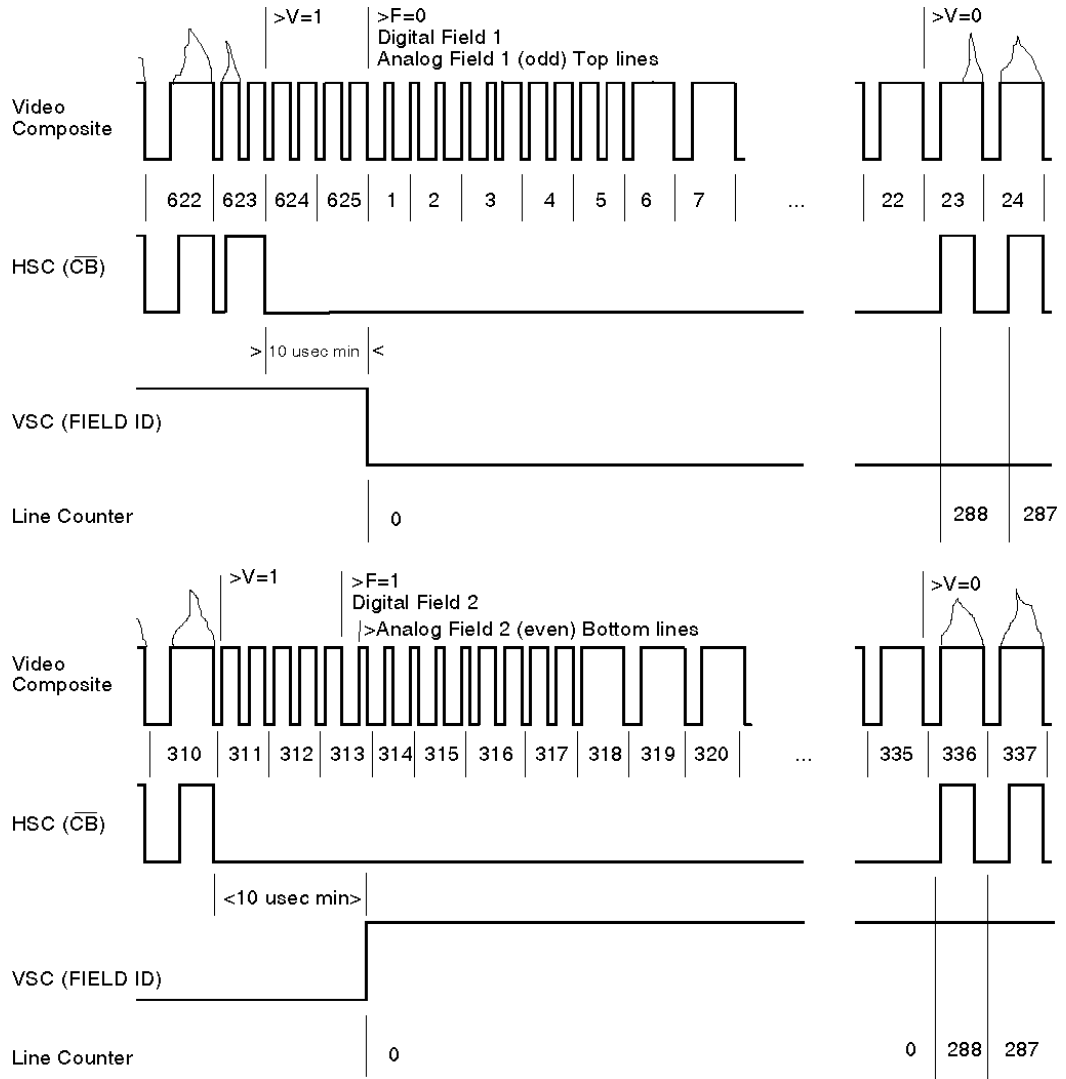
**Figure 6-23. NTSC Vertical Timing using H and V signals. V-Delay is set to 21 in this example.**

**SYNC MODE= 010**

**HSC Polarity= 0 (Active Low)**

**VSC Polarity = 1 (Active High)**

For this mode of operation, VSC must be valid at least one pixel clock (2 cycles of 27 MHz clock) prior to the deactivation (rising edge) of HSC.



**Figure 6-24. PAL Vertical Timing using Composite Blanking and Field ID. V-Delay is set to 0 in this example.**

**SYNC MODE= 001**

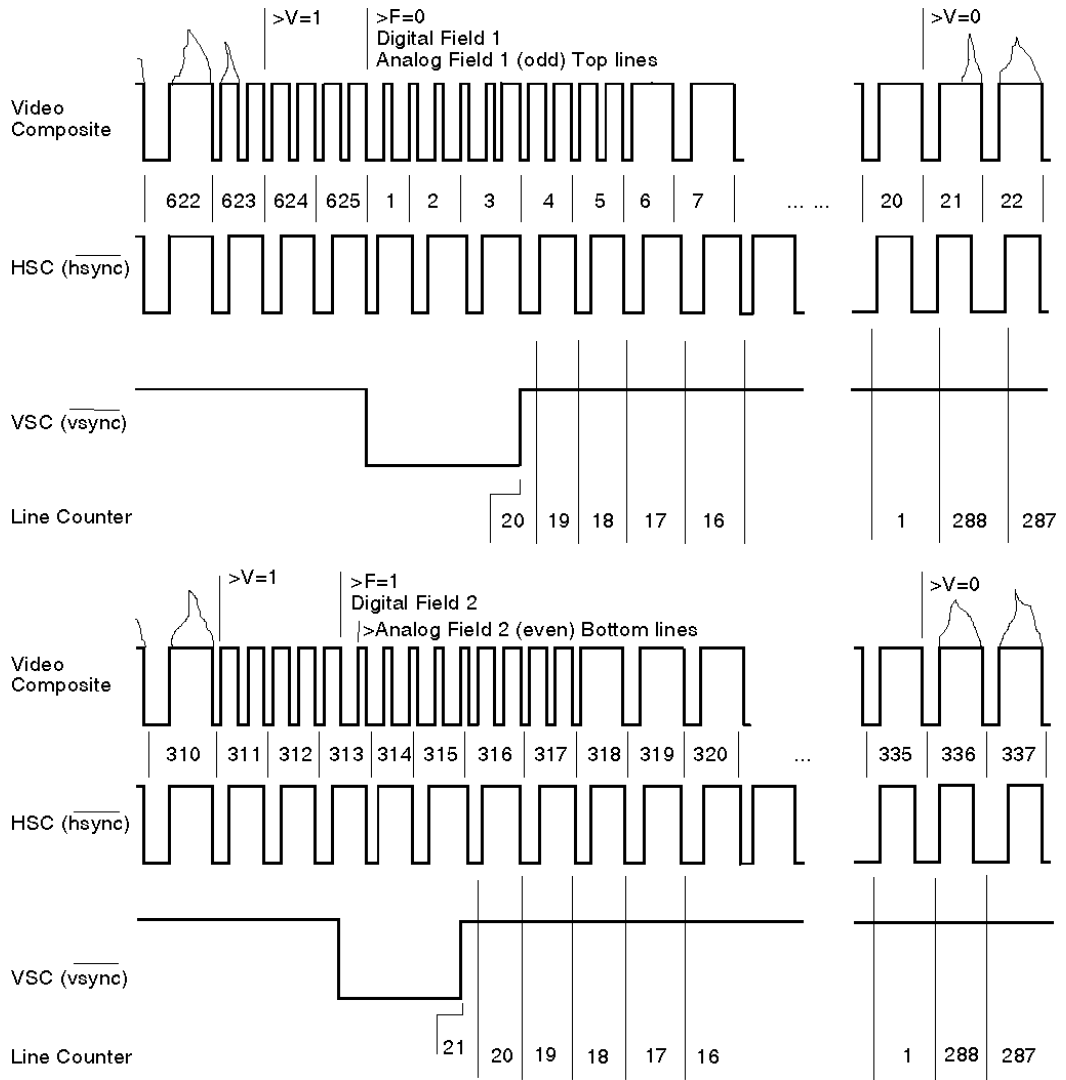
**HSC Polarity= 0 (Active Low)**

**VSC Polarity= 1 (Active High)**

In this mode of operation VSC may change at any time during the vertical blanking interval subject to the following limits. On the front end, VSC may not change earlier than 10 microseconds following the last falling edge (activation) of HSC. At the back end, VSC may change any time prior to, although not simultaneous with, the first rising edge (deactivation) of HSC. If VBI data output is required, VSC must change prior to that line time as well. As shown

## Electrical and Physical Specifications

in the upper set of waveforms in Figure 6-24, when VSC deactivates within the allowed window of the vertical blanking interval, the decoder will recognize this as the start of field-1. The lower set of waveforms show an active VSC transition during the vertical blanking interval, causing the decoder to recognize this as the start of field-2.



**Figure 6-25. PAL Vertical Timing using HSYNC and VSYNC (trailing edge). V-Delay is set to 20 in this example. For field 2, an internal adjustment is made to 21.**

**SYNC MODE= 011**

**HSC Polarity= 0 (Active Low)**

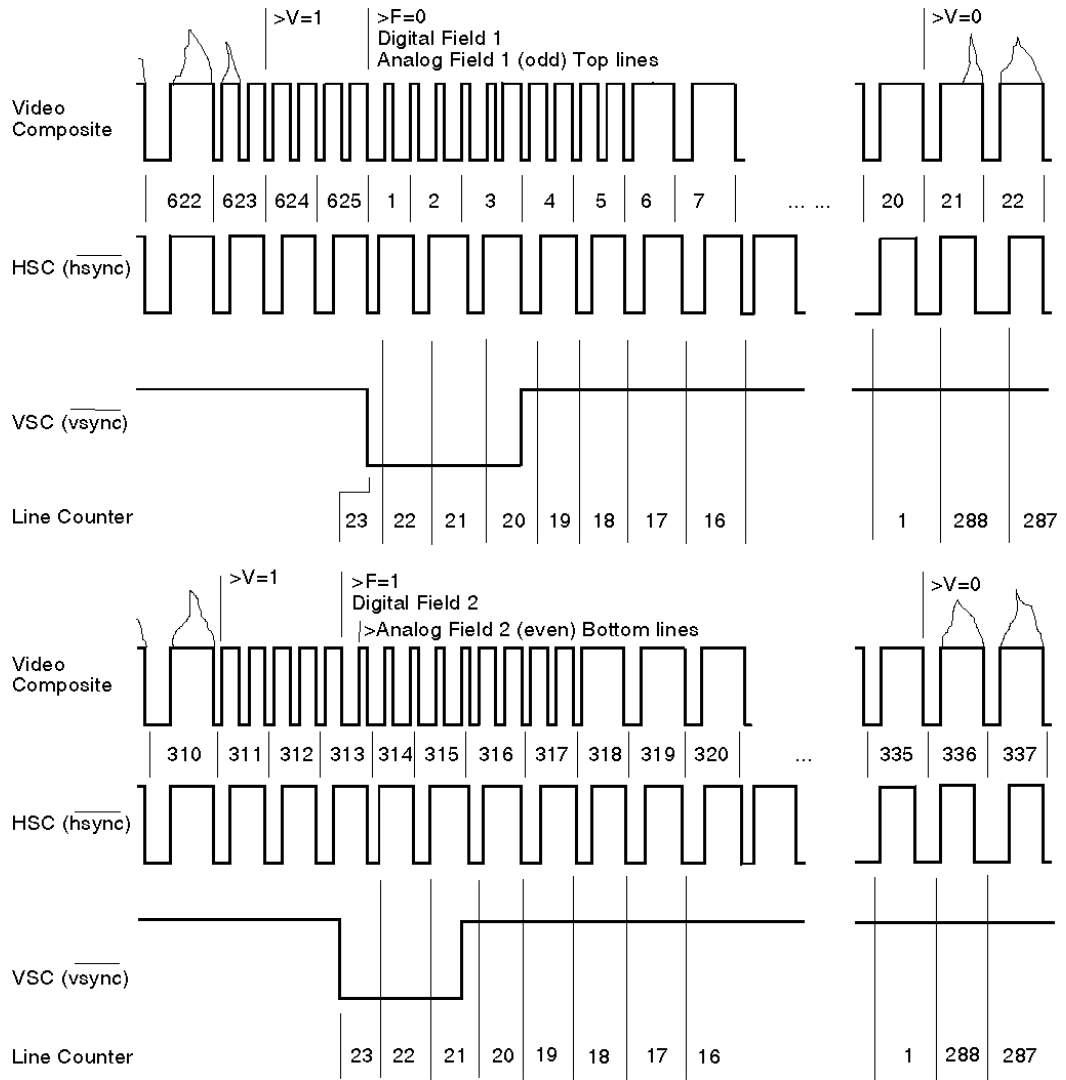
**VSC Polarity= 0 (Active Low)**

The activation (falling edge) of HSC is used as the timing point in this operational mode. The deactivation (rising edge) of VSC must occur within a time window about this point to be recognized. The duration of the time window is plus or minus 64 cycles of the 27 MHz clock, as detailed in Figure 6-20 (NTSC and PAL have same timing window). VSC deactivating outside

## Electrical and Physical Specifications

the 64 cycle window of HSC activating causes the decoder to recognize this as the start of field-1. The lower set of waveforms show an active VSC transition within the 64 cycle window of HSC activating, causing the decoder to recognize this as the start of field-2.





**Figure 6-26. PAL Vertical Timing using HSYNC and VSYNC (leading edge). V-Delay is set to 23 in this example.**

**SYNC MODE= 111**

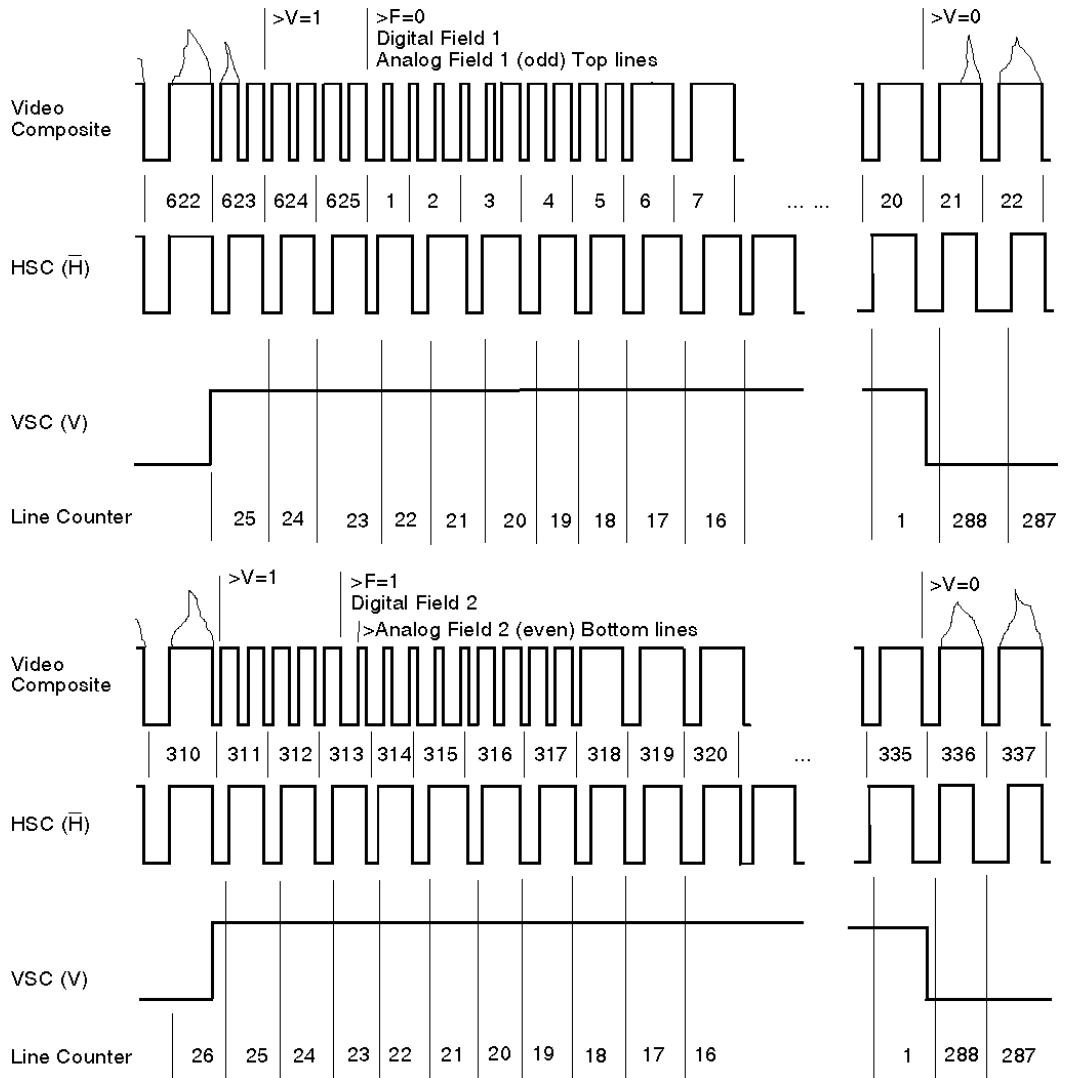
**HSC Polarity= 0 (Active Low)**

**VSC Polarity= 0 (Active Low)**

The activation (falling edge) of HSC is used as the timing point in this operational mode. Activation (falling edge) of VSC must occur within a time window about this point to be recognized. The duration of the time window is plus or minus 64 cycles of the 27 MHz clock as detailed in Figure 6-22 (NTSC and PAL have matching timing windows). When VSC becomes

## Electrical and Physical Specifications

active within the 64 cycle window of HSC activating, this causes the decoder to recognize this as the start of field-1. The lower set of waveforms show an active VSC transition outside the 64 cycle window of HSC activating, causing the decoder to recognize this as the start of field-2.



**Figure 6-27. PAL Vertical Timing using H and V signals. V-Delay is set to 25 in this example. For field 2, an internal adjustment is made to 26.**

**SYNC MODE= 010**

**HSC Polarity= 0 (Active Low)**

**VSC Polarity= 1 (Active High)**

For this mode of operation, VSC must be valid at least one pixel clock (2 cycles of 27 MHz clock) prior to the deactivation (rising edge) of HSC.

Electrical and Physical Specifications

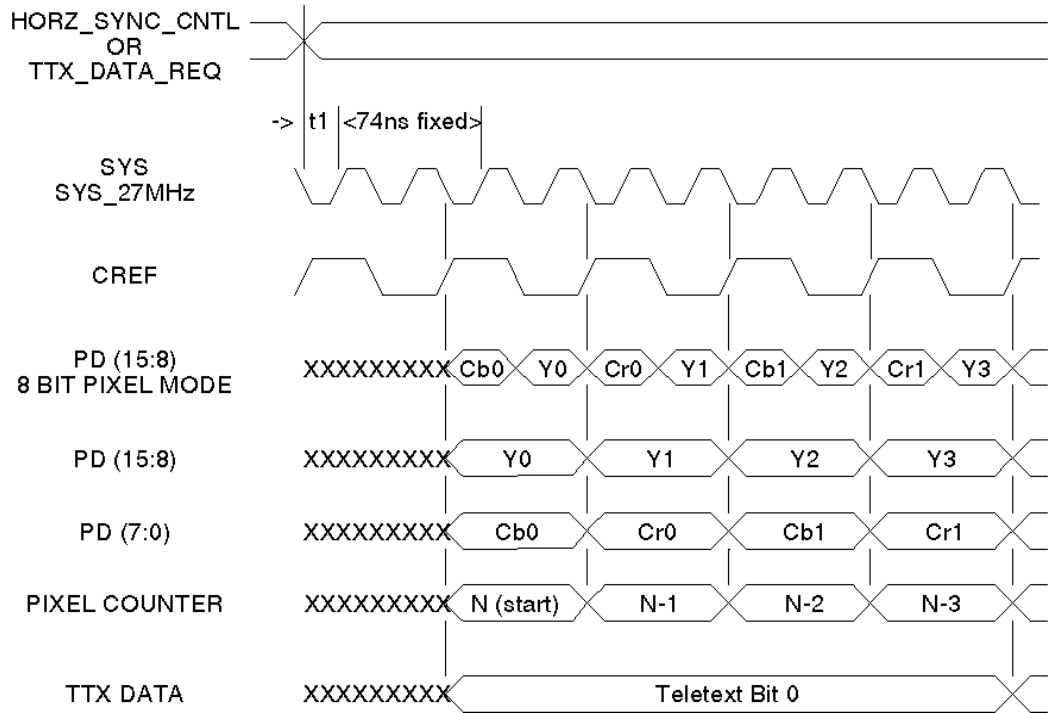
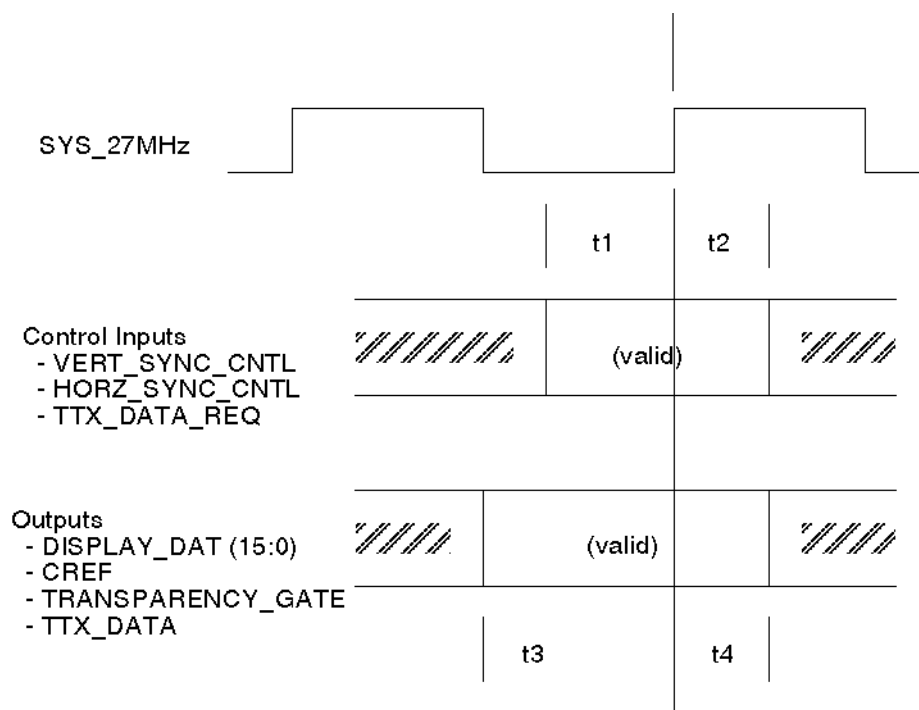


Figure 6-28. Horizontal Timing. This shows the horizontal timing starting with the activation of `HORZ_SYNC_CNTL` both for 16 and 8 bit pixel modes



t1	control input setup time	4 ns min
t2	control input hold time	4 ns min
t3	output setup time	17 ns min
t4	output hold time	6 ns min

**Figure 6-29. Timing Specification. Timing Specifications for the Display Interface.**

6.4.4 Audio Interface Timings

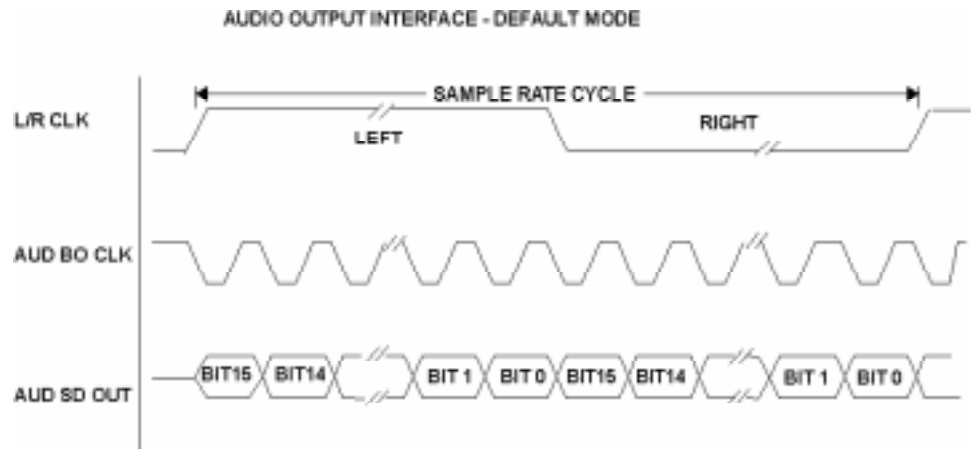


Figure 6-30. Audio Output Interface in Default Mode

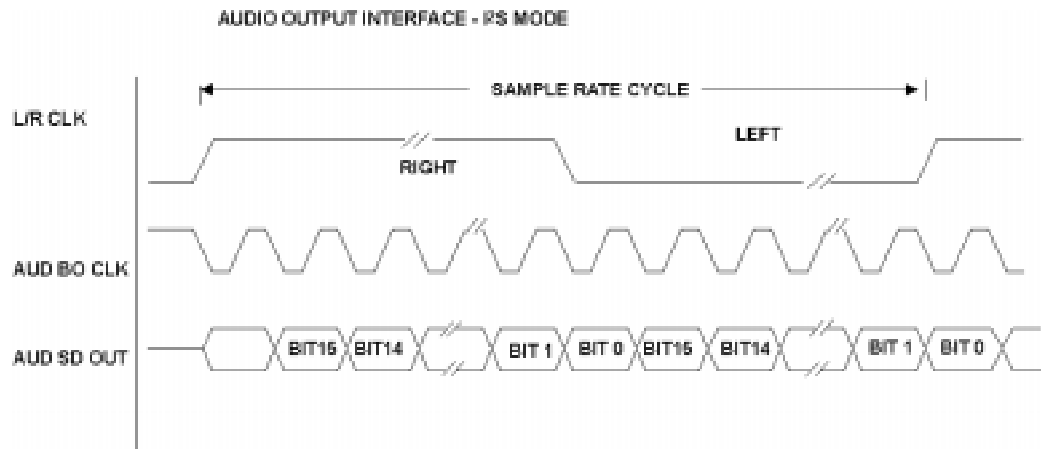


Figure 6-31. Audio Output Interface in I<sup>2</sup>S Mode

---

## Bibliography

This bibliography lists the publications that provide additional information about your system.

The following documents are available in the United States from:

American National Standards Institute (ANSI)  
11 West 42nd Street  
New York, NY 10036

ITU-T Rec. H.222.0 | ISO/IEC 13818-1:1994 Information technology - Generic coding of moving pictures and associated audio - Systems

ITU-T Rec. H.262 | ISO/IEC 13818-2:1994 Information technology - Generic coding of moving pictures and associated audio - Video

ISO/IEC 13818-2 Amendment 3:1996 Information technology - Generic coding of moving pictures and associated audio - Video

ISO/IEC 13818-3:1994. - Generic coding of moving pictures and associated audio - Audio.





# Index

## Numerics

- 3:2 pulldown 4-20
- 4:2:2 profile 2-1
  - requirements 4-37

## A

- A/V sync 4-33
  - audio master sync 4-33
  - STC master sync 4-33
  - sync disabled mode 4-34
  - video master sync 4-33
  - with clock recovery 4-34
  - without clock recovery 4-35
- aspect ratio 4-21
- audio 4-26
  - clock generation 4-30
  - controlling the decoder 4-27
  - decode status 4-29
  - decoder 4-26
  - error concealment 4-36
  - interface 4-30
  - interface signals 3-7
  - interface timing diagrams 6-48
  - master sync 4-33
  - PCM data 4-29
  - tone generation 4-27
- audio decode status 4-29
- audio layer 1-1
- audio master sync 4-33
- audio/video sync. See A/V sync

## B

- B-Picture 1-2

## C

- configuration 5-47
- configure chip command 5-47, 5-48
- controlling the audio decoder 4-27

## D

- data stream 1-1
- description 2-1
  - 4:2:2 profile 2-1
  - features 2-1
  - functional 4-1
  - MPEGCD20 2-1

- MPEGCD21 2-1
  - product 2-1
  - signals 3-1

- display
  - interface controls and timings 4-23
  - video 4-19

- display interface signals 3-7
- display. See also OSD

## DRAM

- 1MB external configuration 4-6
- 2MB external configuration 4-7
- 4MB external configuration 4-8
- configuration 4-5
- controller 4-4
- interface signals 3-5
- interface timing diagrams 6-26
- memory allocations 4-4
- requirements 4-4

## E

- electrical specifications 6-1
- error detection and concealment 4-35
  - audio error concealment 4-36
  - transport errors 4-37
  - video error concealment 4-35

## F

- frame layer 1-2
- freeze frame support 5-49
- functional description 4-1
  - 4:2:2 profile requirements 4-37
  - audio decoder 4-26
  - audio interface 4-30
  - audio/video sync
  - block diagram 4-1
  - DRAM controller 4-4
  - error detection and concealment 4-35
  - host interface 4-2
  - on-screen display 4-12
  - OSD 4-12
  - PLL 4-31
  - video decoder 4-9
  - video display interface 4-19

## G

- global control signals 3-5
- GOP layer 1-2
- group of pictures layer 1-2

## H

- host interface 4-2
  - timing diagrams 6-13

## Index

host interface signals 3-3

## I

image re-sizing 4-20  
I-picture 1-2

## L

level 1-1

## M

macroblock layer 1-4  
MPEG overview 1-1  
MPEG-2 pan/scan support 4-21

## N

no sync 4-34  
normal play support 5-49

## O

OSD 4-12  
    animation support 4-13  
    blending 4-13  
    chroma decimation 4-19  
    colors 4-13  
    control registers 4-18  
    data 4-19  
    data input 4-18  
    header definition 4-13  
    operations 4-18  
    region 4-12  
    resolution 4-13  
    shading 4-13

## P

package specifications 6-2  
PCM data 4-29  
picture in picture 4-23  
picture layer 1-2  
pin assignments 6-4–6-12  
playing PCM data 4-29  
PLL 4-31  
    external components and connections 4-31  
    signals 3-8  
P-Picture 1-2  
profile 1-1

## R

resolution 4-21

## S

Sample Play 5-50  
signals 3-1  
    audio interface 3-7  
    block diagram of 3-2  
    display interface 3-7  
    DRAM interface 3-5  
    global control 3-5  
    host interface 3-3, 4-2  
    PLL 3-8  
single field progressive scan 4-20  
slice layer 1-4  
STC master sync 4-33  
still picture support 5-50  
sync disabled mode 4-34  
sync with clock recovery 4-34  
sync without clock recovery 4-35  
system layer 1-1

## T

teletext 4-25  
timing diagrams 6-13  
    audio interface 6-48  
    DRAM interface 6-26  
    host interface timing diagrams 6-13  
    video display interface 6-31  
tone generation 4-27

## V

VBI 4-25  
vertical blanking interval. See VBI  
video 4-9  
    decoder 4-9  
    display interface timing diagrams 6-31  
    error concealment 4-35  
    master sync 4-33  
    rate buffer 4-11  
video decoder 4-9  
    rate buffer 4-11  
video display  
    3:2 pulldown 4-20  
video display 4-19  
    aspect ratio 4-21  
    image re-sizing 4-20  
    interface controls and timings 4-23  
    MPEG-2 pan/scan support 4-21  
    picture in picture 4-23  
    resolution 4-21  
    single field progressive scan 4-20  
    teletext 4-25  
    VBI  
video layer 1-1  
video master sync 4-33