L64020 DVD Audio/Video Decoder

Technical Manual

LSI LOGIC®

Order Number I14017

Document DB14-000028-00, First Edition (May 1998)
This document describes revision A of LSI Logic Corporation's L64020 DVD Audio/Video Decoder and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

To receive product literature, call us at 1.800.574.4286 (U.S. and Canada); +32.11.300.531 (Europe); 408.433.7700 (outside U.S., Canada, and Europe) and ask for Department JDS; or visit us at http://www.lsilogic.com.

LSI Logic Corporation reserves the right to make changes to any products herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase or use of a product from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or third parties.

In particular, supply of the LSI Logic IC L64020 does not convey a license or imply a right under certain patents and/or other industrial or intellectual property rights claimed by IRT, Dolby, CCETT and Philips, to use this IC in any ready-to-use electronic product. The purchaser is hereby notified that Dolby, Philips, CCETT and IRT are of the opinion that a generally available patent license for such use is required from them. No warranty or indemnity of any sort is provided by LSI Logic regarding patent infringement.

Contents

Figures

Tables

Preface

This book is the primary reference and Technical Manual for the L64020 DVD Audio/Video Decoder. It contains functional descriptions, I/O signal and register descriptions, and includes complete physical and electrical specifications for the L64020.

Audience

This document assumes that you have some familiarity with the ISO/IEC MPEG-2 standard, DVD standard, Dolby Digital standard, microprocessors, and related support devices. The people who benefit from this book are:

- Engineers and managers who are evaluating the L64020 for possible use in a system
- Engineers who are designing the L64020 into a system

Organization

This document has the following chapters and appendixes:

- Chapter 1, **Introduction**, includes an overview of the L64020 Decoder and lists its features.
- Chapter 2, **I/O Signal Descriptions**, describes the input/output signals of the L64020.
- Chapter 3, **Register Summary**, summarizes all of the registers of the L64020 in tabular form with page references to their descriptions in Chapter 4.
- Chapter 4, **Register Descriptions**, identifies and describes all of the register bits and fields of the L64020 accessible from the host processor.

- Chapter 5, **Host Interface**, describes the host interface to the L64020 and to external SDRAM connected to the L64020.
- Chapter 6, **Channel Interface**, describes the processing of the audio/video bitstream as it comes into the L64020 and the various methods which the L64020 uses to handle and recover from input stream errors.
- Chapter 7, **Memory Interface**, describes the SDRAM configurations required by the L64020 and its interface to those memories.
- Chapter 8, **Video Decoder Module**, describes how the video decoder portion of the L64020 supports MPEG-2 Main Profile and Main Level decoding and MPEG-1 Simple Profile and Main Level decoding.
- Chapter 9, **SPU Decoder Module**, describes how the Subpicture Unit (SPU) Decoder module of the L64020 decodes SPU streams.
- Chapter 10, **Video Interface**, describes how video is displayed from decoded frame stores. Also describes the features and operation of the Display Controller and how to program it for proper operation. Includes an overview of the vertical and horizontal post-processing filters.
- Chapter 11, **Audio Decoder Module**, describes how the L64020 processes Dolby Digital, Linear PCM, and MPEG (MUSICAM) input audio streams.
- Chapter 12, **Specifications**, includes the electrical requirements for, AC timing characteristics of, and a pin summary for the L64020. Also contains the pin listings and package outline drawings for the 160-pin PQFP, 176-pin TQFP, and 208-pin mini-BGA L64020 packages.
- Appendix A, **Video/Audio Compression and Decompression Concepts**, an overview of MPEG compression and decompression of audio and video.
- Appendix B, **Glossary of Terms and Abbreviations**

Related Publications

L64108 MPEG-2 Transport with Embedded MIPS CPU (CW4001) and Control Chip Technical Manual, LSI Logic Corporation, DB14-000039-00.

ISO/IEC 13818, Generic Coding of Moving Pictures and Associated Audio (MPEG-2), International Standard. ISO/IEC Copyright Office, Case Postal 56, CH1211 Genève 20, Switzerland.

ISO/IEC 11172 (1993), Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s (MPEG-1), International Standard. ISO/IEC Copyright Office, Case Postal 56, CH1211 Genève 20, Switzerland.

DVD Specifications for Read-Only Disc, August 1996, Toshiba Corporation.

ITU-R BT.601-5 (10/95), Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-screen 16:9 Aspect Ratios, http://www.itu.ch/publications/itu-r/iturbt.htm.

ITU-R BT.656-3 (10/95), Interface for Digital Component Video Signals in 525-line and 625-line Television Systems Operating at the 4:2:2 Level of Recommendation ITU-R BT.601 (Part A), http://www.itu.ch/publications/itu-r/iturbt.htm.

Conventions Used in This Manual

Unless otherwise specified, *MPEG* refers to the MPEG-2 standard.

MSB indicates the most-significant bit or byte. *LSB* indicates the least-significant bit or byte. If bit or byte is not obvious in the context, the term is spelled out.

The first time a word or phrase is defined in this manual, it is *italicized.*

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive. Signals that are active LOW end in an “n.”

Hexadecimal numbers are indicated by the prefix “0x” before the number—for example, 0x32CF. Binary numbers are indicated by the prefix “0b” before the number—for example, 0b0011.0010.1100.1111.

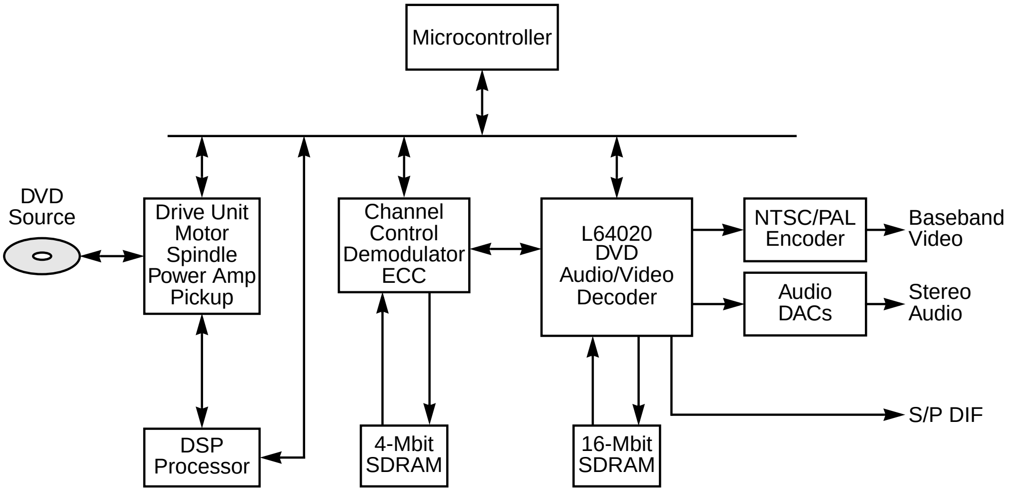
Chapter 1
Introduction

This chapter provides general overview information on the L64020 DVD Audio/Video Decoder chip. The chapter contains the following sections:

1.1 An L64020 Application

Figure 1.1 illustrates the L64020 in a typical DVD/VCD audio/video decoding system. The L64020 is specifically designed for use in digital audio and DVD video decoding systems based on the MPEG-2 algorithm (herein referred to as MPEG).

Figure 1.1 A Typical DVD Player System Block Diagram



The device may be considered a “black box” that receives coded audio and video data and produces a decoded audio and video data stream. LSI Logic has optimized L64020 input/output interfaces for low-cost integration into an embedded application. The L64020 accepts an 8-bit, parallel channel input from a Channel Control Demodulator and, with interaction of a host microcontroller, decompresses and decodes the channel information into separate, serial video and audio streams. The 64020 handles NTSC and PAL formats and provides a Sony/Philips Digital Interface (S/P DIF) formatted output stream.

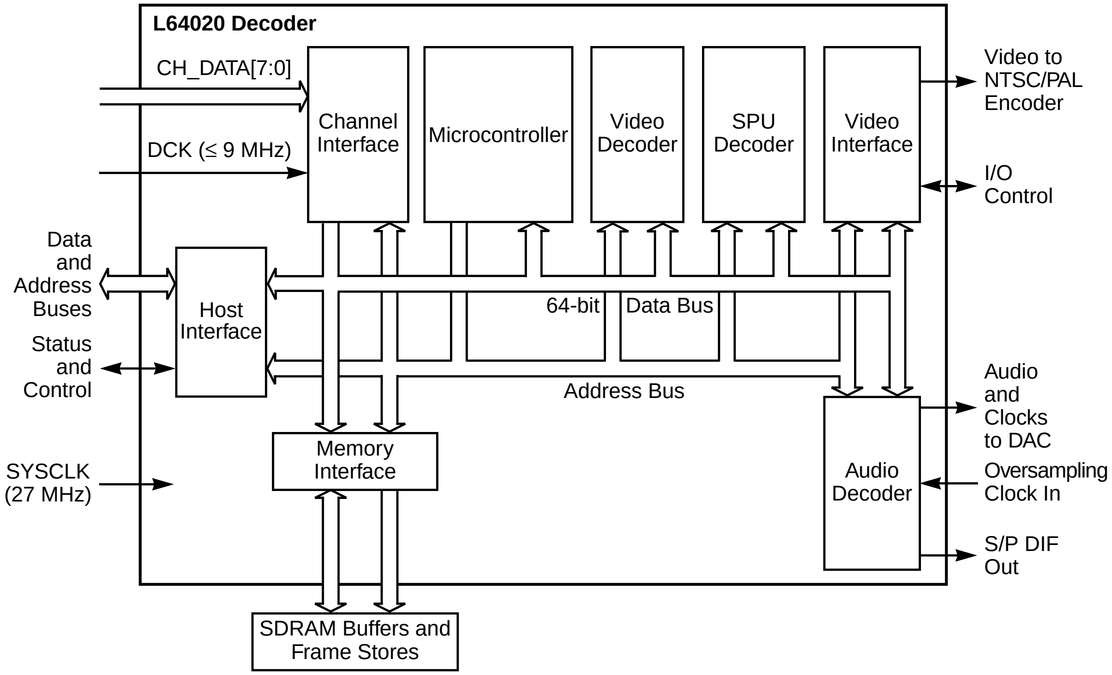
1.2 L64020 Overview

Figure 1.2 shows a block diagram of the L64020. Its major blocks include the:

- Host Interface
- Channel Interface
- Memory Interface
- Video Decoder
- Subpicture Unit (SPU) Decoder
- Video Interface
- Audio Decoder

The Host Interface includes 512, 8-bit registers (some not used), read and write FIFOs, and byte enable logic. The host and L64020 communicate with each other exclusively through the registers. An external interrupt signal from the L64020 alerts the host about internal events, such as picture start code detection. Separate I/O signals are used for handshaking. The L64020 can interface with either an Intel or Motorola type processor by tying an external pin high or low.

Figure 1.2 L64020 Decoder Block Diagram



The read and write FIFOs are used to give the host access to the external SDRAM. The read/write paths are still through registers. The interface supports direct read/write, DMA transfers using an external DMA controller, and block moves within SDRAM. The byte enable logic converts host byte writes to 8-byte words for the write FIFO and 64-bit internal bus and vice versa. The byte enable logic also performs byte switching for little endian hosts.

The Channel Interface accepts byte-wide MPEG streams and a clock. The interface synchronizes to and preparses the incoming stream by stripping system headers and storing them in a dedicated buffer area in SDRAM. The interface also separates the audio and video streams and stores them in dedicated buffer areas in SDRAM. A buffer controller maintains the read and write pointers for the dedicated buffers.

The Memory Interface includes byte enable logic and an address converter. The recommended SDRAM is 16 bits wide, so the byte enable logic performs the conversion between the SDRAM bus and the 8-byte wide internal bus of the L64020. The host and internal microcontroller of the L64020 address SDRAM as if it were 8-byte wide RAM. The address converter changes these addresses to chip selects, bank selects, and column and row addresses for the SDRAM.

The Video Decoder reads the MPEG video elementary stream from the SDRAM buffer, performs postparsing on it, decompresses it, decodes it, and stores it back in SDRAM. The postparser strips off all header information and stores it in internal memory for use in the decoding process. The postparser also strips auxiliary and user data from the stream and stores it in FIFOs that can be read through registers by the host. The decompressed and decoded video is stored back in SDRAM in frame form.

The Subpicture Unit (SPU) Decoder reads the SPU stream from the SDRAM channel buffer and decompresses and decodes it for mixing with normal video. A color palette table in the decoder is initialized with information from the bitstream and used for lookup of SPU colors. The SPU Decoder uses the System Clock Reference (SCR) from the bitstream to synchronize its presentation.

The Video Interface reads the video frames from frame stores in SDRAM, synchronizes them to the vertical and horizontal sync signals from the NTSC/PAL Encoder, and mixes in SPU and On-Screen Display (OSD) information. The interface performs letterboxing, 3:2 pulldown, and pan and scan. It also handles trick modes such as pause, slow play, fast forward, etc.

The Audio Decoder contains a Dolby Digital Decoder, MPEG (Musicam) Decoder, Linear PCM Decoder, Dolby Digital Formatter, MPEG Formatter, Audio DAC Interface, and an S/P DIF (IEC958) Interface. The decoders decompress and decode the audio stream. The decoder outputs can be steered to the DAC or S/P DIF Interface. The formatters convert the encoded and compressed streams to S/P DIF format for the S/P DIF Interface. The host controls the mode of the Audio Decoder; that is, it determines which decoder runs and where its output goes, and which formatter runs. The host can also place the Audio Decoder in the bypass mode and connect inputs from another device directly to the L64020 audio outputs.

The microcontroller is shown on the block diagram since it controls most of the processes of the L64020.

The L64020 is an MPEG-2 Main Level, Main Profile decoder. It handles image sizes up to 720 x 480 pixels with a frame rate of 30 fps for NTSC and up to 720 x 576 pixels at 25 fps for PAL. It can also decode MPEG-1 sequences. The coded data channel may have a sustained bit rate of up to 20 Mbits/second. As the resolution decreases, the amount and bandwidth of SDRAM memory required for frame stores also decreases.

1.2.1 Memory Utilization

The L64020 supports direct connection to commercial SDRAM for use as frame stores, channel buffers, and overlay memory. The L64020 uses frame stores for frame reconstruction and display, separate video and audio channel buffers for rate matching, and zero or more regions for graphical overlay. This storage is combined into a single contiguous memory space accessed over a 16-bit wide bus. In most cases this will be one 1 M x 16-bit SDRAM, for a total memory space of 2 Mbytes. The interface between the L64020 and SDRAM requires no external components. The L64020 pinout allows the connection to SDRAM to be made on a single PCB layer. During normal operation, the L64020 exclusively controls the SDRAM frame stores. However, it is possible to access the SDRAM through the host port on the L64020 for test verification and for access to the overlay stores and channel information.

1.2.2 Error Concealment

The L64020 detects data in the bitstream that does not meet MPEG-2 or Dolby Digital syntax or grammar rules and can flag the data for exception processing. Hardware error handling is limited to error masking and the application of concealment vectors in video. Audio error concealment is limited to muting on errors and searching for error-free frames. The L64020 flags gross errors in the bitstream due to channel buffer overrun or underrun or to nonconformance in the bitstream. The L64020 flags the errors so that they may be masked in the video or the audio output. The host microcontroller may be programmed to execute mechanisms to recover from gross errors.

1.2.3 Mechanical and Electrical

The L64020 is available in a 160-pin Plastic Quad Flat Pack (PQFP), a 176-pin Thin Quad Flat Pack (TQFP), and a 208-pin Mini Ball Grid Array (mini-BGA). It is manufactured using a 0.25-micron CMOS process and is powered by +3.3 Vdc.

1.3 Features

Video Decoder –

- ♦ Fully compliant to Main Profile at Main Level of the MPEG-2 video standard, ISO 13818-2.
- ♦ Decodes an MPEG-2 bitstream, including MPEG-2 Program stream, with private stream support.
- ♦ Decodes an MPEG-1 bitstream as defined in ISO IS 11172, including the MPEG-1 system layer.
- ♦ Operates at image sizes up to ITU-R BT.601 resolution (720 x 480 pixels @ 30 fps for NTSC and 720 x 576 @ 25 fps for PAL).
- ♦ Up to 20 Mbps sustained input channel data rate for program streams and A/V PES streams from a transport demultiplexer.
- ♦ 8-bit parallel dedicated input channel interface.
- ♦ 8-bit luma/chroma output.
- ♦ Complete on-chip channel buffer controller and display buffer controls.
- ♦ Error concealment maintains display of images during channel errors.

Video Interface –

- ♦ Integrates a flexible 256-color, On-Screen Display (OSD) controller.
- ♦ Allows connection to an external OSD generator.
- ♦ Programmable display management.
- ♦ Slave video timing operation.
- ♦ Supports trick modes commonly needed in DVD systems.
- ♦ Integrates postprocessing filters for image resizing (horizontal and vertical).
- ♦ Integrates vertical filter for letterbox format display.
- ♦ Implements Subpicture Unit (SPU) processing on chip.
- ♦ Implements 3:2 pulldown.
- ♦ Supports pan and scan with 1/8-pixel accuracy.
- ♦ Supports 4:2:0 to 4:2:2 sampling filters.

Audio Decoder –

- Combines MPEG and Dolby Digital audio decode with support for Linear PCM data.
- Decodes dual-channel MPEG audio, Layer I and II ISO 13818-2, supporting bit rates of 8 Kbps to 448 Kbps and sampling rates of 16, 22.05, 24, 32, 44.1, and 48 kHz.
- Decodes 5.1 channel Dolby Digital, downmixes to 2 outputs over the entire range of compliant bit rates and sampling rates.
- Supports Linear PCM streams with sample rates of 48 kHz and 96 kHz at 24-bit resolution.
- IEC958 output interface for audio data and Dolby Digital bitstreams.
- Mute on error for concealment.
- Provides an audio “Bypass” mode for interface to a CD audio decoder.

System –

- Programmable preparser accepts PES, ES, and PS streams.
- Direct connection to commodity SDRAM.
- Input/output interfaces are optimized for glueless integration into consumer video systems.
- Operates from a single 27-MHz clock with an additional audio sample clock input.
- Total external memory required for audio and video decoding is 16-Mbit SDRAM for ITU-R BT.601-5 resolution.
- Interfaces to Intel and Motorola (and compatible) 8-bit microcontrollers for initialization, testing, and status monitoring.
- Direct interface to off-the-shelf NTSC/PAL video encoders.
- Direct interface to off-the-shelf audio stereo DACs.
- Available in 160-pin PQFP, 172-pin TQFP, and 208-pin mini-BGA packs.
- Low power 3.3-Volt process.
- TTL-compatible I/O pins.

Chapter 2
I/O Signal Descriptions

This chapter describes the input/output signals of the L64020. The chapter consists of the following sections:

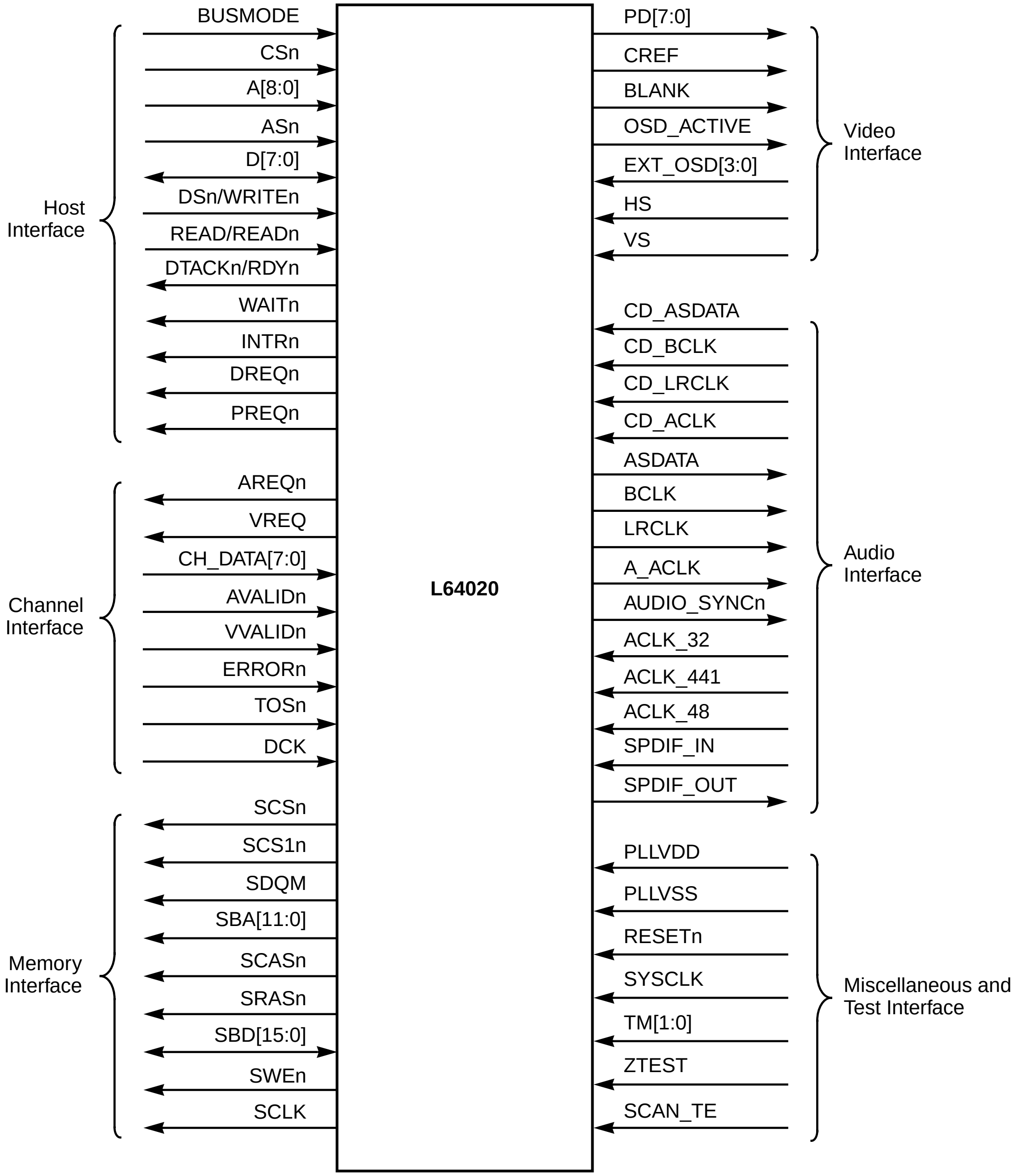
2.1 Signals Organization

The L64020 has six major interfaces:

- Host (8-bit microcontroller interface)
- Channel (8-bit synchronous or asynchronous bitstream data channel)
- Memory (16-bit synchronous SDRAM interface)
- Video (8-bit multiplexed digital video output)
- Audio (serial digital audio output)
- Test

Figure 2.1 shows the signals, their grouping, and their I/O direction. A lower case “n” at the end of a signal name indicates that it is an active LOW signal.

Figure 2.1 L64020 I/O Signals



2.2 Host Interface

BUSMODE **Host Controller Select Pin** **Input**
This pin must be tied to VSS if the host CPU is an Intel processor or to VDD if it is a Motorola processor. The Intel processor uses two separate pins, READn and WRITEn, for read and write transfers. The Motorola processor uses a single read/write signal, READ.

CSn **Chip Select** **Input**
This active-LOW signal indicates an attempt by an external host CPU to access the L64020 either for a read or a write bus cycle. CSn must be asserted for the entire read/write cycle and may held LOW for more than one bus transaction.

A[8:0] **Address** **Input**
Nine-bit address input selects one of 512 internal registers. The address value on these lines is latched on the falling edge of READn in a read cycle and on the falling edge of WRITEn in a write cycle in Intel mode. Motorola mode uses a separate address strobe, ASn.

ASn **Address Strobe** **Input**
Active-LOW address strobe input. This signal is used in Motorola mode to latch the address.

D[7:0] **Host Data Bus** **Bidirectional**
The host uses the D[7:0] bidirectional data bus to program the L64020 and access status and bitstream information during operation. During a read bus cycle, D[7:0] carries valid information from an internal L64020 register. DTACKn/RDYn or WAITn indicate when the data on the bus is valid. In write cycles, the data is latched by the L64020 on the rising edge of DSn/WRITEn.

DSn/WRITEn **Data Strobe/Write Indicator** **Input**
DSn - Motorola Mode
DSn indicates when the host strobes the data in or out of the L64020. Read transactions start when DSn, CSn, and ASn are all LOW. During a write cycle, the L64020 latches the data on the bus on the rising edge of DSn.

WRITEn - Intel Mode
The external host asserts WRITEn to start a write cycle.

READn must be HIGH during a write cycle, and CSn must be LOW during a write cycle. The address is registered on the falling edge of WRITEn. The data is latched by the L64020 on the rising edge of WRITEn.

READ/READn **Read/Write Strobe - Read Indicator** **Input**

READ - Motorola Mode
The Motorola host sets READ HIGH for a read cycle and clears it LOW for a write cycle. CSn must be asserted to select the L64020.

READn - Intel Mode
The Intel host asserts READn and holds WRITEn deasserted to perform a read cycle. The address is registered on the falling edge of READn. CSn must be asserted to select the L64020.

DTACKn/RDYn

Data Acknowledge/Data Ready **3-State Output**

DTACKn - Motorola Mode
The L64020 asserts this signal to indicate to the external host that the current bus transaction (read or write) can be completed. DTACKn is 3-stated if CSn is not asserted. The bus cycle is terminated if the L64020 deasserts DTACKn before the cycle is completed.

RDYn - Intel Mode
The L64020 asserts this signal to indicate to the external host that the current bus transaction (read or write) can be completed. RDYn is 3-stated if CSn is not asserted. The bus cycle is terminated if the L64020 deasserts RDYn before the cycle is completed.

WAITn **Wait** **3-State Output**

This signal may be used instead of DTACKn/RDYn by hosts that require an inverted sense. The L64020 asserts WAITn to indicate that its Host Interface is busy with a read or write bus cycle and it deasserts it when the current cycle is completed. WAITn is 3-stated when CSn is not active.

INTRn **Interrupt** **OD Output**

INTRn is an active-LOW, open-drain, output signal. The L64020 asserts this signal to alert the host that an unmasked interrupt condition has occurred in the chip. The host must read registers 0 through 4 to determine the cause of the interrupt, take the appropriate action, and set the Clear Interrupt Pin bit in Register 6 (page 4-12) to deassert INTRn.

DREQn **DMA Transfer Request** **Output**

The L64020 asserts this signal when it is ready to receive a new byte of data from or transmit a new byte of data to an external DMA controller. The state of DREQn reflects the condition of internal read and write FIFOs. For DMA write cycles, DREQn is deasserted when the write FIFO is not near full (more than one space left) and deasserted when the FIFO is near full. For read cycles, DREQn is asserted when the read FIFO is not near empty (more than one space filled) and deasserted when the FIFO is near empty. The maximum transfer rate over this interface is 20 Mbps in worst case conditions. The peak data rate may increase above this depending on system SDRAM usage.

PREQn **PCM FIFO Request** **Output**

The L64020 asserts this signal when it is ready to receive a new byte of data in the PCM FIFO, i.e., when the FIFO is not near full (less than 25 bytes unread). The PCM FIFO allows the host to send Linear PCM audio samples to the Audio Decoder in the L64020. PREQn can be used as a request signal to an external DMA controller.

2.3 Channel Interface

AREQn **Audio Transfer Request** **Output**

The L64020 asserts AREQn when it is ready to receive a new byte of coded audio data in A/V PES stream mode (from a transport stream demultiplexer) or a new byte of any data in program stream modes. The decoder is ready when the channel input FIFO is not near full.

VREQn **Video Transfer Request** **Output**

The L64020 asserts VREQn when it is ready to receive a new byte of coded audio data in A/V PES stream mode (from a transport stream demultiplexer). The decoder is ready when the channel input FIFO is not near full. VREQn is not used in program stream modes.

CH_DATA[7:0] Channel Data Bus Input

The CH_DATA bus is used to transfer 8-bit, parallel bitstreams into the L64020. The maximum transfer rate over this interface is 20 Mbps in worst case conditions. The peak data rate may increase above this rate depending on system SDRAM usage.

AVALIDn Audio Data Valid Input

The channel device asserts this signal in response to AREQn when the data byte it placed on the CH_DATA bus is valid. The L64020 transfers the byte in when AVALIDn is deasserted. This signal can be used with the DCK input for synchronous transfers.

VVALIDn Video Data Valid Input

The channel device asserts this signal in response to VREQn when the data byte it placed on the CH_DATA bus is valid. The L64020 transfers the byte in when VREQn is deasserted. This signal can be used with the DCK input for synchronous transfers. This signal is used only in the A/V PES stream mode when the channel input is a program from a transport stream demultiplexer. Use the AVALIDn signal for all data bytes in program stream modes.

ERRORn Bitstream Error Input

ERRORn is asserted by the channel device to signal uncorrectable errors in the bitstream and is used by the L64020 to invoke error handling routines. It is latched by the L64020 on the rising edge of AVALIDn or VVALIDn.

TOSn Top of Sector Input

The host asserts TOSn when the Top Of Sector is detected by the DVD channel device. DVD packs are stored in separate sectors on disks. The L64020 uses TOSn for error detection purposes. This signal is latched by the L64020 on the rising edge of AVALIDn (VVALIDn is not used for DVD streams).

DCK Channel Clock Input

The DCK is a free-running clock from the external channel device. It must have a period ≥3x that of SYSCLK (27 MHz). DCK, together with the AVALIDn and VVALIDn signals, is used to write data synchronously to the L64020 channel input.

2.4 Memory Interface

Important: The length of all connections between the L64020 and SDRAM on a PCB layout must be kept as short as possible, must be matched in length and pin load, and the pin load should be less than 50 pF.

SCSn **SDRAM Chip Select** **Output**
The host asserts this signal to select the low address SDRAM chip, the first 2 Mbytes of memory. The recommended SDRAM size for the L64020 is 2048 x 512 x 16 bits.

SCS1n **Second SDRAM Chip Select** **Output**
The host asserts this signal to select the high address SDRAM chip in systems that have more than 2 Mbytes of memory. The high address SDRAM chip must have the same page size as the low address SDRAM chip but does not have to have the same number of pages.

SDQM **SDRAM Control Pin** **Output**
SDQM is an active HIGH output signal for the SDRAM data control mask.

SBA[11:0] **SDRAM Address Bus** **Output**
The row/column multiplexed address bus for SDRAM memory. The L64020's microcontroller and the host address SDRAM as if it were RAM. The Memory Interface converts these addresses to SDRAM format.

SCASn **SDRAM Column Address Select** **Output**
The Memory Interface asserts this signal when the SDRAM column address is on SBA[11:0].

SRASn **SDRAM Row Address Select** **Output**
The Memory Interface asserts this signal when the SDRAM row address is on SBA[11:0].

SBD[15:0] **SDRAM Data Bus** **Bidirectional**
This 16-bit bidirectional data bus is directly connected to 1M x 16 SDRAM(s) for buffering channel data and reconstructed pictures.

SWEn **SDRAM Write Enable** **Output**

The Memory Interface asserts SWEn for SDRAM write cycles and holds it deasserted for SDRAM read cycles.

SCLK **SDRAM 81-MHz Clock** **Output**

The 27-MHz SYSCLK input is multiplied by three using the on-chip PLL to generate the 81-MHz SCLK.

Important: SCLK should be connected through a 33-Ω terminating resistor mounted as close as possible to the SCLK pin of the L64020.

2.5 Video Interface

PD[7:0] **Pixel Data Output Bus** **Output**

The PD[7:0] bus carries the pixel data for the reconstructed pictures. The pixel data is formatted in ITU_R BT.601 Y, Cb, Cr chromanicity.

CREF **Chroma Reference** **Output**

The Video Interface asserts CREF when the Cb component of Chroma is on PD[7:0] and deasserts it at all other times.

BLANK **Blank** **Output**

BLANK is a composite blank output from the L64020 display controller. Its polarity is user-defined.

OSD_ACTIVE **On-Screen Display** **Output**

The Memory Interface asserts this signal to indicate that the on-chip OSD pixel on PD[7:0] is nontransparent. This signal indicates which pixels have mixed OSD.

EXT_OSD[3:0]

Palette Selection Bus **Input**

The host controls an external device (such as a character generator) to write half-bytes across this bus to select colors from a 16-color look-up table in the L64020 to be used for external OSD.

HS **Horizontal Sync** **Input**

HS is the horizontal sync signal from the PAL/NTSC Encoder. HS is used to reset the horizontal counters in the display controller. HS should be synchronous to SYSCLK.

VS **Vertical Sync/Odd-Even Field Indicator** **Input**
VS is the vertical sync signal from the PAL/NTSC Encoder. It can be programmed to be either a conventional vertical sync input or an even/odd field indicator. In the even/odd field indicator mode, the internal display controller counters reset each time VS changes state (at the beginning of each field). The polarity of the field is controlled by the timing of VS relative to HS. VS should be synchronous to SYSCLK.

2.6 Audio Interface

CD_ASDATA **CD Mode Audio Data** **Input**
Unencoded serial audio data from a CD or other storage device. Connected directly to the ASDATA output when the host selects the CD Bypass mode.

CD_BCLK **CD Mode DAC Bit Clock** **Input**
Bit clock from CD player. Connected directly to the BCLK output when the host selects the CD Bypass mode.

CD_LRCLK **CD Mode DAC Left/Right Clock** **Input**
Left/right sample clock from CD player. Connected directly to the LRCLK output when the host selects the CD Bypass mode.

CD_ACLK **CD Mode DAC Clock** **Input**
DAC clock from CD player. Connected directly to the A_ACLK output when the host selects the CD Bypass mode.

ASDATA **Audio Serial Data** **Output**
Serial audio data from the L64020's Audio Decoder in non-bypass modes. The data can be Dolby Digital, MPEG, or Linear PCM audio. Serial audio data from the CD_ASDATA input in CD Bypass mode.

BCLK **DAC Bit Clock** **Output**
Serial data bit clock used by the L64020's DAC Interface to serialize the decoded audio data and by the external DAC to clock it in on the rising edge. BCLK is derived from one of the ACLK_ inputs under host control in normal modes and is the CD_BCLK input in CD Bypass mode.

LRCLK **DAC Left/Right Sample Clock** **Output**

Used to indicate which samples belong to the left and right stereo channels. In default mode, LRCLK is set HIGH when the right channel sample is on the ASDATA pin and cleared to LOW when the left channel sample is on the ASDATA pin. The host can set the Invert LRCLK bit in Register 363 (page 4-98) to invert the sense of the clock (HIGH for left channel, LOW for right).

A_ACLK **DAC Clock** **Output**

This clock is buffered from the selected input ACLK_ (see the following ACLK_ description). In CD-bypass mode, this clock comes directly from the CD_ACLK input pin.

AUDIO_SYNCn

Audio Sync Strobe **Output**

Provides an indication of Audio Decoder synchronization to the bitstream. Use in transport systems requiring hardware sync controls. This is an active LOW pulse at the time of the audio frame decode start.

ACLK_32, ACLK_441, ACLK_48

Audio Reference Clocks **Input**

Host selectable audio reference clocks from which clocks for the external DAC, internal DAC Interface, and internal S/P DIF Interface are derived.

ACLK_32 = 32 kHz * N,
ACLK_441 = 44.1 kHz * N, and
ACLK_48 = 48 kHz * N

where N = 768, 384, 512, or 256.

At least one of the three ACLK_ inputs must be supplied and it must be integrally divisible into the required sample rate clocks. See the ACLK Select bits in Register 363 (page 4-97) and the ACLK Divider Select bits in Register 364 (page 4-98).

SPDIF_IN **External S/P DIF** **Input**

This input is directly connected to the SPDIF_OUT pin when the host selects the S/P DIF Bypass mode.

SPDIF_OUT **S/P DIF Output** **Output**

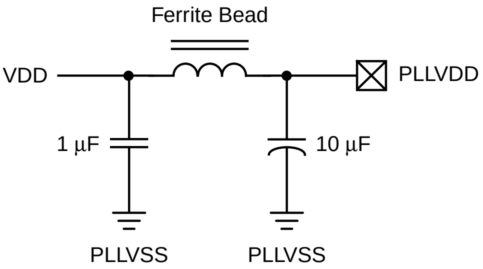
IEC958 formatted output of the L64020’s S/P DIF Interface in normal modes and SPDIF_IN in S/P DIF Bypass mode.

2.7 Miscellaneous and Test Interfaces

PLLVDD **PLL Power Supply** **Input**

This pin provides power (3.3 V) to the on-chip PLL for deriving the 81-MHz SDRAM clock. This power supply pin must be isolated from the digital power plane with the filter shown in Figure 2.2 and only connected at the voltage regulator.

Figure 2.2 PLLVDD Decoupling Circuit



PLLVSS **PLL Ground** **Input**

This pin provides ground to the on-chip PLL for deriving the 81-MHz SDRAM clock. This supply pin must be isolated from the digital ground plane, and only connected at the voltage regulator. It should be decoupled from the PLLVDD pin.

RESETn **Reset** **Input**

When RESETn is asserted, the L64020 resets its internal microcontroller, FIFO controllers, state machines, and registers. The minimum RESETn pulse width is 8 cycles of SYSCLK (8/27 MHz = 300 ns). SYSCLK and the selected ACLK_ (ACLK_32, ACLK_441, or ACLK_48) must be running during reset.

SYSCLK **Device Clock** **Input**

Device clock has a nominal frequency of 27 MHz. Picture reconstruction and video timing are referenced with respect to this clock. SYSCLK also drives the PLL to generate the 81-MHz clock for the SDRAM interface.

TM[1:0] **Test Mode** **Input**
These inputs are used by LSI Logic during manufacturing test. They are not exercised in a customer system. They should both be tied to VSS in the system.

ZTEST **Test Mode** **Input**
Test mode pin. This should be tied to VDD in the system for normal operation.

SCAN_TE **Test Mode** **Input**
Test mode pin. This should be tied to VSS in the system for normal operation.

Chapter 3
Register Summary

Communication with the L64020 Decoder is through 512, 8-bit registers. The registers are named by their decimal address, 0 to 511. They are organized into the eight groups listed in Table 3.1. The registers, fields, and bits in each group are further detailed in Table 3.2 through Table 3.9.

To find a register, field, or bit, use Table 3.1 to find the starting page of the summary table for the group. Then use the summary table to find the page in Chapter 4 on which the register is described.

If you know the name of a field or bit, use the alphabetic index starting on page 3-38 to find the page number on which it is described.

3.1 Summary by Register

Table 3.1 L64020 Register Groupings

Register Number	Register Group	Table Number	Page Reference
0–63	Host Interface Registers	3.2	3-2
64–191	Video Decoder Registers	3.3	3-8
192–223	Memory Interface Registers	3.4	3-16
224–255	Microcontroller Registers	3.5	3-19
256–335	Video Interface Registers	3.6	3-22
336–383	Audio Decoder Registers	3.7	3-27
384–415	RAM Test Registers	3.8	3-32
416–511	SPU Decoder Registers	3.9	3-35

Table 3.2 Host Interface Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
0	0	0	R[1]	0	Decode Status Interrupt	4-2
			W	0	Decode Status Mask	
		1	R[1]	0	Aux/User Data FIFO Ready Interrupt	4-2
			W	0	Aux/User Data FIFO Ready Mask	
		2	R[1]	0	First Slice Start Code Detect Interrupt	4-3
			W	0	First Slice Start Code Detect Mask	
		3	R[1]	0	Sequence End Code Detect Interrupt	4-3
			W	0	Sequence End Code Detect Mask	
		4	R[1]	0	SDRAM Transfer Done Interrupt	4-3
			W	0	SDRAM Transfer Done Mask	
		5	R	0	SPU SCR Compare Interrupt	4-3
			W	0	SPU SCR Compare Mask	
		6	R	0	Audio Sync Recovery Interrupt	4-3
			W	0	Audio Sync Recovery Mask	
		7	R	0	New Field Interrupt	4-3
			W	0	New Field Mask	
1	1	0	R[1]	0	Audio Sync Code Detect Interrupt	4-4
			W	0	Audio Sync Code Detect Mask	
		1	R[1]	0	Picture Start Code Detect Interrupt	4-4
			W	0	Picture Start Code Detect Mask	
		2	R[1]	0	SCR Compare Audio Interrupt	4-4
			W	0	SCR Compare Audio Mask	

(Sheet 1 of 6)

Table 3.2 Host Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
1	1	3	R[1]	0	SPU Start Code Detect Interrupt	4-5
			W	0	SPU Start Code Detect Mask	
		4	R[1]	0	Begin Active Video Interrupt	4-5
			W	0	Begin Active Video Mask	
		5	R[1]	0	Begin Vertical Blank Interrupt	4-5
			W	0	Begin Vertical Blank Mask	
		6	R[1]	0	SCR Overflow Interrupt	4-5
			W	0	SCR Overflow Mask	
		7	R[1]	0	SCR Compare Interrupt	4-5
			W	0	SCR Compare Mask	
2	2	0	R[1]	0	Pack Data Ready Interrupt	4-6
			W	0	Pack Data Ready Mask	
		1	R[1]	0	Audio PES Data Ready Interrupt	4-6
			W	0	Audio PES Data Ready Mask	
		2	R[1]	0	Video PES Data Ready Interrupt	4-6
			W	0	Video PES Data Ready Mask	
		3	R[1]	0	SPU PES Data Ready Interrupt	4-6
			W	0	SPU PES Data Ready Mask	
		4	R[1]	0	Seq End Code in Video Channel Interrupt	4-6
			W	0	Seq End Code in Video Channel Mask	
		5	R[1]	0	DSI/PCI PES Data Ready Interrupt	4-7
			W	0	DSI/PCI PES Data Ready Mask	

(Sheet 2 of 6)

Table 3.2 Host Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
2	2	6	R[1]	0	DTS Audio Event Interrupt	4-7
			W	0	DTS Audio Event Mask	
		7	R[1]	0	DTS Video Event Interrupt	4-7
			W	0	DTS Video Event Mask	
3	3	0	R[1]	0	Audio ES Channel Buffer Overflow Interrupt	4-8
			W	0	Audio ES Channel Buffer Overflow Mask	
		1	R[1]	0	Video ES Channel Buffer Overflow Interrupt	4-8
			W	0	Video ES Channel Buffer Overflow Mask	
		2	R[1]	0	SPU Channel Buffer Overflow Interrupt	4-8
			W	0	SPU Channel Buffer Overflow Mask	
		3	R[1]	0	Data Dump Channel PES Data Ready Interrupt	4-8
			W	0	Data Dump Channel PES Data Ready Mask	
		4	R[1]	0	Audio ES Channel Buffer Underflow Interrupt	4-8
			W	0	Audio ES Channel Buffer Underflow Mask	
		5	R[1]	0	Video ES Channel Buffer Underflow Interrupt	4-9
			W	0	Video ES Channel Buffer Underflow Mask	
		6	R[1]	0	SPU Channel Buffer Underflow Interrupt	4-9
			W	0	SPU Channel Buffer Underflow Mask	
		7		—	Reserved	
4	4	0	R[1]	0	VLC or Run Length Error Interrupt	4-9
			W	0	VLC or Run Length Error Mask	
		1	R[1]	0	Context Error Interrupt	
			W	0	Context Error Mask	

(Sheet 3 of 6)

Table 3.2 Host Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
4	4	2	R[1]	0	Audio CRC or Illegal Bit Error Interrupt	4-10
			W	0	Audio CRC or Illegal Bit Error Mask	
		3	R[1]	0	Audio Sync Error Interrupt	4-10
			W	0	Audio Sync Error Mask	
		4	R[1]	0	SPU Decode Error Interrupt	4-10
			W	0	SPU Decode Error Mask	
		5		—	Reserved	
		6	R[1]	0	Packet Error Interrupt	4-10
			W	0	Packet Error Interrupt Mask	
		7	R[1]	0	S/P DIF Channel Buffer Underflow Interrupt	4-10
			W	0	S/P DIF Channel Buffer Underflow Mask	
5	5	0	R/W	0	Invert Channel Clock	4-11
		1	R/W	0	Channel Request Mode	
		2	R/W	0	Channel Pause	
		3	R/W	0	Channel Bypass Enable	
		4	R	—	AREQ Status	
		5	R	—	VREQ Status	4-12
		7:6		—	Reserved	
6	6	0	W	0	Clear Interrupt Pin	4-12
		7:1		00	Reserved	
7	7	0	R	0	Channel Status	4-12
			W	0	Channel Start/Reset	4-13
		1	R/W	1	Reserved	

(Sheet 4 of 6)

Table 3.2 Host Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
7	7	3:2	R/W	0	Stream Select [1:0]	4-13
		4	R/W	0	SCR Pause	
		5	W	0	Software Reset	4-14
		7:6		0	Reserved	
8	8	7:0		—	Reserved	
9	9	7:0	R/W	00	SCR Value [7:0]	4-14
10	0A	7:0	R/W	00	SCR Value [15:8]	
11	0B	7:0	R/W	00	SCR Value [23:16]	
12	0C	7:0	R/W	00	SCR Value [31:24]	
13	0D	7:0	R/W	FF	SCR Compare/Capture [7:0]	4-14
14	0E	7:0	R/W	FF	SCR Compare/Capture [15:8]	
15	0F	7:0	R/W	FF	SCR Compare/Capture [23:16]	
16	10	7:0	R/W	FF	SCR Compare/Capture [31:24]	
17	11	1:0	R/W	0	SCR Compare/Capture Mode [1:0]	4-15
		2	R/W	0	Capture on Picture Start Code	
		3	R/W	0	Capture on Audio Sync Code	
		4	R/W	0	Capture on Beginning of Active Video	
		5	R/W	0	Capture on Pack Data Ready	4-16
		6	R/W	0	Capture on Audio PES Ready	
		7	R/W	0	Capture on Video PES Ready	
(Sheet 5 of 6)						

Table 3.2 Host Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
18	12	0	R/W	0	Capture on SPU PES Ready	4-16
		1		—	Reserved	
		2	R/W	0	Capture on DSI PES Ready	4-16
		3	R/W	0	Capture on DTS Video	
		4	R/W	0	Capture on DTS Audio	4-17
		7:5		—	Reserved	
19	13	0	R/W	0	Audio Start on Compare	4-17
		1	R/W	0	Video Start on Compare	
		7:2		—	Reserved	
20	14	7:0	R/W	FF	SCR Compare Audio [7:0]	4-17
21	15	7:0	R/W	FF	SCR Compare Audio [15:8]	
22	16	7:0	R/W	FF	SCR Compare Audio [23:16]	
23	17	7:0	R/W	FF	SCR Compare Audio [31:24]	
24–27	18–1B	7:0		—	Reserved	
28	1C	7:0	W	—	Video Channel Bypass Data [7:0]	4-18
29	1D	7:0	W	—	Audio Channel Bypass Data [7:0]	
30–63	1E–3F	7:0		—	Reserved	
(Sheet 6 of 6)						

[1] Reset after read.

Table 3.3 Video Decoder Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
64	40	0	W	0	Reset Aux Data FIFO	4-19
		1:0	R	0	Aux Data FIFO Status [1:0]	
		4:2	R	—	Aux Data Layer ID [2:0]	
		7:5		—	Reserved	
65	41	0	W	0	Reset User Data FIFO	4-20
		1:0	R	0	User Data FIFO Status [1:0]	
		3:2	R	—	User Data Layer ID [1:0]	
		7:4		—	Reserved	
66	42	7:0	R	—	User Data FIFO Output [7:0]	4-21
67	43	7:0	R	—	Aux Data FIFO Output [7:0]	
68	44	0	W	0	Reset Channel Buffer on Error	4-21
		1	W	0	Reset Audio PES Header/System Channel Buffer	
		2	W	0	Reset Video PES Header/SPU Channel Buffer	4-22
		3	W	0	Reset Data Dump Channel Buffer	
		4	W	0	Reset Navi Pack Channel Buffer	
		5	W	0	Reset Video ES Channel Buffer	
		6	W	0	Reset Audio ES Channel Buffer	
		7		—	Reserved	
69	45	0	R/W	0	Enable Video Read Compare DTS	4-22
		2:1	R/W	0	Enable Audio Read Compare DTS [1:0]	4-23
		4:3	R/W	0	Video Numitems/pics Panic Mode Select [1:0]	
		7:5		—	Reserved	

(Sheet 1 of 8)

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
70	46	7:0		—	Reserved	
71	47	7:0		—	Reserved	
72	48	7:0	R/W	—	Video ES Channel Buffer Start Address [7:0][1]	4-24
73	49	5:0	R/W	—	Video ES Channel Buffer Start Address [13:8][1]	
		7:6		—	Reserved	
74	4A	7:0	R/W	—	Video ES Channel Buffer End Address [7:0][1]	4-24
75	4B	5:0	R/W	—	Video ES Channel Buffer End Address [13:8][1]	
		7:6		—	Reserved	
76	4C	7:0	R/W	—	Audio ES Channel Buffer Start Address [7:0][1]	4-25
77	4D	5:0	R/W	—	Audio ES Channel Buffer Start Address [13:8][1]	
		7:6		—	Reserved	
78	4E	7:0	R/W	—	Audio ES Channel Buffer End Address [7:0][1]	4-25
79	4F	5:0	R/W	—	Audio ES Channel Buffer End Address [13:8][1]	
		7:6		—	Reserved	
80	50	7:0	R/W	—	Video PES Header/SPU Channel Buffer Start Address [7:0][1]	4-26
81	51	5:0	R/W	—	Video PES Header/SPU Channel Buffer Start Address [13:8][1]	
		7:6		—	Reserved	
82	52	7:0	R/W	—	Video PES Header/SPU Channel Buffer End Address [7:0][1]	4-26
83	53	5:0	R/W	—	Video PES Header/SPU Channel Buffer End Address [13:8][1]	
		7:6		—	Reserved	
84	54	7:0	R/W	—	Data Dump Channel Buffer Start Address [7:0][1]	4-27

(Sheet 2 of 8)

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
85	55	5:0	R/W	—	Data Dump Channel Buffer Start Address [13:8][1]	4-27
		7:6		—	Reserved	
86	56	7:0	R/W	—	Data Dump Channel Buffer End Address [7:0][1]	4-27
87	57	5:0	R/W	—	Data Dump Channel Buffer End Address [13:8][1]	
		7:6		—	Reserved	
88	58	7:0	R/W	—	Audio PES Header/System Channel Buffer Start Address [7:0][1]	4-28
89	59	5:0	R/W	—	Audio PES Header/System Channel Buffer Start Address [13:8][1]	
		7:6		—	Reserved	
90	5A	7:0	R/W	—	Audio PES Header/System Channel Buffer End Address [7:0][1]	4-28
91	5B	5:0	R/W	—	Audio PES Header/System Channel Buffer End Address [13:8][1]	
		7:6		—	Reserved	
92	5C	7:0	R/W	—	Navi Pack Channel Buffer Start Address [7:0][1]	4-29
93	5D	5:0	R/W	—	Navi Pack Channel Buffer Start Address [13:8][1]	
		7:6		—	Reserved	
94	5E	7:0	R/W	—	Navi Pack Channel Buffer End Address [7:0][1]	4-29
95	5F	5:0	R/W	—	Navi Pack Channel Buffer End Address [13:8][1]	
		7:6		—	Reserved	
96	60	7:0	R	—	Video ES Channel Buffer Write Address [7:0][2]	4-30
97	61	7:0	R	—	Video ES Channel Buffer Write Address [15:8][2]	
98	62	3:0	R	—	Video ES Channel Buffer Write Address [19:16][2]	
		7:4		—	Reserved	

(Sheet 3 of 8)

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
99	63	7:0	R	—	Audio ES Channel Buffer Write Address [7:0]^2	4-30
100	64	7:0	R	—	Audio ES Channel Buffer Write Address [15:8]^2	
101	65	3:0	R	—	Audio ES Channel Buffer Write Address [19:16]^2	
		7:4		—	Reserved	
102	66	7:0	R	—	Video PES Header/SPU Channel Buffer Write Address [7:0]^2	4-31
103	67	7:0	R	—	Video PES Header/SPU Channel Buffer Write Address [15:8]^2	
104	68	3:0	R	—	Video PES Header/SPU Channel Buffer Write Address [19:16]^2	
		7:4		—	Reserved	
105	69	7:0	R	—	Data Dump Channel Buffer Write Address [7:0]^2	4-31
106	6A	7:0	R	—	Data Dump Channel Buffer Write Address [15:8]^2	
107	6B	3:0	R	—	Data Dump Channel Buffer Write Address [19:16]^2	
		7:4		—	Reserved	
108	6C	7:0	R	—	Video ES Channel Buffer Read Address [7:0]^2	4-32
			W	—	Video ES Channel Buffer Compare DTS Address [7:0]	
109	6D	7:0	R	—	Video ES Channel Buffer Read Address [15:8]^2	
			W	—	Video ES Channel Buffer Compare DTS Address [15:8]	
110	6E	3:0	R	—	Video ES Channel Buffer Read Address [19:16])^2	
		2:0	W	—	Video ES Channel Buffer Compare DTS Address [18:16]	
		7:4		—	Reserved	
(Sheet 4 of 8)						

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
111	6F	7:0	R	—	Audio ES Channel Buffer Read Address [7:0][2]	4-33
			W	—	Audio ES Channel Buffer Compare DTS Address [7:0]	
112	70	7:0	R	—	Audio ES Channel Buffer Read Address [15:8][2]	4-33
			W	—	Audio ES Channel Buffer Compare DTS Address [15:8]	
113	71	3:0	R	—	Audio ES Channel Buffer Read Address [19:16][2]	4-33
		2:0	W	—	Audio ES Channel Buffer Compare DTS Address [18:16]	
		7:4		—	Reserved	
114	72	7:0	R	—	Audio PES Header/System Channel Buffer Write Address [7:0][3]	4-34
115	73	7:0	R	—	Audio PES Header/System Channel Buffer Write Address [15:8][3]	
116	74	3:0	R	—	Audio PES Header/System Channel Buffer Write Address [19:16][3]	
		7:4		—	Reserved	
117	75	7:0	R	—	Navi Pack Channel Buffer Write Address [7:0][3]	4-35
118	76	7:0	R	—	Navi Pack Channel Buffer Write Address [15:8][3]	
119	77	3:0	R	—	Navi Pack Channel Buffer Write Address [19:16][3]	
		7:4		—	Reserved	
120	78	7:0	R	—	S/P DIF Channel Buffer Read Address [7:0][3]	4-35
121	79	7:0	R	—	S/P DIF Channel Buffer Read Address [15:8][3]	
122	7A	3:0	R	—	S/P DIF Channel Buffer Read Address [19:16][3]	
		7:4		—	Reserved	
(Sheet 5 of 8)						

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
123	7B	0	W	—	Navi Pause	4-36
		1	W	—	Pack Pause	
		2	R/W	—	Force Sequence End Code	
		3	R/W	—	Flush Audio	
		4	W	—	Navi Pack Counter Enable	
		5	W	—	Navi Pack Counter Decrement	4-37
		7:6	R	—	Navi Pack Counter Output [1:0]	
124	7C	4:0	W	00	MPEG Audio Extension Stream ID [4:0]	4-37
		7:5		—	Reserved	
125–127	7D–7F	7:0		—	Reserved	
128	80	7:0	R	—	Picture Start Code Read Address [7:0][3]	4-38
129	81	7:0	R	—	Picture Start Code Read Address [15:8][3]	
130	82	3:0	R	—	Picture Start Code Read Address [19:16][3]	
		7:4		—	Reserved	
131	83	7:0	R	—	Audio Sync Code Read Address [7:0][3]	4-38
132	84	7:0	R	—	Audio Sync Code Read Address [15:8][3]	
133	85	3:0	R	—	Audio Sync Code Read Address [19:16][3]	
		7:4		—	Reserved	
134	86	7:0	R	00	Video ES Channel Buffer Numitems [7:0][2]	4-39
			W	—	Video Numitems/Pics in Channel Buffer Compare Panic [7:0][2]	

(Sheet 6 of 8)

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
135	87	7:0	R	00	Video ES Channel Buffer Numitems [15:8][2]	4-39
			W	—	Video Numitems/Pics in Channel Buffer Compare Panic [15:8][2]	
136	88	2:0	R	00	Video ES Channel Buffer Numitems [18:16[2]]	4-39
			W	—	Video Numitems/Pics in Channel Buffer Compare Panic [18:16][2]	
		7:3		—	Reserved	
137	89	7:0	R	00	Audio ES Channel Buffer Numitems [7:0][2]	4-40
138	8A	7:0	R	00	Audio ES Channel Buffer Numitems [15:8][2]	
139	8B	2:0	R	0	Audio ES Channel Buffer Numitems [18:16][2]	
		7:3		—	Reserved	
140	8C	7:0	R	00	S/P DIF Channel Buffer Numitems [7:0][2]	4-40
141	8D	7:0	R	00	S/P DIF Channel Buffer Numitems [15:8][2]	
142	8E	2:0	R	0	S/P DIF Channel Buffer Numitems [18:16][2]	
		7:3		—	Reserved	
143	8F	4:0	W	0	Audio Stream ID [4:0]	4-41
		7:5	W	0	Audio Stream Select Enable [2:0]	
144	90	0	W	0	Transport Private Stream Audio	4-42
		7:1		—	Reserved	
145	91	3:0	W	0	Video Stream ID [3:0]	4-42
		5:4	W	0	Video Stream Select Enable [1:0]	
		7:6	R/W	0	Video PES Headers Enable [1:0]	4-43

(Sheet 7 of 8)

Table 3.3 Video Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
146	92	4:0	W	00	SPU Substream ID [4:0]	4-43
		5	W	0	SPU Stream Select Enable	4-43
		6	W	0	Data Dump Stream Select Enable	4-43
		7		—	Reserved	
147	93	1:0	R/W	0	Audio PES Header Enable [1:0]	4-44
		3:2	R/W	0	System Header Enable [1:0]	4-44
		5:4	R/W	0	Pack Header Enable [1:0]	4-45
		7:6	W	0	Navi Pack PES Header Enable [1:0]	4-45
148	94	0	W	0	Top of Sector Detect Enable	4-45
		7:1		—	Reserved	
149	95	0	R	0	Audio Packet Error Status[4]	4-46
		1	R	0	Video Packet Error Status[4]	4-46
		2	R	0	DSI/PCI Packet Error Status[4]	4-46
		3		—	Reserved	
		4	R	0	SPU Packet Error Status[4]	4-46
		7:5		—	Reserved	
150	96	7:0	R	—	Pictures in Video ES Channel Buffer Counter [7:0]	4-46
151	97	7:0	R	—	Pictures in Video ES Channel Buffer Counter [15:8]	
152–191	98–BF	7:0		—	Reserved	
(Sheet 8 of 8)						

1. The channel must be stopped to access these registers. Addresses SDRAM at 256-byte boundaries.
2. SDRAM addresses at 8-byte boundaries. A 1 in the most significant bit indicates that the circular buffer has executed a “wraparound.” Bytes must be read in a least, next, and most significant order.
3. SDRAM addresses at 8-byte boundaries. Bytes must be read in a least, next, and most significant order.
4. Reset after read.

Table 3.4 Memory Interface Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
192	C0	0	R	1	Host Read FIFO Empty	4-47
		1	R	0	Host Read FIFO Full	
		2	R	1	Host Write FIFO Empty	
		3	R	0	Host Write FIFO Full	
		4	R	1	DMA Read FIFO Empty	
		5	R	0	DMA Read FIFO Full	
		6	R	1	DMA Write FIFO Empty	
		7	R	0	DMA Write FIFO Full	
193	C1	0		—	Reserved	
		2:1	R/W	0	DMA Mode [1:0] (idle, DMA, R/W, block move)	4-47
		3	R/W	0	Host SDRAM Transfer Byte Ordering	4-49
		5:4	R/W	0	Refresh Extend [1:0]	
		6	R/W	0	DMA SDRAM Transfer Byte Ordering	
		7		—	Reserved	
194	C2	7:0	R	—	Host SDRAM Read Data [7:0]	4-49
195	C3	7:0	W	00	Host SDRAM Write Data [7:0]	4-50
196	C4	7:0	R/W	00	Host SDRAM Target Address [7:0]	4-50
197	C5	7:0	R/W	00	Host SDRAM Target Address [15:8]	
198	C6	2:0	R/W	0	Host SDRAM Target Address [18:16]	
		7:3		—	Reserved	
199	C7	7:0	R/W	00	Host SDRAM Source Address [7:0] (host ONLY)	4-50
200	C8	7:0	R/W	00	Host SDRAM Source Address [15:8] (host ONLY)	
(Sheet 1 of 3)						

Table 3.4 Memory Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
201	C9	2:0	R/W	0	Host SDRAM Source Address [18:16] (host ONLY)	4-50
		7:3		—	Reserved	
202	CA	7:0	R/W	FF	Block Transfer Count [7:0]	4-51
203	CB	7:0	R/W	FF	Block Transfer Count [15:8]	
204	CC	0	R/W	0	PLL Test	4-51
		2:1		—	Reserved	
		3	R	0	Clk Out of Sync	4-51
		5:4	R/W	1	Control for Programmable Delay Path 1 [1:0]	4-52
		7:6	R/W	1	Control for Programmable Delay Path 2 [1:0]	
205	CD	0	R	0	Phase Locked Status	4-52
		2:1	R	0	Internal Lock Counter State [1:0]	4-53
		5:3	R	0	Internal SDRAM State [2:0]	
		7:6		0	Reserved	
206	CE	1:0	R	—	Internal Phase State (3 cycles before) [1:0]	4-53
		3:2	R	—	Internal Phase State (2 cycles before) [1:0]	
		5:4	R	—	Internal Phase State (1 cycle before) [1:0]	
		7:6	R	0	Internal Phase State (current cycle) [1:0]	
207	CF	7:0	R/W	80	Phase Detect Test High Freq [7:0]	4-54
208	D0	7:0	R/W	00	Phase Detect Test High Freq [15:8]	
209	D1	7:0	R/W	00	Phase Detect Test Low Freq [7:0]	4-54
210	D2	7:0	R/W	01	Phase Detect Test Low Freq [15:8]	
211	D3	7:0	R/W	A2	VCO Test High Freq [7:0]	4-54
212	D4	7:0	R/W	00	VCO Test High Freq [15:8]	

(Sheet 2 of 3)

Table 3.4 Memory Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
213	D5	7:0	R/W	00	DMA SDRAM Target Address [7:0]	4-55
214	D6	7:0	R/W	00	DMA SDRAM Target Address [15:8]	
215	D7	2:0	R/W	0	DMA SDRAM Target Address [18:16]	
		7:3		—	Reserved	
216	D8	7:0	R/W	00	DMA SDRAM Source Address [7:0]	4-55
217	D9	7:0	R/W	00	DMA SDRAM Source Address [15:8]	
218	DA	2:0	R/W	0	DMA SDRAM Source Address [18:16]	
		7:3		—	Reserved	
219	DB	7:0	R	—	DMA SDRAM Read Data [7:0]	4-56
220	DC	7:0	W	00	DMA SDRAM Write Data [7:0]	
221	DD	0	R	—	PLL Phase Detect High Frequency Test Pass	4-56
		1	R	—	PLL Phase Detect Low Frequency Test Pass	
		2	R	—	PLL VCO High Frequency Test Pass	
		3	R	—	PLL VCO Low Frequency Test Pass	
		7:4		—	Reserved	
222	DE	7:0	R/W	B4	VCO Test Low Frequency [7:0]	4-56
223	DF	7:0	R/W	00	VCO Test Low Frequency [15:8]	

(Sheet 3 of 3)

Table 3.5 Microcontroller Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
224	E0	7:0	R/W	—	Anchor Luma Frame Store 1 Base Address [7:0][1]	4-57
225	E1	7:0	R/W	—	Anchor Luma Frame Store 1 Base Address [15:8][1]	
226	E2	7:0	R/W	—	Anchor Chroma Frame Store 1 Base Address [7:0][1]	4-57
227	E3	7:0	R/W	—	Anchor Chroma Frame Store 1 Base Address [15:8][1]	
228	E4	7:0	R/W	—	Anchor Luma Frame Store 2 Base Address [7:0][1]	4-57
229	E5	7:0	R/W	—	Anchor Luma Frame Store 2 Base Address [15:8][1]	
230	E6	7:0	R/W	—	Anchor Chroma Frame Store 2 Base Address [7:0][1]	4-58
231	E7	7:0	R/W	—	Anchor Chroma Frame Store 2 Base Address [15:8][1]	
232	E8	7:0	R/W	—	B Luma Frame Store Base Address [7:0][1]	4-58
233	E9	7:0	R/W	—	B Luma Frame Store Base Address [15:8][1]	
234	EA	7:0	R/W	—	B Chroma Frame Store Base Address [7:0][1]	4-58
235	EB	7:0	R/W	—	B Chroma Frame Store Base Address [15:8][1]	
236	EC	1:0	R	0	Video Skip Frame Status [1:0]	4-59
			W	0	Video Skip Frame Mode [1:0]	
		2	R	0	Video Continuous Skip Status	
			W	0	Video Continuous Skip Mode	
		7:3		—	Reserved	
237	ED	0	R	0	Video Repeat Frame Status	4-60
			W	0	Video Repeat Frame Enable	
		1	R	0	Video Continuous Repeat Frame Status	
			W	0	Video Continuous Repeat Frame Mode	
		7:2		—	Reserved	

(Sheet 1 of 3)

Table 3.5 Microcontroller Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
238	EE	0	R	0	Rip Forward Mode Status	4-61
			W	0	Rip Forward Mode Enable	
		1	R	0	Rip Forward Display Single Step Status	4-62
			W	0	Rip Forward Display Single Step Command	
		3:2	R	—	Current Display Frame [1:0]	
		5:4	R	—	Current Decode Frame [1:0]	
		7:6		—	Reserved	
239	EF	0		—	Reserved	
		1	R/W	0	Host Force Broken Link Mode	4-63
		2	R/W	0	Panic Prediction Enable	
		3	R/W	0	GOP User Data Only	
		4	R/W	0	Concealment Copy Option	4-64
		5	R/W	0	Force Rate Control	
		6	R/W	0	Ignore Sequence End	
		7		—	Reserved	
240	F0	0	R	0	Host Broken Link/Seq Status	4-65
			W	0	Host Search Broken Link/Seq Command	
		7:1		00	Reserved	
241	F1	0	R	—	Q Table Ready	4-65
		1	R/W	—	Intra Q Table	
		7:2	R/W	—	Q Table Address [5:0]	
242	F2	7:0	R	—	Q Table Entry [7:0]	4-66

(Sheet 2 of 3)

Table 3.5 Microcontroller Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
243	F3	7:0	R	—	Microcontroller PC [7:0]	4-66
244	F4	3:0	R	—	Microcontroller PC [11:8]	
		7:4		—	Reserved	
245	F5	7:0	R	—	Revision Number [7:0]	4-66
246	F6	0	W	0	Decode Start/Stop Command	4-66
		7:1		—	Reserved	
247	F7	7:0		—	Reserved	
248	F8	0	R/W	0	Reduced Memory Mode (RMM)	4-67
		7:1		—	Reserved	
249– 255	F9– FF	7:0		—	Reserved	
(Sheet 3 of 3)						

1. SDRAM addresses at 64-byte boundaries.

Table 3.6 Video Interface Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
256–264	100–108	7:0		—	Reserved	
265	109	1:0	R/W	0	OSD Mode [1:0]	4-67
		2		—	Reserved	
		3	R	—	OSD Palette Counter Zero Flag	4-68
			W	0	Clear OSD Palette Counter	
		5:4	R/W	0	Display Override Mode [1:0]	
		7:6	R/W	0	Force Video Background [1:0]	
266	10A	7:0	R/W	23	Programmable Background Y[7:0]	4-69
267	10B	7:0	R/W	D4	Programmable Background Cb[7:0]	
268	10C	7:0	R/W	72	Programmable Background Cr[7:0]	
269	10D	7:0	W	—	OSD Palette Write [7:0]	4-69
270	10E	7:0	R/W	—	OSD Odd Field Pointer [7:0][1]	4-70
271	10F	7:0	R/W	—	OSD Odd Field Pointer [15:8][1]	
272	110	7:0	R/W	—	OSD Even Field Pointer [7:0][1]	4-70
273	111	7:0	R/W	—	OSD Even Field Pointer [15:8][1]	
274	112	3:0	R/W	0	OSD Mix Weight [3:0]	4-70
		4	R/W	0	OSD Chroma Filter Enable	
		5	R/W	0	SPU Chroma Filter Enable	
		6	R/W	0	Horizontal Decimation Filter Enable	
		7		—	Reserved	

(Sheet 1 of 5)

Table 3.6 Video Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
275	113	1:0	R/W	0	Freeze Mode [1:0]	4-71
		2	R/W	1	3:2 Pulldown from Bitstream	
		3	R/W	0	Host Repeat First Field	
		4	R/W	1	Host Top Field First	
		5	R	—	First Field	
		6	R	—	Odd/Not Even Field	4-72
		7	R	—	Top/Not Bottom Field	
276	114	0	R	—	Last Field	4-72
		1	R/W	0	Horizontal Filter Enable	
		2	R/W	—	Horizontal Filter Select	
		6:3	R/W	—	Display Mode [3:0]	
		7	R/W	0	Field Sync Enable	4-73
277	115	7:0	R/W	—	Horizontal Filter Scale [7:0]	4-73
278	116	6:0	R/W	—	Main Reads per Line [6:0]	4-74
		7		—	Reserved	
279	117	2:0	R/W	—	Pan and Scan 1/8 Pixel Offset [2:0]	4-74
		5:3	R/W	—	Pan and Scan Byte Offset [2:0]	
		6	R/W	1	Pan and Scan from Bitstream	
		7	R/W	0	Automatic Field Inversion Correction	
280	118	7:0	R/W	—	Horizontal Pan and Scan Luma/Chroma Word Offset [7:0]	4-75
281	119	7:0	R/W	—	Vertical Pan and Scan Line Offset [7:0]	
(Sheet 2 of 5)						

Table 3.6 Video Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
282	11A	2:0	R/W	1	Vline Count Init [2:0]	4-75
		7:3		—	Reserved	
283	11B	6:0	R/W	—	Override Picture Width [6:0]	4-75
		7		—	Reserved	
284	11C	0	R/W	0	ITU-R BT.656 Mode	4-76
		1	R/W	0	Sync Active Low	
		2		—	Reserved	
		4:3	R/W	2	Pixel State Reset Value [1:0]	4-76
		5	R/W	0	CrCb 2's Complement	
		6	R/W	0	VSYNC Input Type	
		7		—	Reserved	
285	11D	7:0	R/W	—	Display Override Luma Frame Store Start Address [7:0][1]	4-77
286	11E	7:0	R/W	—	Display Override Luma Frame Store Start Address [15:8][1]	
287	11F	7:0	R/W	—	Override Display Chroma Frame Store Start Address [7:0][1]	
288	120	7:0	R/W	—	Override Display Chroma Frame Store Start Address [15:8][1]	
289	121	0		—	Reserved	
		6:1	R/W	2C	Number of Segments in RMM [5:0]	4-77
		7		—	Reserved	
290	122	1:0	W	0	Television Standard Select [1:0]	4-78
		7:2		—	Reserved	

(Sheet 3 of 5)

Table 3.6 Video Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
291–296	123–128	7:0		—	Reserved	
297	129	7:0	R/W	—	Main Start Row [7:0]	4-78
298	12A	7:0	R/W	—	Main End Row [7:0]	
299	12B	2:0	R/W	—	Main Start Row [10:8]	
		3		—	Reserved	
		6:4	R/W	—	Main End Row [10:8]	
		7		—	Reserved	
300	12C	7:0	R/W	—	Main Start Column [7:0]	4-79
301	12D	7:0	R/W	—	Main End Column [7:0]	
302	12E	2:0	R/W	—	Main Start Column [10:8]	
		3		—	Reserved	
		6:4	R/W	—	Main End Column [10:8]	
		7		—	Reserved	
303	12F	4:0	R/W	—	Vcode Zero [4:0]	4-79
		5	R/W	—	Vcode Even [8]	
		6	R/W	—	Vcode Even Plus 1	
		7	R/W	—	Fcode [8]	
304	130	7:0	R/W	—	Vcode Even [7:0]	4-80
305	131	7:0	R/W	—	Fcode [7:0]	
306	132	7:0	R/W	—	SAV Start Column [7:0]	
307	133	7:0	R/W	—	EAV Start Column [7:0]	

(Sheet 4 of 5)

Table 3.6 Video Interface Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
308	134	2:0	R/W	—	SAV Start Column [10:8]	4-80
		3		—	Reserved	
		6:4	R/W	—	EAV Start Column [10:8]	4-80
		7		—	Reserved	
309	135	0	R/W	0	Display Start Command	4-81
		1	R/W	1	SPU Mix Enable	
		7:2		—	Reserved	
310–335	136–14F	7:0		—	Reserved	
(Sheet 5 of 5)						

Table 3.7 Audio Decoder Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
336	150	3:0	R	—	MPEG - bitrate_index [3:0]	4-81
		4	R	—	MPEG - protection_bit	4-82
		6:5	R	—	MPEG - layer_code [1:0]	
		7	R	—	MPEG - ID	4-83
337	151	0	R	—	MPEG - copyright	4-83
		2:1	R	—	MPEG - mode_extension [1:0]	
		4:3	R	—	MPEG - mode [1:0]	4-84
		5	R	—	MPEG - private_bit	
		7:6	R	—	MPEG - sampling_frequency [1:0]	
338	152	4:0		—	Reserved	
		6:5	R	—	MPEG - emphasis [1:0]	4-84
		7	R	—	MPEG - original/copy	4-85
339	153	4:0	R	—	Dolby Digital - dialnorm [4:0]	4-85
		7:5	R	—	Dolby Digital - acmod [2:0]	
340	154	4:0	R	—	Dolby Digital - dialnorm2 [4:0]	4-85
		7:5	R	—	Dolby Digital - bsmod [2:0]	
341	155	0		—	Reserved	
		5:1	R	—	Dolby Digital - mixlevel [4:0]	4-86
		7:6	R	—	Dolby Digital - surmixlev [1:0]	
342	156	0		—	Reserved	
		5:1	R	—	Dolby Digital - mixlevel2 [4:0]	4-86
		7:6	R	—	Dolby Digital - cmixlev [1:0]	

(Sheet 1 of 5)

Table 3.7 Audio Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
343	157	4:0	R	—	Dolby Digital - bsid [4:0]	4-87
		5	R	—	Dolby Digital - lfeon	
		7:6	R	—	Dolby Digital - fscod [1:0]	
344	158	5:0	R	—	Dolby Digital - frmsizecod [5:0]	4-87
		7:6	R	—	Dolby Digital - dsurmod [1:0]	
345	159	7:0	R	—	Dolby Digital - langcod [7:0]	4-88
346	15A	7:0	R	—	Dolby Digital - langcod2 [7:0]	
347	15B	7:0	R	—	Dolby Digital - timecod1 [13:6]	
348	15C	1:0	R	—	Dolby Digital - roomtype [1:0]	4-88
		7:2	R	—	Dolby Digital - timecod1 [5:0]	
349	15D	7:0	R	—	Dolby Digital - timecod2 [13:6]	4-89
350	15E	1:0	R	—	Dolby Digital - roomtype2 [1:0]	
		7:2	R	—	Dolby Digital - timecod2 [5:0]	
351	15F	2:0	R	—	PCM - num_of_audio_ch [2:0]	4-89
		7:3	R	—	PCM - audio_frm_num [4:0]	
352	160	0		—	Reserved	
		1	R	—	PCM - mute_bit	4-90
		3:2	R	—	PCM - emphasis [1:0]	
		5:4	R	—	PCM - quantization [1:0]	
		7:6	R	—	PCM - Fs [1:0]	

(Sheet 2 of 5)

Table 3.7 Audio Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
353	161	4:0		—	Reserved	
		5	R	—	PCM FIFO Empty	4-90
		6	R	—	PCM FIFO Near Full	
		7	R	—	PCM FIFO Full	
354	162	1:0	R	—	Audio Decoder Play Mode Status [1:0]	4-91
		2	R	—	Audio Decoder Soft Mute Status	
		3	R	—	Audio Decoder Reconstruct Error	
		4	R	—	MPEG Multichannel Extension Sync Word Missing	
		7:5		—	Reserved	
355	163	4:0		—	Reserved	
		6:5	R/W	0	Audio Decoder Play Mode [1:0]	4-92
		7	R/W	0	Audio Decoder Start/Stop	
356	164	4:0		—	Reserved	
		6:5	R/W	0	Audio Formatter Play Mode [1:0]	4-93
		7	R/W	0	Audio Formatter Start/Stop	
357	165	4:0		—	Reserved	
		7:5	R/W	0	Audio Decoder Mode Select [2:0]	4-93
358	166	1:0	R/W	2	Dolby Digital Compression Mode [1:0]	4-94
		3:2	R/W	0	Audio Dual-Mono Mode [1:0]	4-95
		4	R/W	0	Dolby Digital Downmix Mode	
		5		—	Reserved	
		6	R/W	0	User Mute Bit	4-96
		7	R/W	1	Mute on Error	

(Sheet 3 of 5)

Table 3.7 Audio Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
359	167	7:0	W	—	PCM FIFO Data In [7:0]	4-96
360	168	7:0	R/W	FF	Dolby Digital/Linear PCM - dynscalehigh [7:0]	
361	169	7:0	R/W	FF	Dolby Digital/Linear PCM - dynscalelow [7:0]	4-97
362	16A	7:0	R/W	FF	PCM Scale [7:0]	
363	16B	1:0	R/W	1	ACLK Select [1:0]	4-97
		2	R/W	0	Invert LRCLK	4-98
		5:3		—	Reserved	
		6	R/W	0	User	4-98
		7	R/W	1	Valid	
364	16C	3:0	R/W	0	ACLK Divider Select [3:0]	4-98
		4	R/W	0	LPCM - Dynamic Range On	4-100
		5	R/W	0	Dolby Digital - Karaoke Center Level On	
		7:6	R/W	0	Dolby Digital - Karaoke Mode [1:0]	
365	16D	1:0		—	Reserved	
		4:2	R/W	0	IEC - Host Emphasis [2:0]	4-101
		5	R/W	0	IEC - Overwrite Emphasis	
		6	R/W	0	IEC - Host Copyright	
		7	R/W	0	IEC - Overwrite Copyright	

(Sheet 4 of 5)

Table 3.7 Audio Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
366	16E	0	R/W	0	Overwrite Category	4-102
		2:1	R/W	0	Host Overwrite Quantization [1:0]	
		3	R/W	0	Overwrite Quantization Enable	
		4	R/W	0	MPEG Formatter Only	4-103
		6:5	R/W	0	Formatter Skip Frame Size [1:0]	
		7	R/W	0	Formatter ES1 Compliant	
367	16F	7:0	R/W	00	Host Category [7:0]	4-103
368	170	0		—	Reserved	
		1	R	—	Pd Data Valid	4-104
		2		—	Reserved	
		4:3	R/W	0	Pd Selection [1:0]	4-104
		7:5	R/W	0	Host Pc Info [2:0]	
369	171	7:0	R/W	00	Host Pd Value [15:8]	4-105
370	172	7:0	R/W	00	Host Pd Value [7:0]	
371– 383	173– 17F	7:0		—	Reserved	

(Sheet 5 of 5)

Table 3.8 RAM Test Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
384	180	7:0	R/W	00	Memory Test Address [7:0]	4-105
385	181	3:0	R/W	0	Memory Test Address [11:8]	
		7:4		—	Reserved	
386	182	1:0	W	0	Operational Mode for RAM Test [1:0]	4-106
		2	R	0	Report End of Test	
			W	0	Initiate Memory Test	
		4:3	W	0	Data Pattern to be Applied to RAM [1:0]	
		5	R/W	0	Memory Test Output Select	
		7:6		—	Reserved	
387	183	0	R	1	MemTest01 Pass/Fail Status[1]	4-107
		1	R	1	MemTest02 Pass/Fail Status[1]	
		2	R	1	MemTest03 Pass/Fail Status[1]	
		3	R	1	MemTest04 Pass/Fail Status[1]	
		4	R	1	MemTest05 Pass/Fail Status[1]	
		5	R	1	MemTest06 Pass/Fail Status[1]	
		6	R	1	MemTest07 Pass/Fail Status[1]	
		7	R	1	MemTest08 Pass/Fail Status[1]	
388	184	0	R	1	MemTest09 Pass/Fail Status[1]	4-107
		1	R	1	MemTest10 Pass/Fail Status[1]	
		2	R	1	MemTest11 Pass/Fail Status[1]	
		3	R	1	MemTest12 Pass/Fail Status[1]	

(Sheet 1 of 3)

Table 3.8 RAM Test Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
388	184	4	R	1	MemTest13 Pass/Fail Status[1]	4-107
		5	R	1	MemTest14 Pass/Fail Status[1]	
		6	R	1	MemTest15 Pass/Fail Status[1]	
		7	R	1	MemTest16 Pass/Fail Status[1]	
389	185	0	R	1	MemTest17 Pass/Fail Status[1]	4-107
		1	R	1	MemTest18 Pass/Fail Status[1]	
		2	R	1	MemTest19 Pass/Fail Status[1]	
		3	R	1	MemTest20 Pass/Fail Status[1]	
		4	R	1	MemTest21 Pass/Fail Status[1]	
		5	R	1	MemTest22 Pass/Fail Status[1]	
		6	R	1	MemTest23 Pass/Fail Status[1]	
		7	R	1	MemTest24 Pass/Fail Status[1]	
390	186	0	R	1	MemTest25 Pass/Fail Status[1]	4-107
		1	R	1	MemTest26 Pass/Fail Status[1]	
		2	R	1	MemTest27 Pass/Fail Status[1]	
		3	R	1	MemTest28 Pass/Fail Status[1]	
		4	R	1	MemTest29 Pass/Fail Status[1]	
		5	R	1	MemTest30 Pass/Fail Status[1]	
		6	R	1	MemTest31 Pass/Fail Status[1]	
		7	R	1	MemTest32 Pass/Fail Status[1]	

(Sheet 2 of 3)

Table 3.8 RAM Test Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
391	187	0	R	1	MemTest33 Pass/Fail Status[1]	4-107
		1	R	1	MemTest34 Pass/Fail Status[1]	
		2	R	1	MemTest35 Pass/Fail Status[1]	
		3	R	1	MemTest36 Pass/Fail Status[1]	
		7:4		—	Reserved	
392	188	0	R	1	MemTest37 Pass/Fail Status[1]	4-107
		1	R	1	MemTest38 Pass/Fail Status[1]	
		2	R	1	MemTest39 Pass/Fail Status[1]	
		6:3		—	Reserved	
		7	R	1	Overall MemTest Pass/Fail Status[1]	4-107
393–415	189–19F	7:0		—	Reserved	
(Sheet 3 of 3)						

[1] Reset after read.

Table 3.9 SPU Decoder Registers

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
416	1A0	0	R/W	0	SPU Decode Start	4-107
		1	R/W	0	SPU Pause	
		2	R/W	0	SPU Display Off	4-108
		3	R/W	0	SPU Display Force Off	
		4	R/W	0	Reset Autofill Counter for SPU Palette	
		5	R/W	0	Jump to Next SPU	
		6	R/W	0	Frame-Based Execution	
		7			Reserved	
417	1A1	3:0	R/W	0	Command Time-Out for SPU [3:0]	4-108
		7:4		—	Reserved	
418	1A2	7:0	R	FF	PTS in Current SPU [7:0]	4-109
419	1A3	7:0	R	FF	PTS in Current SPU [15:8]	
420	1A4	7:0	R	FF	PTS in Current SPU [23:16]	
421	1A5	7:0	R	FF	PTS in Current SPU [31:24]	
422	1A6	7:0	R	FF	PTS in Next SPU [7:0]	4-109
423	1A7	7:0	R	FF	PTS in Next SPU [15:8]	
424	1A8	7:0	R	FF	PTS in Next SPU [23:16]	
425	1A9	7:0	R	FF	PTS in Next SPU [31:24]	
426	1AA	7:0	R	00	SP_DCSQ_STM in Next DCSQ [7:0]	4-110
427	1AB	7:0	R	00	SP_DCSQ_STM in Next DCSQ [15:8]	

(Sheet 1 of 3)

Table 3.9 SPU Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
428	1AC	7:0	R	00	SPU Base Pointer [7:0][1]	4-110
429	1AD	7:0	R	00	SPU Base Pointer [15:8][1]	
430	1AE	3:0	R	0	SPU Base Pointer [19:16][1]	
		7:4		—	Reserved	
431– 445	1AF– 1BD			—	Reserved	
446	1BE	7:0	W	00	Color Palette Data for SPU [7:0]	4-110
447	1BF	0	R/W	0	Highlight Enable	4-111
		7:1		—	Reserved	
448	1C0	7:0	W	00	Highlight Color Info [7:0]	4-111
449	1C1	7:0	W	00	Highight Color Info [15:8]	
450	1C2	7:0	W	00	Highlight Contrast Info [7:0]	4-111
451	1C3	7:0	W	00	Highlight Contrast Info [15:8]	
452	1C4	7:0	W	00	Highlight Area Info [7:0]	4-112
453	1C5	7:0	W	00	Highlight Area Info [15:8]	
454	1C6	7:0	W	00	Highlight Area Info [23:16]	
455	1C7	7:0	W	00	Highlight Area Info [31:24]	
456	1C8	7:0	W	00	Highlight Area Info [39:32]	
457	1C9	7:0	W	00	Highlight Area Info [47:40]	

(Sheet 2 of 3)

Table 3.9 SPU Decoder Registers (Cont.)

Addr (Dec)	Addr (Hex)	Bit(s)	R/W	Default Value (Hex)	Status/Command/Data	Page Ref.
458	1CA	0	R	—	PXD FIFO Underflow	4-112
		1	R	—	Time Stamp Error	
		2	R	—	Syntax Error	4-113
		3	R	—	Unit Error	
		4	R	—	Unit Store Error	
458	1CA	5	R	—	Size Error	4-113
		6	R	—	Sync Word Error	
		7	R	—	Illegal Unit Error Flag	
459	1CB	7:0		—	Reserved	
460	1CC	7:0	R	01	SPU State Machine Info [7:0] (Diagnostics only)	4-114
461–479	1CD–1DF	7:0		—	Reserved	
480–511	1E0–1FF	7:0		—	Reserved	

(Sheet 3 of 3)

1. SDRAM addresses at 8-byte boundaries. A 1 in the most significant bit indicates that the circular buffer has executed a “wraparound.” Bytes must be read in a least, next, and most significant order.

3.2 Alphabetical Listing

Chapter 4
Register Descriptions

This chapter describes the bit and field assignments of all of the non-reserved registers in the L64020. The chapter contains the following sections:

To locate a specific register, field, or bit, use the register summary in Chapter 3.

4.1 Host Interface Registers

Figure 4.1 Register 0 (0x000)

	7	6	5	4	3	2	1	0
Read	New Field Interrupt	Audio Sync Recovery Interrupt	SPU SCR Compare Interrupt	SDRAM Transfer Done Interrupt	Sequence End Code Detect Interrupt	First Slice Start Code Detect Interrupt	Aux/User Data FIFO Ready Interrupt	Decode Status Interrupt
Write	New Field Mask	Audio Sync Recovery Mask	SPU SCR Compare Mask	SDRAM Transfer Done Mask	Sequence End Code Detect Mask	First Slice Start Code Detect Mask	Aux/User Data FIFO Ready Mask	Decode Status Mask

Decode Status Interrupt **0**

This bit is set when the video decode status changes from stopped to running (0 to 1) and cleared when the status changes from running to stopped (1 to 0). Either status change causes assertion of the INTRn interrupt signal to the host if not masked. The 0 to 1 transition occurs on a picture start code boundary after channel start. It is linked in timing to the last field of the display system. The decode status is updated internally and may change when one of the following events is recognized by the internal microcontroller:

1. A write to the Decode Start/Stop Command register (page 4-66) by the host.

2. When the Video Start on Compare register (page 4-17) is set by the host and a compare occurs. In this case, the status goes from stopped to running.

Reading this register does NOT change the Decode Status bit.

INTRn is not asserted if the host sets the mask bit.

Aux/User Data FIFO Ready Interrupt **1**

When set, indicates there is new data in the Aux or User Data FIFO ready to be read. A NOT ready (0) to ready (1) change causes assertion of the INTRn signal if not masked. The status of the Aux Data FIFO (page 4-19) and User Data FIFO (page 4-20) can be read to determine which has valid data. The bit is cleared on reading. Even though data remains in the FIFOs, no further interrupts are generated.

INTRn is not asserted if the host sets the mask bit.

First Slice Start Code Detect Interrupt **2**

This bit is set when the decoder detects the first slice start code after the picture layer. INTRn is asserted unless the host sets the mask bit.

Sequence End Code Detect Interrupt **3**

This bit is set when the decoder detects a sequence end code. INTRn is asserted unless the host sets the mask bit.

SDRAM Transfer Done Interrupt **4**

This bit is set when an SDRAM block move is completed. INTRn is asserted unless the host sets the mask bit.

SPU SCR Compare Interrupt **5**

This bit is set when the System Clock Reference (SCR) catches up to and equals the Presentation Time Stamp (PTS) extracted from the SPU PES header. This signals the SPU Decoder to start. The INTRn interrupt signal to the host is also asserted unless the host sets the mask bit.

Audio Sync Recovery Interrupt **6**

The audio sync recovery bit is set when sync is re-established after any errors. The conditions which trigger it are:

1. for MPEG, when three good frames are detected after synchronization was lost.

2. for Dolby Digital, when the first good frame is detected after the sync was lost. There is no sync recovery triggered during the power-on condition.

This bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

New Field Interrupt **7**

This bit is set after a short delay after the termination of the Vertical Sync pulse from the PAL/NTSC Encoder. INTRn is also asserted unless the host sets the mask bit.

Figure 4.2 Register 1 (0x001)

	7	6	5	4	3	2	1	0
Read	SCR Compare Interrupt	SCR Overflow Interrupt	Begin Vertical Blank Interrupt	Begin Active Video Interrupt	SPU Start Code Detect Interrupt	SCR Compare Audio Interrupt	Picture Start Code Detect Interrupt	Audio Sync Code Detect Interrupt
Write	SCR Compare Mask	SCR Overflow Mask	Begin Vertical Blank Mask	Begin Active Video Mask	SPU Start Code Detect Mask	SCR Compare Audio Mask	Picture Start Code Detect Mask	Audio Sync Code Detect Mask

Audio Sync Code Detect Interrupt **0**

This bit is set when the Audio Decoder detects a valid audio sync code. The interrupt is intended to be used for synchronization of presentation units. This is achieved by sampling the System Clock Reference (SCR) using the capture register function of the SCR. Also at this time, the decoder samples the channel read pointers and maintains the audio sync code read address and the picture start code address. These addresses are the current read pointers which are generally 48 addresses higher than the picture start code and 8 addresses higher than the audio sync code (due to the size of the top of channel FIFOs). These can be related to the channel buffer address stored at the time of the Packetized Elementary Stream (PES) packet header when the packet entered the system to allow correlating the packet to the particular picture or audio frame contained in that packet.

This bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

Picture Start Code Detect Interrupt **1**

This bit is set when the decoder detects a picture start code in the bitstream. The bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

SCR Compare Audio Interrupt **2**

This bit is set when the System Clock Reference (SCR) Compare Audio value in Registers 20, 21, 22, and 23 (page 4-17) matches the current SCR value. The SCR Compare Audio value is different from the main SCR Compare value.

This bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

SPU Start Code Detect Interrupt **3**

This bit is set when a Subpicture Unit (SPU) start code is detected by the SPU Decoder Module. This bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

Begin Active Video Interrupt **4**

The Video Interface module sets this bit and asserts INTRn (if not masked) at the beginning of active video. This time is defined by the vertical blanking code (Vcode) in the Start of Active Video/End of Active Video (SAV/EAV) timing codes programmed into the Video Interface.

This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Begin Vertical Blank Interrupt **5**

The Video Interface module sets this bit and asserts INTRn (if not masked) at the beginning of the vertical blanking interval. This time is defined by the Vcode in the Start of Active Video/End of Active Video (SAV/EAV) timing codes programmed into the Video Interface.

This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

SCR Overflow Interrupt **6**

This bit is set and when the System Clock Reference (SCR) counter (page 4-14) overflows. This bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

SCR Compare Interrupt **7**

This bit is set when the System Clock Reference (SCR) Compare mode is enabled and a match between the value stored in the SCR Compare/Capture registers (page 4-14) and the current value of the SCR occurs.

This bit is cleared when read. INTRn is also asserted unless the host sets the mask bit.

Figure 4.3 Register 2 (0x002)

	7	6	5	4	3	2	1	0
Read	DTS Video Event Interrupt	DTS Audio Event Interrupt	DSI/PCI PES Data Ready Interrupt	Seq End Code in Video Channel Interrupt	SPU PES Data Ready Interrupt	Video PES Data Ready Interrupt	Audio PES Data Ready Interrupt	Pack Data Ready Interrupt
Write	DTS Video Event Mask	DTS Audio Event Mask	DSI/PCI PES Data Ready Mask	Seq End Code in Video Channel Mask	SPU PES Data Ready Mask	Video PES Data Ready Mask	Audio PES Data Ready Mask	Pack Data Ready Mask

Pack Data Ready Interrupt **0**

This bit is set and INTRn is asserted (if not masked) by the preparser when it detects the start of a pack. The interrupt alerts the host that the pack header, system header, and first packet pointer are in the channel buffer. This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Audio PES Data Ready Interrupt **1**

This bit is set and INTRn is asserted (if not masked) by the preparser when it detects an audio PES packet. This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Video PES Data Ready Interrupt **2**

This bit is set and INTRn is asserted (if not masked) by the preparser when it detects a video PES packet. This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

SPU PES Data Ready Interrupt **3**

This bit is set and INTRn is asserted (if not masked) by the preparser when it detects an SPU PES packet in the channel. This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Seq End Code in Video Channel Interrupt **4**

This bit is set and INTRn is asserted (if not masked) by the preparser when it detects a sequence end code in the video channel. This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

DSI/PCI PES Data Ready Interrupt **5**

This bit is set and INTRn is asserted (if not masked) by the preparser when it detects a DSI or PCI PES packet. This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

DTS Audio Event Interrupt **6**

When the chip is in the Audio Read Compare mode (Register 69, bits 1 and 2, page 4-23), the channel buffer controller generates a single cycle pulse when the read pointer in the channel buffer matches a preset value (Registers 111, 112, and 113, page 4-33). At the pulse, an internal state machine waits for an audio sync code, sets this bit, and then generates an interrupt by asserting the INTRn output signal. The interrupt is used for audio/video synchronization.

This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

DTS Video Event Interrupt **7**

When the chip is in the Video Read Compare mode (Register 69, bit 0, page 4-22), the channel buffer controller generates a single cycle pulse when the read pointer in the channel buffer matches to a preset value (Registers 108, 109, and 110, page 4-32). At the pulse, an internal state machine waits for a picture start code, sets this bit, and then generates an interrupt by asserting the INTRn output signal. The interrupt is used for audio/video synchronization.

This bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Figure 4.4 Register 3 (0x003)

	7	6	5	4	3	2	1	0
Read	Reserved	SPU Channel Buffer Underflow Interrupt	Video ES Channel Buffer Underflow Interrupt	Audio ES Channel Buffer Underflow Interrupt	Data Dump Channel PES Data Ready Interrupt	SPU Channel Buffer Overflow Interrupt	Video ES Channel Buffer Overflow Interrupt	Audio ES Channel Buffer Overflow Interrupt
Write	Reserved	SPU Channel Buffer Underflow Mask	Video ES Channel Buffer Underflow Mask	Audio ES Channel Buffer Underflow Mask	Data Dump Channel PES Data Ready Mask	SPU Channel Buffer Overflow Mask	Video ES Channel Buffer Overflow Mask	Audio ES Channel Buffer Overflow Mask

Audio ES Channel Buffer Overflow Interrupt **0**

This bit is set and INTRn is asserted (if not masked) when the Audio ES channel buffer in SDRAM overflows. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Video ES Channel Buffer Overflow Interrupt **1**

This bit is set and INTRn is asserted (if not masked) when the Video ES channel buffer in SDRAM overflows. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

SPU Channel Buffer Overflow Interrupt **2**

This bit is set and INTRn is asserted (if not masked) when the SPU channel buffer in SDRAM overflows. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Data Dump Channel PES Data Ready Interrupt **3**

This bit is set and INTRn is asserted (if not masked) when the preparser detects data written to the Data Dump channel buffer in SDRAM. INTRn is not asserted if the host sets the mask bit.

Audio ES Channel Buffer Underflow Interrupt **4**

This bit is set and INTRn is asserted (if not masked) when the Audio ES channel buffer in SDRAM underflows (becomes empty). The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Video ES Channel Buffer Underflow Interrupt **5**

This bit is set and INTRn is asserted (if not masked) when the Video ES channel buffer in SDRAM underflows (becomes empty). The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

SPU Channel Buffer Underflow Interrupt **6**

This bit is set and INTRn is asserted (if not masked) when the SPU channel buffer in SDRAM underflows (becomes empty). The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Reserved **7**

Figure 4.5 Register 4 (0x004)

	7	6	5	4	3	2	1	0
Read	S/P DIF Channel Buffer Underflow Interrupt	Packet Error Interrupt	Reserved	SPU Decode Error Interrupt	Audio Sync Error Interrupt	Audio CRC or Illegal Bit Error Interrupt	Context Error Interrupt	VLC or Run Length Error Interrupt
Write	S/P DIF Channel Buffer Underflow Mask	Packet Error Interrupt Mask	Reserved	SPU Decode Error Mask	Audio Sync Error Mask	Audio CRC or Illegal Bit Error Mask	Context Error Mask	VLC or Run Length Error Mask

VLC or Run Length Error Interrupt **0**

This bit is set and INTRn is asserted (if not masked) when an illegal variable length code (VLC) is detected in the bitstream, for example:

1. when a start code is found in an unexpected location in the bitstream, or

2. when there is an error in the run-length parameters supplied to the IDCT unit.

The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Context Error Interrupt **1**

This bit is set and INTRn is asserted (if not masked) when the Video Decoder detects a parameter in the bitstream that is not consistent with the context, e.g., an illegal value. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Audio CRC or Illegal Bit Error Interrupt **2**

This bit is set and INTRn is asserted (if not masked) by the Audio Decoder when it detects a CRC or illegal bit error. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Audio Sync Error Interrupt **3**

This bit is set and INTRn is asserted (if not masked) when an audio sync code is not in the expected location in the bitstream. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

SPU Decode Error Interrupt **4**

This bit is set and INTRn is asserted (if not masked) when an error is detected in the SPU data. The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Reserved **5**

Packet Error Interrupt **6**

This bit is set and INTRn is asserted (if not masked) when the preparser detects an error while processing packet data. When this interrupt occurs, the host should read the Packet Error Status register (page 4-46) to determine in which packet the error occurred.

The Packet Error Interrupt bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

S/P DIF Channel Buffer Underflow Interrupt **7**

This bit is set and INTRn is asserted (if not masked) when the S/P DIF read pointer in the Audio ES channel buffer catches up to the write pointer (all buffer data read to the S/P DIF Formatter).

The bit is cleared when read. INTRn is not asserted if the host sets the mask bit.

Figure 4.6 Register 5 (0x005)

7	6	5	4	3	2	1	0
Reserved		VREQ Status	AREQ Status	Channel Bypass Enable	Channel Pause	Channel Request Mode	Invert Channel Clock

Invert Channel Clock **R/W 0**

When this bit is set, the internal DCK is inverted from the external DCK clock. By default, the host interface accepts the DCK and ACLK signals and ORs them together to generate the internal VALID signal. This assumes that channel data is available immediately after the rising edge of DCK. For systems in which the data is available immediately after the falling edge of DCK, this bit needs to be set so that the internal VALID signal can be generated on the falling edge of DCK. Asynchronous systems can tie DCK to ground.

Channel Request Mode **R/W 1**

By default, the L64020 expects an external device to sample the REQn (AREQn and VREQn) signals synchronously with the system clock of the L64020. If the external device requires the REQn signals to be synchronous with the external device clock (DCK), then the Channel Request Mode bit needs to be set. In this mode, the channel internal request is sampled twice, first by the rising edge of internal DCK and then by the falling edge of internal DCK, before being sent out as a REQn signal.

Channel Pause **R/W 2**

Setting this bit prevents the channel request signals (AREQn and VREQn) from being asserted so channel data is not transferred into the L64020. The external host must clear this bit to reassert the REQn signals.

Channel Bypass Enable **R/W 3**

Setting this bit allows the host to write data directly to the channel, bypassing the parallel channel input port. Video ES or Audio ES channel data can be written into Registers 28 or 29 respectively (page 4-18) when in this mode. At reset, this register defaults to 0, i.e., no bypass.

AREQ Status **R 4**

This bit is set when the AREQn signal in the chip is asserted. This bit position is read only.

VREQ Status **R 5**

This bit is set when the VREQn signal in the chip is asserted. This bit position is read only.

Reserved **[7:6]**

Figure 4.7 Register 6 (0x006)

7	6	5	4	3	2	1	0
Reserved							Clear Interrupt Pin

Clear Interrupt Pin **W 0**

This bit is used to clear the interrupt signal, INTRn, of previous pending interrupts. In normal operation, events in the L64020 can cause INTRn to be asserted if the event mask is cleared. The bits in the interrupt registers (register 0 through 4) are cleared when read by the host. However, INTRn remains asserted until all the interrupt registers are read (all bits cleared) and the Clear Interrupt Pin bit is set.

This separate control is provided for systems with priority interrupts since this will allow the driver software to exit the interrupt handler before completion and service higher priority interrupts. While INTRn is still asserted, the interrupt handler returns to the interrupt routine for the L64020 when it is again the highest priority interrupt.

Reserved **[7:1]**

Figure 4.8 Register 7 (0x007)

7	6	5	4	3	2	1	0
Reserved		Software Reset	SCR Pause	Stream Select [1:0]		Reserved	Channel Status R Channel Start/Reset W

Channel Status **R 0**

This bit indicates the status of the channel at any time. At reset or power-up, this bit is cleared to indicate that the channel is stopped. When the Channel Start command is issued (host writes a 1 to this bit position), the L64020 microcontroller updates this bit to a 1 indicating that the channel start command has been acknowledged and the channel has started. When a Channel Reset command is issued (host writes a 0 to this bit position) the L64020 microcontroller updates this bit to a 0 indicating acknowledgment of the Channel Reset command and that the channel is currently stopped.

Channel Start/Reset **W 0**

Setting this bit starts the channel. Clearing it stops the channel.

Reserved **R/W 1**

The default value of this bit is 1 and should NOT be overwritten with 0.

Stream Select [1:0] **R/W [3:2]**

The host must program these bits to set up the L64020 for the format of the input bitstream as shown in the following table.

Stream Select [1:0]	Bitstream Format
0b00	A/V PES Packets
0b01	MPEG-1 System or MPEG-2 Program Stream
0b10	(Not defined)
0b11	A/V Elementary Streams

A 0b11 in these bits causes the L64020 to skip packet searching and byte count matching. Video data is taken in at the first start code. Subsequent start codes re-establish the byte alignment. Audio data is not byte aligned in the channel buffer.

For 0b00 through 0b10, the L64020 parses from the packet layer and resynchronizes the preparser to the packet and pack layer start codes on any packet layer errors.

SCR Pause **R/W 4**

When set, this bit prevents the SCR Counter (Figure 4.9) from incrementing. However, the SCR Counter can still be written to by the host (override). When this bit is cleared, the SCR Counter operates in normal mode, i.e., it increments with the system clock. At power-on and reset, this bit is initialized to 0.

Software Reset **W 5**

When set by the host, this bit causes the L64020 to reset (reinitialize). The effect is the same as asserting the hard reset signal of the chip, RESETn. This reset function generates a 10-clock cycle reset pulse that resets all internal modules. All host register values are reinitialized and need to be reconfigured by the host for proper operation.

Reserved **[7:6]**

Register 8 (0x008) **Reserved** **[7:0]**

Figure 4.9 Registers 9–12 (0x009–0x00C) SCR Value [31:0]

	7	0
Reg. 9 LSB	SCR Value [7:0] R/W	
Reg. 10	SCR Value [15:8] R/W	
Reg. 11	SCR Value [23:16] R/W	
Reg. 12 MSB	SCR Value [31:24] R/W	

These registers contain the current value of the System Clock Reference (SCR) Counter. The host must read Register 9, the LSB, first. This captures the upper 24 bits and writes them into Registers 10, 11, and 12. The host must set the SCR Pause bit in Register 8 before writing to these registers.

Figure 4.10 Registers 13–16 (0x00D–0x010) SCR Compare/Capture [31:0]

	7	0
Reg. 13 LSB	SCR Compare/Capture [7:0] R/W	
Reg. 14	SCR Compare/Capture [15:8] R/W	
Reg. 15	SCR Compare/Capture [23:16] R/W	
Reg. 16 MSB	SCR Compare/Capture [31:24] R/W	

At reset, these registers are initialized to 0xFFFF.FFFF. They can be configured in two ways. If the SCR Compare/Capture Mode in Register

17 is set to 0b10, the host can write in any value to generate an interrupt when the SCR Counter reaches that value.

If the SCR Compare/Capture Mode is set to 0b01, the L64020 captures the SCR Counter value at an event specified by the host and writes the SCR value to these registers. The capture can be triggered when any one of the bits in Registers 17 or 18 is set and the corresponding event occurs.

Figure 4.11 Register 17 (0x011)

7	6	5	4	3	2	1	0
Capture on Video PES Ready	Capture on Audio PES Ready	Capture on Pack Data Ready	Capture on BAV	Capture on Audio Sync Code	Capture on Picture Start Code	SCR Compare/Capture Mode [1:0]	

SCR Compare/Capture Mode [1:0] **R/W 1:0**

The value of these two bits sets the operating mode of Registers 13, 14, 15, and 16 as shown in the following table.

Mode Bits	Mode
0b00	No compare and capture. SCR overflow works.
0b01	Capture
0b10	Compare
0b11	Reserved

Capture on Picture Start Code **R/W 2**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects the Picture Start Code.

Capture on Audio Sync Code **R/W 3**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects the Audio Sync Code.

Capture on Beginning of Active Video (BAV) **R/W 4**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects the Beginning of Active Video.

Capture on Pack Data Ready **R/W 5**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects Pack Data Ready.

Capture on Audio PES Ready **R/W 6**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects Audio PES Ready.

Capture on Video PES Ready **R/W 7**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects Video PES Ready.

Figure 4.12 Register 18 (0x012)

7	5	4	3	2	1	0
Reserved		Capture on DTS Audio	Capture on DTS Video	Capture on DSI PES Ready	Reserved	Capture on SPU PES Ready

Capture on SPU PES Ready **R/W 0**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects Subpicture Unit (SPU) PES Ready.

Reserved **1**

Capture on DSI PES Ready **R/W 2**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects Data Search Information (DSI) PES Ready.

Capture on DTS Video **R/W 3**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects Decode Time Stamp (DTS) Video.

Capture on DTS Audio **R/W 4**

When this bit is set and the L64020 is in the Capture Mode, the SCR Counter value is captured and written to Registers 13 through 16 when the preparser detects DTS Audio.

Reserved **[7:5]**

Figure 4.13 Register 19 (0x013)

7 … 2	1	0
Reserved	Video Start on Compare	Audio Start on Compare

Audio Start on Compare **R/W 0**

When the L64020 is in the Compare Mode, setting this bit generates a single-cycle, autostart pulse for starting the Audio Decoder when the current value of the SCR Counter is equal to the value in the SCR Compare Audio register. This autostart pulse also clears the Audio Start on Compare bit. The Audio Decoder must be in Pause Mode for the autostart signal to be effective.

Video Start on Compare **R/W 1**

When the L64020 is in the Compare Mode, setting this bit generates a single-cycle, autostart pulse to start the Video Decoder when current value of the SCR Counter is equal to the value in the SCR Compare register. This bit is cleared after the autostart signal is generated.

Reserved **[7:2]**

Figure 4.14 Registers 20–23 (0x014–0x017) SCR Compare Audio [31:0]

	7 … 0
Reg. 20 LSB	SCR Compare Audio [7:0] R/W
Reg. 21	SCR Compare Audio [15:8] R/W
Reg. 22	SCR Compare Audio [23:16] R/W
Reg. 23 MSB	SCR Compare Audio [31:24] R/W

When the Audio Start on Compare bit in Register 19 (Figure 4.13) is set, the SCR Compare/Capture mode is Compare, and the SCR Counter reaches the value in these registers, an autostart pulse is generated to start the Audio Decoder.

The compare also sets the SCR Compare Audio Interrupt bit (bit 2 in Register 1, page 4-4) and asserts the INTRn signal to the host if not masked. The Audio Start on Compare bit is cleared when the compare event occurs.

Registers 24–27 (0x018–0x01B) Reserved [7:0]

Figure 4.15 Register 28 (0x01C) Video Channel Bypass Data [7:0]

7	0
Video Channel Bypass Data [7:0] W	

Setting the Channel Bypass Enable bit (bit 3 in Register 5, page 4-11) allows the host to write data directly to the video channel through this register, bypassing the parallel channel input port.

Figure 4.16 Register 29 (0x01D) Audio Channel Bypass Data [7:0]

7	0
Audio Channel Bypass Data [7:0] W	

Setting the Channel Bypass Enable bit (bit 3 in Register 5, page 4-11) allows the host to write data directly to the audio channel through this register, bypassing the parallel channel input port.

Registers 30–63 (0x01E–0x03F) Reserved [7:0]

4.2 Video Decoder Registers

Figure 4.17 Register 64 (0x040)

<table>
<tr><td>7</td><td>5</td><td>4</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2" rowspan="2">Reserved</td><td colspan="2" rowspan="2">Aux Data Layer ID [2:0]</td><td colspan="2">Aux Data FIFO Status [1:0]
R</td></tr>
<tr><td>Read Only</td><td>Reset Aux Data FIFO
W</td></tr>
</table>

Aux Data FIFO Status [1:0] **R [1:0]**

The states of these bit indicate the status of the Aux Data FIFO as shown in the following table. Once “overrun” (0b11) occurs, the status stays at overrun until the register is read.

Bits	Status
0b00	Empty
0b01	Data ready
0b10	Full
0b11	Overrun

Reset Aux Data FIFO **W 0**

Writing a 1 to this bit resets the Aux Data FIFO to empty. Any data in the FIFO at this time is lost.

Aux Data Layer ID [2:0] **R [4:2]**

The Aux Data Layer ID indicates the layer origin of the physical parameter of the current Aux Data FIFO output. Reading the ID does NOT change the FIFO status. Reading the current byte in the Auxiliary Data FIFO Output register (page 4-21) may change the Aux Data Layer ID. The host should always read this layer ID register before reading the FIFO output register. The IDs for the layers are defined in the following table.

Bits	Layer
0b100	Packet
0b000	Sequence
0b001	Group of pictures
0b010	Picture
0b111	Extension layer (picture or sequence)

Reserved **[7:5]**

Figure 4.18 Register 65 (0x41)

<table>
<tr><td>7</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2" rowspan="2">Reserved</td><td colspan="2" rowspan="2">User Data Layer ID [1:0]</td><td colspan="2">User Data FIFO Status [1:0]
R</td></tr>
<tr><td>Read Only</td><td>Reset User Data FIFO
W</td></tr>
</table>

User Data FIFO Status [1:0] **R [1:0]**

The following table shows the user data FIFO status codes and their meanings. Once “overrun” (11) occurs it stays at overrun until the status is read.

Bits	Status
0b00	Empty
0b01	Data ready
0b10	Full
0b11	Overrun

Reset User Data FIFO **W 0**

Writing a 1 to this bit position resets the User Data FIFO to empty. Any data currently in the FIFO is lost.

User Data Layer ID [1:0] **R [3:2]**

The User Data Layer ID bits indicate the layer origin of the user data or extra data at the current User Data FIFO output. Reading the ID does NOT change the FIFO status. The IDs for the four layers are defined in the following table.

Bits	Layer
0b00	Sequence
0b01	Group of pictures
0b10	Picture
0b11	Slice

Reserved **[7:4]**

Figure 4.19 Register 66 (0x042) User Data FIFO Output [7:0]

<table>
<tr><th>7</th><th>0</th></tr>
<tr><td colspan="2">User Data FIFO Output [7:0]
Read Only</td></tr>
</table>

User data can be read out by the host one byte at a time through this read port. When a byte is read, the next byte in the FIFO is loaded into the register. See also Register 65.

Figure 4.20 Register 67 (0x043) Aux Data FIFO Output [7:0]

<table>
<tr><th>7</th><th>0</th></tr>
<tr><td colspan="2">Aux Data FIFO Output [7:0]
Read Only</td></tr>
</table>

Auxiliary data can be read out by the host microprocessor one byte at a time through this read port. When a byte is read, the next byte in the FIFO is loaded into the register. See also Register 64.

Figure 4.21 Register 68 (0x044)

7	6	5	4	3	2	1	0
Reserved	Reset Audio ES Channel Buffer	Reset Video ES Channel Buffer	Reset Navi Pack Channel Buffer	Reset Data Dump Channel Buffer	Reset Video PES Header/SPU Channel Buffer	Reset Audio PES Header/ System Channel Buffer	Reset Channel Buffer on Error

Reset Channel Buffer on Error **W 0**

Setting this bit causes the preparser to reset all channel buffers if it detects a packet sync error. If this bit is cleared, the preparser does not reset the channel buffers on a packet sync error.

Reset Audio PES Header/System Channel Buffer **W 1**

Setting this bit resets the write pointer of the Audio PES Header/System channel buffer to the buffer start address. A read pointer is not maintained for this buffer.

Reset Video PES Header/SPU Channel Buffer **W 2**

Setting this bit resets the write pointer of the Video PES Header/SPU channel buffer to the buffer start address. A read pointer is not maintained for this buffer.

Reset Data Dump Channel Buffer **W 3**

Setting this bit resets the write pointer of the Data Dump channel buffer to the buffer start address.

Reset Navi Pack Channel Buffer **W 4**

Setting this bit resets the write pointer of the Navi Pack channel buffer to the buffer start address.

Reset Video ES Channel Buffer **W 5**

Setting this bit resets the read and write pointers of the Video ES channel buffer to the buffer start address.

Reset Audio ES Channel Buffer **W 6**

Setting this bit resets the Audio ES channel buffer read and write pointers and the S/P DIF read pointer to the buffer start address.

Reserved **7**

Figure 4.22 Register 69 (0x045)

7 5	4 3	2 1	0
Reserved	Video Numitems/Pics Panic Mode Select [1:0]	Enable Audio Read Compare DTS [1:0]	Enable Video Read Compare DTS

Enable Video Read Compare DTS **R/W 0**

When this bit is set, the Video ES channel buffer read pointer is compared with the Video ES Channel Buffer Compare DTS Address written in Registers 108, 109, and 110 (page 4-32) by the host. When the two addresses match, the DTS Video Event Interrupt bit (Register 2, bit 7, page 4-7) is set and an interrupt is generated, if not masked, by asserting the INTRn output signal. This can be used as an aid to audio/video synchronization by the host software. When INTRn is asserted, the host should read Registers 0 through 4 to determine the cause of the interrupt, take the necessary action, and deassert INTRn by setting the Clear Interrupt Pin bit (Register 6, bit 0, page 4-12).

Enable Audio Read Compare DTS [1:0] R/W [2:1]

The bit encoding and meanings are shown in the following table.

Bits	Description
0b00	Disable compare
0b01	Audio decoder read pointer compare
0b10	IEC958 (S/P DIF) read pointer compare
0b11	Reserved

When these bits are configured for a compare, the selected Audio ES channel buffer read pointer is compared with the Audio ES Channel Buffer Compare DTS Address written in Registers 111, 112, and 113 (page 4-33) by the host. When the two addresses match, the DTS Audio Event Interrupt bit (Register 2, bit 6, page 4-7) is set and an interrupt is generated, if not masked, by asserting the INTRn output signal. This can be used as an aid to audio/video synchronization by the host software.

When INTRn is asserted, the host should read Registers 0 through 4 to determine the cause of the interrupt, take the necessary action, and deassert INTRn by setting the Clear Interrupt Pin bit (Register 6, bit 0, page 4-12).

Video Numitems/Pics Panic Mode Select [1:0] R/W [4:3]

This field allows the host to select a “panic” mode as shown in the following table.

Bits	Description
0b00	Disable panic feature
0b01	Video numitems panic mode
0b10	Pics-in-channel panic mode
0b11	Reserved

When enabled in either the Video Numitems Panic Mode or the Pics-in-channel Panic Mode, the Video Decoder suspends decoding when the number of items (64-bit words) or the number of complete encoded and compressed pictures in the Video ES channel buffer falls below the Video Channel Numitems threshold value written in Registers 134, 135, and 136 (page 4-39) by the host. This helps to handle potential video channel underflow situations gracefully without interrupting the host. During a panic situation, the display is frozen (freeze field) on the last picture displayed before the panic was recognized.

Reserved **[7:5]**

Registers 70 and 71 (0x046 and 0x047) Reserved [7:0]

Figure 4.23 Registers 72 and 73 (0x048 and 0x049) Video ES Channel Buffer Start Address [13:0]

	7	6	5	0
Reg. 72 LSB	Video ES Channel Buffer Start Address [7:0] R/W			
Reg. 73 MSB	Reserved		Video ES Channel Buffer Start Address [13:8] R/W	

These registers allow the host to program the Video ES channel buffer start address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2 M x 16 RAM. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.24 Registers 74 and 75 (0x04A and 0x04B) Video ES Channel Buffer End Address [13:0]

	7	6	5	0
Reg. 74 LSB	Video ES Channel Buffer End Address [7:0] R/W			
Reg. 75 MSB	Reserved		Video ES Channel Buffer End Address [13:8] R/W	

These registers allow the host to program the Video ES channel buffer end address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2 M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.25 Registers 76 and 77 (0x04C and 0x04D) Audio ES Channel Buffer Start Address [13:0]

	7	6	5	0
Reg. 76 LSB	Audio ES Channel Buffer Start Address [7:0] R/W			
Reg. 77 MSB	Reserved		Audio ES Channel Buffer Start Address [13:8] R/W	

These registers allow the host to program the Audio ES channel buffer start address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.26 Registers 78 and 79 (0x04E and 0x04F) Audio ES Channel Buffer End Address [13:0]

	7	6	5	0
Reg. 78 LSB	Audio ES Channel Buffer End Address [7:0] R/W			
Reg. 79 MSB	Reserved		Audio ES Channel Buffer End Address [13:8] R/W	

These registers allow the host to program the Audio ES channel buffer end address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.27 Registers 80 and 81 (0x050 and 0x051) Video PES Header/SPU Channel Buffer Start Address [13:0]

	7	6	5	0
Reg. 80 LSB	Video PES Header/SPU Channel Buffer Start Address [7:0] R/W			
Reg. 81 MSB	Reserved		Video PES Header/SPU Channel Buffer Start Address [13:8] R/W	

These registers allow the host to program the Video PES Header/SPU channel buffer start address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.28 Registers 82 and 83 (0x052 and 0x053) Video PES Header/SPU Channel Buffer End Address [13:0]

	7	6	5	0
Reg. 82 LSB	Video PES Header/SPU Channel Buffer End Address [7:0] R/W			
Reg. 83 MSB	Reserved		Video PES Header/SPU Channel Buffer End Address [13:8] R/W	

These registers allow the host to program the Video PES Header/SPU channel buffer end address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.29 Registers 84 and 85 (0x054 and 0x055) Data Dump Channel Buffer Start Address [13:0]

	7	6	5	0
Reg. 84 LSB	Data Dump Channel Buffer Start Address [7:0] R/W			
Reg. 85 MSB	Reserved		Data Dump Channel Buffer Start Address [13:8] R/W	

These registers allow the host to program the data dump channel buffer start address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.30 Registers 86 and 87 (0x056 and 0x057) Data Dump Channel Buffer End Address [13:0]

	7	6	5	0
Reg. 86 LSB	Data Dump Channel Buffer End Address [7:0] R/W			
Reg. 87 MSB	Reserved		Data Dump Channel Buffer End Address [13:8] R/W	

These registers allow the host to program the data dump channel buffer end address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.31 Registers 88 and 89 (0x058 and 0x059) Audio PES Header/System Channel Buffer Start Address [13:0]

	7	6	5	0
Reg. 88 LSB	Audio PES Header/System Channel Buffer Start Address [7:0] R/W			
Reg. 89 MSB	Reserved		Audio PES Header/System Channel Buffer Start Address [13:8] R/W	

These registers allow the host to program the Audio PES Header/System channel buffer start address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.32 Registers 90 and 91 (0x05A and 0x05B) Audio PES Header/System Channel Buffer End Address [13:0]

	7	6	5	0
Reg. 90 LSB	Audio PES Header/System Channel Buffer End Address [7:0] R/W			
Reg. 91 MSB	Reserved		Audio PES Header/System Channel Buffer End Address [13:8] R/W	

These registers allow the host to program the Audio PES Header/System channel buffer end address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.33 Registers 92 and 93 (0x05C and 0x05D) Navi Pack Channel Buffer Start Address [13:0]

<table>
<tr><th></th><th>7</th><th>6</th><th>5</th><th>0</th></tr>
<tr><td>Reg. 92
LSB</td><td colspan="4">Navi Pack Channel Buffer Start Address [7:0]
R/W</td></tr>
<tr><td>Reg. 93
MSB</td><td colspan="2">Reserved</td><td colspan="2">Navi Pack Channel Buffer Start Address [13:8]
R/W</td></tr>
</table>

These registers allow the host to program the Navi Pack channel buffer start address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.34 Registers 94 and 95 (0x05E and 0x05F) Navi Pack Channel Buffer End Address [13:0]

<table>
<tr><th></th><th>7</th><th>6</th><th>5</th><th>0</th></tr>
<tr><td>Reg. 94
LSB</td><td colspan="4">Navi Pack Channel Buffer End Address [7:0]
R/W</td></tr>
<tr><td>Reg. 95
MSB</td><td colspan="2">Reserved</td><td colspan="2">Navi Pack Channel Buffer End Address [13:8]
R/W</td></tr>
</table>

These registers allow the host to program the Navi Pack channel buffer end address. The address is entered as if it were the upper 14 bits of a 21-bit address for a conventional 2M x 16 RAM address. The Memory Interface of the L64020 converts the address to an SDRAM address at a 256-byte boundary in SDRAM. This register should only be updated while the channel is stopped (reset).

Figure 4.35 Registers 96–98 (0x060–0x062) Video ES Channel Buffer Write Address [19:0]

	7 4	3 0
Reg. 96 LSB	Video ES Channel Buffer Write Address [7:0] Read Only	
Reg. 97	Video ES Channel Buffer Write Address [15:8] Read Only	
Reg. 98 MSB	Reserved	Video ES Channel Buffer Write Address [19:16] Read Only

These registers contain the current write pointer address of the Video ES channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 97 and 98. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 98) indicates that the write pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.36 Registers 99–101 (0x063–0x065) Audio ES Channel Buffer Write Address [19:0]

	7 4	3 0
Reg. 99 LSB	Audio ES Channel Buffer Write Address [7:0] Read Only	
Reg. 100	Audio ES Channel Buffer Write Address [15:8] Read Only	
Reg. 101 MSB	Reserved	Audio ES Channel Buffer Write Address [19:16] Read Only

These registers contain the current write pointer address of the Audio ES channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 100 and 101. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 101) indicates that the write pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.37 Registers 102–104 (0x066–0x068) Video PES Header/SPU Channel Buffer Write Address [19:0]

	7	4	3	0
Reg. 102 LSB	Video PES Header/SPU Channel Buffer Write Address [7:0] Read Only			
Reg. 103	Video PES Header/SPU Channel Buffer Write Address [15:8] Read Only			
Reg. 104 MSB	Reserved		Video PES Header/SPU Channel Buffer Write Address [19:16] Read Only	

These registers contain the current write pointer address of the Video PES Header/SPU channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 103 and 104. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 104) indicates that the write pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.38 Registers 105–107 (0x069–0x06B) Data Dump Channel Buffer Write Address [19:0]

	7	4	3	0
Reg. 105 LSB	Data Dump Channel Buffer Write Address [7:0] Read Only			
Reg. 106	Data Dump Channel Buffer Write Address [15:8] Read Only			
Reg. 107 MSB	Reserved		Data Dump Channel Buffer Write Address [19:16] Read Only	

These registers contain the current write pointer address of the data dump channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 106 and 107. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 107) indicates that the write pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.39 Registers 108–110 (0x06C–0x06E) Video ES Channel Buffer Read Address [19:0]

	7 4	3 0
Reg. 108 LSB	Video ES Channel Buffer Read Address [7:0] Read Only	
Reg. 109	Video ES Channel Buffer Read Address [15:8] Read Only	
Reg. 110 MSB	Reserved	Video ES Channel Buffer Read Address [19:16] Read Only

These registers contain the current read pointer address of the Video ES channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 109 and 110. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 110) indicates that the read pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.40 Registers 108–110 (0x06C–0x06E) Video ES Channel Buffer Compare DTS Address [19:0]

	7 4	3 0
Reg. 108 LSB	Video ES Channel Buffer Compare DTS Address [7:0] Write	
Reg. 109	Video ES Channel Buffer Compare DTS Address [15:8] Write	
Reg. 110 MSB	Reserved	Video ES Channel Buffer Compare DTS Address [18:16] Write

The host can write a Video ES channel buffer address in these registers to be compared with the current read pointer address of the Video ES channel buffer. When the current read pointer address matches the contents of the registers and the chip is in the Video Read Compare Mode (Register 69, bit 0 set, page 4-22), the DTS Video Event Interrupt bit (Register 2, bit 7, page 4-7) is set and, if the interrupt is not masked, the INTRn output signal is asserted.

This can be used by the host as an aid to audio/video synchronization. When INTRn is asserted, the host should read Registers 0 through 4 to determine the cause of the interrupt, take the necessary action, and deassert INTRn by setting the Clear Interrupt Pin bit (Register 6, bit 0, page 4-12).

Figure 4.41 Registers 111–113 (0x06F–0x071) Audio ES Channel Buffer Read Address [19:0]

	7 4	3 0
Reg. 111 LSB	Audio ES Channel Buffer Read Address [7:0] Read Only	
Reg. 112	Audio ES Channel Buffer Read Address [15:8] Read Only	
Reg. 113 MSB	Reserved	Audio ES Channel Buffer Read Address [19:16] Read Only

These registers contain the current read pointer address of the Audio ES channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 112 and 113. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 113) indicates that the read pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.42 Registers 111–113 (0x06F–0x071) Audio ES Channel Buffer Compare DTS Address [19:0]

	7 3	2 0
Reg. 111 LSB	Audio ES Channel Buffer Compare DTS Address [7:0] Write	
Reg. 112	Audio ES Channel Buffer Compare DTS Address [15:8] Write	
Reg. 113 MSB	Reserved	Audio ES Channel Buffer Compare DTS Address [19:16] Write

The host can write an audio channel address in these registers to be compared with one of the current read pointer addresses of the Audio

ES channel buffer. When the selected current read pointer address matches the contents of the registers and the chip is in one of the Audio Read Compare modes (Register 69, bits 1 and 2, page 4-23), the DTS Audio Event Interrupt bit (Register 2, bit 6, page 4-7) is set and an interrupt is generated, if not masked, by asserting the INTRn output signal.

This can be used by the host as an aid to audio/video synchronization. When INTRn is asserted, the host should read Registers 0 through 4 to determine the cause of the interrupt, take the necessary action, and deassert INTRn by setting the Clear Interrupt Pin bit (Register 6, bit 0, page 4-12).

Figure 4.43 Registers 114–116 (0x072–0x074) Audio PES Header/System Channel Buffer Write Address [19:0]

<table>
<tr><th></th><th>7</th><th>4</th><th>3</th><th>0</th></tr>
<tr><td>Reg. 114
LSB</td><td colspan="4">Audio PES Header/System Channel Buffer Write Address [7:0]
Read Only</td></tr>
<tr><td>Reg. 115</td><td colspan="4">Audio PES Header/System Channel Buffer Write Address [15:8]
Read Only</td></tr>
<tr><td>Reg. 116
MSB</td><td colspan="2">Reserved</td><td colspan="2">Audio PES Header/System Channel Buffer Write Address [19:16]
Read Only</td></tr>
</table>

These registers contain the current write pointer address of the Audio PES Header/System channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 115 and 116. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 116) indicates that the write pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.44 Registers 117–119 (0x075–0x077) Navi Pack Channel Buffer Write Address [19:0]

	7 4	3 0
Reg. 117 LSB	Navi Pack Channel Buffer Write Address [7:0] Read Only	
Reg. 118	Navi Pack Channel Buffer Write Address [15:8] Read Only	
Reg. 119 MSB	Reserved	Navi Pack Channel Buffer Write Address [19:16] Read Only

These registers contain the current write pointer address of the Navi Pack channel buffer. The LSB should be read first, since this captures the next significant byte and MSB in Registers 118 and 119. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 119) indicates that the write pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.45 Registers 120–122 (0x078–0x07A) S/P DIF Channel Buffer Read Address [19:0]

	7 4	3 0
Reg. 120 LSB	S/P DIF Channel Buffer Read Address [7:0] Read Only	
Reg. 121	S/P DIF Channel Buffer Read Address [15:8] Read Only	
Reg. 122 MSB	Reserved	S/P DIF Channel Buffer Read Address [19:16] Read Only

These registers contain the current address of the S/P DIF (IEC958) read pointer in the Audio ES channel buffer. The LSB should be read first since this captures the next significant byte and MSB in Registers 121 and 122. These should then be read immediately to ensure that the correct captured value is read. When set, the most significant bit (bit 3 of Register 122) indicates that the read pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.46 Register 123 (0x07B)

7	6	5	4	3	2	1	0
Navi Pack Counter Output [1:0]		Navi Pack Counter Decrement	Navi Pack Counter Enable	Flush Audio	Force Sequence End Code	Pack Pause	Navi Pause

Navi Pause **W 0**

When this bit is set by the host, the stream parser in the L64020 pauses after the last Navi packet data is completely stored in the SDRAM buffer assigned to it. This function is designed especially for use with the Force Sequence End Code feature (Register 123, bit 2) and the Flush Audio feature (Register 123, bit 3). The stream parser stops at the next Navi pack boundary.

Pack Pause **W 1**

When this bit is set by the host, the stream parser in the device pauses after the last pack header data is completely stored in the SDRAM buffer assigned to it. This function is designed especially for use with the Force Sequence End Code feature (Register 123, bit 2) and the Flush Audio feature (Register 123, bit 3). The stream parser stops at the next pack header boundary.

Force Sequence End Code **R/W 2**

When this bit is set, the stream parser forces a Video Sequence End Code into the video channel just before the next Navi pack boundary if the Navi Pause bit is set and/or just before the pack header boundary if the Pack Pause bit is set. Note that the Sequence End Code is followed by sufficient zero bytes to flush all the video data from internal FIFOs into the video decode pipeline.

Flush Audio **R/W 3**

When this bit is set, the stream parser ensures that all audio data is written to the Audio ES channel buffer just before the next Navi pack boundary if the Navi Pause bit is set and/or the next Pack Header boundary if the Pack Pause bit is set.

Navi Pack Counter Enable **W 4**

When this bit is set, the Navi Pack Counter is enabled. The stream parser checks the counter before storing the next Navi pack. If the counter indicates that 0 or 1 Navi pack has been stored by the stream parser but not yet read by the host (counter values 0 or 1), the stream parser still reads the next Navi pack. When the counter indicates that there are two Navi packs stored but not read by the host (counter value 2), the stream parser pauses at the next Navi pack start point. At this point, the stream parser expects the host to read a Navi pack from the DSI channel in SDRAM and decrement the counter by writing to Register 123, bit 5, before storing the next Navi pack. Thus, the Navi Pack Counter Enable function ensures that no Navi packs are lost, i.e., not read by the host before being overwritten by the next Navi pack. The DSI channel should be programmed to a size that will accommodate information from at least two Navi packs.

Navi Pack Counter Decrement **W 5**

When this bit is set, the Navi Pack Counter is decremented by 1. See the description of the previous bit for more detail.

Navi Pack Counter Output [1:0] **R [7:6]**

This read only register enables the host to read out the value of the Navi Pack Counter. See the description of the previous two bits for more detail.

Figure 4.47 Register 124 (0x07C)

7	5	4	0
Reserved		MPEG Audio Extension Stream ID [4:0]	

MPEG Audio Extension Stream ID [4:0] **W [4:0]**

This register can be used by the host to program the extension stream ID for multichannel MPEG audio bitstreams. This register is used only if Register 143, bits 5–7 (page 4-41), are set to mode 0b101, MPEG multichannel audio stream select enable. The Audio Decoder provides only an S/P DIF (IEC958) formatted output for multichannel MPEG audio bitstreams.

Reserved **[7:5]**

Registers 125–127 (0x07D–0x07F) Reserved **[7:0]**

Figure 4.48 Registers 128–130 (0x080–0x082) Picture Start Code Read Address [19:0]

	7 4	3 0
Reg. 128 LSB	Picture Start Code Read Address [7:0] Read Only	
Reg. 129	Picture Start Code Read Address [15:8] Read Only	
Reg. 130 MSB	Reserved	Picture Start Code Read Address [19:16] Read Only

These registers contain the address of the video channel read pointer captured at the time that a picture start code is decoded from the bitstream by the decoder. When set, the most significant bit (bit 3 of Register 130) indicates that the read pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.49 Registers 131–133 (0x083–0x085) Audio Sync Code Read Address [19:0]

	7 4	3 0
Reg. 131 LSB	Audio Sync Code Read Address [7:0] Read Only	
Reg. 132	Audio Sync Code Read Address [15:8] Read Only	
Reg. 133 MSB	Reserved	Audio Sync Code Read Address [19:16] Read Only

These registers contain the address of the audio channel read pointer captured at the time that an audio sync code is decoded from the bitstream by the Audio Decoder. When set, the most significant bit (bit 3 of Register 133) indicates that the read pointer has wrapped around from the end address to the start address of the buffer.

Figure 4.50 Registers 134–136 (0x086–0x088) Video ES Channel Buffer Numitems [18:0]

	7	3	2	0
Reg. 134 LSB	Video ES Channel Buffer Numitems [7:0] Read Only			
Reg. 135	Video ES Channel Buffer Numitems [15:8] Read Only			
Reg. 136 MSB	Reserved		Video ES Channel Buffer Numitems [18:16] Read Only	

These registers contain the number of items (64-bit words) remaining to be read in the Video ES channel buffer. The LSB should be read first since this captures the next significant byte and MSB in Registers 135 and 136. These should then be read immediately to ensure that the correct captured value is read.

Figure 4.51 Registers 134–136 (0x086–0x088) Video Numitems/Pics in Channel Buffer Compare Panic [18:0]

	7	3	2	0
Reg. 134 LSB	Video Numitems/Pics in Channel Buffer Compare Panic [7:0] Write			
Reg. 135	Video Numitems/Pics in Channel Buffer Compare Panic [15:8] Write			
Reg. 136 MSB	Reserved		Video Numitems/Pics in Channel Buffer Compare Panic [18:16] Write	

The host can write to these registers to program the threshold value to be used for the panic feature. For a description of the panic feature, see Register 69, bits 3 and 4, page 4-23. The threshold units depends on the setting of the bits in Register 69. When they are set to 0b01, the threshold is in terms of the number of items (64-bit words) in the Video ES channel buffer. When the bits in Register 69 are set to 0b10, the threshold is in terms of the number of picture start codes present in the Video ES channel buffer.

Figure 4.52 Registers 137–139 (0x089–0x08B) Audio ES Channel Buffer Numitems [18:0]

<table>
<tr><th></th><th>7</th><th>3</th><th>2</th><th>0</th></tr>
<tr><td>Reg. 137
LSB</td><td colspan="4">Audio ES Channel Buffer Numitems [7:0]
Read Only</td></tr>
<tr><td>Reg. 138</td><td colspan="4">Audio ES Channel Buffer Numitems [15:8]
Read Only</td></tr>
<tr><td>Reg. 139
MSB</td><td colspan="2">Reserved</td><td colspan="2">Audio ES Channel Buffer
Numitems [18:16]
Read Only</td></tr>
</table>

These registers contain the number of items (64-bit words) remaining to be read from the Audio ES channel buffer to the selected Audio Decoder. The LSB should be read first, since this captures the next significant byte and MSB in Registers 138 and 139. These should then be read immediately to ensure that the correct captured value is read.

Figure 4.53 Registers 140–142 (0x08C–0x08E) S/P DIF Channel Buffer Numitems [18:0]

<table>
<tr><th></th><th>7</th><th>3</th><th>2</th><th>0</th></tr>
<tr><td>Reg. 140
LSB</td><td colspan="4">S/P DIF Channel Buffer Numitems [7:0]
Read Only</td></tr>
<tr><td>Reg. 141</td><td colspan="4">S/P DIF Channel Buffer Numitems [15:8]
Read Only</td></tr>
<tr><td>Reg. 142
MSB</td><td colspan="2">Reserved</td><td colspan="2">S/P DIF Channel Buffer
Numitems [18:16]
Read Only</td></tr>
</table>

These registers contain the number of items (64-bit words) remaining to be read from the Audio ES channel buffer to the MPEG or Dolby Digital S/P DIF Formatter. The LSB should be read first since this captures the next significant byte and MSB in Registers 141 and 142. These should then be read immediately to ensure that the correct captured value is read.

Figure 4.54 Register 143 (0x08F)

7 5	4 0
Audio Stream Select Enable [2:0]	Audio Stream ID [4:0]

Audio Stream ID [4:0] **W [4:0]**

This field is used to select a particular audio stream in the type enabled by the following Audio Stream Select Enable field.

Audio Stream Select Enable [2:0] **W [7:5]**

These bits select the type of audio stream that is to be processed by the L64020 according to the following table.

Audio Stream Select Enable	Description
0b000	Always discard (off). No audio data is put in channel.
0b001	MPEG ID selected[1]
0b010	Linear PCM Stream ID selected
0b011	Dolby Digital Stream ID selected
0b100	All MPEG Audio IDs[2]
0b101	MPEG audio multichannel with extension[3]
0b110–0b111	Always discard (off)

[1] In mode 0b001 (MPEG ID selected), the MPEG audio stream is assumed to be MPEG-1 audio or MPEG-2 audio without extensions. The audio stream comes in a single ID programmed in the Audio Stream ID field above.

[2] In mode 0b100 (All MPEG Audio IDs), no audio streams are filtered out. Use this mode when you know that only one audio stream is in the incoming bitstream.

[3] In mode 0b101 (MPEG audio multichannel with extension), the main audio stream ID is programmed in the Audio Stream ID field above and the extension is programmed in the MPEG Audio Extension Stream ID field in Register 124 (page 4-37).

Figure 4.55 Register 144 (0x090)

7	1	0
Reserved		Transport Private Stream Audio

Transport Private Stream Audio **W 0**

When this bit is cleared to 0, MPEG audio is stored in the Audio ES channel buffer. When the bit is set, private 1 stream (Dolby Digital) audio is stored in the Audio ES channel buffer.

Reserved **[7:1]**

Figure 4.56 Register 145 (0x091)

7	6	5	4	3	0
Video PES Headers Enable [1:0]		Video Stream Select Enable [1:0]		Video Stream ID [3:0]	

Video Stream ID [3:0] **W [3:0]**

In this field, the host enters the ID of the video stream to be processed by the decoder. See the Video Stream Select Enable field following.

Video Stream Select Enable [1:0] **W [5:4]**

These bits select whether just the video stream with the ID entered in bits [3:0] of this register or any MPEG video stream is recognized and processed by the L64020. The coding is described in the following table.

Video Stream Select Enable	Description
0b00	Always discard
0b01	MPEG ID selected
0b10	All Video Stream IDs stored
0b11	Always discard

Use select enable 0b10 when you know that only one stream ID is being input.

Video PES Headers Enable [1:0] **R/W [7:6]**

The coding of this field determines which Video PES headers, if any, are stored and processed.

Video PES Headers Enable	Description
0b00	Write no Video PES headers
0b01	Write one header if PTS or DTS. This mode is reset internally to mode 0b00 above after the successful completion of a write.
0b10	Write all headers
0b11	Always store with PTS

Figure 4.57 Register 146 (0x092)

7	6	5	4	0
Reserved	Data Dump Stream Select Enable	SPU Stream Select Enable	SPU Substream ID [4:0]	

SPU Substream ID [4:0] **W [4:0]**

This field is used in conjunction with the SPU Stream Select Enable to determine which SPU Substream ID is parsed and decoded.

SPU Stream Select Enable **W 5**

When this bit is set, the SPU substream with ID matching that in the SPU Substream ID field above is processed by the L64020. When the bit is cleared, all SPU streams are discarded and not processed.

Data Dump Stream Select Enable **W 6**

When this bit is set, the Data Dump Substreams are processed by the L64020. When the bit is cleared, the substreams are discarded.

Reserved **7**

Figure 4.58 Register 147 (0x093)

7	6	5	4	3	2	1	0
Navi Pack PES Header Enable [1:0]		Pack Header Enable [1:0]		System Header Enable [1:0]		Audio PES Header Enable [1:0]	

Audio PES Header Enable [1:0] **R/W [1:0]**

The host can configure these bits to have the L64020 store or not store audio PES headers as shown in the following table.

Audio PES Header Enable	Description
0b00	Write no headers
0b01	Write one header if PTS or DTS present. This mode is reset internally back to mode 0b00 above on successful completion of write.
0b10	Write all Audio PES headers
0b11	Always store with PTS

System Header Enable [1:0] **R/W [3:2]**

The host can configure these bits to have the L64020 store or not store system headers as shown in the following table.

System Header Enable	Description
0b00	Always discard
0b01	Write one header. This mode is reset internally back to mode 0b00 on successful completion of write.
0b10	Write all headers
0b11	Always discard

Pack Header Enable [1:0] R/W [5:4]

The host can configure these bits to have the L64020 store or not store pack headers as shown in the following table.

Pack Header Enable	Description
0b00	Write no headers
0b01	Write one header. This mode is reset internally back to mode 0b00 on successful completion of write.
0b10	Write all headers
0b11	Always discard

Navi Pack PES Header Enable [1:0] W [7:6]

The host can configure these bits to have the L64020 store or not store DSI and PCI PES headers as shown in the following table.

Navi Pack PES Header Enable	Description
0b00	Write no DSI or PCI headers (OFF)
0b01	Reserved
0b10	Reserved
0b11	Write PCI and DSI headers and packets.

Figure 4.59 Register 148 (0x094)

7 1	0
Reserved	Top of Sector Detect Enable

Top of Sector Detect Enable W 0

When this bit is set, the preparser monitors the Top of Sector (TOSn) input signal. If a pack start code does not occur at the same time as the Top of Sector signal is asserted, the preparser signals an error condition. When this bit is cleared, the preparser ignores the Top of Sector signal.

Reserved [7:1]

Figure 4.60 Register 149 (0x095)

7	5	4	3	2	1	0
Reserved		SPU Packet Error Status	Reserved	DSI/PCI Packet Error Status	Video Packet Error Status	Audio Packet Error Status

Audio Packet Error Status **R 0**

When set, this bit indicates that an error was detected by the preparser while parsing through an audio packet. This bit is cleared on reading this register.

Video Packet Error Status **R 1**

When set, this bit indicates that an error was detected by the preparser while parsing through a video packet. This bit is cleared on reading this register.

DSI/PCI Packet Error Status **R 2**

When set, this bit indicates that an error was detected by the preparser while parsing through a DSI or PCI packet. This bit is cleared on reading this register.

Reserved **3**

SPU Packet Error Status **R 4**

When set, this bit indicates that an error was detected by the preparser while parsing through an SPU packet. This bit is cleared on reading this register.

Reserved **[7:5]**

Figure 4.61 Registers 150 and 151 (0x096 and 0x097) Pictures in Video ES Channel Buffer Counter [15:0]

	7 … 0
Reg. 150 LSB	Pictures in Video ES Channel Buffer Counter [7:0] Read Only
Reg. 151 MSB	Pictures in Video ES Channel Buffer Counter [15:8] Read Only

These registers allow the host to read the number of pictures currently in the Video ES channel buffer.

Registers 152–191 (0x098–0x0BF) Reserved **[7:0]**

4.3 Memory Interface Registers

Figure 4.62 Register 192 (0x0C0)

7	6	5	4	3	2	1	0
DMA Write FIFO Full	DMA Write FIFO Empty	DMA Read FIFO Full	DMA Read FIFO Empty	Host Write FIFO Full	Host Write FIFO Empty	Host Read FIFO Full	Host Read FIFO Empty

These read-only bits contain the empty/full status of the four, 8 x 64-bit, internal FIFOs used during host/SDRAM read or write operations. The host should read this register before transfers to avoid reading from an empty FIFO or writing to a full FIFO. Refer to Section 5.4, “SDRAM Access,” for more details on host/SDRAM transfer operations. The FIFO empty bits are set and the FIFO full bits are cleared at reset.

Figure 4.63 Register 193 (0x0C1)

7	6	5	4	3	2	1	0
Reserved	DMA Transfer Byte Ordering	Refresh Extend [1:0]		Host SDRAM Transfer Byte Ordering	DMA Mode [1:0]		Reserved

Reserved **0**

DMA Mode [1:0] **R/W [2:1]**

Defines the state of the DMA Transfer Request (DREQn) output signal per the following table.

DMA Mode [1:0]	Description
0b00	DMA Idle (DREQn = 1)
0b01	DMA Read (DREQn = read FIFO near empty)
0b10	DMA Write (DREQn = write FIFO near full)
0b11	Block Move (DREQn =1)

During DMA transfers, the external DMA controller should use the DREQn output signal to determine whether or not another 64-bit word can be transferred.

DMA Idle: This setting is used to hold DREQn deasserted and prevent the external DMA controller from transferring any data to or from SDRAM.

DMA Read: The on-chip SDRAM controller continuously fills the internal 8 x 64-bit DMA read FIFO with data read from the specified SDRAM source address. The SDRAM address is automatically incremented until the read FIFO is near full. Separate FIFOs and address registers are available for DMA and host reads. The DMA controller can retrieve the next available read byte from the DMA SDRAM Read Data register (Register 219, page 4-56).

During DMA Read Mode, DREQn is asserted only when there are more SDRAM data words in the read FIFO for reading.

DMA Write: The DMA controller writes data to the DMA SDRAM Write Data register (Register 220, page 4-56). Every 8 bytes written are formed into a 64-bit word and transferred to the internal 8 x 64 write FIFO. Note that separate counters and addresses are maintained for host and DMA write operations. The on-chip SDRAM controller continuously empties the write FIFO and transfers the data to the specified SDRAM target address. The target address is automatically incremented for every 64-bit word transferred from the write FIFO to the SDRAM.

During DMA Write Mode, DREQn is asserted only when there is more space in the write FIFO for writing. It is the responsibility of the external DMA controller to keep track of the DMA transfer count. On receiving DMA done from the external DMA controller, the external host should always check for read FIFO full (for DMA read) or write FIFO empty (for DMA write) before returning the DMA Mode back to Idle.

Block Move: The host specifies the source address, the target address and a block transfer count. The internal SDRAM controller uses the read FIFO to continuously load SDRAM data from the source address and empties the contents of the read FIFO to the target address. When the total number of transferred words reaches the block transfer count, the SDRAM Transfer Done Interrupt bit in Register 0, bit 4 (page 4-3) is set, INTRn is asserted if the interrupt is not masked, and the DMA Mode is reset to Idle, 00. Refer to Section 5.4, “SDRAM Access,” for further detail.

During block moves, the DREQn signal is held high. DMA and block moves may NOT occur simultaneously.

Host SDRAM Transfer Byte Ordering
Little Endian/Big Endian **R/W 3**

This bit must be set if the host operates in big endian mode, i.e., with byte 0 in bits [63:56] and byte 7 in bits [7:0]. Since the L64020 operates in big endian mode, no byte swapping occurs at the host interface. If the host is little endian, this bit must be cleared to enable byte swapping.

Refresh Extend [1:0] **R/W [5:4]**

These bits specify the multiplying factor for SDRAM refreshes. The table below lists the number of refresh cycles per refresh period (1 refresh period per macroblock during reconstruction).

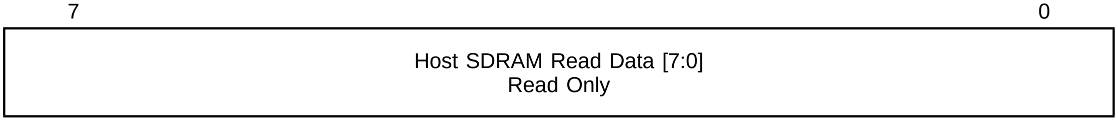
Refresh Extend	Refresh Cycles
0b00	2 (default)
0b01	4
0b10	16
0b11	1 (Reserved for LSI internal use only.)

DMA SDRAM Transfer Byte Ordering
Little Endian/Big Endian **6**

This bit must be set if the external DMA controller operates in big endian mode, i.e., with byte 0 in bits [63:56] and byte 7 in bits [7:0]. Since the L64020 operates in big endian mode, no byte swapping occurs at the host interface. If the DMA controller is little endian, this bit must be cleared to enable byte swapping.

Reserved **7**

Figure 4.64 Register 194 (0x0C2) Host SDRAM Read Data [7:0]



Host SDRAM Read Data [7:0]
Read Only

This register stores the next byte to be read by the host during a host read from SDRAM.

Figure 4.65 Register 195 (0x0C3) Host SDRAM Write Data [7:0]

7	0
Host SDRAM Write Data [7:0] Write	

The host writes the next byte to be read into SDRAM in this register during a host write to SDRAM.

Figure 4.66 Registers 196–198 (0x0C4–0x0C6) Host SDRAM Target Address [18:0]

	7	3	2	0
Reg. 196 LSB	Host SDRAM Target Address [7:0] R/W			
Reg. 197	Host SDRAM Target Address [15:8] R/W			
Reg. 198 MSB	Reserved		Host SDRAM Target Address [18:16] R/W	

For a host write to SDRAM, the host must write the starting SDRAM address in this register. This address is automatically incremented after eight bytes are transferred to SDRAM through Register 195 and the internal, 8 x 64, write FIFO. The host should update the SDRAM target address only when the write FIFO is empty.

Figure 4.67 Registers 199–201 (0x0C7–0x0C9) Host SDRAM Source Address [18:0]

	7	3	2	0
Reg. 199 LSB	Host SDRAM Source Address [7:0] R/W			
Reg. 200	Host SDRAM Source Address [15:8] R/W			
Reg. 201 MSB	Reserved		Host SDRAM Source Address [18:16] R/W	

For a host read from SDRAM, the host must write the starting SDRAM address in this register. This address is automatically incremented after eight bytes are transferred to the internal, 8 x 64, read FIFO. The host should update the SDRAM source address only when the read FIFO is full, allowing a clean flush of the read FIFO. When updating the SDRAM source address, the LSB of the address should be written last. This triggers the refill of the read FIFO at the new address.

Figure 4.68 Registers 202 and 203 (0x0CA and 0x0CB) Block Transfer Count [15:0]

	7 … 0
Reg. 202 LSB	Block Transfer Count [7:0] R/W
Reg. 203 MSB	Block Transfer Count [15:8] R/W

For an SDRAM block move, the host writes the number of 64-bit words to be moved in these registers. During the move, these registers contain the number of words left to transfer. These registers are not used during a host SDRAM read/write. The Block Transfer Count defaults to 0xFFFF at reset.

Figure 4.69 Register 204 (0x0CC)

7	6	5	4	3	2	1	0
Control for Programmable Delay Path 2 [1:0]		Control for Programmable Delay Path 1 [1:0]		Clk Out of Sync	Reserved		PLL Test

PLL Test **R/W 0**

When this bit is set, it initiates the PLL test. Results are stored in Register 221 (page 4-56).

Reserved **[2:1]**

Clk Out of Sync **R 3**

When set, indicates that some of the 27-MHz and 81-MHz clocks are no longer synchronized. Used for diagnostic purposes.

Control for Programmable Delay Path 1 [1:0] R/W [5:4]

This register controls the selection of delay cells on the 81-MHz clock fed back from the SCLK pin.

Control Bits	Description
0b00	None
0b01	del05 x 1
0b10	del05 x 2
0b11	del05 x 3

Note that the delays are in units of del05, a delay of 0.5 ns at nominal conditions (nominal process factor, 25 ˚C, and V_{DD} = +3.3 V).

Control for Programmable Delay Path 2 [1:0] R/W [7:6]

This register controls the selection of delay cells on the incoming 81-MHz clock in scan test mode or bypass mode. This register is only used in diagnostic mode and during manufacturing test.

Control Bits	Description
0b00	None
0b01	del1 x 1
0b10	del1 x 2
0b11	del1 x 3

Note that the delays are in units of del1, a delay of 1.0 ns at nominal conditions (nominal process factor, 25 ˚C, and V_{DD} = 3.3 V).

Figure 4.70 Register 205 (0x0CD)

7	6	5	3	2	1	0
Reserved		Internal SDRAM State [2:0]		Internal Lock Counter State [1:0]		Phase Locked Status

Phase Locked Status R 0

When this bit is set, the two internal clocks (81 MHz and 27 MHz) are synchronized.

Internal Lock Counter State [1:0] **R [2:1]**

Used to monitor synchronization of the 81-MHz and 27-MHz clocks (diagnostics only).

Bits [2:1]	Description
0b00	No sync yet
0b01	Got sync for 1 cycle
0b10	Got sync for 2 cycles
0b11	Got sync for at least 3 cycles

Internal SDRAM State [2:0] **R [5:3]**

Used to monitor the internal SDRAM state (diagnostics only).

Bits [5:3]	Description
0b000	Waiting for 2 clocks to reach synchronization
0b001	SDRAM initialization (precharge both banks)
0b010	SDRAM Initialization (first 8 refreshes)
0b011	SDRAM Initialization (set mode register)
0b100	SDRAM Initialization (second 8 refreshes)
0b101	SDRAM ready to operate

Reserved **[7:6]**

Figure 4.71 Register 206 (0x0CE)

7	6	5	4	3	2	1	0
Internal Phase State (current cycle) [1:0] Read Only		Internal Phase State (1 cycles before) [1:0] Read Only		Internal Phase State (2 cycles before) [1:0] Read Only		Internal Phase State (3 cycles before) [1:0] Read Only	

When the two internal clocks reach synchronization, the internal phase state should be looping through states 01, 10, and 11. If a 00 state is ever reached, it indicates that the synchronization has been lost. These registers are used for diagnostic purposes only.

Figure 4.72 Registers 207–212 (0x0CF–0x0D4)

	7 … 0
Reg. 207 LSB	Phase Detect Test High Freq [7:0] R/W
Reg. 208 MSB	Phase Detect Test High Freq [15:8] R/W
Reg. 209 LSB	Phase Detect Test Low Freq [7:0] R/W
Reg. 210 MSB	Phase Detect Test Low Freq [15:8] R/W
Reg. 211 LSB	VCO Test High Freq [7:0] R/W
Reg. 212 MSB	VCO Test High Freq [15:8] R/W

Registers 207–212, and 222 and 223 (page 4-56) set the parameters for testing the Phase-Locked Loop (PLL). The PLL test is run by setting bit 0, PLL Test, in Register 204 (page 4-51). The results from the PLL test can be read from Register 221 (page 4-56). Tests are run on the phase detector and the VCO. The PLL passes the test if the frequency of the PLL falls between the high frequency value and the low frequency value. The PLL test interrupts the system clock and should not be attempted when the chip is running.

Note: Registers 207 through 212 are included for LSI Logic’s testing purposes. Do not write to the registers without specific directions from LSI Logic.

Figure 4.73 Registers 213–215 (0xD5–0x0D7) DMA SDRAM Target Address [18:0]

	7 3	2 0
Reg. 213 LSB	DMA SDRAM Target Address (dma+block) [7:0] R/W	
Reg. 214	DMA SDRAM Target Address (dma+block) [15:8] R/W	
Reg. 215 MSB	Reserved	DMA SDRAM Target Address (dma+block) [18:16] R/W

During DMA Write and Block Move, the DMA SDRAM Target Address is the starting address where future DMA writes will take place. This address is automatically incremented after a 64-bit word is transferred to SDRAM from the internal 8 x 64 write FIFO. The DMA SDRAM Target Address should be updated by the host only when the write FIFO is empty.

Figure 4.74 Registers 216–218 (0xD8–0x0DA) DMA SDRAM Source Address [18:0]

	7 3	2 0
Reg. 216 LSB	DMA SDRAM Source Address (dma+block) [7:0] R/W	
Reg. 217	DMA SDRAM Source Address (dma+block) [15:8] R/W	
Reg. 218 MSB	Reserved	DMA SDRAM Source Address (dma+block) [18:16] R/W

During DMA Read and Block Move, the DMA SDRAM Source Address is the starting address where future DMA reads will take place. This address is automatically incremented after a 64-bit word is transferred from SDRAM to the internal 8 x 64 read FIFO. The DMA SDRAM Source Address should be updated by the host only when the DMA read FIFO is full, allowing a clean flush of the read FIFO. When updating the DMA SDRAM Source Address, it should be written in MSB to LSB order. This triggers the refill of the read FIFO at the new address.

Figure 4.75 Register 219 (0x0DB) DMA SDRAM Read Data [7:0]

7 0
DMA SDRAM Read Data (DMA only) [7:0] Read Only

During DMA read, the next byte from SDRAM to be read by the external DMA is placed in this register.

Figure 4.76 Register 220 (0x0DC) DMA SDRAM Write Data [7:0]

7 0
DMA SDRAM Write Data (DMA only) [7:0] W

During DMA write, the external DMA writes the next byte to be written to SDRAM in this register.

Figure 4.77 Register 221 (0x0DD)

7 4	3	2	1	0
Reserved	PLL VCO Low Frequency Test Pass Read Only	PLL VCO High Frequency Test Pass Read Only	PLL Phase Detect Low Frequency Test Pass Read Only	PLL Phase Detect High Frequency Test Pass Read Only

Register 221 holds the results of the PLL test. See also Registers 204 bit 0 (page 4-51), 207–212 (page 4-54), and 222–223.

Figure 4.78 Registers 222 and 223 (0x0DE and 0x0DF) VCO Test Low Frequency [15:8]

	7 0
Reg. 222 LSB	VCO Test Low Frequency [7:0] R/W
Reg. 223 MSB	VCO Test Low Frequency [15:8] R/W

See Registers 204, bit 0 (page 4-51), and 207–212 (page 4-54).

4.4 Microcontroller Registers

Figure 4.79 Registers 224 and 225 (0x0E0 and 0x0E1) Anchor Luma Frame Store 1 Base Address [15:0]

	7	0
Reg. 224 LSB	Anchor Luma Frame Store 1 Base Address [7:0] R/W	
Reg. 225 MSB	Anchor Luma Frame Store 1 Base Address [15:8] R/W	

These registers contain the start address of the Anchor Luma Frame Store 1. The resolution of this address is in units of 64-bytes of SDRAM memory.

Figure 4.80 Registers 226 and 227 (0x0E2 and 0x0E3) Anchor Chroma Frame Store 1 Base Address [15:0]

	7	0
Reg. 226 LSB	Anchor Chroma Frame Store 1 Base Address [7:0] R/W	
Reg. 227 MSB	Anchor Chroma Frame Store 1 Base Address [15:8] R/W	

These registers contain the start address of the Anchor Chroma Frame Store 1. The resolution of this address is in units of 64-bytes of SDRAM memory.

Figure 4.81 Registers 228 and 229 (0x0E4 and 0x0E5) Anchor Luma Frame Store 2 Base Address [15:0]

	7	0
Reg. 228 LSB	Anchor Luma Frame Store 2 Base Address [7:0] R/W	
Reg. 229 MSB	Anchor Luma Frame Store 2 Base Address [15:8] R/W	

These registers contain the start address of the Anchor Luma Frame Store 2. The resolution of this address is in units of 64-bytes of SDRAM memory.

Figure 4.82 Registers 230 and 231 (0x0E6 and 0x0E7) Anchor Chroma Frame Store 2 Base Address [15:0]

	7	0
Reg. 230 LSB	Anchor Chroma Frame Store 2 Base Address [7:0] R/W	
Reg. 231 MSB	Anchor Chroma Frame Store 2 Base Address [15:8] R/W	

These registers contain the start address of the Anchor Chroma Frame Store 2. The resolution of this address is in units of 64-bytes of SDRAM memory.

Figure 4.83 Registers 232 and 233 (0x0E8 and 0x0E9) B Luma Frame Store Base Address [15:0]

	7	0
Reg. 232 LSB	B Luma Frame Store Base Address [7:0] R/W	
Reg. 233 MSB	B Luma Frame Store Base Address [15:8] R/W	

These registers contain the start address of the B Luma Frame Store. The resolution of this address is in units of 64-bytes of SDRAM memory.

Figure 4.84 Registers 234 and 235 (0x0EA and 0x0EB) B Chroma Frame Store Base Address [15:0]

	7	0
Reg. 234 LSB	B Chroma Frame Store Base Address [7:0] R/W	
Reg. 235 MSB	B Chroma Frame Store Base Address [15:8] R/W	

These registers contain the start address of the B Chroma Frame Store. The resolution of this address is in units of 64-bytes of SDRAM memory.

Figure 4.85 Register 236 (0x0EC)

	7 3	2	1 0
Read	Reserved	Video Continuous Skip Status	Video Skip Frame Status [1:0]
Write	Reserved	Video Continuous Skip Mode	Video Skip Frame Mode [1:0]

Video Skip Frame Status [1:0] R [1:0]

In one-time skip mode (see the description of bit 2 in this register), the microcontroller clears these bits to let the host know that the skip has been completed. In this mode, the microcontroller skips through the bitstream for one picture of the correct type (see Video Skip Frame Mode following) without decoding it, and then starts decoding one frame ahead in the bitstream.

Video Skip Frame Mode [1:0] W [1:0]

These bits command the microcontroller to skip the next I, P, or B frame according to the following table. When in continuous skip mode (see the description of bit 2 in this register), the host needs to reset this register to 0b00, i.e., normal play to stop the skip.

Skip Frame Bits	Skip Frame Mode
0b00	None (normal play)
0b01	Skip B frame
0b10	Skip P or B frame
0b11	Skip any frame

Video Continuous Skip Status R 2

When set, indicates that the skip mode is continuous. When cleared, indicates that the decoder is in single-skip mode.

Video Continuous Skip Mode W 2

This bit controls the behavior of the video skip mode. If this bit is set, the video skip mode is continuous, i.e., the decoder continues to skip the selected picture types until the host resets the skip mode to 0b00 (normal play). If this bit is cleared by the host, then the skip is treated as one-time, that is, one picture of the selected type is skipped and the skip mode is reset back to 0b00 or normal play by the microcontroller.

Reserved **[7:3]**

Figure 4.86 Register 237 (0x0ED)

	7 2	1	0
Read	Reserved	Video Continuous Repeat Frame Status	Video Repeat Frame Status
Write	Reserved	Video Continuous Repeat Frame Mode	Video Repeat Frame Enable

Video Repeat Frame Status **R 0**

Shows status of Video Repeat Frame Enable Mode. In one-time repeat mode (see the description of bit 1 in this register) when the repeat has been completed, the microcontroller clears this bit to let the host know that the repeat has been completed.

Video Repeat Frame Enable **W 0**

Setting this bit commands the microcontroller to repeat the next I, P, or B frame. In this mode, the microcontroller stops decoding and displays the same frame a second time. When in continuous repeat mode (see the description of bit 1 in this register), the repeat frame continues until the host resets this bit to a 0.

Video Continuous Repeat Frame Status **R 1**

Indicates the status of the Video Continuous Repeat Frame Mode as described below.

Video Continuous Repeat Frame Mode **W 1**

This bit controls the behavior of the video repeat frame function. When cleared, one frame is repeated and then the Video Repeat Frame Enable bit is cleared by the microcontroller. When this bit is set, frames are repeated continuously until the host clears the Video Repeat Frame Enable bit to end the operation.

Reserved **[7:2]**

Figure 4.87 Register 238 (0x0EE)

	7	6	5	4	3	2	1	0
Read	Reserved		Current Decode Frame [1:0]		Current Display Frame [1:0]		Rip Forward Display Single Step Status	Rip Forward Mode Status
Write	Reserved						Rip Forward Display Single Step Command	Rip Forward Mode Enable

Rip Forward Mode Status **R 0**

Indicates the status of the Rip Forward Mode as described for the enable bit following.

Rip Forward Mode Enable **W 0**

Setting this bit enables the Rip Forward Mode. In this mode, the decoder processes pictures as fast as it can without regard to the status of the display, i.e., the rate control for the decode with respect to the vertical sync of the display is turned off. The rate control for the decode is governed by the Rip Forward Display Single Step Command (bit 1 in this register). The microcontroller monitors the single step command bit after it has received a picture start code and processed a picture header. The decode for that picture only proceeds if the single step command bit is set. The single step command bit is cleared on reading by the microcontroller. Rip Forward Mode is intended to be used in an application where not every picture that is decoded needs to be displayed. The picture to be displayed is specified in separate registers. These registers are decoded in the Video Interface module. (See Register 265 bits 4 and 5, Display Override Mode, and Registers 285, 286, 287, and 288, Override Display Start Addresses for luma and chroma on page 4-76). When Display Override Mode is active, the host must also specify the Override Picture Width (Register 283, bits [6:0], page 4-75). Pan and Scan from the bitstream must be disabled (bit 6 in Register 279 must be cleared, page 4-74) and pan-scan values (if any) must be supplied by the host. Bit 2 in Register 275 (page 4-71), 3:2 Pulldown from Bitstream, must be cleared, and the Host Repeat First Field bit (bit 3 in

Register 275) and Host Top Field First bit (bit 4 in Register 275) must be used by the host to specify the display completely.

Rip Forward Display Single Step Status **R 1**

Indicates the status of the Rip Forward Display Single Step Command as described in the following bit.

Rip Forward Display Single Step Command **W 1**

This bit should be set by the host after it has made the decision on whether or not to display the current picture. The L64020's microcontroller waits for this bit to be set before continuing with picture reconstruction in Rip Forward Mode. This register is only applicable when the Rip Forward Mode Enable bit (bit 0 in this register) is set. The microcontroller clears this bit when it reads it.

Current Display Frame [1:0] **R [3:2]**

These bits indicate which frame store is being used for displaying the current frame as shown in the following table.

Current Display Frame	Description
0b00	Anchor 1
0b01	Anchor 2
0b10	B
0b11	Reserved

Current Decode Frame [1:0] **R [5:4]**

These bits indicate which frame store is being used for reconstruction as shown in the following table.

Current Decode Frame	Description
0b00	Anchor 1
0b01	Anchor 2
0b10	B
0b11	Reserved

Reserved **[7:6]**

Figure 4.88 Register 239 (0x0EF)

7	6	5	4	3	2	1	0
Reserved	Ignore Sequence End	Force Rate Control	Concealment Copy Option	GOP User Data Only	Panic Prediction Enable	Host Force Broken Link Mode	Reserved

Reserved **0**

Host Force Broken Link Mode **R/W 1**

Setting this bit disables the display of certain B pictures in a Group of Pictures (GOP) by forcing the Broken Link bit to 1. If the bitstream indicates an open GOP and this bit is set, then any B pictures before the first I picture in the GOP are skipped. The Video Interface, however, stays synchronized with the display since only one B picture is skipped per frame display period. This is different from merely skipping B pictures which skips each B picture as fast as possible and does not control the rate of the skip with respect to the display.

Panic Prediction Enable **R/W 2**

Setting this bit causes the motion compensation module to go into a panic mode. In this mode, the motion compensation module performs motion compensation using only the forward vector set and ignores the backward vectors for B pictures. Similarly for dual-prime motion vectors, only same parity prediction is performed. This mode is intended to be used in situations where limited SDRAM bandwidth slows down picture reconstruction to the point where the picture starts tearing since reconstruction is not able to keep pace with the display. Eliminating the backward vectors reduces the demand made on the SDRAM bandwidth by reducing accesses to the reference anchor store in the SDRAM. This should alleviate any tearing problems. However, this does not completely guarantee solving any bandwidth problems that may exist with slow SDRAM devices.

GOP User Data Only **R/W 3**

When this bit is set, the Video Decoder recognizes only user data supplied in the GOP layer of the MPEG-1/ MPEG-2 video stream. User data of other layers, if present, is discarded by the decoder and is not written to the user FIFO. This feature is designed to accept only line 21 data (closed-caption data) in a DVD system. By discarding other user data layers, the processing overhead on the host controller in a DVD system is reduced significantly. The default value of this register at start-up is 0, which means user data of all layers is available to the host in the user FIFO.

Concealment Copy Option **R/W 4**

When set, this bit overrides any concealment vectors that may have been present in the original MPEG video bitstream. Normally, MPEG specifies that these vectors be used to conceal any errors that may be detected by the decoder. When this bit is set, copying from the previously decoded valid picture is used instead of applying the concealment vectors. This register is cleared by default (i.e., after reset or power-up).

Force Rate Control **R/W 5**

When this bit is set, the decoder controls the rate of the decoding process based on the display rate. When this bit is cleared, the decoder controls the rate of the decoding process only if it is accessing the same frame store as the display process.

Ignore Sequence End **R/W 6**

When this bit is set, the last picture of a sequence is not displayed at the sequence end code but at the beginning of the next sequence. When this bit is cleared, the decoder displays the last picture of the sequence at the sequence end code.

Reserved **7**

Figure 4.89 Register 240 (0x0F0)

	7	1	0
Read	Reserved		Host Broken Link/Seq Status
Write	Reserved		Host Search Broken Link/Seq Command

Host Broken Link/Seq Status **R 0**

Indicates the status of the bitstream search described for the following command.

Host Search Broken Link/Seq Command **W 0**

Setting this bit causes the decoder to skip bits in the bitstream until it finds the next GOP or sequence header. When it finds a GOP or sequence header, the L64020's microcontroller sets the Host Broken Link/Seq Status bit.

Reserved **[7:1]**

Figure 4.90 Register 241 (0x0F1)

7	2	1	0
Q Table Address [5:0]		Intra Q Table	Q Table Ready

Q Table Ready **R 0**

When set, this bit indicates that the Q table is ready to be read by the host.

Intra Q Table **R/W 1**

When set, this bit indicates that the Q table is an intra table. When cleared, it indicates a non-intra Q table.

Q Table Address [5:0] **R/W [7:2]**

The host writes the address of the Q table entry to be accessed in these six bits. See Section 8.2.8, "Host Access of Q Table Entries."

Figure 4.91 Register 242 (0x0F2) Q Table Entry [7:0]

7	0
Q Table Entry [7:0] Read Only	

The Q table entry addressed in the previous register is available to the host in this register.

Figure 4.92 Register 243 and 244 (0x0F3 and 0x0F4) Microcontroller PC [11:0]

	7	4	3	0
Reg. 243 LSB	Microcontroller PC [7:0] Read Only			
Reg. 244 MSB	Reserved		Microcontroller PC [11:8] Read Only	

Internal microcontroller Program Counter (PC). Note that the LSB should always be read before the MSB to ensure correct capture of the microcontroller PC value.

Figure 4.93 Register 245 (0x0F5) Revision Number [7:0]

7	0
Revision Number [7:0] Read Only	

The value in this register is the revision number of the L64020.

Figure 4.94 Register 246 (0x0F6)

7	1	0
Reserved		Decode Start/Stop Command Write

Setting this bit causes the Video Decoder to start decode/reconstruction of pictures. Clearing the bit stops the decode process but does not stop the channel. The current status of decoding can be monitored by reading the Decode Status Interrupt bit (Register 0, bit 0, page 4-2).

Register 247 (0x0F7) Reserved [7:0]

Figure 4.95 Register 248 (0x0F8) Reduced Memory Mode (RMM) Bit

<table>
<tr><th>7</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Reserved</td><td>Reduced Memory Mode (RMM)
R/W</td></tr>
</table>

When set, this bit enables the Reduced Memory Mode (RMM) required for PAL resolution (720 x 576). This mode has the capability to use less than one frame store SDRAM memory space for B pictures provided some restrictions are met. These restrictions include no “repeat-first-field” for B pictures.

RMM is achieved by dynamically allocating segments of memory to the reconstruction and display processes. Re-use of segments is possible in the case of B pictures, thus reducing the frame store memory requirement for B pictures. When this bit is cleared, the chip is in regular memory mode.

Registers 249–255 (0x0F9–0x0FF) Reserved [7:0]

4.5 Video Interface Registers

Registers 256–264 (0x100–0x108) Reserved [7:0]

Figure 4.96 Register 265 (0x109)

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Force Video Background [1:0]</td><td colspan="2">Display Override Mode [1:0]</td><td>Clear OSD Palette Counter</td><td>Reserved</td><td colspan="2">OSD Mode [1:0]</td></tr>
</table>

OSD Mode [1:0] R/W [1:0]

The On-Screen Display (OSD) Mode field determines the source of OSD data, either internal or external, per the following table.

Mode Bits	Description
0b00	No OSD (Disabled)
0b01	Internal OSD (Contiguous)
0b10	Internal OSD (Linked List)
0b11	External OSD

Reserved **2**

OSD Palette Counter Zero Flag **R 3**

This bit is set to inform the host that the OSD Palette Counter is cleared to zero. The host should check this bit before starting to write the OSD Palette.

Clear OSD Palette Counter **W 3**

When this bit is set, the counter that controls access to the OSD palette is cleared.

Display Override Mode [1:0] **R/W [5:4]**

When this field is set to modes 0b01 and 0b10, the display start address from the internal video decoder can be overridden, i.e., controlled by the external host. In these modes, the start address of the frame store from which to display is taken from Registers 285 and 286 for Luma (page 4-77) and Registers 287 and 288 for Chroma. The resolution of the start addresses is 64 bytes of SDRAM data. When set to mode 0b00 (normal mode), the start address of the display controller is supplied by the internal video decoder.

Mode Bits	Description
0b00	Normal Mode (no override)
0b01	Frame Mode
0b10	First Field Only
0b11	Reserved

Force Video Background [1:0] **R/W [7:6]**

These bits control the background and allow the host to set modes that can force the background to any color according to the following table:

Mode Bits	Description
0b00	No Background (default)
0b01	Video Black
0b10	Video Blue/User programmed
0b11	Video on Blue

Figure 4.97 Registers 266–268 (0x10A and 0x10C) Programmable Background Y/Cb/Cr [7:0]

	7	0
Reg. 266	Programmable Background Y[7:0] R/W	
Reg. 267	Programmable Background Cb[7:0] R/W	
Reg. 268	Programmable Background Cr[7:0] R/W	

When the Force Video Background Mode (bits 6 and 7 in Register 265) is 0b10, the Y, Cb, and Cr values for the background are specified in Registers 266, 267, and 268, respectively. The default values (i.e., values in the register at startup/reset) are Y=35, Cb=212, and Cr=114. These values correspond to 75% Amplitude, 100% Saturated Blue.

Figure 4.98 Register 269 (0x10D) OSD Palette Write [7:0]

7	0
OSD Palette Write [7:0] W	

This register is for writing the On-Screen Display (OSD) palette into the Color Lookup Table (CLUT) in on-chip RAM for the External OSD Mode. The host must disable the OSD Mode (Register 265), write the OSD Mix Weight into bits [3:0] of Register 274, clear the OSD palette counter (Register 265, bit 3), and then write the color palette into this register. The palette should be written in Color 0 LSB, Color 0 MSB, Color 1 LSB,..., Color 15 MSB order. The internal OSD palette counter is automatically incremented as each byte is written into the register. After the palette is completed, the host can then change the OSD Mode to external to start OSD.

Figure 4.99 Registers 270–273 (0x10E–0x111) OSD Odd/Even Field Pointers [15:0]

	7 0
Reg. 270 LSB	OSD Odd Field Pointer [7:0] R/W
Reg. 271 MSB	OSD Odd Field Pointer [15:8] R/W
Reg. 272 LSB	OSD Even Field Pointer [7:0] R/W
Reg. 273 MSB	OSD Even Field Pointer [15:8] R/W

The host can program the addresses of the OSD Odd/Even Field Pointers into these registers. The addresses are in 64-byte resolution.

Figure 4.100 Register 274 (0x112)

7	6	5	4	3 0
Reserved	Horizontal Decimation Filter Enable	SPU Chroma Filter Enable	OSD Chroma Filter Enable	OSD Mix Weight [3:0]

OSD Mix Weight [3:0] **R/W [3:0]**

This register is used by the host to specify the OSD mix weight in external OSD Mode only. When using internal OSD, the mix weight is specified in the OSD header stored in SDRAM. When programmed to zero, the mixed video output is 100% video and 0% OSD. When programmed to 0xF, the mixed video output is 1/16 video and 15/16 OSD.

OSD Chroma Filter Enable **R/W 4**

When this bit is set, the chroma enhancement filter is enabled for OSD overlay images.

SPU Chroma Filter Enable **R/W 5**

When this bit is set, the SPU chroma filter is enabled.

Horizontal Decimation Filter Enable **R/W 6**

When this bit is set, the 2:1 horizontal decimation filter is enabled.

Reserved **7**

Figure 4.101 Register 275 (0x113)

7	6	5	4	3	2	1	0
Top/Not Bottom Field	Odd/Not Even Field	First Field	Host Top Field First	Host Repeat First Field	3:2 Pulldown from Bitstream	Freeze Mode [1:0]	

Freeze Mode [1:0] **R/W [1:0]**

These bits select the Freeze Mode according to the following table.

Freeze Mode	Description
0b00	Normal
0b01	Freeze Frame
0b10	Freeze Last Field
0b11	Freeze First Field and Hold

3:2 Pulldown from Bitstream **R/W 2**

Setting this bit causes the L64020 to decode pulldown control from the MPEG-2 syntax in the bitstream. Clearing this bit allows the host to control pulldown. The default value for this bit is 1 (at power-up or chip reset).

Host Repeat First Field **R/W 3**

When this bit is set, the first displayed field in a frame is repeated during the third field time. This is the primary mechanism for performing 3:2 pulldown from the host interface. The default value for this register is 0.

Host Top Field First **R/W 4**

When this bit is set, the first displayed field in a frame is the top field (or odd field lines). This bit is used in conjunction with the Host Repeat First Field for controlling 3:2 pulldown. The default value for this register is 1.

First Field **R 5**

This bit is set to indicate that the current field being displayed is the first field of the frame and cleared when it is the last field. Normally, this bit and the Last Field bit in the next register toggle as the current field alternates. In 3:2 pulldown, both the First Field bit and Last Field bit are cleared when the current field is the middle field.

Odd/Not Even Field **R 6**

The display controller sets this bit at the first horizontal sync after a vertical sync during an odd field. This bit is cleared at the first horizontal sync after a vertical sync during an even field.

Top/Not Bottom Field **R 7**

This bit is set at the first horizontal sync after a vertical sync when top-field data is being displayed. This bit is cleared at the first horizontal sync after a vertical sync when bottom-field data is being displayed.

Figure 4.102 Register 276 (0x114)

<table>
<tr><th>7</th><th>6</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Field Sync Enable</td><td colspan="2">Display Mode [3:0]</td><td>Horizontal Filter Select</td><td>Horizontal Filter Enable</td><td>Last Field</td></tr>
</table>

Last Field **R 0**

When set, this bit indicates that the current field being displayed is the last field in the frame.

Horizontal Filter Enable **R/W 1**

Setting this bit enables the horizontal interpolation filter.

Horizontal Filter Select **R/W 2**

This bit sets the frequency response of the output filter to one of two preprogrammed values. When this bit is 1, frequency response ‘A’ is selected; when the bit is 0, frequency response ‘B’ is selected. See Section 10.6, “Display Modes and Vertical Filtering,” for more details.

Display Mode [3:0] **R/W [6:3]**

The host should set these bits based on the characteristics of the source video as shown in Table 4.1. These bits cause the display controller to operate in one of 12 different post-processing modes. Refer to Section

10.6, “Display Modes and Vertical Filtering,” for descriptions of each of these modes.

Table 4.1 Display Mode Selection Table

	Display Mode [3:0]											
Parameter	**0x0**	**0x1**	**0x2**	**0x3**	**0x4**	**0x5**	**0x6**	**0x7**	**0x8**	**0x9**	**0xA**	**0xB**
Field Structure Picture							x	x	x		x	x
Frame Structure Picture	x	x	x	x	x	x	x	x	x	x	x	x
16:9 Aspect Ratio				x					x			
4:3 Aspect Ratio	x	x	x		x	x	x	x		x	x	x
SIF Res. (240/288 lines)	x	x	x	x						x	x	
Full Res. (480/576 lines)					x	x	x	x	x			x
RMM					x	x	x	x	x		x	x

Field Sync Enable **R/W 7**

When this bit is set, the timing generator syncs to any field rather than just the even field for the purpose of enabling the decoder.

Figure 4.103 Register 277 (0x115) Horizontal Filter Scale [7:0]

7	0
Horizontal Filter Scale [7:0] R/W	

The host writes this value to set the interpolator for the output pixels when the filter is enabled. The equation for deriving the Horizontal Filter Scale value is:

$$\text{Horizontal Filter Scale} = \begin{cases} \left\lceil \dfrac{\text{Source Width}}{\text{Target Width}} \times 256 \right\rceil, \\ 0 \text{ if Source Width} = \text{Target Width} \end{cases}$$

where $\lceil x \rceil$ means to round the value x to the smallest integer larger than x.

Figure 4.104 Register 278 (0x116)

7	6 0
Reserved	Main Reads per Line [6:0]

Main Reads per Line [6:0] **R/W [6:0]**

This register is programmed with the number of reads required to construct a scan line for display. The value programmed is the number of pixels ÷ 8. Note that the number of display pixels may differ from main reads per line because of the horizontal interpolation filter.

Reserved **7**

Figure 4.105 Register 279 (0x117)

7	6	5 3	2 0
Automatic Field Inversion Correction	Pan and Scan from Bitstream	Pan and Scan Byte Offset [2:0]	Pan and Scan 1/8 Pixel Offset [2:0]

Pan and Scan 1/8 Pixel Offset [2:0] **R/W [2:0]**

The subpixel offset of the first pixel on which the display begins on each scan line.

Pan and Scan Byte Offset [2:0] **R/W [5:3]**

The byte number within an 8-byte word on which the display begins on each scan line.

Pan and Scan from Bitstream **R/W 6**

When set to 1, this bit enables the decoder to decode the pan and scan parameters from the bitstream. Clearing this bit allows the host to specify the pan and scan 1/8 pixel and byte offsets.

Automatic Field Inversion Correction **R/W 7**

When this bit is set and field inversion is detected, the Display Controller displays the next frame starting at display line two in the frame store.

Figure 4.106 Register 280 (0x118) Horizontal Pan and Scan Luma/Chroma Word Offset [7:0]

7 0
Horizontal Pan and Scan Luma/Chroma Word Offset [7:0] R/W

The word number on each scan line on which the display begins. Since the Display Controller supports both positive and negative horizontal pan and scan, this register only needs to apply an offset of up to ±90 words (720 pixels). These eight bits provide a signed (2’s complement) pan and scan offset of ±127 words.

Figure 4.107 Register 281 (0x119) Vertical Pan and Scan Line Offset [7:0]

7 0
Vertical Pan and Scan Line Offset [7:0] R/W

This register specifies the number of line pairs per field to pan vertically.

Figure 4.108 Register 282 (0x11A)

7 3	2 0
Reserved	Vline Count Init [2:0]

Vline Count Init [2:0] **R/W [2:0]**

This field contains the value to which the vertical line count initializes at the start of new field.

Reserved **[7:3]**

Figure 4.109 Register 283 (0x11B)

7	6 0
Reserved	Override Picture Width [6:0]

Override Picture Width [6:0] **R/W [6:0]**

This field contains the picture width of the override frame store in 8-pixel increments. In other words, this field should be programmed with picture width in pixels ÷ 8. This field is used only when the Display Override Mode bits (Register 265, bits 4 and 5, page 4-68) are set to 0b00 or 0b01.

Reserved **7**

Figure 4.110 Register 284 (0x11C)

7	6	5	4	3	2	1	0
Reserved	VSYNC Input Type	CrCb 2's Complement	Pixel State Reset Value [1:0]		Reserved	Sync Active Low	ITU-R BT.656 Mode

ITU-R BT.656 Mode **R/W 0**

When this bit is set, the L64020 sends out a 4-word code for the start and end of active video at blanking time.

Sync Active Low **R/W 1**

When this bit is set, the L64020 expects active low Horizontal and Vertical Sync inputs. If the bit is cleared, the chip expects active high sync inputs. The host should set this bit to match the sense of the sync inputs from the NTSC/PAL encoder.

Reserved **2**

Bit must be kept cleared.

Pixel State Reset Value [1:0] **R/W [4:3]**

The pixel state machine is initialized by the Horizontal Sync pulse. The initial state of this field is programmed by the host. This allows the host to adjust the pixel state timing such that the main region starts on the Cb state. This state machine follows this sequence; Cb, Y, Cr, Ys, Cb or Cr, Y, Cb, Ys, Cr. The Pixel State Reset Values are calculated using the following formula:

Pixel State Reset Value = (Main Start Column + 2) mod 4

CrCb 2's Complement **R/W 5**

When this bit is set, the chroma components are converted to 2's-complement values with the centers at 0 instead of 128. This is done by effectively inverting the MSB of the Cr and Cb values.

VSYNC Input Type **R/W 6**

When this bit is set, the Vertical Sync pulse is an Even/Not Odd field input. When this bit is cleared, the Vertical Sync input is a pulse.

Reserved **7**

Figure 4.111 Registers 285–288 (0x11D–0x120) Display Override Luma/Chroma Frame Store Start Addresses [15:0]

	7 0
Reg. 285 LSB	Display Override Luma Frame Store Start Address [7:0] R/W
Reg. 286 MSB	Display Override Luma Frame Store Start Address [15:8] R/W
Reg. 287 LSB	Display Override Chroma Frame Store Start Address [7:0] R/W
Reg. 288 MSB	Display Override Chroma Frame Store Start Address [15:8] R/W

The host can write to these registers to override the display picture luma and chroma frame store start addresses when the Display Override Mode (Register 265, bits 4 and 5, page 4-68) is set to Frame (0b01) or First Field Only (0b10).

Figure 4.112 Register 289 (0x121)

7	6 1	0
Reserved	Number of Segments in RMM [5:0]	Reserved

Reserved **0**

Number of Segments in RMM [5:0] **R/W [6:1]**

This register can be programmed by the host for the number of memory segments available for B pictures in Reduced Memory Mode (RMM). Each segment can store 8 lines of the picture. The maximum number of segments allowed is 54.

Reserved **7**

Figure 4.113 Register 290 (0x122)

7	2	1	0
Reserved		Television Standard Select [1:0]	

Television Standard Select [1:0] **W [1:0]**

The host can write to this register to select the TV standard per the following table:

Select [1:0]	Description
0b00	User programmed
0b01	NTSC (USA version)
0b10	PAL
0b11	Reserved

Reserved **[7:2]**

Registers 291–296 (0x123–0x128) Reserved **[7:0]**

Figure 4.114 Registers 297–299 (0x129–0x12B) Main Start/End Rows [10:0]

	7	6	4	3	2	0
Reg. 297 LSB	Main Start Row [7:0] R/W					
Reg. 298 LSB	Main End Row [7:0] R/W					
Reg. 299 MSBs	Reserved	Main End Row [10:8] R/W		Reserved	Main Start Row [10:8] R/W	

The host can write to these registers to program the start and end row numbers for the Main region on the display. The values entered are the number of lines from the start of a top or bottom field.

Figure 4.115 Registers 300–302 (0x12C–0x12E) Main Start/End Columns [10:0]

	7	6	4	3	2	0
Reg. 300 LSB	Main Start Column [7:0] R/W					
Reg. 301 LSB	Main End Column [7:0] R/W					
Reg. 302 MSBs	Reserved	Main End Column [10:8] R/W		Reserved	Main Start Column [10:8] R/W	

The host can write to these registers to program the start and end column numbers for the Main region on the display. The values entered are the number of system clocks from the horizontal sync signal.

Figure 4.116 Register 303 (0x12F)

7	6	5	4	0
Fcode [8]	Vcode Even Plus 1	Vcode Even [8]	Vcode Zero [4:0]	

Vcode Zero [4:0] **R/W [4:0]**

The host can write to this field to program the number of offset lines in the odd/even field beginning from the new field to the line where the Vcode of the Start of Active Video/End of Active Video (SAV/EAV) changes from 1 to 0. This the same for both odd and even fields.

Vcode Even [8] **R/W 5**

Most significant bit of Vcode Even. See the description in Register 304.

Vcode Even Plus 1 **R/W 6**

In the case of NTSC, the number of offset lines for the Vcode in the odd field is one greater than the even field. The host can program a one line difference by setting this bit. This feature is not required for PAL since the number of lines remains the same.

Fcode [8] **R/W 7**

Most significant bit of Fcode. See description in Register 305.

Figure 4.117 Register 304 (0x130) Vcode Even [7:0]

7	0
Vcode Even [7:0] R/W	

The host can write to this register and bit 5 of Register 303 to program the number of offset lines in the even field beginning from the new field to the line where the Vcode changes from 0 to 1. If the Vcode Even Plus 1 bit is cleared, this value is the same for both fields.

Figure 4.118 Register 305 (0x131) Fcode [7:0]

7	0
Fcode [7:0] R/W	

The host can write to this register and bit 7 of Register 303 to program the number of lines in a field beginning with the new field to the toggle of the Fcode of the SAV/EAV. This assumes that the Fcode will be the same for both fields.

Figure 4.119 Registers 306–308 (0x132–0x134) SAV/EAV Start Columns [10:0]

	7	6	4	3	2	0
Reg. 306 LSB	SAV Start Column [7:0] R/W					
Reg. 307 LSB	EAV Start Column [7:0] R/W					
Reg. 308 MSBs	Reserved	EAV Start Column [10:8] R/W		Reserved	SAV Start Column [10:8] R/W	

The host can write to these registers to define the start of SAV and EAV in terms of the number of system clocks from the horizontal sync.

Figure 4.120 Register 309 (0x135)

7 2	1	0
Reserved	SPU Mix Enable	Display Start Command

Display Start Command **R/W 0**

Setting this bit causes the display unit to start operation.

SPU Mix Enable **R/W 1**

When this bit is set, mixing of SPU display output with the regular display output is enabled. When the bit is cleared, mixing of the SPU output is disabled. Note that this does not affect SPU decoding, only mixing of the SPU output.

Reserved **[7:2]**

Registers 310–335 (0x136–0x14F) Reserved **[7:0]**

4.6 Audio Decoder Registers

Figure 4.121 Register 336 (0x150)

7	6 5	4	3 0
MPEG - ID	MPEG - layer_code [1:0]	MPEG - protection_bit	MPEG - bitrate_index [3:0]

MPEG - bitrate_index [3:0] **R [3:0]**

MPEG mode bitrate_index parsed from the bitstream. Table 4.2 shows the decoding of the index by layer. The 0x0 code indicates any fixed bitrate not included in the table where fixed means that a frame contains either N or N+1 slots, depending on the value of the padding bit.

Table 4.2 MPEG Bitrate Index Table

bitrate_index [3:0]	Layer I Bitrate (kbps)	Layer II Bitrate (kbps)	Layer III Bitrate (kbps)
0x0	free		
0x1	32	32	32
0x2	64	48	40
0x3	96	56	48
0x4	128	64	56
0x5	160	80	64
0x6	192	96	80
0x7	224	112	96
0x8	256	128	112
0x9	288	160	128
0xA	320	192	160
0xB	352	224	192
0xC	384	256	224
0xD	416	320	256
0xE	448	384	320
0xF	Not used		

MPEG - protection_bit R 4

MPEG mode protection bit parsed from the bitstream. A 0 bit means that redundancy has been added in the bitstream to facilitate error detection and concealment. A 1 bit indicates that no redundancy has been added.

MPEG - layer_code [1:0] R [6:5]

MPEG mode bitstream layer parsed from the bitstream. Indicates the MPEG layer in the bitstream per the following table:

layer_code Bits	MPEG Layer
0b00	Reserved
0b01	Layer III
0b10	Layer II
0b11	Layer I

MPEG - ID **R 7**

MPEG mode ID parsed from the bitstream. A 1 bit indicates MPEG-1 audio. A 0 bit indicates a low-sampling-rate MPEG-2 bitstream.

Figure 4.122 Register 337 (0x151)

7	6	5	4	3	2	1	0
MPEG - sampling_frequency [1:0]		MPEG - private_bit	MPEG - mode [1:0]		MPEG - mode_extension [1:0]		MPEG - copyright

MPEG - copyright **R 0**

MPEG mode copyright bit from the bitstream. A 1 bit indicates that the audio is copyright protected. A 0 bit indicates no copyright.

MPEG - mode_extension [1:0] **R [2:1]**

MPEG mode_extension [1:0] parsed from the bitstream. These bits are used in the joint_stereo mode. In Layer I and II, they indicate which subbands are in intensity stereo per the following table:

mode_extension Bits	Subbands in Intensity Stereo
0b00	4 to 31, bound = 4
0b01	8 to 31, bound = 8
0b10	12 to 31, bound = 12
0b11	16 to 31, bound = 16

In Layer III, the mode_extension bits indicate which type of joint_stereo coding method is used per the following table. The frequency ranges over which intensity stereo and ms_stereo are applied are implicit in the algorithm.

mode_extension Bits	Intensity Stereo	Ms Stereo
0b00	Off	Off
0b01	On	Off
0b10	Off	On
0b11	On	On

Note: The mode “stereo” is used if the mode bits specify stereo or the mode bits specify joint_stereo and the mode_extension bits are 0b00.

MPEG - mode [1:0] **R [4:3]**

MPEG mode [1:0] from the bitstream. These bits specify the mode according to the following table. In Layers I and II, the joint_stereo mode is intensity stereo. In Layer III, the joint_stereo mode is intensity stereo and/or ms_stereo.

Mode Bits	Mode
0b00	Stereo
0b01	Joint_stereo (intensity and/or ms-stereo)
0b10	Dual_channel
0b11	Single_channel

In all Layer I modes except joint_stereo, the bound equals 32. In all Layer II modes except joint_stereo, the bound equals sblimit (subband limit). In joint-stereo mode, the bound is determined by the mode extension.

MPEG - private_bit **R 5**

MPEG mode private_bit parsed from the bitstream. This bit is not used by ISO/IEC.

MPEG - sampling_frequency [1:0] **R [7:6]**

MPEG mode sampling_frequency parsed from the bitstream per the following table:

sampling_ frequency Bits	Sampling Frequency (kHz)
0b00	44.1
0b01	48
0b10	32
0b11	Reserved

Figure 4.123 Register 338 (0x152)

7	6 5	4 0
MPEG - original/copy	MPEG - emphasis [1:0]	Reserved

Reserved **[4:0]**

MPEG - emphasis [1:0] **R [6:5]**

MPEG mode emphasis from the bitstream.

MPEG - original/copy **R 7**

MPEG mode original/copy bit from the bitstream. A 1 bit indicates an original and a 0 bit indicates a copy.

Figure 4.124 Register 339 (0x153)

7 5	4 0
Dolby Digital - acmod [2:0]	Dolby Digital - dialnorm [4:0]

Dolby Digital - dialnorm [4:0] **R [4:0]**

Dolby Digital mode dialnorm parsed from the bitstream. Values of 1 to 31 indicate that the average dialogue level is -1 to -31 dB below digital 100%. The value 0 is reserved.

Dolby Digital - acmod [2:0] **R [7:5]**

Dolby Digital mode acmod from the bitstream. The audio coding mode bits indicates the number of front and surround channels in the bitstream.

Figure 4.125 Register 340 (0x154)

7 5	4 0
Dolby Digital - bsmod [2:0]	Dolby Digital - dialnorm2 [4:0]

Dolby Digital - dialnorm2 [4:0] **R [4:0]**

Dolby Digital mode dialnorm2 parsed from the bitstream. Same as dialnorm but for channel 2 in dual-mono mode.

Dolby Digital - bsmod [2:0] **R [7:5]**

Dolby Digital mode bsmod from the bitstream.These bits indicated the bitstream mode or type of service conveyed in the stream, such as, music and effects, dialogue, karaoke, etc.

Figure 4.126 Register 341 (0x155)

<table><tr><td>7</td><td>6</td><td>5</td><td>1</td><td>0</td></tr><tr><td colspan="2">Dolby Digital - surmixlev [1:0]</td><td colspan="2">Dolby Digital - mixlevel [4:0]</td><td>Reserved</td></tr></table>

Reserved **0**

Dolby Digital - mixlevel [4:0] **R [5:1]**
Dolby Digital mode mixlevel from the bitstream. Indicates the absolute acoustic sound pressure level of an individual channel during the final mixing session. Typically not used within the decoder but by other parts of the reproduction equipment.

Dolby Digital - surmixlev [1:0] **R [7:6]**
Dolby Digital mode surmixlev from the bitstream. The surround mix level bits indicate the nominal downmix level of the surround channels.

Figure 4.127 Register 342 (0x156)

<table><tr><td>7</td><td>6</td><td>5</td><td>1</td><td>0</td></tr><tr><td colspan="2">Dolby Digital - cmixlev [1:0]</td><td colspan="2">Dolby Digital - mixlevel2 [4:0]</td><td>Reserved</td></tr></table>

Reserved **0**

Dolby Digital - mixlevel2 [4:0] **R [5:1]**
Dolby Digital mode mixlevel2 from the bitstream. Same as mixlevel but for channel 2 in dual-mono mode.

Dolby Digital - cmixlev [1:0] **R [7:6]**
Dolby Digital mode cmixlev from the bitstream. The center mix level bits indicate the nominal downmix level of the center channel with respect to the left and right channels.

Figure 4.128 Register 343 (0x157)

7	6	5	4	0
Dolby Digital - fscod [1:0]		Dolby Digital - lfeon	Dolby Digital - bsid [4:0]	

Dolby Digital - bsid [4:0] **R [4:0]**

Dolby Digital mode bsid from the bitstream. The bitstream ID bits are set for the Dolby Digital version syntax supported by the audio encoder. Decoders built to a lower version syntax will mute the audio. The version current at this writing is 8 = 0b01000.

Dolby Digital - lfeon **R 5**

Dolby Digital mode lfeon from the bitstream. This bit is a 1 if the low frequency effects (LFE or subwoofer) channel is on and is a 0 if the LFE channel is off.

Dolby Digital - fscod [1:0] **R [7:6]**

Dolby Digital mode fscod from the bitstream. These bits indicate the audio sampling frequency.

Figure 4.129 Register 344 (0x158)

7	6	5	0
Dolby Digital - dsurmod [1:0]		Dolby Digital - frmsizecod [5:0]	

Dolby Digital - frmsizecod [5:0] **R [5:0]**

Dolby Digital mode frmsizecod from the bitstream. The frame size code and the sampling frequency code (fscod) can be used to calculate the number of 2-byte words in the bitstream before the next sync word.

Dolby Digital - dsurmod [1:0] **R [7:6]**

Dolby Digital mode dsurmod from the bitstream. These bits indicate whether or not the program has been encoded in Dolby Surround when operating in two-channel mode. They are not used by the decoder but by other components of the sound system.

Figure 4.130 Registers 345 and 346 (0x159 and 0x15A) Dolby Digital - langcod/langcod2 [7:0]

	7	0
Reg. 345	Dolby Digital - langcod [7:0] Read Only	
Reg. 346	Dolby Digital - langcod2 [7:0] Read Only	

Dolby Digital mode langcod and langcod2 from the bitstream. These two 8-bit fields indicate which language(s) have been encoded in channel 1 and channel 2.

Figure 4.131 Register 347 (0x15B) Dolby Digital - timecod1 [13:6]

7	0
Dolby Digital - timecod1 [13:6] Read Only	

Dolby Digital mode timecod1 bits [13:6]. Bits [5:0] are in Register 348. The 14-bit time code indicates the audio synchronization time in hours, minutes, and 8-second intervals up to 24 hours.

Figure 4.132 Register 348 (0x15C)

7	2	1	0
Dolby Digital - timecod1 [5:0]		Dolby Digital - roomtyp [1:0]	

Dolby Digital - roomtyp [1:0] R [1:0]

Dolby Digital mode roomtyp from the bitstream. These two bits indicate the type and calibration of the room used for the final audio mixing session. The bits are not used by the decoder but by other components of the audio system.

Dolby Digital - timecod1 [5:0] R [7:2]

Dolby Digital mode timecod1 [5:0]. Most significant 8 bits are in Register 347.

Figure 4.133 Register 349 (0x15D) Dolby Digital - timecod2 [13:6]

<table>
<tr><th>7</th><th>0</th></tr>
<tr><td colspan="2">Dolby Digital - timecod2 [13:6]
Read Only</td></tr>
</table>

Dolby Digital mode timecod2 bits [13:6]. Bits [5:0] are in Register 350. These 14 bits extend the resolution of timecod1 to seconds (0 to 7), frames (0 to 29), and 1/64 of a frame (0 to 63).

Figure 4.134 Register 350 (0x15E)

<table>
<tr><th>7</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Dolby Digital - timecod2 [5:0]</td><td colspan="2">Dolby Digital - roomtype2 [1:0]</td></tr>
</table>

Dolby Digital - roomtype2 [1:0] **R [1:0]**

Dolby Digital mode roomtype2 from the bitstream. These bits have the same meaning as the roomtyp bits but apply to channel 2 in dual-mono mode.

Dolby Digital - timecod2 [5:0] **R [7:2]**

Dolby Digital mode timecod2 [5:0]. Most significant 8 bits are in Register 349.

Figure 4.135 Register 351 (0x15F)

<table>
<tr><th>7</th><th>3</th><th>2</th><th>0</th></tr>
<tr><td colspan="2">PCM - audio_frm_num [4:0]</td><td colspan="2">PCM - num_of_audio_ch [2:0]</td></tr>
</table>

PCM - num_of_audio_ch [2:0] **R [2:0]**

PCM num_of_audio_ch parameter from Linear PCM bitstream.

PCM - audio_frm_num [4:0] **R [7:3]**

PCM audio_frm_num parameter from Linear PCM bitstream.

Figure 4.136 Register 352 (0x160)

7	6	5	4	3	2	1	0
PCM - Fs [1:0]		PCM - quantization [1:0]		PCM - emphasis [1:0]		PCM - mute_bit	Reserved

Reserved **0**

PCM - mute_bit **R 1**
PCM mute_bit parameter from Linear PCM bitstream.

PCM - emphasis [1:0] **R [3:2]**
PCM emphasis parameter from Linear PCM bitstream.

PCM - quantization [1:0] **R [5:4]**
PCM quantization parameter from Linear PCM bitstream.

PCM - Fs [1:0] **R [7:6]**
PCM Fs parameter from Linear PCM bitstream.

Figure 4.137 Register 353 (0x161)

7	6	5	4				0
PCM FIFO Full	PCM FIFO Near Full	PCM FIFO Empty	Reserved				

Reserved **[4:0]**

PCM FIFO Empty **R 5**
This bit is set when the PCM FIFO is empty.

PCM FIFO Near Full **R 6**
This bit is set when the PCM FIFO is near full, i.e., contains 25 bytes or more of unread data.

PCM FIFO Full **R 7**
This bit is set when the PCM FIFO is full.

Figure 4.138 Register 354 (0x162)

7	5	4	3	2	1	0
Reserved		MPEG Multichannel Extension Sync Word Missing	Audio Decoder Reconstruct Error	Audio Decoder Soft Mute Status	Audio Decoder Play Mode Status [1:0]	

Audio Decoder Play Mode Status [1:0] R [1:0]

This field indicates the status of the audio decoder as shown in the following table. This field is valid only when the Dolby Digital, MPEG, or Linear PCM Decoder is enabled. The field is reset to 0b00 (pause mode) when the L64020 is powered up and reset.

Audio Decoder Play Mode Status Bits	Description
0b00	Pause
0b01	Normal play
0b10	Decimate (fast)
0b11	Interpolate (slow)

Audio Decoder Soft Mute Status R 2

This bit is set by the Audio Decoder and the audio output is muted when any of the following occurs:

1. The PCM - soft_mute_bit in Register 252 is set from the bitstream or the Audio Interface.

2. The User Soft Mute bit in Register 358 is set by the host.

3. The Mute on Error bit in Register 358 is set and errors are detected.

Audio Decoder Reconstruct Error R 3

This bit is set when an error is encountered by the audio decoder while reconstructing a frame. Reading this bit clears it.

MPEG Multichannel Extension Sync Word Missing R 4

This bit is set when the multichannel extension synchronization word can not be found during the decode process of the MPEG audio multichannel bitstream. Reading this bit clears it.

Reserved [7:5]

Figure 4.139 Register 355 (0x163)

7	6	5	4	0
Audio Decoder Start/Stop	Audio Decoder Play Mode [1:0]		Reserved	

Reserved **[4:0]**

Audio Decoder Play Mode [1:0] **R/W [6:5]**

In MPEG and Dolby Digital modes, these bits command the audio decoder to pause, play at normal speed, play at a faster rate, or play at a slower rate according to the following table. During these modes the audio decoder presents 15/16 of the normal frame data to the output PCM filter (fast) or 17/16 of the normal frame data to the output PCM filter (slow) within 1 normal frame decode.

Audio Decoder Play Mode Bits	Description
0b00	Pause
0b01	Normal play
0b10	Decimate (fast)
0b11	Interpolate (slow)

The field is reset to pause mode. In pause mode, the audio decoder stops parsing the bitstream and maintains the current state so that reselecting normal play at a later time does not cause the decoder to lose sync to the bitstream. These bits are only valid when the Audio Decoder Start/Stop bit in this register is set.

The Linear PCM Decoder responds as above except that the rate for fast playback is 7/8 of normal and the rate for slow playback is 9/8 times normal.

Audio Decoder Start/Stop **R/W 7**

When this bit is set, the selected audio decoder (MPEG, Dolby Digital, or Linear PCM) starts decoding. Clearing this bit stops the decoder and flushes the data from the Audio ES channel buffer.

Figure 4.140 Register 356 (0x164)

7	6 5	4 0
Audio Formatter Start/Stop	Audio Formatter Play Mode [1:0]	Reserved

Reserved **[4:0]**

Audio Formatter Play Mode [1:0] **R/W [6:5]**

These bits command the Dolby Digital Formatter, MPEG Formatter, and the S/P DIF Interface to either perform normal play or pause. These bits are only valid when the Audio Formatter Start/Stop bit in this register is set.

Audio Formatter Play Mode Bits	Description
0b00	Pause
0b01	Normal play
0b10	Reserved
0b11	Reserved

Audio Formatter Start/Stop **R/W 7**

Setting this bit starts the selected formatter (MPEG or Dolby Digital) and S/P DIF Interface. Clearing this bit stops the formatter and interface.

Important: The host must clear the Audio Formatter Start/Stop bit before selecting Audio Module Mode 0b000, 0b001, 0b100, or 0b101 (see Table 4.3). That is, formatters must be stopped before selecting nonformatter modes and not started unless the mode is changed to include a formatter.

Figure 4.141 Register 357 (0x165)

7 5	4 0
Audio Decoder Mode Select [2:0]	Reserved

Reserved **[4:0]**

Audio Decoder Mode Select [2:0] **R/W [7:5]**

These bits control the selection of modes that are allowable in the Audio Decoder according to Table 4.3.

See the Important note following the description of the Audio Formatter Start/Stop bit, bit 7 in Register 356.

Table 4.3 Audio Decoder Modes

Select Bits	DAC Interface	S/P DIF Interface
0b000	MPEG Decoder	MPEG Decoder output PCM samples converted to IEC958 format
0b001	Dolby Digital Decoder	Dolby Digital Decoder output PCM samples converted to IEC958 format
0b010	MPEG Decoder	MPEG Formatter
0b011	Dolby Digital Decoder	Dolby Digital Formatter
0b100	PCM Decoder	Linear PCM samples converted to IEC958 format. NOTE: If the sample frequency in the Linear PCM bitstream is 96 kHz, then the IEC958 output is derived from an on-chip filter that converts from 96-kHz to 48-kHz sample frequency.
0b101	PCM Decoder output decimated through on-chip filter to convert from 96-kHz to 48-kHz sample rate. This mode should only be set if the output is for a DAC that supports 48-kHz sample frequency only.	Same as DAC, converted to IEC958 format.
0b110	CD Bypass	S/P DIF bypass
0b111	PCM FIFO	PCM FIFO

Figure 4.142 Register 358 (0x166)

7	6	5	4	3	2	1	0
Mute on Error	User Mute Bit	Reserved	Dolby Digital Downmix Mode	Audio Dual-Mono Mode [1:0]		Dolby Digital Compression Mode [1:0]	

Dolby Digital Compression Mode [1:0] R/W [1:0]

These bits select the Dolby Digital dynamic range compression mode according to the following table.

Compression Mode Bits	Description
0b00	Custom analog mode (no dialogue normalization)
0b01	Custom digital mode (digital dialogue normalization)
0b10	Line-out mode (default)
0b11	RF mode

Refer to the Dolby Digital specification for a complete description of the different modes. The default mode is line-out (0b10).

Audio Dual-Mono Mode [1:0] R/W [3:2]

These bits select which audio channel (left/right) the dual-mono data is sent out. For Dolby Digital dual-mono streams only, the outputs are attenuated by 3 dB in modes 0b01 and 0b10 and by 6 dB in mode 0b11. The default at power up and reset is the 0b00 stereo mode.

Audio Dual-Mono Mode	Audio Dual Mono Output Mode
0b00	Stereo: L channel on L speaker, R channel on R speaker (default)
0b01	Right: R channel on L and R speaker
0b10[1]	Left: L channel on L and R speaker
0b11	Mix mono

[1] This selection should be used for mono bitstreams to avoid noise on the right speaker.

Dolby Digital Downmix Mode R/W 4

This bit sets the downmix mode per the following table.

Dolby Digital Downmix Mode Bits	Dolby Digital Stereo Output Format
0	Conventional stereo mix (default)
1	Dolby surround stereo mix

Refer to the Dolby Digital specification for a complete description of the downmix mode. The default value of this register is 0 (conventional stereo mix).

Reserved **5**

User Mute Bit **R/W 6**

When this bit is set, the audio outputs are muted.

Mute on Error **R/W 7**

When this bit is set, the MPEG and Dolby Digital audio outputs are muted for any of the following conditions:

- ♦ audio CRC error
- ♦ audio illegal bit error
- ♦ audio sync error
- ♦ audio reconstruction error

The default value of this bit is 1 (soft mute enabled)

Note: The Linear PCM output is always muted when errors occur regardless of the setting of this bit.

Figure 4.143 Register 359 (0x167) PCM FIFO Data In [7:0]

7	0
PCM FIFO Data In [7:0] W	

The host should issue four consecutive write operations for each pair of PCM samples to be played at the output when the PCM FIFO Mode is enabled (see Audio Decoder Mode Selection [2:0] on page 4-93). The PCM data should be written to this register in the following order: Left Channel LSB, Left Channel MSB, Right Channel LSB, and Right Channel MSB. This register is write only.

Figure 4.144 Register 360 (0x168) Dolby Digital/Linear PCM - dynscalehigh [7:0]

7	0
Dolby Digital/Linear PCM - dynscalehigh [7:0] R/W	

This is an 8-bit, fractional, scale factor for scaling the dynrng value coded in the Dolby Digital bitstreams. The dynscalehigh factor is applied when the dynrng value in the Dolby Digital bitstream indicates a negative dB gain. dynscalehigh = 0x00 disables the dynrng scaling intended in the bitstream; dynscalehigh = 0xFF applies the full dynamic range scaling coded in the bitstream. Intermediate values (dynscalehigh = 0x01, ... , 0xFE) scale the dynrng value by factors of 2/256 to 255/256. The dynscalehigh setting can be used to effectively boost the dynamic range of the program. The default value of this register is 0xFF. This register also can be used by the host to perform similar scaling for Linear PCM bitstreams.

Figure 4.145 Register 361 (0x169) Dolby Digital/Linear PCM - dynscalelow [7:0]

7	0
Dolby Digital/Linear PCM - dynscalelow [7:0] R/W	

This is an 8-bit, fractional, scale factor for scaling the dynrng value coded in Dolby Digital bitstreams when it is a positive dB gain. See dynscalehigh in the previous register.

Figure 4.146 Register 362 (0x16A) PCM Scale [7:0]

7	0
PCM Scale [7:0] R/W	

This is an 8-bit, fractional, scale factor for scaling output PCM from any of the audio decoders. PCM Scale = 0x00 mutes the audio output; PCM Scale = 0xFF keeps the output PCM scaled as decoded. Intermediate values (0x01, ... , 0xFE) scale the output PCM by factors of 2/256 to 255/256. The default value of this register is 0xFF.

Figure 4.147 Register 363 (0x16B)

7	6	5	3	2	1	0
Valid	User	Reserved		Invert LRCLK	ACLK Select [1:0]	

ACLK Select [1:0] **R/W [1:0]**

These bits select the external audio clock used for generating the DAC and S/P DIF clocks. Note that N in the table below stands for 768, 512, 384, or 256 according to the ACLK_ multiple(s) provided. See also the ACLK Divider Select bits, bits [3:0] in Register 364.

ACLK Select	External Clock Used
0b00	ACLK_441 (44.1 kHz * N)
0b01	ACLK_48 (48 kHz * N)
0b10	ACLK_32 (32 kHz * N)
0b11	Reserved

The default value is 01 (48 kHz).

Invert LRCLK **R/W 2**

The audio LRCLK output signal polarity indicates to the external audio DAC to which channel, left or right, the current audio sample belongs. The default setting of this bit is 0 which means that right samples are output when LRCLK is high and left samples are output when LRCLK is low. Setting this bit inverts the LRCLK sense. Set or clear this bit according to the requirements of your audio DAC.

Reserved **[5:3]**

User **R/W 6**

The value of the User bit to be packed in the IEC958 (S/P DIF) output. The default is 0.

Valid **R/W 7**

The data Valid bit to be packed in the IEC958 (S/P DIF) output. The bit is set when the S/P DIF output is from a formatter in the Audio Decoder and is cleared when the output is from one of the audio decoders.

Figure 4.148 Register 364 (0x16C)

7 6	5	4	3 0
Dolby Digital - Karaoke Mode [1:0]	Dolby Digital - Karaoke Center Level On	LPCM - Dynamic Range On	ACLK Divider Select [3:0]

ACLK Divider Select [3:0] **R/W [3:0]**

The host sets these bits to select clock divider values which derive the S/P DIF interface BCLK, DAC interface BCLK, and external DAC A_ACLK from the selected

ACLK_ input (bits 0 and 1 in Register 363). The divider values depend on ACLK_ availability, the input audio sampling frequency (Fs), the sample resolution (16/24/32 bits per sample), and the external DAC capabilities. The L64020 supports sampling rates of 32, 44.1, and 48 kHz for MPEG and Dolby Digital, and 48 and 96 kHz for Linear PCM. The equations for the derived clocks are:

S/P DIF BCLK = Fs * 32 bits per sample * 2 channels * 2
biphase marks = Fs * 128

DAC BCLK = Fs * 32 bits per sample * 2 channels
= Fs * 64

Ext DAC A_ACLK = Fs * 32 bits per sample * K
= Fs * 256 or Fs * 384
where K = the oversampling factor.

The available divider settings are listed in Table 4.4. Use the following cases as selection criteria:

- Case I: All of the ACLK_ inputs are available. Select the ACLK_ which is an integer multiple of the input sampling frequency using bits 0 and 1 in Register 363. Then use the 0x0 through 0x4 ACLK Divider Select code that matches the Fs-multiple of the ACLK_. For example, if the input sampling frequency is 32 kHz and ACLK_32 = 512 * 32 kHz, use the 0x2 ACLK Divider Select code.
- Case IIA: The Linear PCM bitstream with a sampling frequency of 96 kHz is selected and the external DAC supports 96-kHz sampling frequency. ACLK_48 at a multiple of 512 or 768 must be available and it must be selected. Use divider code 0x5 for ACLK = 768 * 48 or code 0x6 for ACLK = 512 * 48.
- Case IIB: The Linear PCM bitstream with a sampling frequency of 96 kHz is selected but the external DAC does not support 96-kHz sampling frequency. ACLK_48 must be available and it must be selected. Set the Audio Decoder Mode Select field (Register 357, bits [7:5], page 4-93) to 0b101 to decimate the output samples to 48 kHz. Use the 0x0 through 0x4 divider code that matches the ACLK_48 multiple.
- Case III: The input sampling rate is 32 kHz but ACLK_32 is not available. Select ACLK_48 and the 0xC through 0xF divider code that matches the

ACLK_48 multiple to derive the 32-kHz clocks from ACLK_48.

Note: The CD bypass mode has a dedicated ACLK input pin called CD_ACLK.

Table 4.4 ACLK Divider Select [3:0] Code Definitions

ACLK Divider Select [3:0]	ACLK Input	S/P DIF Interface BCLK	DAC Interface BCLK	DAC A_ACLK
0x0	768*Fs	128*Fs=ACLK÷6	64*Fs=ACLK÷12	256*Fs=ACLK÷3
0x1	768*Fs	128*Fs=ACLK÷6	64*Fs=ACLK÷12	384*Fs=ACLK÷2
0x2	512*Fs	128*Fs=ACLK÷4	64*Fs=ACLK÷8	256*Fs=ACLK÷2
0x3	384*Fs	128*Fs=ACLK÷3	64*Fs=ACLK÷6	384*Fs=ACLK÷1
0x4	256*Fs	128*Fs=ACLK÷2	64*Fs=ACLK÷4	256*Fs=ACLK÷1
0x5	768*48	128*48=ACLK÷6	64*96=ACLK÷6	384*96=ACLK÷1
0x6	512*48	128*48=ACLK÷4	64*96=ACLK÷4	256*96=ACLK÷1
0x7–0xB	Not Used			
0xC	768*48	128*32=ACLK÷9	64*32=ACLK÷18	384*32=ACLK÷3
0xD	512*48	128*32=ACLK÷6	64*32=ACLK÷12	256*32=ACLK÷3
0xE	512*48	128*32=ACLK÷6	64*32=ACLK÷12	384*32=ACLK÷2
0xF	256*48	128*32=ACLK÷3	64*32=ACLK÷6	256*32=ACLK÷1

LPCM - Dynamic Range On **R/W 4**

Setting this bit in Linear PCM Mode enables the dynamic range feature of the Linear PCM bitstream. When the bit is cleared, dynamic range control is off and the PCM samples recovered from the bitstream are not multiplied by the gain value. The default value of this bit is 0.

Dolby Digital - Karaoke Center Level On **R/W 5**

When this bit is set, the karaoke center guide melody is on. See the following description for the karaoke downmix equations.

Dolby Digital - Karaoke Mode [1:0] **R/W [7:6]**

The karaoke downmix equations are:

$$L_{ko} = (1.0 \times L) + (\text{clevel} \times M) + (\alpha_1 \times V1) + (\beta_1 \times V2)$$

$$R_{ko} = (1.0 \times R) + (\text{clevel} \times M) + (\alpha_2 \times V1) + (\beta_2 \times V2)$$

where the guide melody M is turned on or off with the Karaoke Center Level On bit in this register and the Vocal 1 (V1) and Vocal 2 (V2) coefficients are selected by the mode as in the following table.

Karaoke Mode	**Karaoke Output**	α_1	α_2	β_1	β_2
0b00	No vocal output	0	0	0	0
0b01	Vocal 1 only	0.7	0.7	0	0
0b10	Vocal 2 only	0	0	0.7	0.7
0b11	Vocal 1 + Vocal 2	1.0	0	0	1.0

Figure 4.149 Register 365 (0x16D)

7	6	5	4	2	1	0
IEC - Overwrite Copyright	IEC - Host Copyright	IEC - Overwrite Emphasis	IEC - Host Emphasis [2:0]		Reserved	

Reserved **[1:0]**

IEC - Host Emphasis [2:0] **R/W [4:2]**

When the overwrite emphasis bit (bit 5 in this register) is set, the value in the host emphasis field is used instead of the emphasis value in the bitstream.

IEC - Overwrite Emphasis **R/W 5**

When this bit is set, the value in bits [4:2] of this register are used instead of the emphasis value in the bitstream. The default value of this bit is 0.

IEC - Host Copyright **R/W 6**

When the overwrite copyright bit (bit 7 in this register) is set, the value of the Host Copyright bit is used instead of the copyright value in the bitstream.

IEC - Overwrite Copyright **R/W 7**

When this bit is set, the value in bit 6 of this register is used instead of the copyright value in the bitstream. The default value of this bit is 0.

Figure 4.150 Register 366 (0x16E)

7	6	5	4	3	2	1	0
Formatter ES1 Compliant	Formatter Skip Frame Size [1:0]		MPEG Formatter Only	Overwrite Quantization	Host Quantization [1:0]		Overwrite Category

Overwrite Category **R/W 0**

When this bit is set, the category code in the channel status word of the S/P DIF frame is overridden by the value written in Register 367 (page 4-103) by the host. When this bit is cleared, the default category codes are shown in the following table:

Data Format	Default Category Code
PCM Sample	0x00
Digital Data	0x98

Host Quantization [1:0] **R/W [2:1]**

The host can specify the quantization value here and override the value in the bitstream if the Overwrite Quantization Enable bit (bit 3 in this register) is set. For example, the host may want to force 16-bit quantization even though the input bitstream has 20-bit resolution. The audio decoder appropriately truncates or extends PCM samples to achieve this. The following table shows the encoding of the quantization values.

Bits [2:1]	Host Quantization
0b00	16 bit
0b01	20 bit
0b10	24 bit
0b11	Not used

Overwrite Quantization **R/W 3**

When the host sets this bit, the value of the quantization parameter is specified by the Host Overwrite Quantization bits (bits 1 and 2 in this register]). When this bit is cleared (default), the quantization is as specified in the bitstream.

MPEG Formatter Only **R/W 4**

When this bit is set, the MPEG Formatter runs stand-alone without the MPEG Decoder being activated.

Formatter Skip Frame Size [1:0] **R/W [6:5]**

These bits control the behavior of the MPEG and Dolby Digital Formatters. To achieve synchronization between the decoders and the outputs sent to the IEC958 (S/P DIF) Interface, the formatter may decide to skip entire frames or to pause (wait) for a frame. The decision to skip or to wait is made by the formatter after comparing the fullness levels of the respective internal buffers (FIFOs) that feed the input MPEG and Dolby Digital bitstream to the decoders and formatters. When the difference in fullness levels of the two buffers goes beyond a certain threshold, the audio output of the DAC and S/P DIF will not be considered as acceptably synchronized. When this happens, the formatters take appropriate action, either skipping a frame or waiting until the two get back in synchronization. The only setting for these thresholds is 0b00, 2 frames for MPEG and 1 frame for Dolby Digital.

Formatter ES1 Compliant **R/W 7**

When this bit is set, the behavior of the formatters is compliant to that of ES1. When this bit is cleared, it is compliant with the Dolby Digital specification.

Figure 4.151 Register 367 (0x16F) Host Category [7:0]

7	0
Host Category [7:0] R/W	

This value can be set by the host to override the existing category code when the Overwrite Category bit (bit 0 in Register 366) is set.

Figure 4.152 Register 368 (0x170)

7	5	4	3	2	1	0
Host Pc Info [2:0]		Pd Selection [1:0]		Reserved	Pd Data Valid	Reserved

Reserved **0**

Pd Data Valid **R 1**

When the Pd Selection bits (3 and 4 in this register) are 0b10 (host force mode) and the host writes a Host Pd Value to Registers 369 and 370, this bit is set. When the internal MPEG Audio Formatter reads the existing Host Pd Value, this bit is cleared. ThIs provides the host a means of detecting exactly when the previous data was used and when it is safe to set the Host Pd Value for the next IEC958 frame.

Reserved **2**

Pd Selection [1:0] **R/W [4:3]**

This value in this field (see the following table) determines from where the MPEG Audio Formatter picks up the value of the Pd parameter.

Bits [4:3]	Description
0b00	Previous audio packet
0b01	Base packet without extension
0b10	Host force
0b11	Reserved

Host Pc Info [2:0] **R/W [7:5]**

The host writes the Pc info to be loaded into the MPEG burst preamble Host Pc info field bits [10:8] into this field.

Figure 4.153 Registers 369 and 370 (0x171 and 0x172) Host Pd Value [15:0]

	7	0
Reg. 369 MSB	Host Pd Value [15:8] R/W	
Reg. 370 LSB	Host Pd Value [7:0] R/W	

When the Pd Selection bits (3 and 4 in Register 368) are 0b10 (host force mode), the host writes a Host Pd Value for the MPEG burst preamble into these registers.

Registers 371–383 (0x173–0x17F)
Reserved for diagnostic use [7:0]

4.7 RAM Test Registers

Figure 4.154 Registers 384 and 385 (0x180 and 0x181) Memory Test Address [11:0]

	7	4	3	0
Reg. 384 MSB	Memory Test Address [7:0] R/W			
Reg. 385 LSB	Reserved		Memory Test Address [11:8] R/W	

The host writes an address to these registers for a host-controlled testing of a single address (bits [1:0] of Register 386 set to 0b01). During automated test modes, these registers are updated by the L64020 to indicate the progress of the tests.

Figure 4.155 Register 386 (0x182)

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Reserved</td><td>Memory Test Output Select</td><td colspan="2">Data Pattern to be Applied to RAM [1:0]</td><td>Report End of Test/Initiate Memory Test</td><td colspan="2">Operational Mode for RAM Test [1:0]</td></tr>
</table>

Operational Mode for RAM Test [1:0] — W [1:0]

The host writes to this field to specify the type of memory test to be run according to the following table.

Operational Mode [1:0]	Description
0b00	Normal (no test)
0b01	Host-controlled testing of memories for a single address
0b10	Automated RAM test
0b11	Automated ROM test

Report End of Test — R 2

This bit is cleared by the L64020 at the conclusion of the memory test.

Initiate Memory Test — W 2

The host sets this bit to start the memory test specified by bits 0 and 1 of this register.

Data Pattern to be Applied to RAM [1:0] — W [4:3]

This field contains the 2-bit repeated pattern to be applied during the automated RAM test. These bits are set by the L64020 during automated RAM test and should be set by the host during host-controlled testing of RAM (mode 0b01 of the memory test).

Memory Test Output Select — R/W 5

Setting this bit enables the overall memory test pass/fail status to assert the AREQn output signal of the L64020 for test pass.

Note: This bit should be set only when a memory test is to be run. This bit defaults to 0 at reset and should be maintained at 0 during normal functional mode.

Reserved — [7:6]

Figure 4.156 Registers 387–392 (0x183–0x188) Memory Test Pass/Fail Status Bits

	7	6	5	4	3	2	1	0
Reg. 387	MemTest08	MemTest07	MemTest06	MemTest05	MemTest04	MemTest03	MemTest02	MemTest01
Reg. 388	MemTest16	MemTest15	MemTest14	MemTest13	MemTest12	MemTest11	MemTest10	MemTest09
Reg. 389	MemTest24	MemTest23	MemTest22	MemTest21	MemTest20	MemTest19	MemTest18	MemTest17
Reg. 390	MemTest32	MemTest31	MemTest30	MemTest29	MemTest28	MemTest27	MemTest26	MemTest25
Reg. 391	Reserved				MemTest36	MemTest35	MemTest34	MemTest33
Reg. 392	Overall Mem Test Pass/Fail Status	Reserved				MemTest39	MemTest38	MemTest37

Each bit in the above read-only registers indicates the pass/fail status of a memory in the L64020; a 1 for pass and a 0 for failed.

Registers 393–415 (0x189–0x19F) Reserved [7:0]

4.8 SPU Decoder Registers

Figure 4.157 Register 416 (0x1A0)

7	6	5	4	3	2	1	0
Reserved	Frame-Based Execution	Jump to Next SPU	Reset Autofill Counter for SPU Palette	SPU Display Force Off	SPU Display Off	SPU Pause	SPU Decode Start

SPU Decode Start R/W 0

Setting this bit causes the SPU Decoder to start decoding. Clearing the bit resets the decoder. Note that the SPU channel is not controlled by this bit and will not stop even if it is overflowing.

SPU Pause R/W 1

Setting this bit causes the SPU Decoder to stop checking timing. Therefore, the SPU Decoder will not decode the next DCSQ or unit. The default value of this bit at power-up and reset is 0.

SPU Display Off **R/W 2**

Setting this bit causes the SPU Decoder to clear the SPU window. However, any SPU windows displayed by the FSTA_DSP command are not cleared.

SPU Display Force Off **R/W 3**

Setting this bit causes the SPU Decoder to clear the SPU window even if it is displayed by the FSTA_DSP command.

Reset Autofill Counter for SPU Palette **R/W 4**

Setting this bit causes the SPU palette write address counter to be reset to the Y0 address. The address increments when the host writes to Register 446 (page 4-110).

Jump to Next SPU **R/W 5**

Setting this bit causes the SPU Decoder to stop decoding the current unit and to clear the SPU window on the next new field. The decoder then waits until the start of the next unit. This bit is cleared automatically by the SPU Decoder when it stops decoding the current unit.

Frame-Based Execution **R/W 6**

When this bit is set, the SPU Decoder executes a DCSQ on the beginning of only odd fields. When the bit is cleared, the SPU Decoder executes a DCSQ on the beginning of every field (even or odd). This bit does not affect the timing comparison set up in Register 417.

Reserved **7**

Figure 4.158 Register 417 (0x1A1)

7 4	3 0
Reserved	Command Time-Out for SPU [3:0]

Command Time-Out for SPU [3:0] **R/W [3:0]**

This is the error margin for DSCQ time-stamp analysis. If the register contains a nonzero value, the following inequality is used to determine time-out:

$$0 < \text{Time-Out} < \frac{STC - PTS}{1024} - SP\_DCSQ\_STM$$

If this register is set to zero, no time-out error interrupts are generated.

Reserved **[7:4]**

Figure 4.159 Registers 418–421 (0x1A2–0x1A5) PTS in Current SPU [31:0]

	7 … 0
Reg. 418 LSB	PTS in Current SPU [7:0] Read Only
Reg. 419	PTS in Current SPU [15:8] Read Only
Reg. 420	PTS in Current SPU [23:16] Read Only
Reg. 421 MSB	PTS in Current SPU [31:24] Read Only

Registers 418 through 421 contain the PTS in the current unit.

Figure 4.160 Registers 422–425 (0x1A6–0x1A9) PTS in Next SPU [31:0]

	7 … 0
Reg. 422 LSB	PTS in Next SPU [7:0] Read Only
Reg. 423	PTS in Next SPU [15:8] Read Only
Reg. 424	PTS in Next SPU [23:16] Read Only
Reg. 425 MSB	PTS in Next SPU [31:24] Read Only

Registers 422 through 425 contain the PTS in the next unit. If the next unit is not in the channel buffer, this value is equal to the PTS in Current SPU.

Figure 4.161 Registers 426 and 427 (0x1AA and 0x1AB) SP_DCSQ_STM in Next DCSQ [15:0]

	7	0
Reg. 426 LSB	SP_DCSQ_STM in Next DCSQ [7:0] Read Only	
Reg. 427 MSB	SP_DCSQ_STM in Next DCSQ [15:8] Read Only	

Registers 426 and 427 contain the SP_DCSQ_STM in the next DCSQ.

Figure 4.162 Registers 428–430 (0x1AC–0x1AE) SPU Base Pointer [19:0]

	7	4	3	0
Reg. 428 LSB	SPU Base Pointer [7:0] Read Only			
Reg. 429	SPU Base Pointer [15:8] Read Only			
Reg. 430	Reserved		SPU Base Pointer [19:16] Read Only	

These registers contain the SPU base pointer word address in which SPUSZ (SPU size) is located.

Registers 431–445 (0x1AF–0x1BD) Reserved [7:0]

Figure 4.163 Register 446 (0x1BE) Color Palette Data for SPU [7:0]

7	0
Color Palette Data for SPU [7:0] W	

This register is used by the host to load the SPU palette. Each write to this register increments the internally maintained address counter. The fill order is Y0, Cr0, Cb0, Y1, ... See the description of bit 4 in Register 416 (page 4-108) on how to initialize the color palette fill operation.

Figure 4.164 Register 447 (0x1BF)

7	1	0
Reserved		Highlight Enable

Highlight Enable **R/W 0**

When this bit is set, the SPU Decoder highlights the area from the next field.

Reserved **[7:1]**

Figure 4.165 Registers 448 and 449 (0x1C0 and 0x1C1) Highlight Color Info [15:0]

	7	0
Reg. 448 LSB	Highlight Color Info [7:0] W	
Reg. 449 MSB	Highlight Color Info [15:8] W	

These registers supply the color information for highlight. The bit order is as follows:

emphasis-2 color
emphasis-1 color
pattern color
background color

Figure 4.166 Registers 450 and 451 (0x1C2 and 0x1C3) Highlight Contrast Info [15:0]

	7	0
Reg. 450 LSB	Highlight Contrast Info [7:0] W	
Reg. 451 MSB	Highlight Contrast Info [15:8] W	

These registers supply the contrast information for highlight. The bit order is as follows:

emphasis-2 contrast
emphasis-1 contrast
pattern contrast
background contrast

Figure 4.167 Registers 452–457 (0x1C4–0x1C9) Highlight Area Info [47:0]

<table>
<tr><th></th><th>7</th><th>0</th></tr>
<tr><td>Reg. 452
LSB</td><td colspan="2">Highlight Area Info [7:0]
W</td></tr>
<tr><td>Reg. 453</td><td colspan="2">Highlight Area Info [15:8]
W</td></tr>
<tr><td>Reg. 454</td><td colspan="2">Highlight Area Info [23:16]
W</td></tr>
<tr><td>Reg. 455</td><td colspan="2">Highlight Area Info [31:24]
W</td></tr>
<tr><td>Reg. 456</td><td colspan="2">Highlight Area Info [39:32]
W</td></tr>
<tr><td>Reg. 457
MSB</td><td colspan="2">Highlight Area Info [47:40]
W</td></tr>
</table>

These registers supply the area information for highlight. The bit allocation is the same as the button position information in the *DVD Specifications for Read-Only Disc*.

Figure 4.168 Register 458 (0x1CA)

7	6	5	4	3	2	1	0
Illegal Unit Error Flag	Sync Word Error	Size Error	Unit Store Error	Unit Error	Syntax Error	Time Stamp Error	PXD FIFO Underflow

PXD FIFO Underflow **R 0**

This bit is set when there is SDRAM bandwidth shortage. When this error occurs, the SPU Decoder clears the display immediately and resumes display from the next field.

Time Stamp Error **R 1**

This bit is set when elapsed time from the PTS exceeds the limit time set by the Command Time-out for SPU (bits [3:0] in Register 417, page 4-108). When this error occurs, the SPU Decoder clears the display immediately, skips the current unit, and then waits for the next PTS.

Syntax Error **R 2**

This bit is set when the SPU Decoder identifies the reserved code for DCCMD or an incorrect code for the PXCD end code. When this error occurs, the SPU Decoder clears the display immediately, skips the current unit, and then waits for the next PTS.

Unit Error **R 3**

This bit is set when the unit error flag is inserted by the preparser. When this occurs, the SPU Decoder clears the display immediately, skips the current unit, and then waits for the next PTS.

Unit Store Error **R 4**

This bit is set when the current PTS is valid if the whole SPU data is not stored in the SDRAM. When this error occurs, the SPU Decoder clears the display immediately and then searches for the next sync word.

Size Error **R 5**

This bit is set when the channel indicates an error in the SPUSZ field. When this error occurs, the SPU Decoder clears the display immediately and then searches for the next sync word.

Sync Word Error **R 6**

This bit is set when the SPU Decoder identifies an incorrect sync word. When this error occurs, the SPU Decoder clears the display immediately and then searches for the next sync word.

Illegal Unit Error Flag **R 7**

This bit is set when SPU Decoder identifies an incorrect code for the unit. When this error occurs, the SPU Decoder clears the display immediately and then searches for the next sync word.

Register 459 (0x1CB) Reserved [7:0]

Figure 4.169 Register 460 (0x1CC) SPU State Machine Info [7:0]

7	0
SPU State Machine Info [7:0] Read Only	

This register is for diagnostics only.

Registers 461–479 (0x1CD–0x1DF) Reserved for future expansion of SPU Module [7:0]

Registers 480–511 (0x1E0–0x1FF) Reserved for future expansion of the L64020 [7:0]

Chapter 5
Host Interface

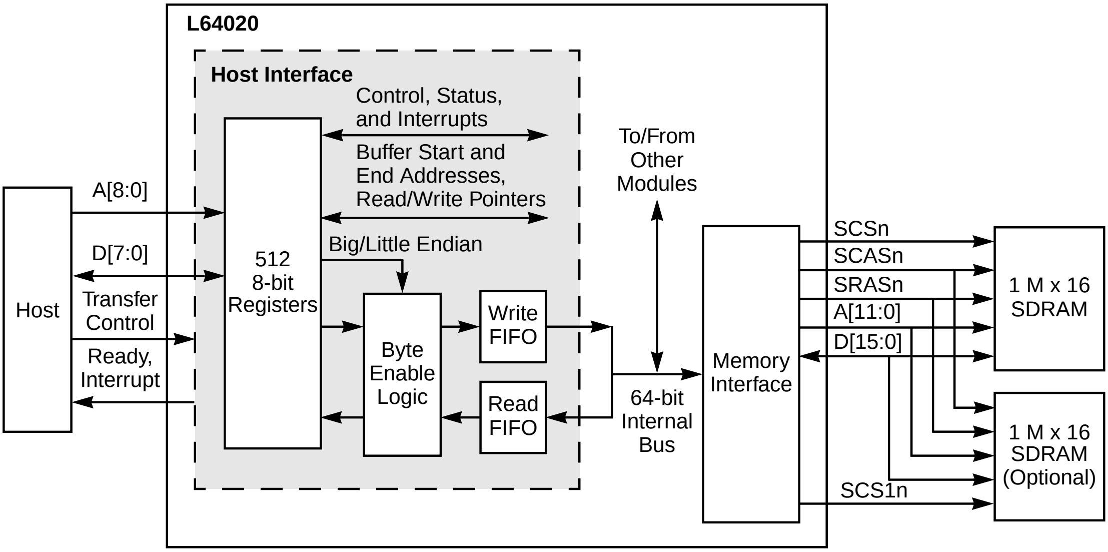
This chapter describes the host's interface to the L64020 chip and external SDRAM. Refer to Chapter 7 for a complete description of the interface between the L64020 and external SDRAM. This chapter includes the following sections:

5.1 Overview

Figure 5.1 shows a block diagram of the Host Interface. The host communicates with the L64020 and external SDRAM through 512, 8-bit registers. (All of the registers are not currently used; some are reserved for future changes to the chip.) The chip provides a 9-bit input address bus, A[8:0], to reach all 512 registers and a register-wide (8-bit), bidirectional data bus, D[7:0]. Refer to Chapter 3 for a summary of the registers and Chapter 4 for descriptions of the registers.

The host accesses external SDRAM through a set of registers, byte enabling logic for big/little endian control, read and write FIFOs, and the Memory Interface block of the L64020 chip.

Figure 5.1 Host Interface Block Diagram



5.2 Interface Signals

The host interface is configurable for either an Intel or a Motorola processor. Table 5.1 lists the signals for each processor. The configuration selection is made by tying the BUSMODE pin of the chip to a VDD (+3.3 V) or a VSS (ground) pin.

Table 5.1 Host Interface Signals

Signal	L64020 Direction	Intel Mode	Motorola Mode
BUSMODE	Input	Tied low (logic 0).	Tied high (logic 1).
A[8:0]	Input	A[8:0]	A[8:0]
ASn	Input	ASn	ASn
D[7:0]	Input/Output	D[7:0]	D[7:0]
DSn/WRITEn	Input	WRITEn	DSn
CSn	Input	CSn	CSn
(Sheet 1 of 2)			

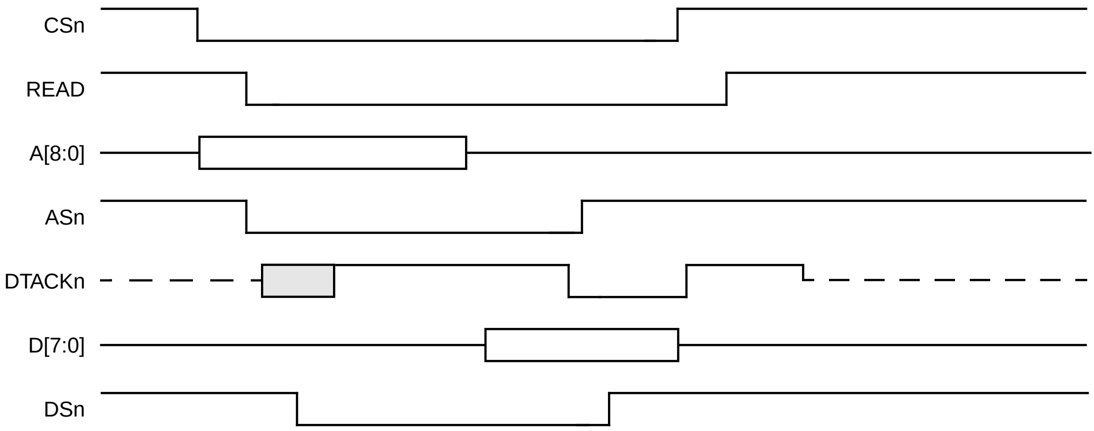
Table 5.1 Host Interface Signals (Cont.)

Signal	L64020 Direction	Intel Mode	Motorola Mode
READ/READn	Input	READn	READ
DTACKn/RDYn	Output (3-state)	RDYn	DTACKn
WAITn/WTN	Output (3-state)	WTN	WAITn
INTRn	Output (open drain)	INTRn	INTRn
DREQn	Output	DREQn	DREQn
PREQn	Output	PREQn	PREQn
(Sheet 2 of 2)			

Figure 5.2 shows the interface signal timing for a Motorola mode write cycle. The host asserts the chip select (CSn) signal to inform the L64020 that it wishes to read or write. The host then drives READ low to signal that it is a write cycle and asserts ASn to strobe the address onto the interface address bus, A[8:0].

When the L64020 detects CSn active, it drives its DTACKn output high to inform the host that it is not ready for a read or write and low when it is ready. In the example shown, the decoder initially set DTACKn high to delay the cycle. After DTACKn goes low and if the data is stable, the host deasserts DSn to strobe the data into the decoder.

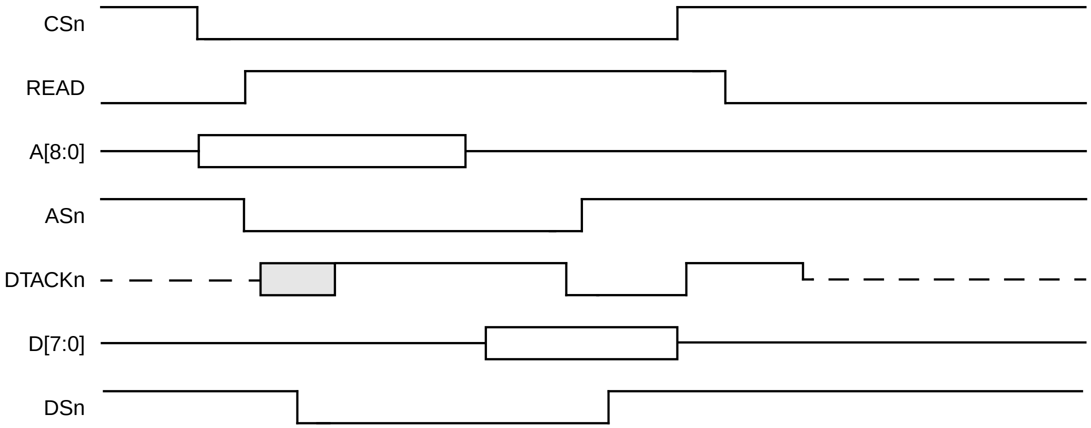
Figure 5.2 Motorola Mode Write Timing



The cycle can be terminated by the L64020 setting DTACKn high or by the host deasserting CSn. When CSn is deasserted, the L64020 3-states its DTACKn output.

The Motorola mode read timing is shown in Figure 5.3. The read timing is very similar to that for write. The only difference is that the READ signal is asserted for the cycle.

Figure 5.3 Motorola Mode Read Timing



The write and read timing for Intel host processors are shown in Figure 5.4 and Figure 5.5. Intel processors use separate read/write signals. The address is strobed onto A[8:0] at the negative-going edge of the read/write signal, and the data is strobed into the L64020 on the positive-going edge of the read/write signal. The address can be placed on the bus even though the L64020 is not ready for the transfer. The host holds the WRITEn signal asserted until the L64020 asserts DTACKn. If the decoder does not respond within 107.5 ns of the falling edge of WRITEn, the host aborts the write. For a read, the host waits for 144.5 ns from the falling edge of READn for DTACKn to be asserted before aborting the operation.

Figure 5.4 Intel Mode Write Timing

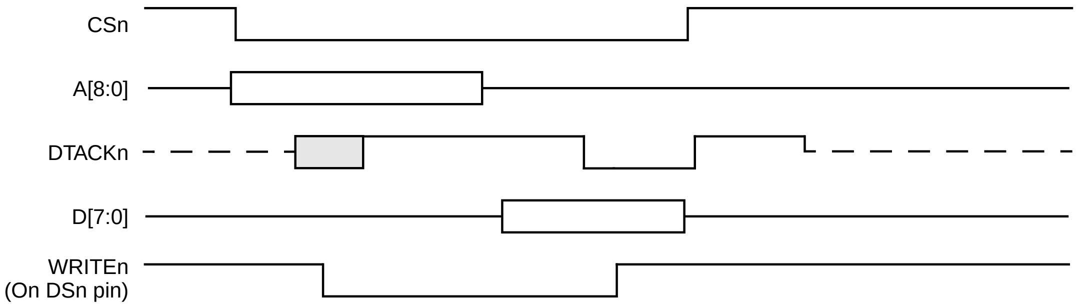
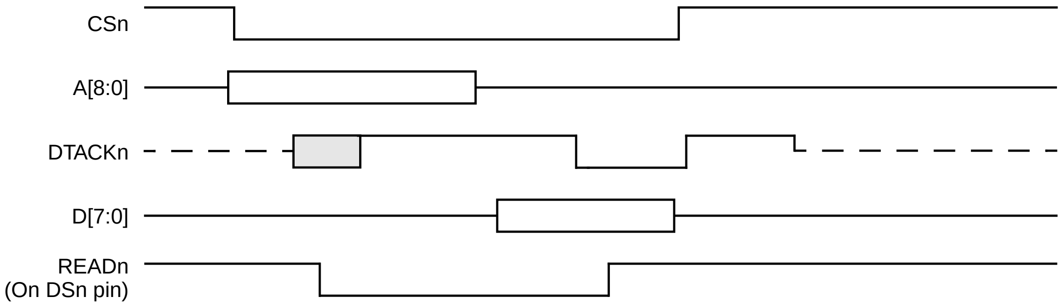


Figure 5.5 Intel Mode Read Timing



5.3 Register Access and Functions

The registers of the L64020 Decoder are accessed when the host places their address (0x000 through 0x1FF) on the A[8:0] input lines of the chip and starts a read or write operation.

5.3.1 General Functions

The registers contain status bits and fields, control bits and fields, SDRAM buffer pointers for bitstream header fields and data, System Clock Reference (SCR) capture and compare values and control bits, and host to SDRAM access addresses and data. The latter group are described in the following section. A complete summary of all of the registers is included in Chapter 3 and detailed descriptions of all register bits and fields are provided in Chapter 4.

When any of the interrupt bits in the first few registers are set, the L64020 also asserts the INTRn output signal to the host. The INTRn signal alerts the host to read the interrupt registers to determine the reason for the interrupt and take the necessary action. Any of these interrupts can be masked to prevent the assertion of INTRn for that condition.

The control bits and fields allow the host to determine the modes of operation of the chip. Many of these also serve to show the current status of the chip.

As the input bitstream is parsed, address pointers to the different header fields and data elements in the SDRAM buffers are written to registers by the chip as information to the host. The host can write to some of these registers to specify the start and end addresses of the various header and data buffers in SDRAM.

5.3.2 SCR Registers

The L64020 contains a 32-bit, free-running counter for maintaining a System Clock Reference. This counter is incremented every 300 clock cycles and serves as the basic time reference for the device, including the automatic synchronization function of the SPU decoder. The SCR counter has a number of features which enhance the synchronization of audio and video. Figure 5.6 shows the functional operation of the SCR counter circuits.

The general operating mode depends on the host's setting of the SCR Compare/Capture Mode bits in Register 17. The bit assignments and modes are listed in Table 5.2.

Table 5.2 SCR Compare/Capture Mode Bits

Mode Bits	Mode
0b00	No compare and capture happens. SCR overflow works.
0b01	Capture mode
0b10	Compare mode
0b11	Reserved

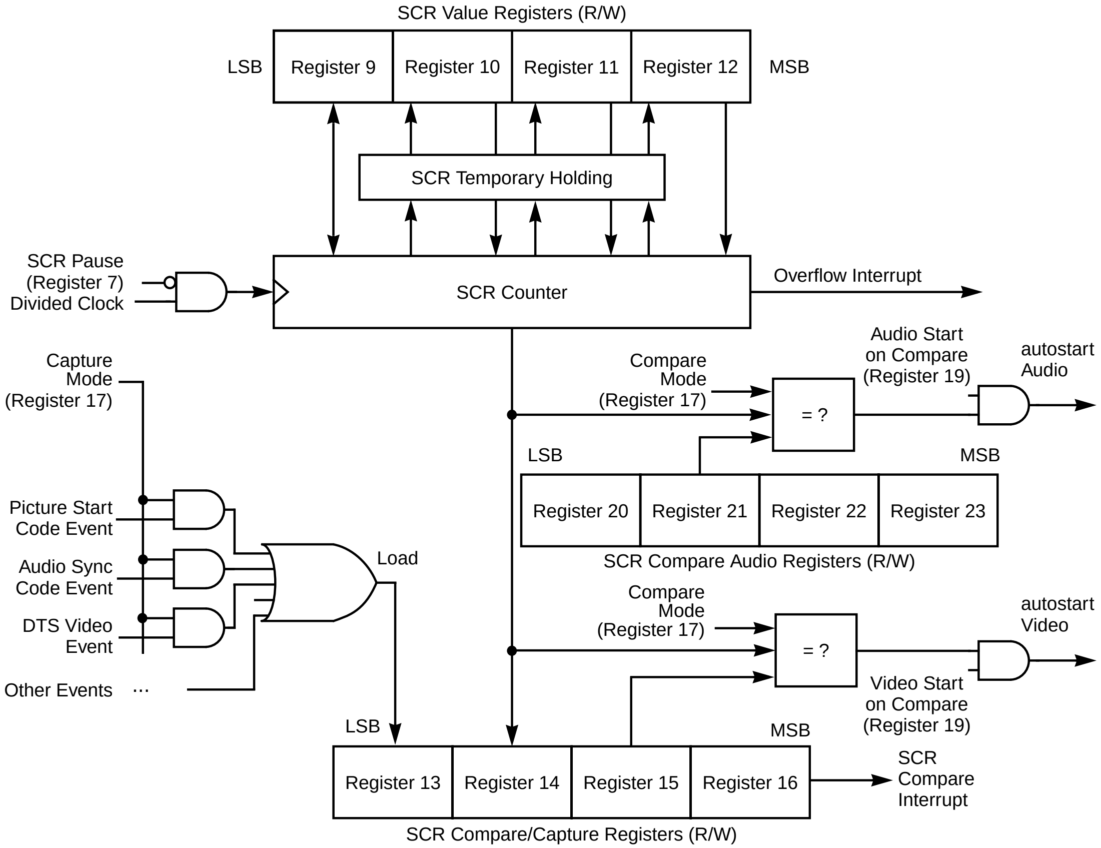
In the No Compare and Capture mode, the SCR counter can be read, paused, and loaded by the host through the SCR Value registers. The

L64020 only keeps the LSB in Register 9 updated. When the host reads the LSB, the upper three bytes of the counter are captured and written to Registers 10, 11, and 12. To load a value into the counter, the host must set the SCR Pause bit in Register 7, write the new counter value in the SCR Value registers, and then clear the SCR Pause bit. The SCR counter then increments from the value in the SCR Value registers.

The pause function of the SCR counter also pauses the automatic synchronization circuitry of the SPU Decoder, and can be used for slow forwarding of the SPU Decoder.

Also in this mode, when the SCR counter overflows, the SCR Overflow Interrupt bit in Register 1 is set and the INTRn output to the host is asserted if not masked by the host for this interrupt.

Figure 5.6 Operation of the SCR Counter



In the Capture Mode, the host can select an event in the bitstream to use for capturing the value of the SCR counter. When the preparser in the chip detects the selected event, the SCR counter value is loaded into the SCR Compare/Capture registers. The host can read these registers at some later time to determine exactly when the event occurred. The following events can be selected for SCR capture:

- Picture Start Code
- Audio Sync Code
- Beginning of Active Video (BAV)
- Pack Data Ready
- Audio PES Ready
- Video PES Ready
- SPU PES Ready
- DSI PES Ready
- DTS Video
- DTS Audio

The Compare Mode can be used to generate an interrupt, start the video decoder, or start the audio decoder. To generate an interrupt, the host writes the desired compare value into the SCR Compare/Capture registers. When the SCR counter reaches the compare value, the INTRn output signal to the host is asserted. To start video decoding on compare, the host writes the compare value as just described and sets the Video Start on Compare bit in Register 19. When the SCR counter reaches the compare value, an autostart signal is sent to the video decoder.

To start audio on compare, the host writes the compare value in the SCR Compare Audio registers (Registers 20 through 23) and sets the Audio Start on Compare bit in Register 19. This generates an autostart to the audio Decoder when the SCR counter reaches the compare value.

Note that the video decoder must be stopped prior to the autostart video signal and the audio decoder must be PAUSED prior to the autostart audio signal. The autostart functions commonly are used at the start-up time of the disk and also are used to restart the audio after a pause encountered by a seamless playback break on a DVD disk.

Note: The compare/capture circuits can only be used in one mode at a time. Usually, they are initially set to the No Compare or Capture mode so the PCR value in the first System header can be loaded into the SCR counter. Then they are placed in the autostart on Compare mode while the channel buffers are filling up. Once the decoding has started, the SCR circuits can be placed in the Capture mode to monitor the progress of decoding based on incoming events.

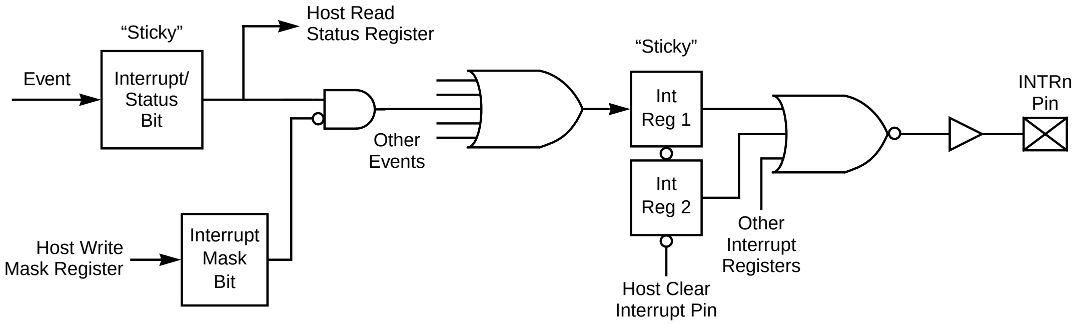
5.3.3 Interrupt Registers

In addition to the SCR Compare/Capture events, the L64020 uses other events (single cycle internal pulses occurring at a specific time) to tell the host when critical items have happened in the decoder. These events are needed in various systems to signal error conditions, channel buffer conditions, A/V sync information, and general data flow through the decoder. The events can be used as interrupts or simply as status information.

Registers 0 through 4 (see Chapter 4) contain 38 status/interrupt bits. These bits are set by the L64020 when their corresponding event occurs and the INTRn interrupt output signal is asserted to the host if the event is not masked.

Figure 5.7 shows the interrupt structure. The event sets an interrupt/status bit in one of the host-accessible registers. If the interrupt mask for that bit is not set by the host, the event is ORed with other events to set one of the inaccessible IntReg registers. The outputs of these registers are ORed and the result is inverted to assert the INTRn output signal low.

Figure 5.7 Interrupt Structure



When the host detects INTRn asserted, it reads all of the interrupt/status registers to determine the cause of the interrupt and take any necessary action. The host read clears the interrupt/status bit but does not clear the associated IntReg. To deassert INTRn, the host must set the Clear Interrupt Pin bit in Register 6.

Note that, if an unmasked interrupt/status bit is still set at the time the Clear Interrupt Pin bit is set, INTRn deasserts for 1 clock cycle and then returns to its active state immediately, indicating pending events. The host must read all of the interrupt/status registers prior to setting the Clear Interrupt Pin bit.

This type of interrupt structure is designed for use in systems with multiple interrupt priorities such that the interrupt routines can exit at any point to service higher priority interrupts. As soon as all higher priority interrupts are serviced and disabled, the L64020 becomes the next highest priority as the interrupt pin can be left active. In this manner, the interrupt hardware provides a mechanism to leave and return to the interrupt service routine cleanly.

5.4 SDRAM Access

The SDRAM controller in the L64020 provides three methods of SDRAM access by the host:

- Host Read/Write
- DMA Read/Write
- SDRAM Block Move

5.4.1 Host Reads/Writes

For host/SDRAM read/writes, the host loads a 19-bit address into the Host SDRAM Target Address (page 4-50) for writes and the Host SDRAM Source Address (page 4-50) for reads. These addresses are written as if they are the upper 19 bits of a 21-bit SDRAM address. The most significant bit asserts either CS or CS1 to select one of the SDRAM chips. The remaining bits are converted to row and column addresses by the Memory Interface. Since the internal data bus of the L64020 is 64 bits wide, the SDRAM is set up to transfer a block of four, 16-bit words at each access. It does so by setting the two least significant column address bits to 00 to start and then incrementing them to transfer the four words.

The host has access to two 8-bit registers for SDRAM transfers, the Host SDRAM Write Data register (page 4-50) and the Host SDRAM Read Data register (page 4-49). The host can transfer the 8-byte data block through the registers in big or little endian order by setting or clearing the Host SDRAM Transfer Byte Ordering bit in Register 193 (page 4-49). The L64020 operates in big endian mode, i.e., byte 0 occupies the upper bits of the word and byte 8 occupies the lower bits.

The transfers are paced by the FIFO status bits in Register 192 (page 4-47). The host must read the status bits before writing or reading the next 8 bytes to or from the data registers and before starting a new transfer.

5.4.1.1 Host Read

The host read mode uses the SDRAM Source Address (Registers 199 through 201) as the SDRAM pointer for reading. This address is autoincremented after a word is loaded from SDRAM into the on-chip FIFO.

Figure 5.8 shows the flow for host reads from and writes to SDRAM. The host begins a SDRAM read operation by setting or clearing the Host SDRAM Byte Ordering bit (if necessary) to change the endian mode and then writing the Host SDRAM Source Address. Typically, the Host SDRAM Byte Ordering bit is set or cleared by the host at initialization and not changed again. When the host writes in the LSB of the source address, the L64020 automatically resets the pointers of the host read

FIFO (Figure 5.1) and begins to fill the FIFO with new data from the source address.

After setting the source address, the host must check the Host Read FIFO Empty status bit. If the host read FIFO is not empty, the host may read 1 byte from the Host SDRAM Read Data register. The host may continue to read from this register until 8 bytes have been read from the host read FIFO. After 8 bytes are read, the FIFO read pointer is automatically incremented and the host can continue to read data.

When the host is finished with the current host SDRAM read operation, it must wait for the Host Read FIFO Full bit to be set before beginning any new SDRAM operation (host r/w, DMA r/w, or block move.)

5.4.1.2 Host Write

The host write operation proceeds similarly to the host read operation. The host begins a SDRAM write operation by setting or clearing the Host SDRAM Byte Ordering bit (if necessary) to change the endian mode and then writing the Host SDRAM Target Address, LSB last.

The host can then begin to write bytes to the Host SDRAM Write Data register. The host can continue to write bytes to the write register as long as the Host Write FIFO Full bit is not set.

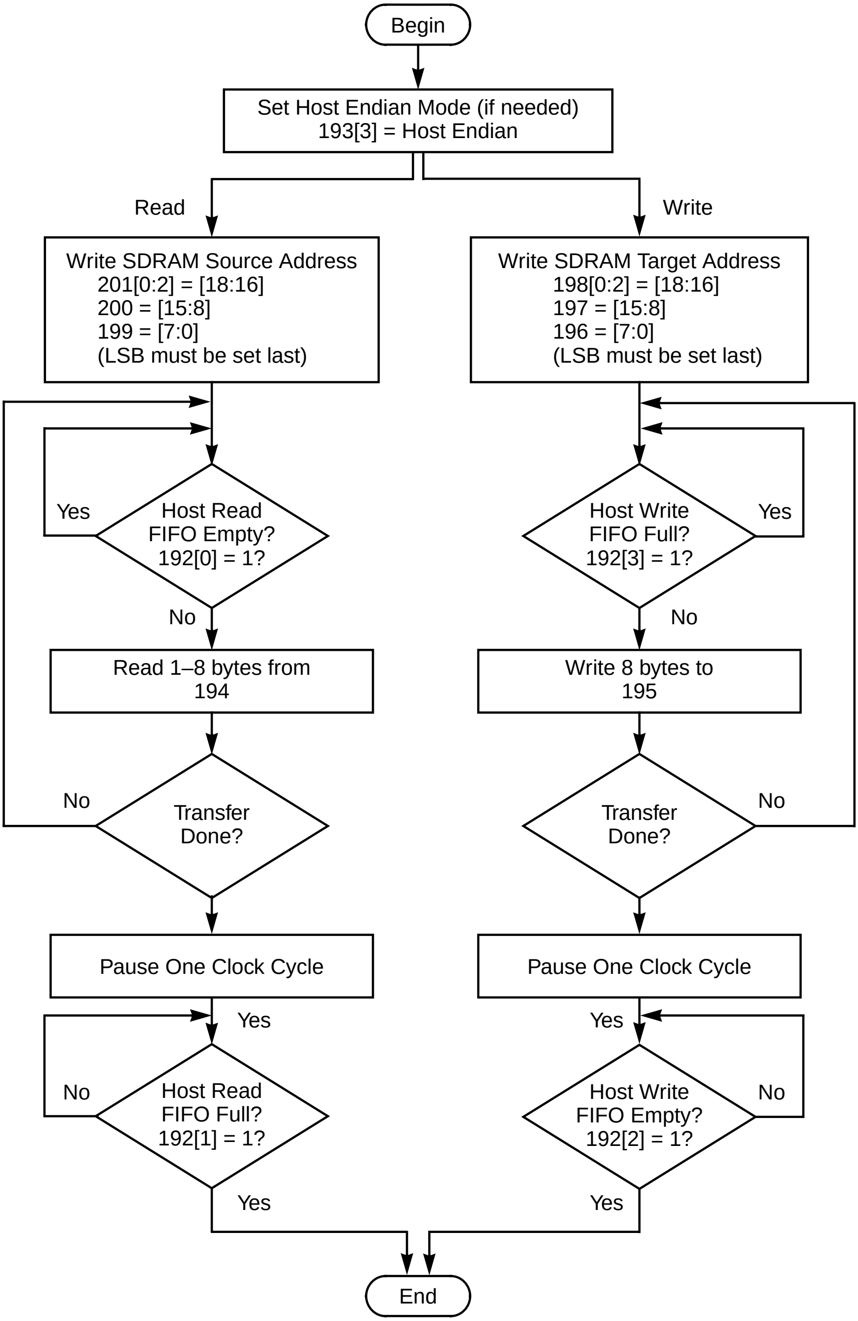
The L64020 only writes data out of the host write FIFO when a complete 8-byte (64-bit) word is available.

Caution: If the host attempts to write less than eight bytes of data to SDRAM, the data will not be transferred to SDRAM. The host can continue to transfer blocks of eight bytes as long as the Host Write FIFO Full bit is not set.

When the host is finished with the current host SDRAM write operation, it must wait for the Host Write FIFO Empty bit to be set before beginning any new SDRAM operation (host r/w, DMA r/w, or block move.)

Note in Figure 5.8 that the host source and target addresses must be entered with the LSB last. Writing the LSB of the source or target address causes the host read or write FIFO to reset, respectively. Also, note that the FIFO status registers require 1 clock cycle to update. There should be at least 1 clock cycle separating the last read/write command and checking the FIFO status registers.

Figure 5.8 Host Read/Write Flowchart



5.4.2 Host DMA SDRAM Transfers

Host DMA transfers to/from SDRAM through the L64020 are supported with the DMA Transfer Request (DREQn) output signal. This signal is asserted to the host during DMA reads when the DMA RdFIFO contains more than one 64-bit word and during DMA writes when the DMA WrFIFO has space left for more than one 64-bit word. Control of the DREQn signal is determined by the setting of the DMA Mode bits in Register 193 as shown in Table 5.3.

Table 5.3 DMA Mode Bits

Register 193[2:1]	**DMA Mode**
0b00	DMA Idle; DREQn = 1
0b01	DMA Read; DREQn = RdFIFO near-empty
0b10	DMA Write; DREQn = WrFIFO near-full
0b11	Block Move; DREQn = 1

The DREQn signal can be used as an input to a host DMA controller that accepts a level-sensitive DREQn input. The DMA controller can read a few bytes beyond the end timing of the DREQn pin and still function correctly. Note that the DREQn signal will continue to request data transfer for a read or write operation after the DMA controller has reached the terminal count. The L64020 is not responsible for monitoring the DMA terminal count.

Host DMA read/write operations may proceed in parallel with standard host read/write operations. The registers, FIFOs, and counters for DMA and host operations are completely independent. Since there is only one physical data access port on the L64020, the host must arbitrate host read/write and DMA read/write operations through it.

Block move operations may NOT proceed in parallel with host read/write operations or DMA read/write operations. The block move corrupts any data left in the FIFOs at the time the block move begins.

5.4.2.1 DMA Read

The system can use a dual-address DMA controller with a nonincrementing source address for DMA read operations. For a DMA read (refer to Figure 5.9), the host first sets the DMA Mode to Idle to prevent DMA operation until everything is ready. This holds DREQn to the host deasserted. Next, the host sets the DMA Transfer Byte Ordering bit to the DMA controller's endian, if necessary. Then the host writes the SDRAM starting address of the transfer to the DMA SDRAM Source Address registers. When the LSB of the source address is written into its register, the L64020 flushes the DMA RdFIFO and starts refilling it from the source address. To start the read, the host sets the DMA Mode bits to Read. Since there should be more than one 8-byte word in the RdFIFO at this time, DREQn is asserted to the host.

The external DMA controller then starts reading the data bytes from the DMA SDRAM Read Data register. DREQn will remain asserted so long as there are at least two words in the RdFIFO. The L64020 SDRAM controller automatically increments the source address after each 8-byte word is read from the SDRAM into the DMA RdFIFO.

The external DMA controller is responsible for setting the initial transfer count and decrementing it after reading each 8-byte word. The SDRAM controller will continue to increment the SDRAM address and transfer bytes into the DMA RdFIFO until the FIFO is full or the host changes the DMA Mode. In a normal DMA read, the DMA controller must stop reading bytes from the DMA SDRAM Read Data register when its transfer count reaches zero even though DREQn is still asserted. The L64020 SDRAM controller fills the RdFIFO if it is not already full. After the transfer count reaches zero, the host must read the DMA Read FIFO Full bit. When the hosts detects that the full bit is set, it should set the DMA Mode to Idle to deassert DREQn.

Note: The L64020 requires one clock cycle after the DMA RdFIFO is full to set the DMA RdFIFO Full bit. The host should wait for one clock cycle before reading the bit.

5.4.2.2 DMA Write

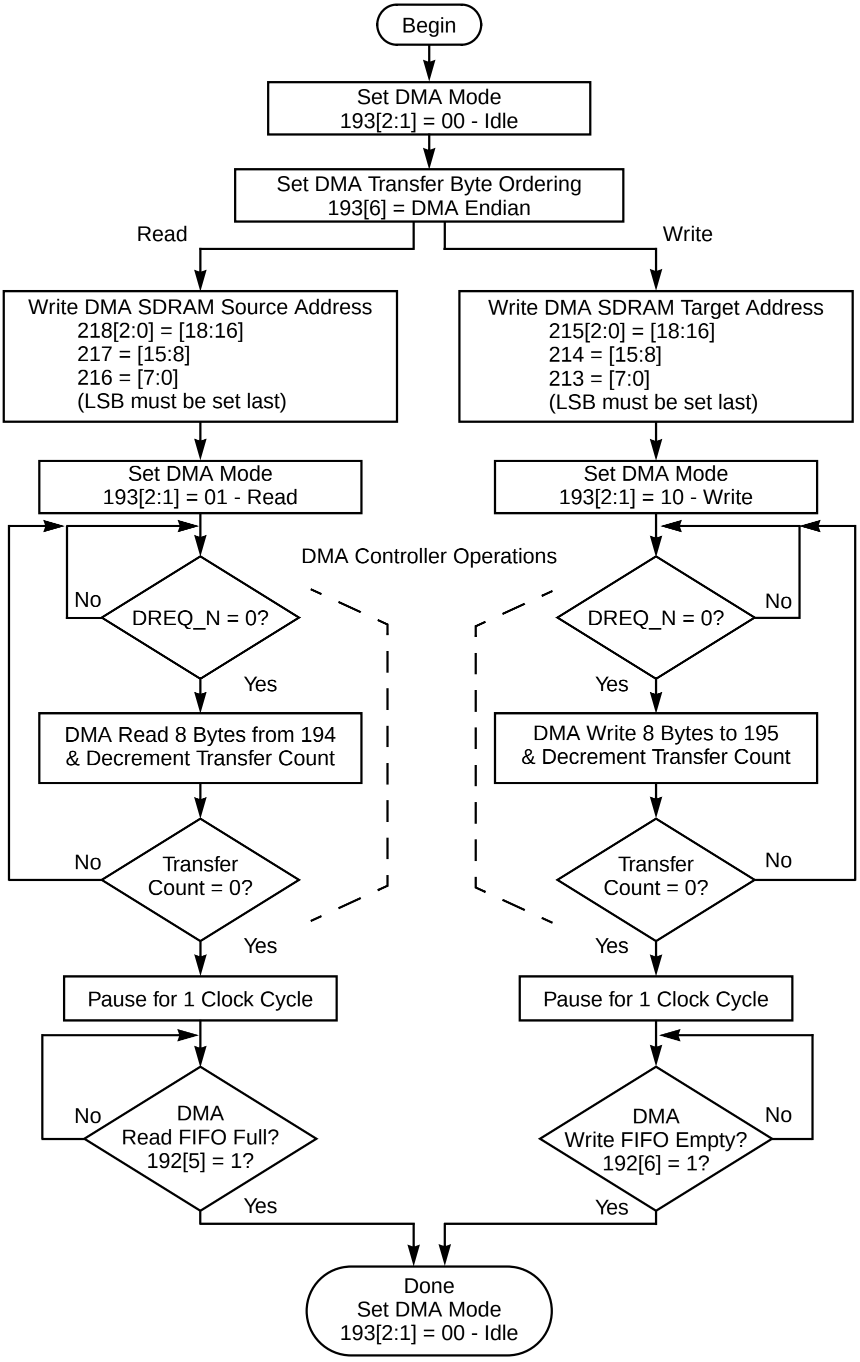
The DMA SDRAM write operation is very similar to the DMA SDRAM read operation as shown in Figure 5.9. The host sets the DMA Mode to Idle, sets the endian mode if necessary, writes the SDRAM target address into the DMA SDRAM Target Address registers, and sets the DMA Mode to Write. This causes the L64020 to assert the DREQn signal.

The host's DMA controller can then start writing bytes into the DMA SDRAM Write Data register. Each set of eight bytes are loaded into the DMA WrFIFO in the proper endian order. After the second 8-byte word is in the WrFIFO, the L64020 SDRAM controller starts writing the words to SDRAM as 16-bit words starting at the target address. The SDRAM controller automatically increments the target address for each new 16-bit word.

Again, the DMA controller is responsible for maintaining the transfer count. It continues to write bytes in as long as there is more than one 8-byte space left in the WrFIFO until the transfer count reaches zero. The host must then wait until the DMA Write FIFO Empty bit is set and then return the DMA Mode to Idle.

Caution: The L64020 only writes data out of the DMA WrFIFO when a complete 8-byte (64-bit) word is available. If the DMA controller attempts to write less than eight bytes of data to SDRAM or less than eight bytes in the last word, the data is not transferred to SDRAM and is lost when the FIFO pointers are next reset.

Figure 5.9 DMA SDRAM Read/Write Flowchart



5.4.2.3 DMA Bandwidth

During DMA, the L64020 can support a bandwidth of a sustained rate of approximately 2.5 Mbytes/sec. During the transfer of data, the rate can increase for short periods of time.

5.4.3 SDRAM Block Move

The DRAM block move, flowcharted in Figure 5.10, allows the host to specify a block of data to be copied from one SDRAM location to another SDRAM location.

Important: Some care should be taken when executing a SDRAM block move. The SDRAM block move should not be performed in parallel with any other SDRAM transfer; host read, host write, DMA read, or DMA write. All SDRAM transfers should be completed before a block move is started. The block move should be completed before any other SDRAM transfer is attempted.

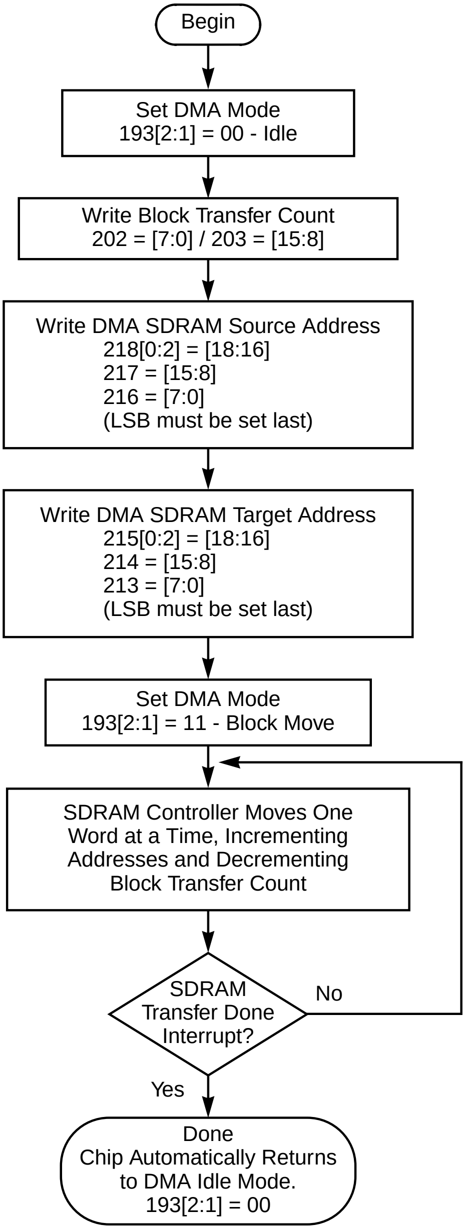
Block moves cannot be executed on data blocks smaller than one 64-bit SDRAM word.

It is the host's responsibility to ensure that the transfer count agrees with the difference between the start and end addresses.

To perform a block move, the host first sets the DMA Mode to Idle. Then it writes the number of contiguous 64-bit words to be moved into the Block Transfer Count registers. The host writes the move from address into the DMA SDRAM Source Address registers and the move to address into the DMA SDRAM Target Address registers. To start the move, the host sets the DMA Mode to Block Move.

The L64020 SDRAM controller performs the block move and sets the SDRAM Transfer Done Interrupt bit at the completion. If the SDRAM Transfer Done Interrupt is not masked by the host, the L64020 also asserts the INTRn signal to the host to notify it of the move completion. The L64020 automatically sets the DMA Mode to Idle at the move completion.

Figure 5.10 Block Move Flowchart


Begin
Set DMA Mode
193[2:1] = 00 - Idle
Write Block Transfer Count
202 = [7:0] / 203 = [15:8]
Write DMA SDRAM Source Address
218[0:2] = [18:16]
217 = [15:8]
216 = [7:0]
(LSB must be set last)
Write DMA SDRAM Target Address
215[0:2] = [18:16]
214 = [15:8]
213 = [7:0]
(LSB must be set last)
Set DMA Mode
193[2:1] = 11 - Block Move
SDRAM Controller Moves One Word at a Time, Incrementing Addresses and Decrementing Block Transfer Count
SDRAM Transfer Done Interrupt?
No
Yes
Done
Chip Automatically Returns to DMA Idle Mode.
193[2:1] = 00

Chapter 6
Channel Interface

This chapter describes the processing of the system stream through the Channel Interface. It describes how the preparser operates on the input stream, demultiplexes the various components, and writes them to the appropriate buffers in SDRAM. Various methods of handling and recovering from input stream errors are also discussed.

This chapter consists of the following sections:

6.1 Overview

The L64020 can process the following types of input streams:

1. Audio/Video PES packets for one program produced by transport decoder devices such as the L64008.
2. MPEG-1 System or MPEG-2 Program streams. This also includes DVD format streams that contains video, audio, SPU, and navigation (DSI/PCI) packs.
3. Audio/Video Elementary Streams (ES).

The host writes a 2-bit, Stream Select code into Register 7 to configure the L64020 for the input stream type.

MPEG system syntax governs the transfer of data from the encoder to the decoder. A system stream typically consists of a number of elementary streams (for example, video and audio streams) that are combined (multiplexed) to form a program stream. A program is defined as a set of elementary streams that share the same system clock reference and therefore can be decoded synchronously. In MPEG-1, there are only two levels of hierarchy in the system syntax. In MPEG-2, there are four levels of hierarchy in the system syntax. The following table shows the system streams for MPEG-1 and MPEG-2 system syntax.

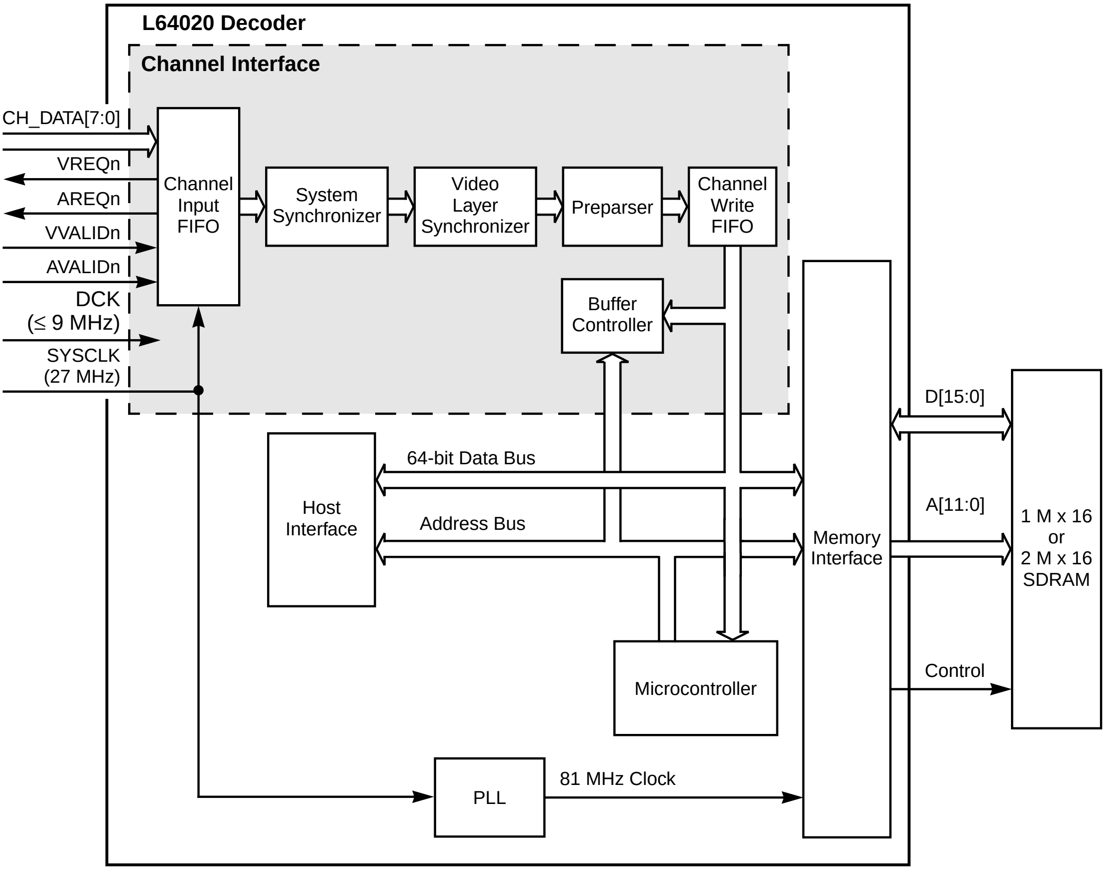
Table 6.1 Levels of Hierarchy in MPEG-1 and MPEG-2 System Syntax

MPEG-1 Streams	MPEG-2 Streams
	Transport Stream
System Stream	Program Stream
	Packetized Elementary Stream (PES)
Elementary Stream	Elementary Stream

MPEG-2 introduced the Packetized Elementary Stream (PES) to allow multiple streams and multiple programs to be combined in a single stream. An MPEG-2 system may transmit either a program stream that contains PES packets for a single program, or a transport stream that contains PES packets for multiple, possibly unrelated, programs. An MPEG-2 system decoder therefore must be able to accept PES data from a transport stream or from a program stream. The crucial difference between these two is that a program stream contains variable-length PES packets, but a transport multiplexer reforms input PES packets into fixed, 188-byte packets (184 payload bytes and 4 header bytes).

The Channel Interface, shown in Figure 6.1, includes an Input FIFO, System Synchronizer, Video Layer Synchronizer, Preparser, Channel Write FIFO, and a Buffer Controller.

Figure 6.1 Channel Interface Block Diagram



6.2 Interface Signals Operation

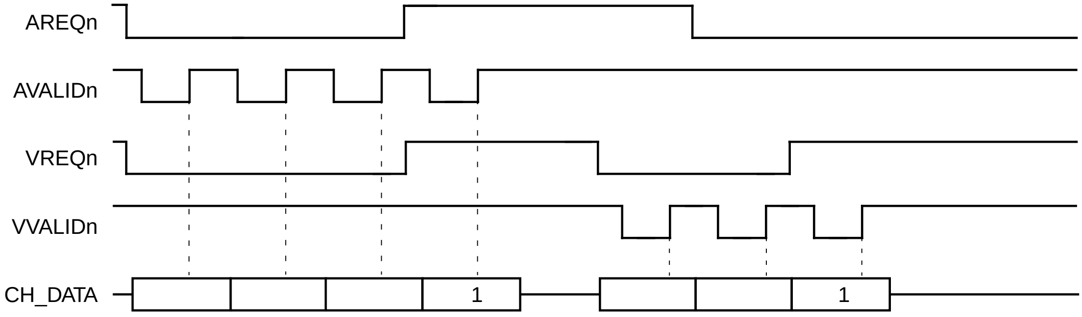
The L64020's Channel Interface can be connected to a variety of devices. The interface is capable of accepting one byte of data in 3Tc time (Tc = 1/27 MHz = 37 ns). This provides a transfer rate of nine Mbytes/s. The interface can be configured for asynchronous or synchronous mode with the Channel Request Mode bit in Register 5 (page 4-11). In asynchronous mode, the DCK input pin is tied to VSS and channel data bytes are paced into the L64020 with the REQ and VALID signals. In synchronous mode, the DCK input from the connecting device is used by the L64020 to gate the REQ signals and is gated by the VALID signals. The gated DCK strobes the data bytes in.

6.2.1 Asynchronous Mode

The timing for this mode is shown in Figure 6.2. The decoder asserts the AREQn or VREQn signal when it is ready for more audio or video data. Both the AREQn and VREQn requests are used for elementary streams and A/V PES streams from a transport decoder. Only the AREQn is used for program stream inputs.

The connecting device places the requested data on the CH_DATA[7:0] bus and asserts and deasserts its AVALIDn or VVALIDn output in response. Again, both VALIDn inputs are used for elementary streams and A/V PES streams; only the AVALIDn input is used for program stream inputs. A byte of data is strobed into the input FIFO of the L64020 on each rising edge of the VALIDn signal while its corresponding REQn signal is low. One extra byte can be strobed in after the REQn signal is deasserted. Note that both the AREQn and VREQn signals can be asserted at the same time. In that case, it is up to the connecting device to decide whether to strobe audio or video in.

Figure 6.2 Asynchronous Channel Interface Timing



1. One extra byte okay for asynchronous AREQn/VREQn.

The constraints of this mode are:

1. AVALIDn and VVALIDn should never be low at the same time. The valid byte on CH_DATA[7:0] is either audio or video.
2. Any VALIDn rising edge to VALIDn rising edge must be separated by ≥ 3Tc. This allows the synchronizing logic of the L64020 time to resynchronize, and the input channel FIFO time to deassert the AREQn/VREQn signals and prevent overflow conditions.

3. The system must respect the function of the AREQn/VREQn signals. The timing restriction above will allow enough space within the input channel FIFO to allow an external synchronizer on the AREQn/VREQn signals. This allows writing data beyond AREQn/VREQn rising edge by 1 byte.
4. The DCK pin of the decoder must be tied to V_{SS}, and the Invert Channel Clock bit in Register 5 (page 4-11) must be cleared.

6.2.2 Synchronous VALIDn Inputs

When the DCK input is connected, the L64020 uses it to internally synchronize the input VALIDn signals before they strobe data in. This mode is recommended for connecting devices that do not have clean AVALIDn/VVALIDn signals.

The synchronizing circuits in the L64020 are shown in Figure 6.3. When DCK is not connected in from the upstream device, AVALIDn and VVALIDn strobe audio and video bytes in from the CH_DATA[7:0] bus on their rising edges. When DCK is supplied, it is gated through when either VALIDn signal is asserted. The gated rising edges of DCK then strobe data in. When the Invert Channel Clock bit is set, DCK is inverted through the exclusive OR before being gated by the VALIDn signals. The timing for synchronous valid signals is shown in Figure 6.4.

Figure 6.3 VALIDn Input Synchronization Circuits

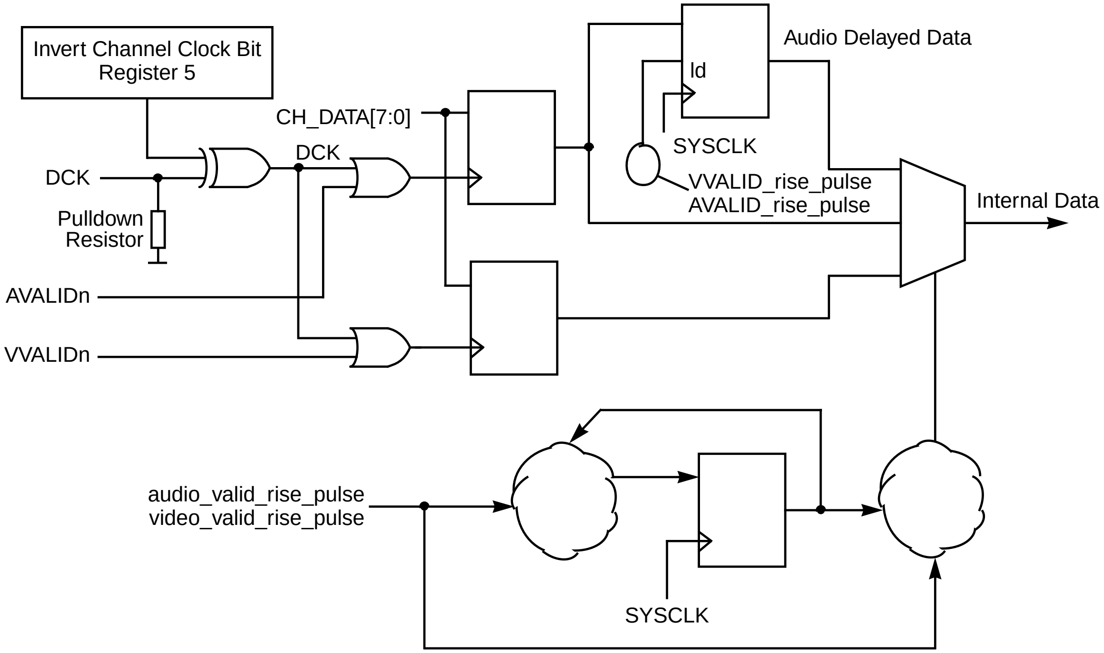
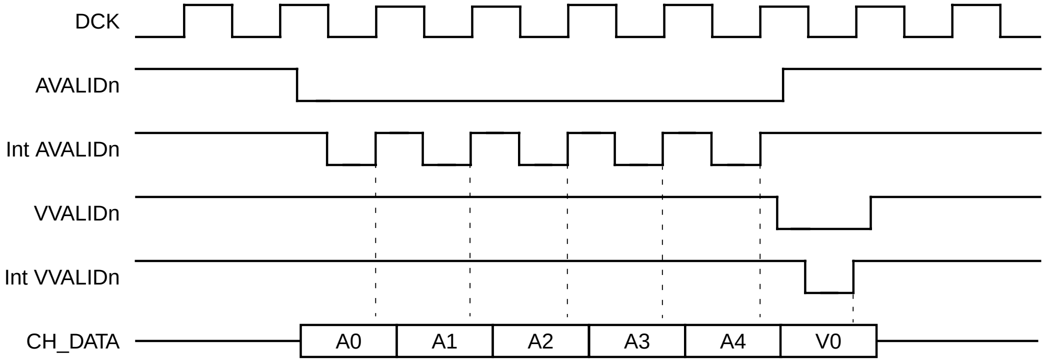


Figure 6.4 Synchronous Valid Signals Timing



The constraints on synchronous valid signal mode are:

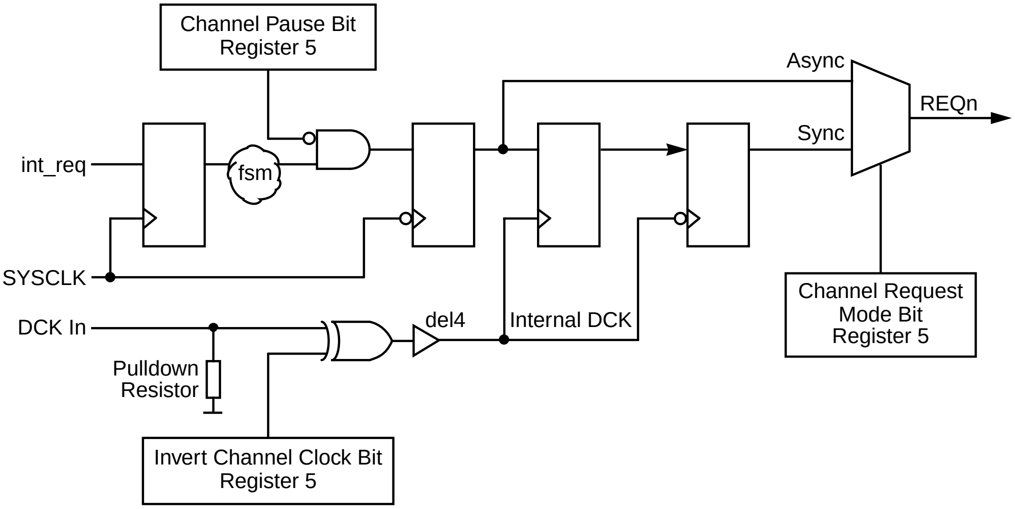
1. AVALIDn and VVALIDn should never be low at the same time. The valid byte on CH_DATA[7:0] is either audio or video.

2. It is recommended that the AVALIDn and VVALIDn outputs are registered on the rising edge of DCK (if DCK is in normal mode, i.e., not inverted mode.)
3. The minimum period of DCK must be 3Tc(Tc = 1/27 MHz = 37 ns). This allows the internal synchronizing logic time to resynchronize, and allows the input channel FIFO time to assert and deassert the AREQn/VREQn signals and prevent overflow conditions.
4. The system must respect the function of the AREQn/VREQn signals. The timing restriction above will allow enough space within the input channel FIFO to allow an external synchronizer on the AREQn/ VREQn signals. This allows writing data beyond AREQn/VREQn rising edge by 1 byte.

6.2.3 Synchronous REQn Outputs

If the upstream device requires that the DREQn outputs of the L64020 be synchronized to DCK, setting the Channel Request Mode bit configures the L64020 appropriately. The REQn circuits for one request signal in the decoder are shown in Figure 6.5.

Figure 6.5 L64020 REQn Circuits



The internal request (int_req) signal is generated by the channel input FIFO controller on the L64020 and indicates available room in the on-chip buffers and the SDRAM channel buffers. The internal request signal is always registered by the L64020 SYSCLK. Normally, REQn signals are asserted even when the channel is stopped to prevent upstream device overflow. The host can set the Channel Pause bit to block the int_req. If not, the SYSCLK-registered int_req is routed through the output multiplexer to the appropriate REQn pin.

When the Channel Request Mode bit is set by the host, the Sync input to the multiplexer is selected. As was shown in Figure 6.3, the DCK input can be inverted or not through the exclusive OR. In either case, the internal request is registered by a rising and falling internal DCK to avoid metastability. The external AREQn/VREQn signals always change at the falling edge of the Internal DCK. Refer to Chapter 12 for exact timing.

6.2.4 Channel Bypass Mode

When the Channel Bypass Enable bit in Register 5 (page 4-11) is set, the L64020 reads audio and video data in from the host through the A/V Channel Bypass Data registers (page 4-18). In this mode, the parallel data channel input port and the AVALIDn and VVALIDn input signals are ignored. The AREQn and VREQn output signals still function normally and can be used by the host as DMA control handshake signals. When either is asserted, the internal microcontroller watches the corresponding Channel Bypass Data register for activity. The Channel Bypass Data registers can accept one additional byte after the AREQn or VREQn signals are deasserted.

6.2.5 Channel Pause

When the Channel Pause bit in Register 5 (page 4-11) is set, the REQn outputs of the L64020 are held deasserted. This does not stop the channel processing inside the L64020. This function is intended to be used to force a pause of either transport decoder devices or channel decoder devices that respect the AREQn and VREQn signals. While paused, the host can change stream IDs at known boundaries, but it cannot change any of the address ranges or the setup of the Preparser read and write pointers.

6.3 Preparser

For A/V PES and program streams, the Preparser strips the packets of headers and writes the headers and packet data payloads into separate buffer areas in the off-chip SDRAM memory. The host writes the start and end addresses of each of the buffer areas into registers. The internal microcontroller transfers these addresses to the Buffer Controller. The Buffer Controller maintains current read and write pointers for each buffer area defined. When the Preparser strips an item out of the bitstream, the microcontroller gets the current write pointer to the buffer area for that item and writes the item into the buffer. The microcontroller also writes the LSB of the item's address pointer to the appropriate register. If the host reads the LSB, the Buffer Controller writes the next pointer address byte and the MSB to the register. The Buffer Controller and the host registers used to program these buffer areas are explained in detail in Section 6.4, "Channel Buffer Controller," page 6-40.

6.3.1 Host Selection of Streams and Headers

The host has control over which streams are preparsed and if headers are stored. The register bits that define the preparse operation are discussed here. It is assumed in the Preparser descriptions that follow that the particular stream and header has been selected or enabled.

The host selects the video stream to be decoded by setting the Video Stream Select Enable bits in Register 145 and entering a 4-bit Video Stream ID in the same register (page 4-42). The Video Stream Select Enable codes are listed in Table 6.2.

Table 6.2 Video Stream Select Enable Bits

Video Stream Select Enable	Description
0b00	Discard all video packets
0b01	MPEG ID selected
0b10	All Video Stream IDs stored
0b11	Discard all video packets

Host Registers 143[4:0] store the audio stream ID. This is used in conjunction with the audio stream select enable (Register 143[7:5]) to select which audio stream IDs are selected for decoding. Table 6.3 illustrates the options available in selecting audio streams.

Table 6.3 Audio Stream Select Enable Bits

Audio Stream Select Enable	Description
0b000	Always discard (off). No audio data is put in channel.
0b001	MPEG ID selected[1]
0b010	Linear PCM Stream ID selected[1]
0b011	Dolby Digital Stream ID selected[1]
0b100	All MPEG Audio IDs[2]
0b101	MPEG audio multichannel with extension[3]
0b110–0b111	Always discard (off)

1. In mode 0b001 (MPEG ID selected), the MPEG audio stream is assumed to be MPEG-1 audio or MPEG-2 audio without extensions. In modes 0b001 through 0b011, the audio stream ID is programmed in bits [4:0] of Register 143.
2. Mode 0b100 is used when only one audio stream ID is in the bitstream. The decoder ignores the contents of bits [4:0] in Register 143.
3. In mode 0b101 (MPEG Audio multichannel with extension), the main audio stream ID is programmed in Register 143[4:0], while the extension stream ID is programmed in Register 124[4:0].

The Pack Header Enable bits in Register 147 (page 4-44) determine whether pack headers are parsed and written to the Audio PES Header/System Channel Buffer. The available selections are listed in

Table 6.4. Note that the 0b01 code lets the host selectively parse the headers.

Table 6.4 Pack Header Enable Bits

Pack Header Enable	Description
0b00	Write no headers.
0b01	Write one header. This mode is reset internally back to mode 0b00 above on successful completion of the write.
0b10	Write all headers.
0b11	Write no headers.

The System Header Enable bits in Register 147 (page 4-44) determine whether system headers are parsed and written to the Audio PES Header/System Channel Buffer. The available selections are listed in Table 6.5. Note that the 0b01 code lets the host selectively parse the headers.

Table 6.5 System Header Enable Bits

System Header Enable	Description
0b00	Write no headers.
0b01	Write one header. This mode is reset internally back to mode 0b00 above on successful completion of the write.
0b10	Write all headers.
0b11	Write no headers.

The Video PES Header Enable bits in Register 145 (page 4-42) determine whether video headers are parsed and written to the Audio PES Header/System Channel Buffer or the Video PES Header/SPU

Channel Buffer. The available selections are listed in Table 6.6. Note that the 0b01 code lets the host selectively parse the headers.

Table 6.6 Video PES Header Enable Bits

Video PES Enable	Description
0b00	Write no video PES headers.
0b01	Write one header if PTS or DTS is present. This mode is reset internally to mode 0b00 above after successful completion of the write.
0b10	Write all headers.
0b11	Write all video PES headers if PTS or DTS is present.

The Audio PES Header Enable bits in Register 147 (page 4-44) determine if and when audio headers are parsed and written to the Audio PES Header/System Channel Buffer. The available selections are listed in Table 6.7. Note that the 0b01 code lets the host selectively parse the headers.

Table 6.7 Audio PES Header Enable Bits

Audio PES Header Enable	Description
0b00	Write no headers.
0b01	Write one header if PTS or DTS is present. This mode is reset internally back to mode 0b00 above on successful completion of the write.
0b10	Write all audio PES headers.
0b11	Write all audio PES headers if PTS or DTS is present.

When a DVD stream is selected, the host can elect to disregard Subpicture Unit (SPU) streams (Register 146, bit 5 cleared, page 4-43) or parse and store an SPU stream with the substream ID set in bits 4 through 0 of Register 146.

Also in DVD streams, bit 6 in Register 146 (page 4-43) can be set by the host to parse and store Data Dump streams or cleared to ignore them.

Data dump is optional and refers to private stream 1 and a substream ID = 0b0100.1000.

For DVD streams, the Navi Pack PES Header Enable bits in Register 147 (page 4-44) determine if and when DSI and PSI headers and packets are written to the Navi Pack Channel Buffer. The available selections are listed in Table 6.8.

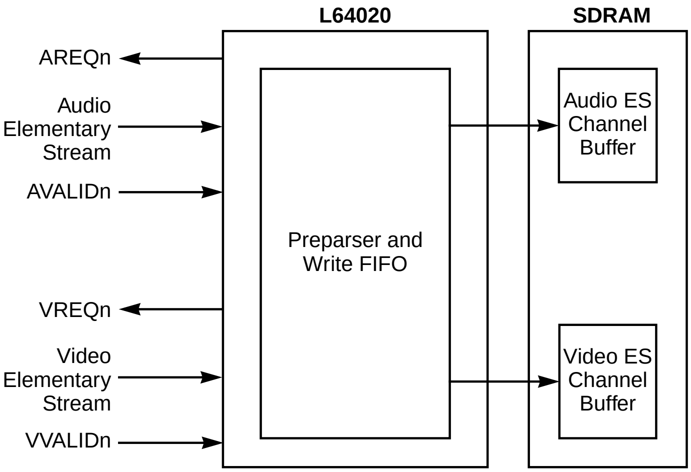
Table 6.8 Navi Pack PES Header Enable Bits

Navi Pack PES Header Enable	Description
0b00	Write no DSI or PCI headers.
0b01	Reserved
0b10	Reserved
0b11	Write PCI and DSI headers and packets.

6.3.2 Elementary Streams

In the Elementary Stream mode, the video and audio input streams are not preparsed but written as is (unmodified) into the buffer areas earmarked for them in the external SDRAM as shown in Figure 6.6.

Figure 6.6 Elementary Stream Buffering



The start and end addresses of each of the buffers are programmed by the host in the registers listed in Table 6.9.

Table 6.9 Buffer Start and End Address Registers for ES Mode

Addresses	Registers	Page Ref.
Video ES Channel Buffer Start Address	72 and 73	4-24
Video ES Channel Buffer End Address	74 and 75	4-24
Audio ES Channel Buffer Start Address	76 and 77	4-25
Audio ES Channel Buffer End Address	78 and 79	4-25

These registers hold the upper 14 bits of the buffer addresses. The SDRAM Controller programs the address bits so that the addresses are on 256-byte boundaries. The host can write to these registers only when the channel is stopped.

The buffers are maintained as circular FIFOs. The current read and write pointers for each of the buffers are written to registers (listed in Table 6.10) and available to the host. Actually, only the LSB registers are continually updated. When the host reads the LSB, the next byte and the MSB registers are then updated. Also, the number of items in each channel is provided in host registers:

Table 6.10 Buffer Write and Read Pointer Registers in ES Mode

Pointer	Registers	Page Ref.
Video ES Channel Buffer Write Address	96–98	4-30
Audio ES Channel Buffer Write Address	99–101	4-30
Video ES Channel Buffer Read Address	108–110	4-32
Audio ES Channel Buffer Read Address	111–113	4-33
S/P DIF Channel Buffer Read Address	120–122	4-35

The read and write pointer registers each contain 20 bits. The most significant bit is set when the pointer wraps around to the beginning of the buffer and cleared when the host next reads the register. The next 19 bits are the actual address on 64-bit boundaries since SDRAM operations are always in bursts of four 16-bit words.

The Audio Decoder and the S/P DIF (IEC958) Formatter both read from the Audio ES channel buffer so a read pointer is maintained for both; the Audio ES Channel Buffer Read Address and the S/P DIF Channel Buffer Read Address.

The number of items (64-bit words) remaining to be read in each of these buffers is written to the registers listed in Table 6.11 and available to the host. Again, only the LSB registers are continually updated. The Next and MSB registers are updated when the host reads the LSB.

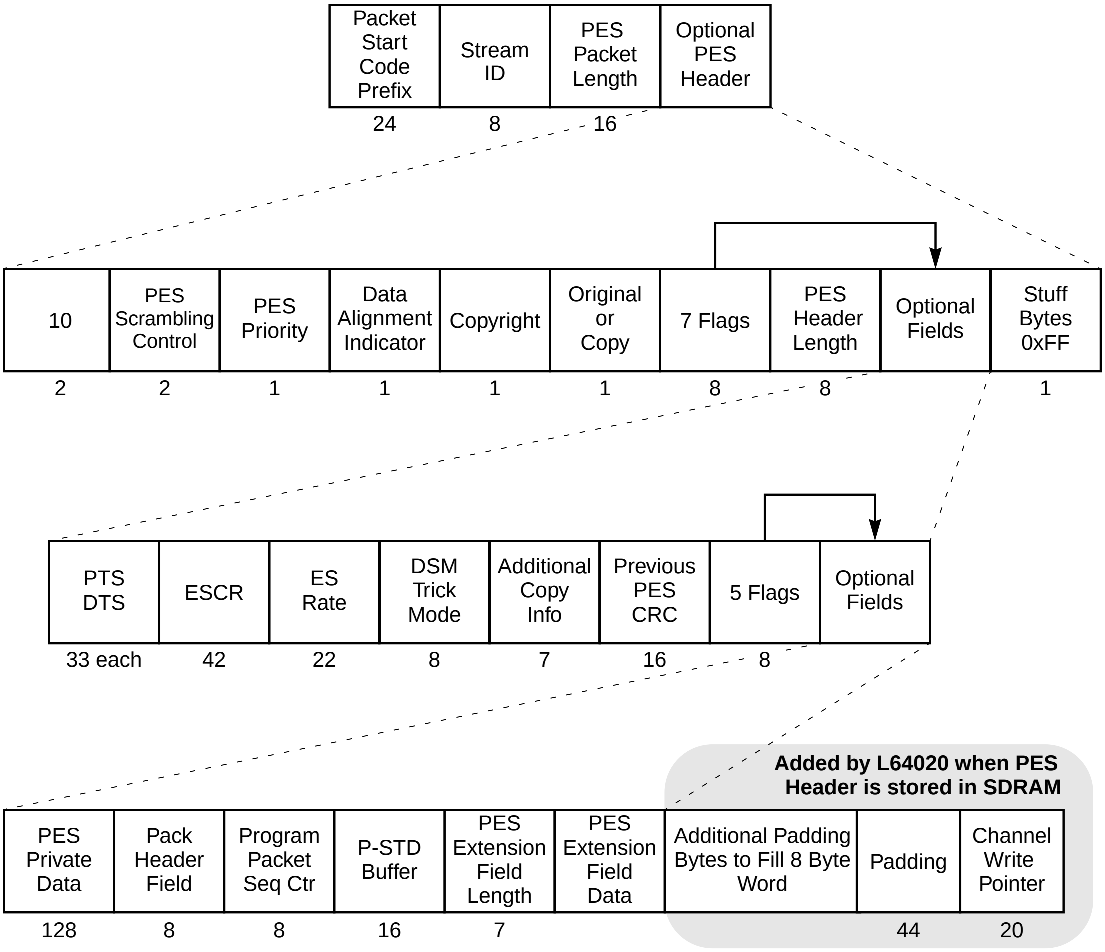
Table 6.11 Number of Items in Buffers in ES Mode

Buffer No. of Items	Registers	Page Ref.
Video ES Channel Buffer Numitems	134–136	4-39
Audio ES Channel Buffer Numitems	137–139	4-40
S/P DIF Channel Buffer Numitems	140–142	4-40

6.3.3 PES Packet Structure

Since the Preparser does strip headers out of packets in other modes, it is useful to look at a PES packet before discussing those modes. Figure 6.7 shows the packet structure. The Packet Start Code is a string of 23 zeros followed by a logic one. The next byte, the Stream ID, identifies the type of data that is in the packet. The Packet Length field specifies the number of bytes following to the end of the packet. The Preparser stores the remainder of the packet header, the PES header, in a buffer in SDRAM with padding to make it a multiple of 8 bytes. The write pointer to the beginning of that buffer area is padded out to 8 bytes and stored in the same buffer following the PES header.

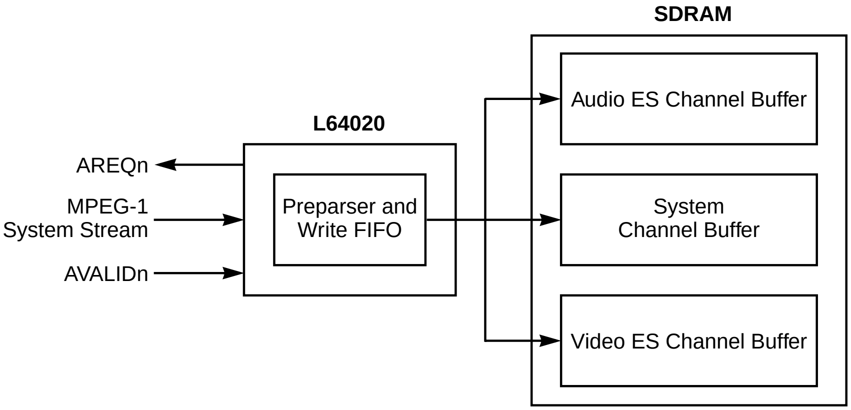
Figure 6.7 PES Packet Structure



6.3.4 Preparsing an MPEG-1 System Stream

In addition to audio and video channel buffers, a System Channel Buffer is allocated in SDRAM for MPEG-1 streams. This buffer is used to hold headers. When the decoder encounters any System Start Code, it synchronizes to that start code, if it is not already in sync. The Preparser then moves the system header into the System Channel Buffer and looks for the beginning of the first packet.

Figure 6.8 Preparsing an MPEG-1 System Stream



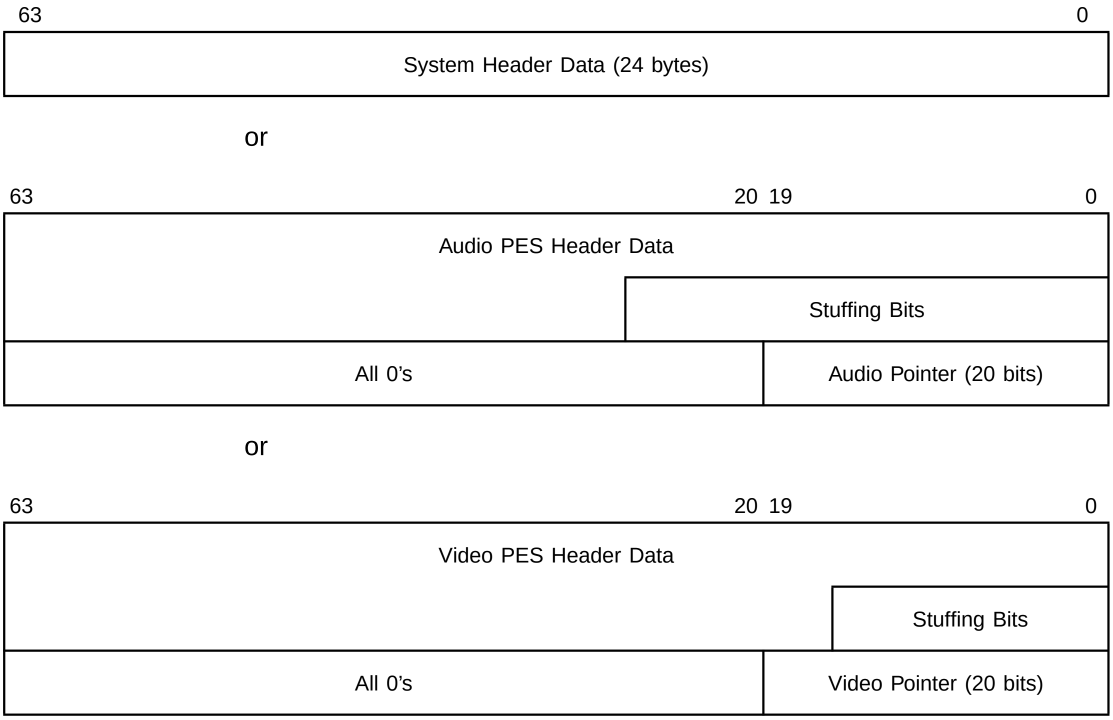
The data flow is shown in Figure 6.8. Since audio and video packets are multiplexed in the stream, only the AREQn and AVALIDn are used. When the Preparser recognizes a Packet Start Code, it checks to see whether the packet contains audio or video data, and whether the Stream Select field matches the Stream Select code written into Register 7 (page 4-12) by the host. If the Preparser does not find a match, it discards the packet. For accepted packets, the Preparser uses the Packet Length field to determine where the packet ends. This is necessary to avoid mistakenly parsing the possible emulation of start codes in audio packet data.

If the stream ID indicates an audio stream, the Preparser skips any packet stuffing bytes and moves the remainder of the packet header into the System PES Channel Buffer. The chip sets the Audio PES Data Ready Interrupt (Register 2, page 4-6) and asserts the INTRn output signal, if the interrupt is not masked, to indicate to the host that the packet header is in the System Channel Buffer. The Preparser then samples the current write pointer for the Audio ES Channel Buffer and moves its value into the System Channel Buffer after the packet header. The host can subsequently use this value for system synchronization. The Preparser then moves the packet payload into the Audio ES Channel Buffer. The Preparser uses the Packet Length field in the packet header to determine the end of the audio data payload.

If the stream ID is a video stream, the Preparser skips any packet stuffing bytes and moves the remainder of the packet header into the System Channel Buffer. INTRn is asserted if not masked and the Video PES Data Ready Interrupt (Register 2, page 4-6) is set. The Preparser then samples the current write pointer for the Video ES Channel Buffer and stores its value in the System Channel Buffer after the packet header. The Preparser then moves the packet payload into the Video ES Channel Buffer. Note that the Preparser must be able to parse the packet header because there is no header length field.

Figure 6.9 shows the mapping of the header data and payload pointers in the System Channel Buffer. The header data is written into the buffer in 64-bit words (four 16-bit bursts). The 20-bit pointers are aligned to the MSB and preceded by 44 zero bits to round out the word.

Figure 6.9 System PES Channel Buffer Map for MPEG-1 Streams



The only error that the Preparser can detect is a mismatch between the packet length field and the next packet start code. If this occurs, the Preparser generates an interrupt and optionally clears the buffers. For a complete description of the MPEG-1 system stream syntax, the reader is referred to *ISO/IEC 11172*.

The registers for the Audio and Video ES Channel Buffers are those described for Elementary Stream Mode. Table 6.12 lists the registers associated with the System Channel Buffer.

> Note: These registers are also used for the Audio PES Header Channel Buffer when the input stream is an A/V PES stream from a transport demultiplexer.

The start and end addresses are the upper 16 bits for alignment on 256-bit boundaries. The host must read the LSB of the write pointer first to get the next bytes of the pointer updated. There is no read pointer for this buffer.

Table 6.12 SDRAM Addresses - Audio PES Header/System Channel Buffer

Addresses	Registers	Page Ref.
Audio PES Header/System Channel Buffer Start Address	88 and 89	4-28
Audio PES Header/System Channel Buffer End Address	90 and 91	4-28
Audio PES Header/System Channel Buffer Write Address	114–116	4-34

6.3.5 Preparsing a Program Stream without DVD

Preparsing an MPEG-1 or 2 program stream without DVD is very similar to the MPEG-1 system stream case shown in Figure 6.8. The differences are that the program stream is divided into packs and then packets, and the PES packet header contains a header length field. The Preparser reads this field to determine the number of header bytes to store in the Audio PES Header/System Channel Buffer. The pack headers are also mapped into the buffer in the same manner as for the system header in Figure 6.9. Storing a pack header causes the chip to assert INTRn, if not masked, and to set the Pack Data Ready Interrupt bit in Register 2 (page 4-6).

The structure of a PES packet is shown in Figure 6.7. A description of MPEG-2 program syntax can be found in *ISO/IEC 13818-1*.

6.3.6 DVD Stream Structure

This section presents an overview of the structure of a DVD stream as related to preparsing by the L64020. A DVD stream contains the layers shown in Figure 6.10. The stream is divided into separate streams for audio, video, subpicture data, and navigation data. Each pack contains one or more packets associated with a particular stream, i.e., there are audio packets, video packets, etc.

Figure 6.10 Overview of DVD Stream Layers

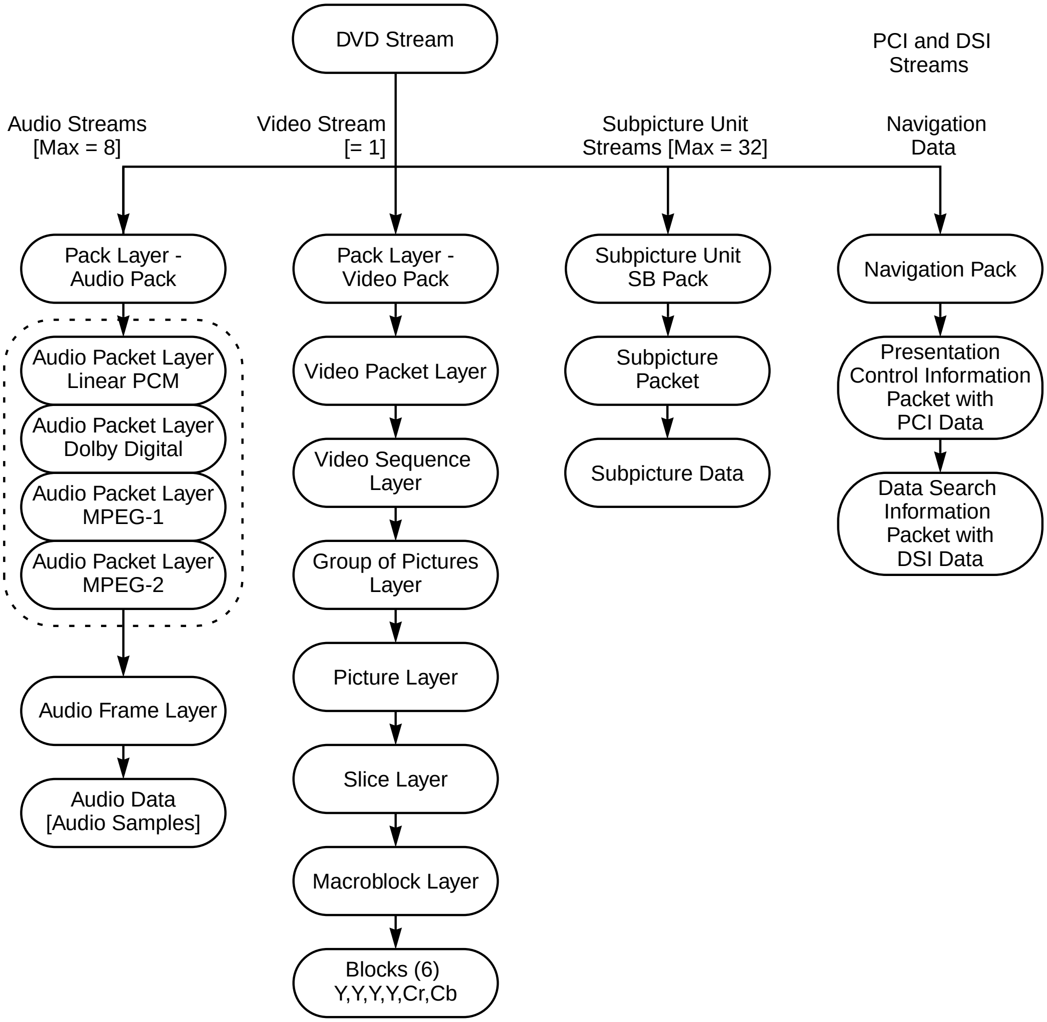


Figure 6.11 shows the general structure of a pack along with methods of padding in case the pack length is less than 2048 bytes.

Figure 6.11 General Structure of DVD Pack

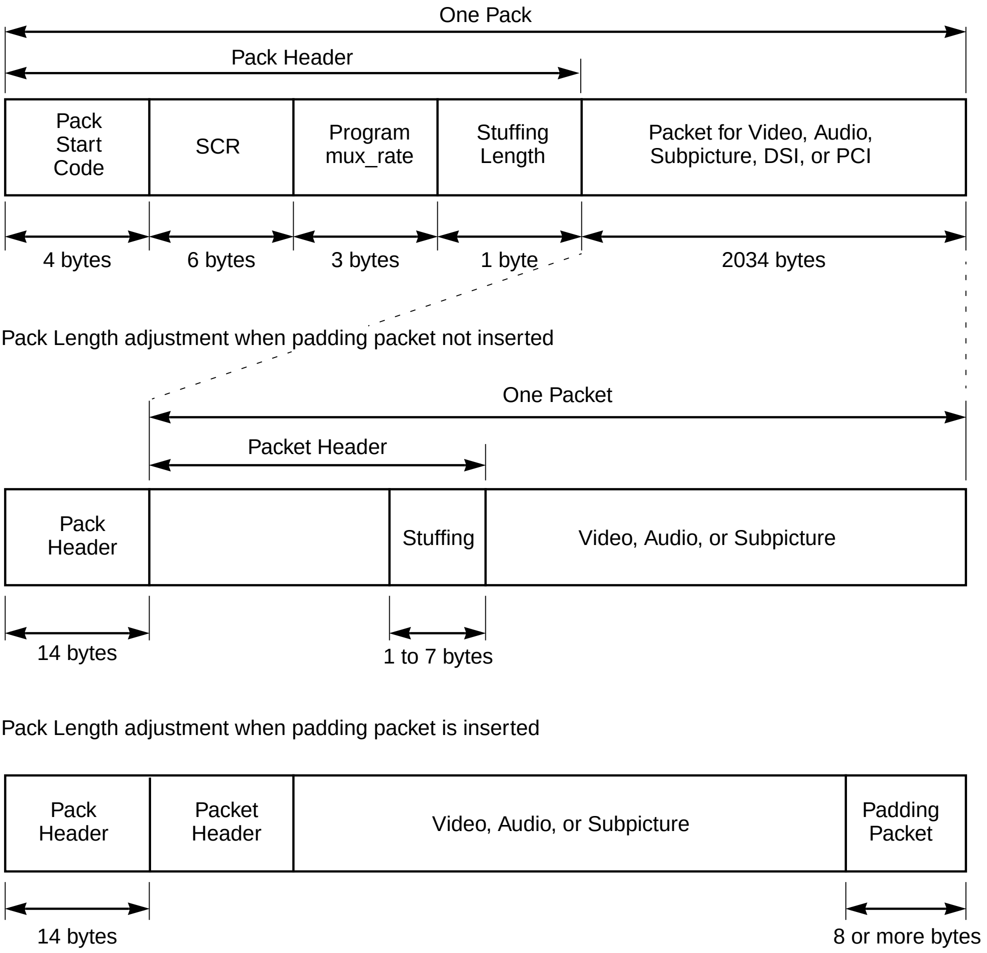


Figure 6.12 illustrates the structures of the navigation pack, video pack, and subpicture pack.

Figure 6.12 Navigation, Video, and Subpicture Pack Structures

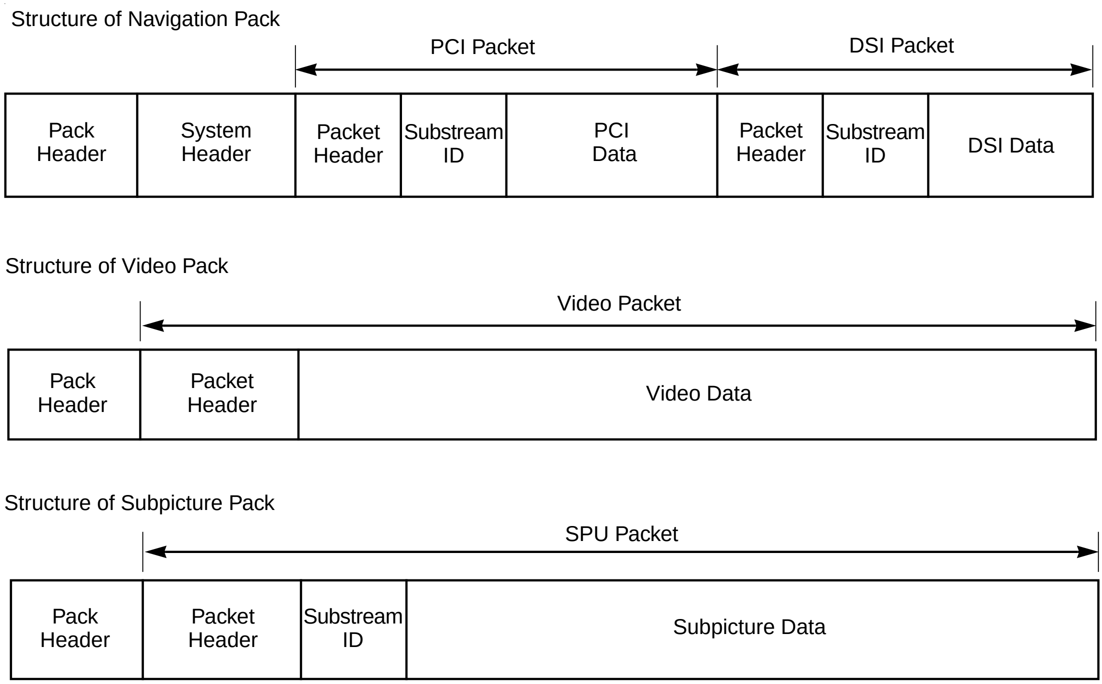
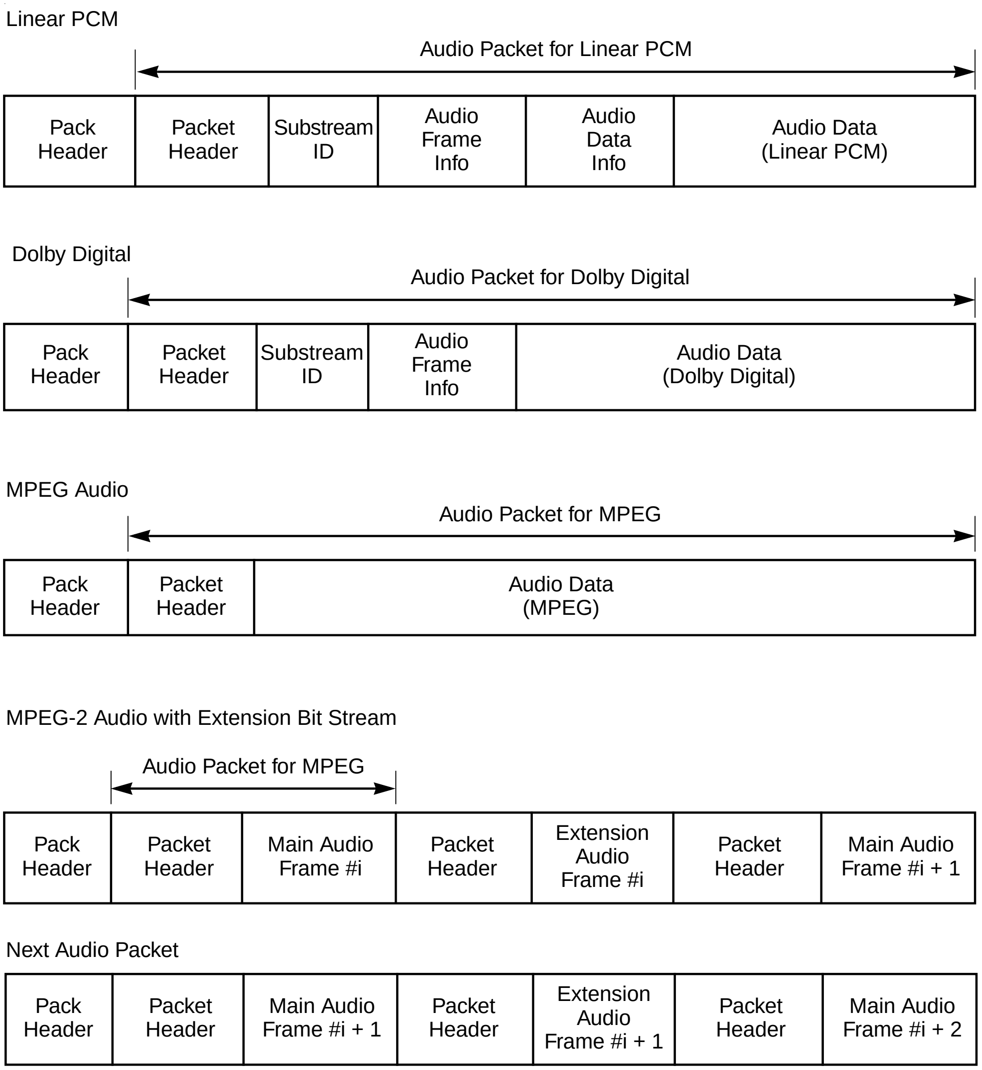


Figure 6.13 illustrates the structure of the Audio Pack.

Figure 6.13 Audio Pack Structures



6.3.7 Preparsing an MPEG-2 Program Stream with DVD

For DVD streams, the L64020 supports the types of packets listed in Table 6.13.

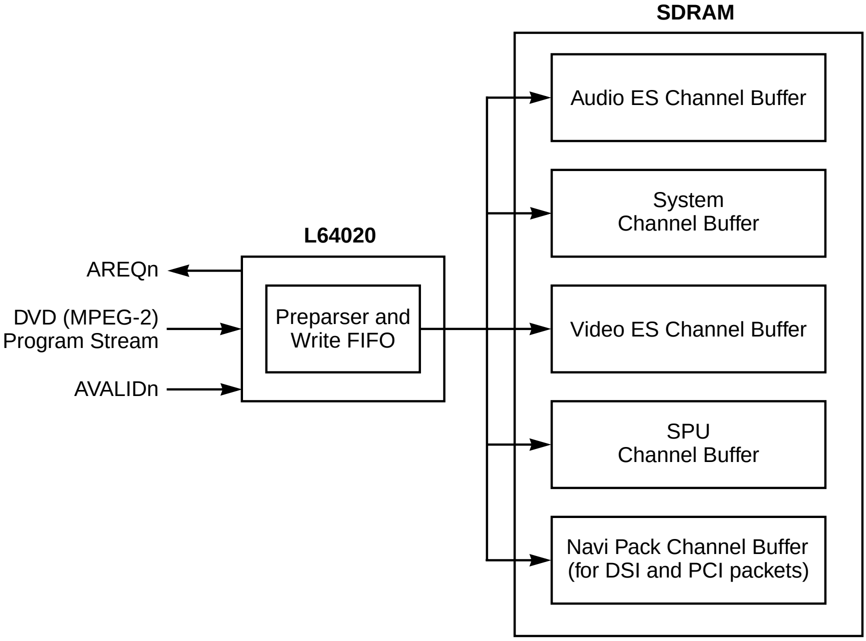
Table 6.13 DVD Packet Types with Stream and Substream IDs

Stream Type	Packet	Stream ID	Private Stream Type	Substream ID
Navigation	PCI	BFh	2	0b0000.0000
Navigation	DSI	BFh	2	0b0000.0001
Video	Video	E0-EFh	NA	NA
Audio - Linear PCM	Audio	BDh	1	0b1010.0xxx
Audio - Dolby Digital	Audio	BDh	1	0b1000.0xxx
Audio - MPEG	Audio	0b110x.0xxx	NA	NA
Subpicture Unit	Subpicture	BDh	1	0b001x.xxxx

The Preparser operation is shown in Figure 6.14. Two additional buffers are required; the SPU Channel Buffer for Subpicture Unit packets and the Navi Pack Channel Buffer for Data Search Information and Presentation Control Information (PCI) packets.

An optional Data Dump Channel Buffer can be used to parse incoming channel data that has the same syntax as a PES packet with private stream 1 and substream ID = 0b0100.1000. This particular substream ID is not present in the DVD specification. One application of the Data Dump Channel Buffer is to store initial lead-in data at the beginning of DVD playback. It is the host’s responsibility to ensure correct PES packet syntax and provide the correct substream ID as shown in Table 6.13. No error checking is performed by the L64020 Decoder for any streams processed to this buffer. This buffer can be turned off by clearing the Data Dump Stream Select Enable bit in Register 146 (page 4-43).

Figure 6.14 Preparsing an MPEG-2 DVD Stream



The start address, end address, read pointer, and write pointer registers for the Audio ES, Video ES, and System Channel Buffers are those listed for Elementary and MPEG-1 streams in Table 6.9 and Table 6.10. The registers associated with the remaining buffers are shown in Table 6.14.

Note: The Video PES Header/SPU registers are also used for the Video PES Header Channel Buffer when the input stream is an A/V PES stream from a transport demultiplexer.

Table 6.14 DVD Stream Buffer Registers

Addresses	Registers	Page Ref.
Video PES Header/SPU Channel Buffer Start Address	80 and 81	4-26
Video PES Header/SPU Channel Buffer End Address	82 and 83	4-26
Video PES Header/SPU Channel Buffer Write Address	102–104	4-31
Navi Pack Channel Buffer Start Address	92 and 93	4-29
Navi Pack Channel Buffer End Address	94 and 95	4-29
Navi Pack Channel Buffer Write Address	117–119	4-35
Data Dump Channel Buffer Start Address	84 and 85	4-27
Data Dump Channel Buffer End Address	86 and 87	4-27
Data Dump Channel Buffer Write Address	105–107	4-31

Figure 6.15 is the map of the System Channel Buffer for DVD streams. As for MPEG-1 and MPEG-2 without DVD, stuffing bits are added to the end of the headers to round out the last word to 64 bits if necessary. Address pointers are aligned to the MSB with leading zeros to complete the 64-bit words.

Figure 6.15 System Channel Buffer Map for DVD Streams

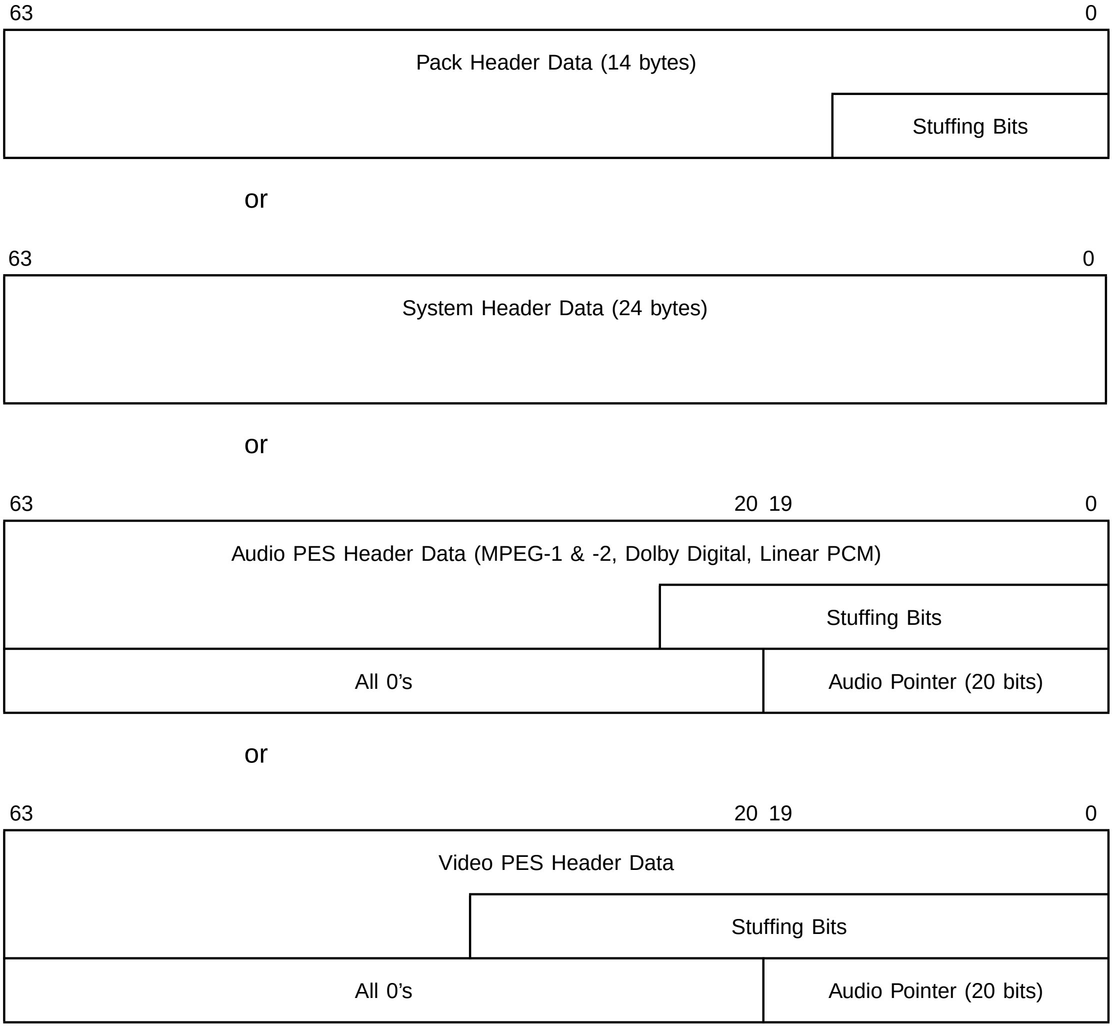
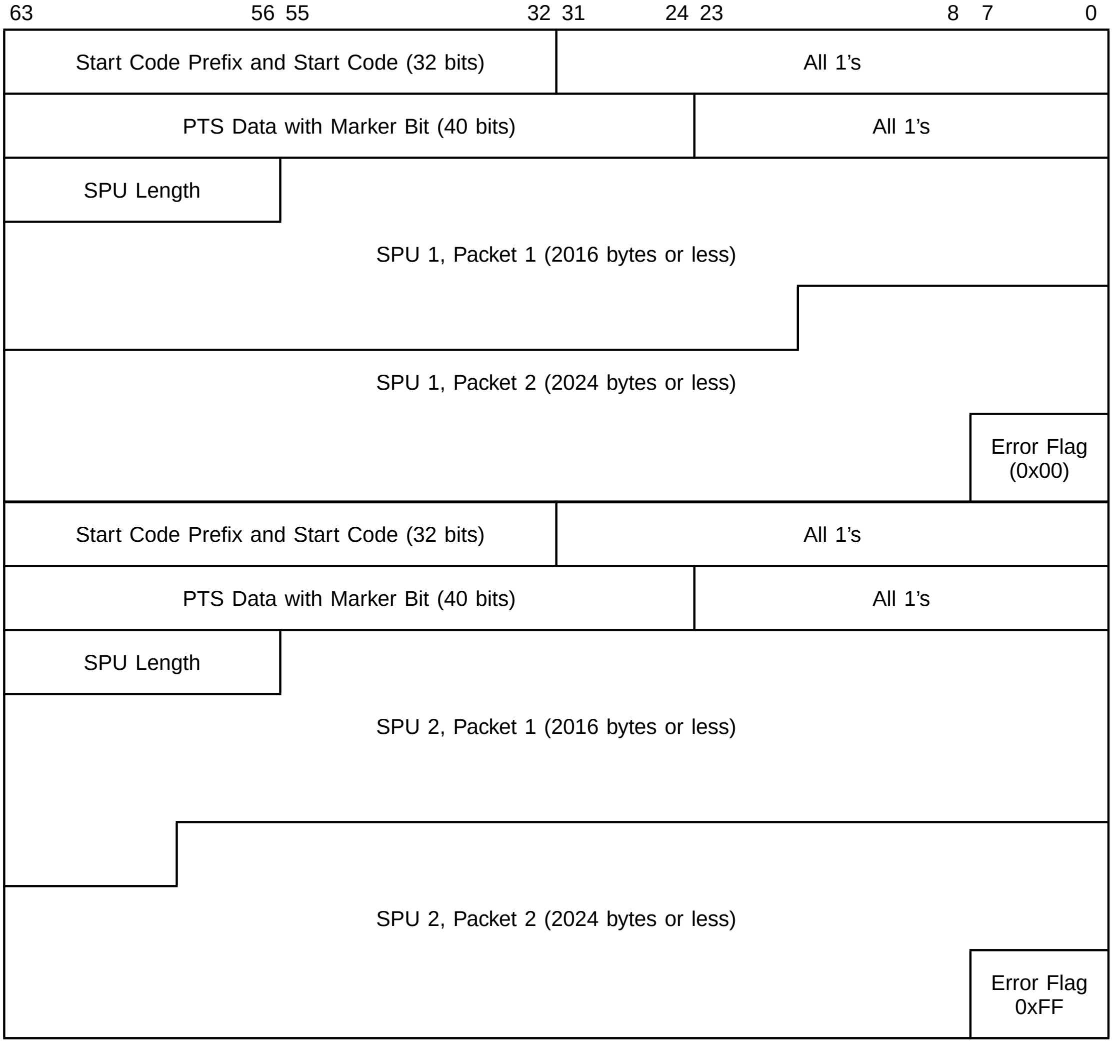


Figure 6.16 shows the map of the SPU Channel Buffer for a DVD stream.

Figure 6.16 SPU Channel Buffer Map for DVD Streams


63
56 55
32 31
24 23
8 7
0
Start Code Prefix and Start Code (32 bits)
All 1’s
PTS Data with Marker Bit (40 bits)
All 1’s
SPU Length
SPU 1, Packet 1 (2016 bytes or less)
SPU 1, Packet 2 (2024 bytes or less)
Error Flag (0x00)
Start Code Prefix and Start Code (32 bits)
All 1’s
PTS Data with Marker Bit (40 bits)
All 1’s
SPU Length
SPU 2, Packet 1 (2016 bytes or less)
SPU 2, Packet 2 (2024 bytes or less)
Error Flag 0xFF

The SPU channel buffer stores the PTS data of the SPU header and the SPU packet data. SPU PTS data is stored with the start code prefix and the start code. Both have their upper bytes stuffed with 1’s to form 64-bit words. If any error exists in an SPU, the error flag in the last packet is set to 0xFF. If there is no error, the error flag is left at 0x00. If there is an error between the SPU Length field and the actual SPU length, the

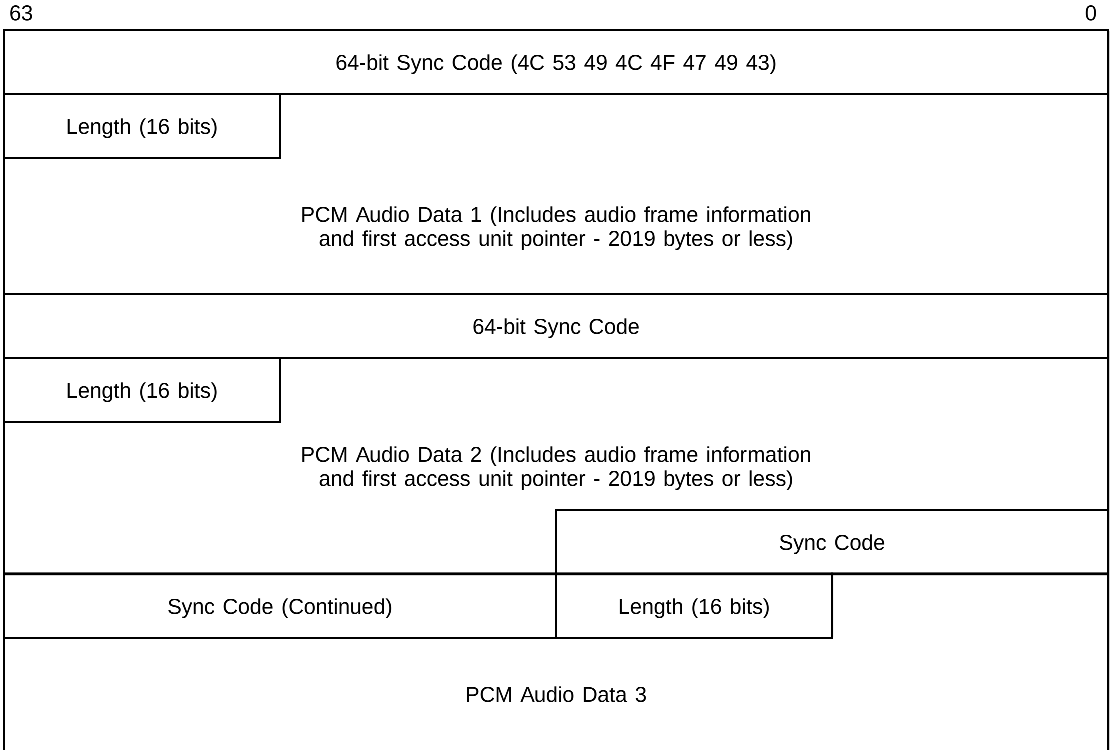
SPU Length field is set to 0x0000. The SPU decoder is designed to read this data format automatically.

The Audio ES Channel Buffer can contain any of the following audio streams:

1. Linear PCM audio
2. MPEG-1/MPEG-2 audio
3. Dolby Digital audio.

Figure 6.17 shows the mapping of Linear PCM information in the Audio ES Channel Buffer.

Figure 6.17 Audio ES Channel Buffer Map for Linear PCM Audio


63
0
64-bit Sync Code (4C 53 49 4C 4F 47 49 43)
Length (16 bits)
PCM Audio Data 1 (Includes audio frame information and first access unit pointer - 2019 bytes or less)
64-bit Sync Code
Length (16 bits)
PCM Audio Data 2 (Includes audio frame information and first access unit pointer - 2019 bytes or less)
Sync Code
Sync Code (Continued)
Length (16 bits)
PCM Audio Data 3

The Preparser adds a sync code and a data length field to the input Linear PCM packet header to provide the on-chip audio decoder with better error recovery features.

Figure 6.18 illustrates the mapping of the Audio ES Channel Buffer for MPEG-1 or MPEG-2 audio.

Figure 6.18 Audio ES Channel Buffer MAP for MPEG Audio

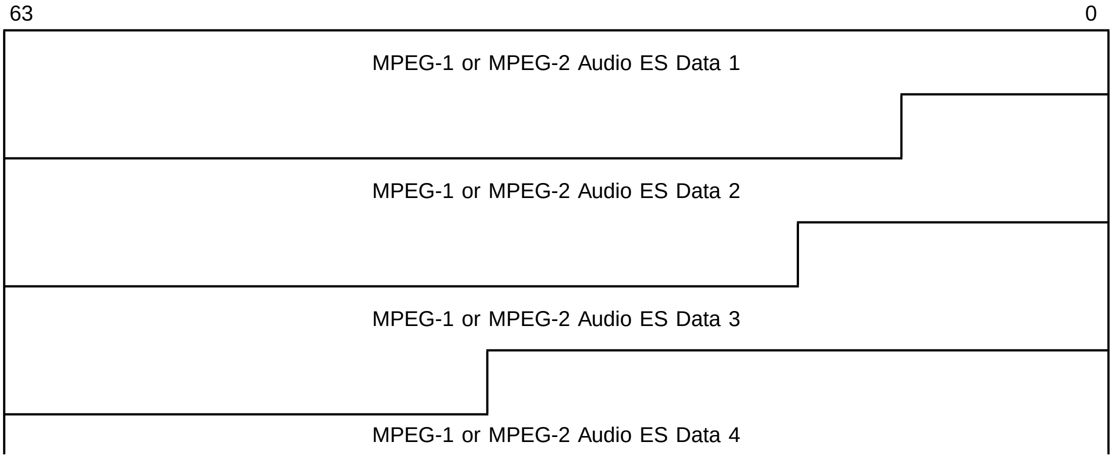

63
0
MPEG-1 or MPEG-2 Audio ES Data 1
MPEG-1 or MPEG-2 Audio ES Data 2
MPEG-1 or MPEG-2 Audio ES Data 3
MPEG-1 or MPEG-2 Audio ES Data 4

Figure 6.19 shows the contents of the Audio ES Channel Buffer for Dolby Digital audio.

Figure 6.19 Audio ES Channel Buffer Map for Dolby Digital

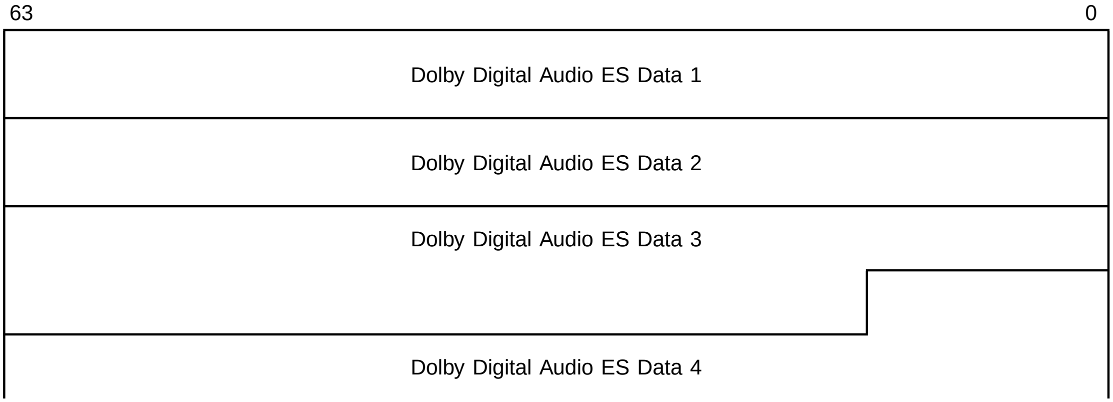

63
0
Dolby Digital Audio ES Data 1
Dolby Digital Audio ES Data 2
Dolby Digital Audio ES Data 3
Dolby Digital Audio ES Data 4

Figure 6.20 shows the Video ES Channel Buffer. There is no word alignment between the current Elementary Stream data and the next Elementary Stream data boundary.

Figure 6.20 Video ES Channel Buffer Map

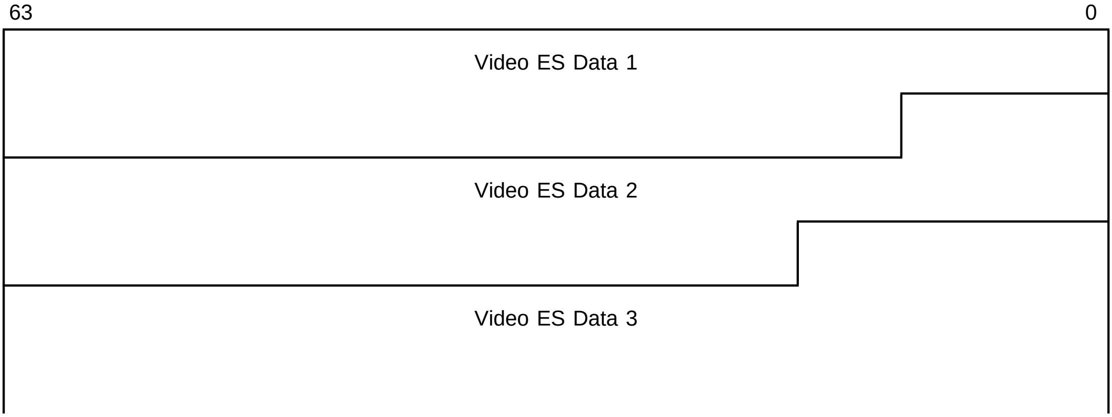
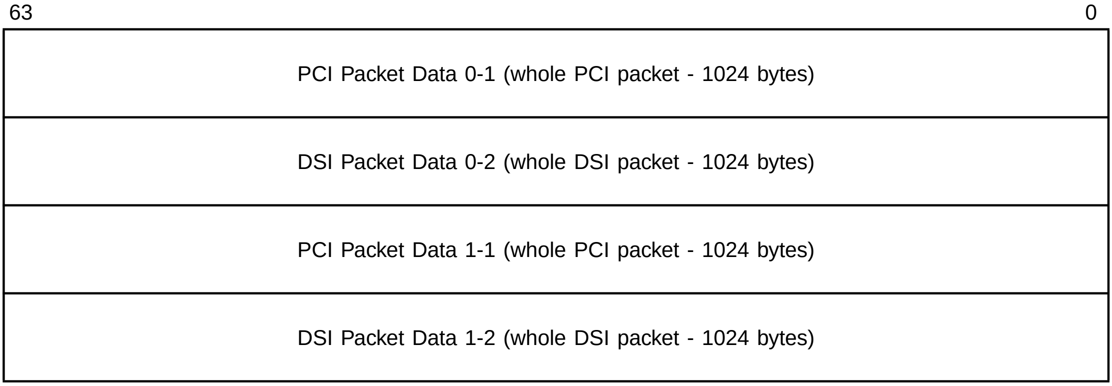


Figure 6.21 shows a map of the Navi Pack Channel Buffer. The buffer stores all PCI and DSI packet data of private_stream_2. PCI and DSI packet data is word aligned after the last PCI/DSI data stored.

Note: This data is NOT used by the L64020 but is made available for the host to read for system use. The DVD navigator is controlled by software.

Figure 6.21 Navi Pack Channel Buffer Map



63 0

PCI Packet Data 0-1 (whole PCI packet - 1024 bytes)

DSI Packet Data 0-2 (whole DSI packet - 1024 bytes)

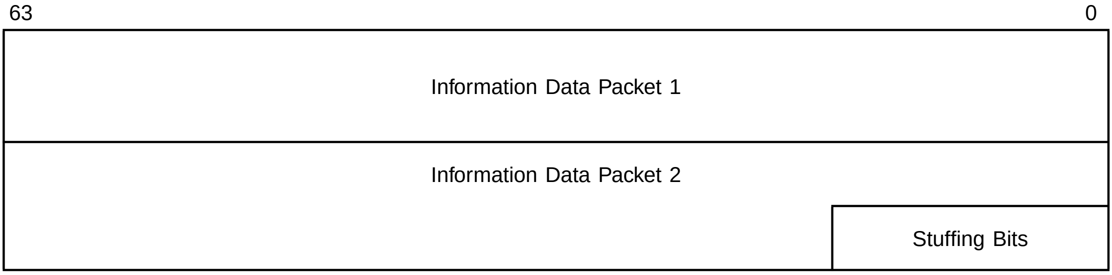
PCI Packet Data 1-1 (whole PCI packet - 1024 bytes)

DSI Packet Data 1-2 (whole DSI packet - 1024 bytes)

Figure 6.22 shows a map of the Data Dump Channel Buffer.

Note: This data is NOT used by the decoder device, but is made available for the host to read for system use.

Figure 6.22 Data Dump Channel Buffer Map



6.3.8 Navi Pack Processing

There are several registers available to the host for implementing Navi pack video/audio pause. The host can set the Navi Pause bit in Register 123 (page 4-36) to pause the channel input at the end of the last Navi packet or set the Pack Pause bit in Register 123 to pause the channel input after the last pack header is stored in its channel buffer. Once the channel is paused, the host can set the Force Sequence End Code bit in Register 123 to force a video sequence end code into the stream just before the next Navi packet or pack header boundary.

When either pause bit is set, the host can also set the Flush Audio bit in Register 123 to stuff sufficient zero bits into the Audio ES channel buffer before the next Navi packet or pack header boundary to flush all audio data into the Audio Decoder.

Several registers are also available for monitoring the number of Navi packs in the channel buffer. The host can set the Navi Pack Counter Enable bit in Register 123 to keep track of the number of stored Navi packs. When the counter reaches 3 (4 packs in the buffer), the channel input is automatically paused until the host reads a Navi pack from the channel buffer and decrements the counter by setting the Navi Pack Counter Decrement bit in Register 123. The host can also read the Navi Pack Counter Output field in Register 123 at any time to determine how many packs are in the buffer.

6.3.9 Error Handling in Program Streams

This section describes how the preparser responds to errors it detects in the input bitstream and to the assertion of the ERRORn input signal by the channel device.

6.3.9.1 Pack Header

In DVD systems, each pack in the stream is stored on a separate sector of the disk. The Top of Sector (TOS) signal supplied by the DVD disk subsystem provides a cross check for the start of a pack.

If the Top of Sector Detect Enable bit in Register 148 (page 4-45) is set when the Preparser detects a pack start code, it looks to see if the TOS signal is asserted. If TOS is not asserted, the Preparser skips input data until TOS is asserted.

6.3.9.2 System Header

Error Check Point: All header data (except the start code and packet length field)

Description: If the ERRORn signal is asserted during the processing of header data, the error data is stored as normal and the Preparser starts the search for the next start code.

6.3.9.3 Private_2 Stream

Error Check Point: Packet data

Description: If the ERRORn signal is asserted during the processing of packet data, the Video Packet Error Status bit in Register 149 (page 4-46) is set, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, and INTRn is asserted if the interrupt is not masked. The byte(s) in error are stored as normal.

6.3.9.4 MPEG Video Stream

MPEG-1 – Error Check Point: After start code detection, header data and packet data

Description: If the ERRORn signal is asserted at start code detection, the whole packet is skipped and the Preparser resynchronizes to the next start code.

If there is a syntax error in the header data, the Video Packet Error Status bit in Register 149 (page 4-46) is set, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, and INTRn is asserted if the interrupt is not masked. The Preparser skips the remainder of the packet after the error and resynchronizes to the next start code.

If the ERRORn signal is asserted while processing packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, and INTRn is asserted if the interrupt is not masked. The error data is removed and sequence error start codes (0x0000.01B4) are injected into the packet.

MPEG-2 – Error Check Point: After start code, zero packet length, and packet data

Description: If the ERRORn signal is asserted during the start code, the whole packet is skipped and the Preparser resynchronizes to the next start code.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, and INTRn is asserted if the interrupt is not masked. The error data is removed and sequence error start codes (0x0000.01B4) are injected into the packet.

6.3.9.5 MPEG Audio Stream

MPEG-1 Audio – Error Check Point: After start code detection, header data and packet data

Description: If the ERRORn signal is asserted during the start code, the whole packet is skipped and the Preparser resynchronizes to the next start code.

If there is a syntax error in the header, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, and INTRn is asserted if the interrupt is not masked. The remainder of the packet is skipped and the Preparser resynchronizes to the next start code.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, INTRn is asserted if the interrupt is not masked, and the error data is stored anyway.

MPEG-2 Audio – Error Check Point: At start code detection and during packet data

Description: If the ERRORn signal is asserted during the start code, the whole packet is skipped and the Preparser resynchronizes to the next start code.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, INTRn is asserted if the interrupt is not masked, and the error data is stored anyway.

6.3.9.6 Dolby Digital Audio Stream

Error Check Point: Packet data and packet length in the first 4 bytes of the packet data

Description: In case of syntax errors in the packet length field (zero packet length), the Preparser searches for the next start code and resynchronizes to it.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, INTRn is asserted if the interrupt is not masked, and the error data is stored anyway.

6.3.9.7 Linear PCM Audio Stream

Error Check Point: Packet data and packet length in first byte of packet data

Description: If there are syntax errors in the packet length field (zero packet length), the Preparser searches for the next start code and resynchronizes to it.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, INTRn is asserted if the interrupt is not masked, and the error data is stored anyway.

6.3.9.8 Subpicture Unit Stream

Error Check Point: Packet data and SPU unit length

Description: If there is an error in the size of the SPU packet data, the SPU Decode Error bit in Register 4 is set, 0x0000 is written in size field, and the entire SPU data after the size field is removed.

If the received SPU unit length is longer than actual unit length, the SPU Decode Error bit in Register 4 is set. Word alignment is performed by inserting padding 0’s before the next valid start code.

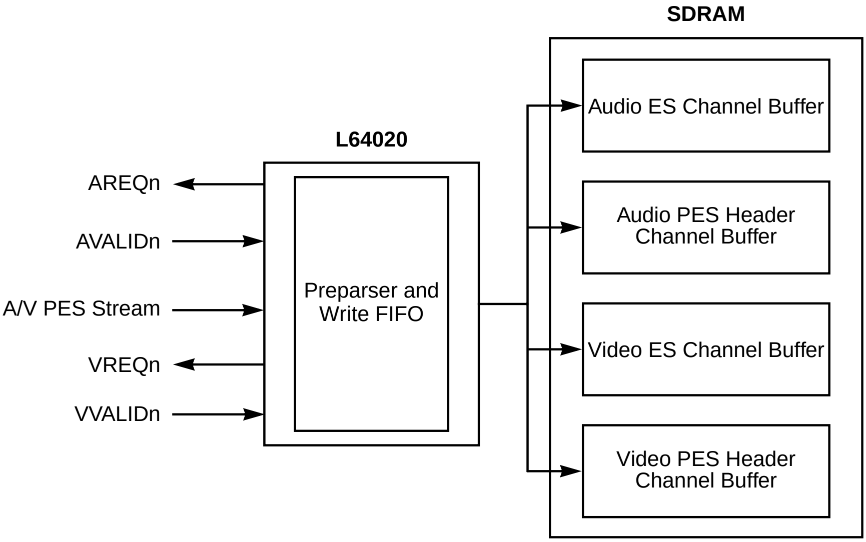
If the received SPU unit length is shorter than the actual unit length, the SPU Decode Error bit in Register 4 is set and the data after the unit length field is removed.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit in Register 4 (page 4-10) is set, INTRn is asserted if the interrupt is not masked, error bytes are injected into the packet data area to end of the unit (before word alignment).

6.3.10 Preparsing A/V PES Packets from a Transport Stream

The L64020 accepts two interleaved PESs (one video, one audio) from a transport stream demultiplexer. The transport stream is different from the program stream because the PES packets are repacketized into fixed 188-byte (184 payload and 4 header) packets. The PES packets may be split at non-PES packet boundaries. This means that there must be duplicate states for parsing the two PES streams and two PES header buffers.

Figure 6.23 Parsing an Audio/Video PES Transport Stream



The payloads of transport packets that contain PES data are presented over the parallel channel interface. The AVALIDn or VVALIDn strobe indicates the type of the elementary stream. Audio PES streams are parsed as described in the MPEG-2 Program Stream case. The Preparser stores the PES header in the Audio PES Header Channel Buffer, INTRn is asserted if not masked, and the Audio PES Data Ready Interrupt bit in Register 2 (page 4-6) is set. The Preparser stores the audio stream in the Audio ES Channel Buffer. Video PES streams are preparsed as described in the MPEG-2 Program Stream case. The Preparser stores the video PES header in the Video PES Header Channel Buffer, INTRn is asserted if not masked, and the Video PES Data Ready Interrupt bit in Register 2 is set. The Preparser stores the video stream in the Video ES Channel Buffer. As in the MPEG-1 and MPEG-2 Program Stream cases, the Preparser adds the buffer write pointers for the start of the audio and video streams after the headers in the PES header buffers. The purpose of the write pointer is to allow the host software to connect the PES header with the next access unit such as a video frame.

The start address, end address, read pointer, and write pointer registers for the Audio ES, Video ES, and Audio PES Header Channel Buffers are those listed for Elementary and MPEG-1 streams in Table 6.9 and Table 6.10.

Note: The registers for the Audio PES Header Channel Buffer are the same as those used for the System Channel Buffer in MPEG stream modes.

The registers associated with the Video PES Header Channel Buffer are shown in Table 6.15.

Table 6.15 Video PES Header/SPU Channel Buffer Registers

Address	Registers	Page Ref.
Video PES Header/SPU Channel Buffer Start Address	80 and 81	4-26
Video PES Header/SPU Channel Buffer End Address	82 and 83	4-26
Video PES Header/SPU Channel Buffer Write Address	102–104	4-31

6.3.11 Error Handling in A/V PES Mode

The paragraphs in this section list the error check points, and describe the errors at the check points and their handling for MPEG-1 and MPEG-2 audio and video.

6.3.11.1 Transport MPEG-1 Audio

Error Check Point: Start codes, header data, and packet data

Description: If the ERRORn signal is asserted during the start code, the Preparser skips the whole packet and resynchronizes to the next start code.

If there are syntax errors during the header data, the Packet Error Interrupt bit is set, INTRn is asserted if the interrupt is not masked, and the remainder of the packet after the error is skipped.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit is set, INTRn is asserted if the interrupt is not masked, and the data in error is stored anyway.

6.3.11.2 Transport MPEG-2 Audio

Error Check Point: Start codes and packet data

Description: If the ERRORn signal is asserted during the start code, the whole packet is skipped and the Preparser resynchronizes to the next start code.

If the ERRORn signal is asserted during packet data, the Packet Error Interrupt bit is set, INTRn is asserted if the interrupt is not masked, and the data in error is stored anyway.

6.3.11.3 Transport MPEG-1 Video

Error Check Point: Start codes, header data, and packet data

Description: If the ERRORn signal is asserted during the start code, the whole packet is skipped and the preparser resynchronizes to the next start code.

If there are syntax errors during the header data, the Packet Error Interrupt bit is set, INTRn is asserted if the interrupt is not masked, the remainder of the packet after the error is skipped, and the Preparser resynchronizes to the next start code.

If the ERRORn signal is asserted during the packet data, the Packet Error Interrupt bit is set, INTRn is asserted if the interrupt is not masked, and sequence error start codes (0x0000.01B4) are substituted for the errored data.

6.3.11.4 Transport MPEG-2 Video

Error Check Point: Start codes, zero packet length, and packet data

Description: If the ERRORn signal is asserted during the start code, the whole packet is skipped and the Preparser resynchronizes to the next start code.

If zero packet length is detected (transport mode is an exception), the packet data until the next start code is stored. If the error occurs in the start code search routine, the Packet Error Interrupt bit is set, INTRn is asserted if not masked, error codes are injected, and then the Preparser resynchronizes.

If the ERRORn signal is asserted during the packet data, the Packet Error Interrupt bit is set, INTRn is asserted if the interrupt is not masked, and sequence error start codes (0x0000.01B4) are substituted for the errored data.

6.4 Channel Buffer Controller

The Channel Buffer Controller manages the various buffers in the external SDRAM. It reads and stores the start and end addresses of each of the buffer areas. It maintains a write pointer for each buffer and a read pointer for those that the internal microcontroller needs to access. It updates the registers holding the read and write pointers. It also keeps track of the number of 64-bit words in the Audio ES Channel Buffer and the number of words or pictures in the Video ES Channel Buffer that have not been read or decoded by the microcontroller and reports the numbers to registers for access by the host. These functions are described in Section 6.3.7, “Preparsing an MPEG-2 Program Stream with DVD.”

The Channel Buffer Controller has other features that aid the host in system operation. These features include the ability to reset each of the buffers individually, support for extracting an actual decode time stamp, and control signals to handle cases of video channel underflow.

6.4.1 Buffer Reset

Each of the buffers can be reset on an individual basis, i.e., without affecting the other buffers. Resetting a buffer returns its read and write pointers to the buffer start address. A buffer is reset when the host sets the corresponding bit in Register 68 (page 4-21). When bit 0 in the register is set, all defined buffers are reset when a packet sync error is Compare and Extract DTS.

The Channel Buffer Controller provides a compare function for extracting actual Decode Time Stamp (DTS) values, i.e., the actual time when a picture or audio frame has started decoding. The host registers associated with this function are listed in Table 6.16.

Table 6.16 Compare DTS Register Bits and Fields

Function	Registers	Page Ref.
Enable Video Read Compare DTS	69	4-22
Enable Audio Read Compare DTS	69	4-23
Video ES Channel Buffer Compare DTS Address	108–110	4-32
Audio ES Channel Buffer Compare DTS Address	111–113	4-33

When the Enable Video Read Compare DTS bit is set, the value in the Video ES Channel Buffer Compare DTS Address registers is constantly compared with the current value of the video channel read pointer. As soon as a match is detected, a signal is generated that triggers a state machine. When the state machine detects a Picture Start Code, the INTRn output to the host is asserted, if not masked, and the DTS Video Event Interrupt bit in Register 2 (page 4-7) is set.

In an actual situation, the host, when alerted, would read the packet header and the start address of a packet payload from the Audio PES Header/System Channel Buffer and write that address to the Video ES Channel Buffer Compare DTS registers. At the first Picture Start Code after the read pointer for the Audio PES Header/System Channel Buffer reached the compare address, the host would be alerted to the start of decoding for that picture. The host would then read the value of the SCR counter as the DTS.

In the case of audio, the host can select the read pointer for the Audio Decoder or the S/P DIF Formatter by setting the Enable Audio Read Compare bits in Register 69 so that synchronization can be maintained against either one. When the compare produces a match, INTRn is asserted if not masked and the DTS Audio Event Interrupt bit (page 4-7) is set.

The Picture Start Code Read Address (Registers 128–130, page 4-38) and the Audio Sync Code Read Address (Registers 131–133, page 4-38) can be used in conjunction with the Picture Start Code Detect Interrupt bit and the Audio Sync Code Detect Interrupt bit (both in Register 1, page 4-4).

6.4.2 Detecting Potential Underflow Conditions in the Video Channel

As previously mentioned, the Channel Buffer Controller keeps track of the number of items (64-bit words) and pictures in the Video ES Channel Buffer and reports these to the host through a set of registers. The Channel Buffer Controller can also be configured by the host to alert the internal microcontroller when the Video ES Channel Buffer does not contain enough unread data to construct an entire picture.

To enable this feature, the host writes a numitems/pics threshold value in Registers 134–136 (see Table 6.17) and sets the Video Numitems/ Pics Panic Mode Select bits to alert the microcontroller when either the number of items or pictures falls below the threshold. The microcontroller then takes suitable action, which may include suspending reconstruction in order for the video channel to build up. The display is frozen (field freeze) on the previously reconstructed picture during the period that reconstruction is suspended.

Table 6.17 Video Channel Underflow Control Registers

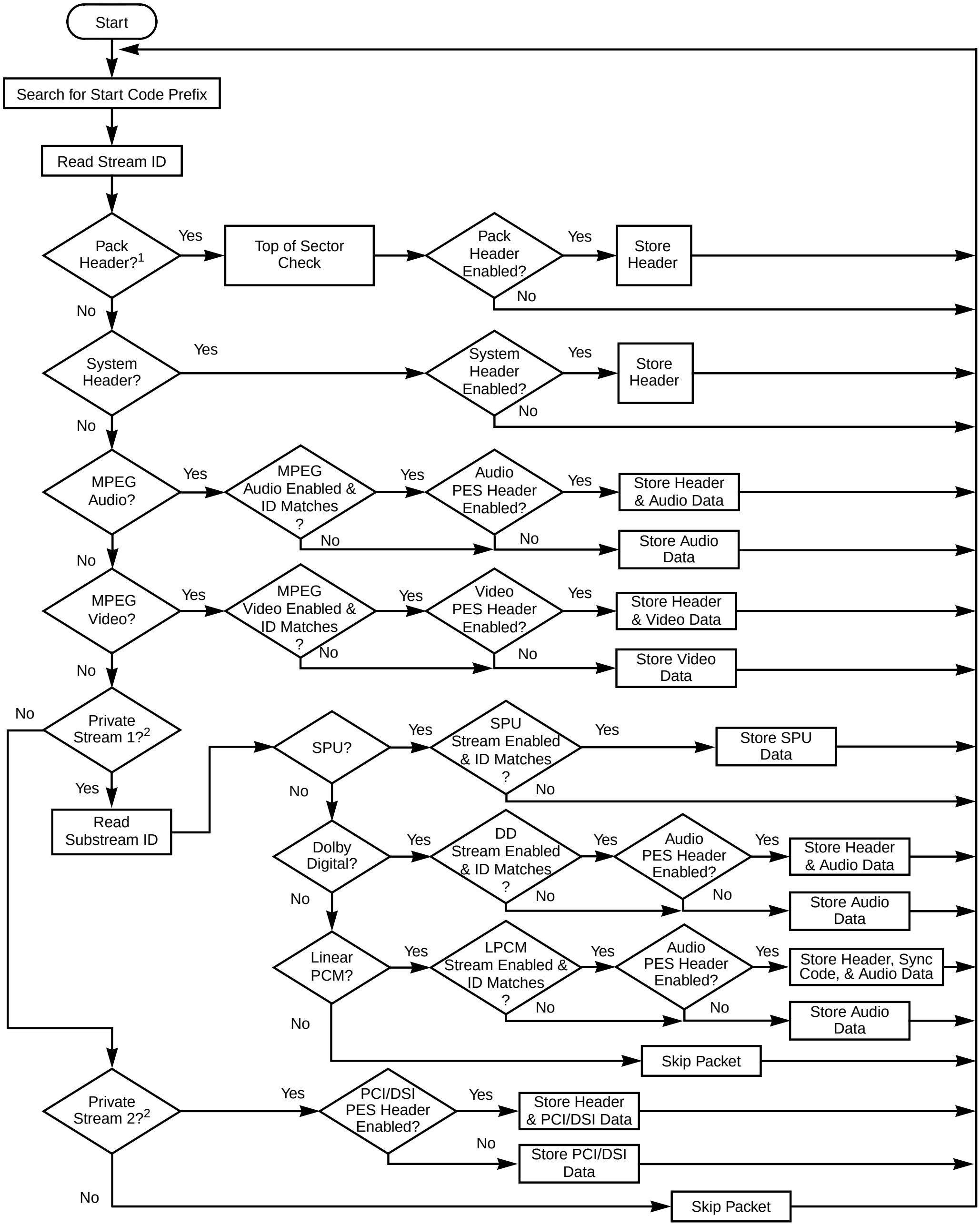
Function	Registers	Page Ref.
Video Numitems/Pics Panic Mode Select	69	4-23
Video Numitems/Pics in Channel Compare Panic	134–136	4-39

The host can read the video numitems at any given time from Registers 134–136 (page 4-39) and the number of pics in the channel at any time from Registers 150 and 151 (page 4-46).

6.5 Summary

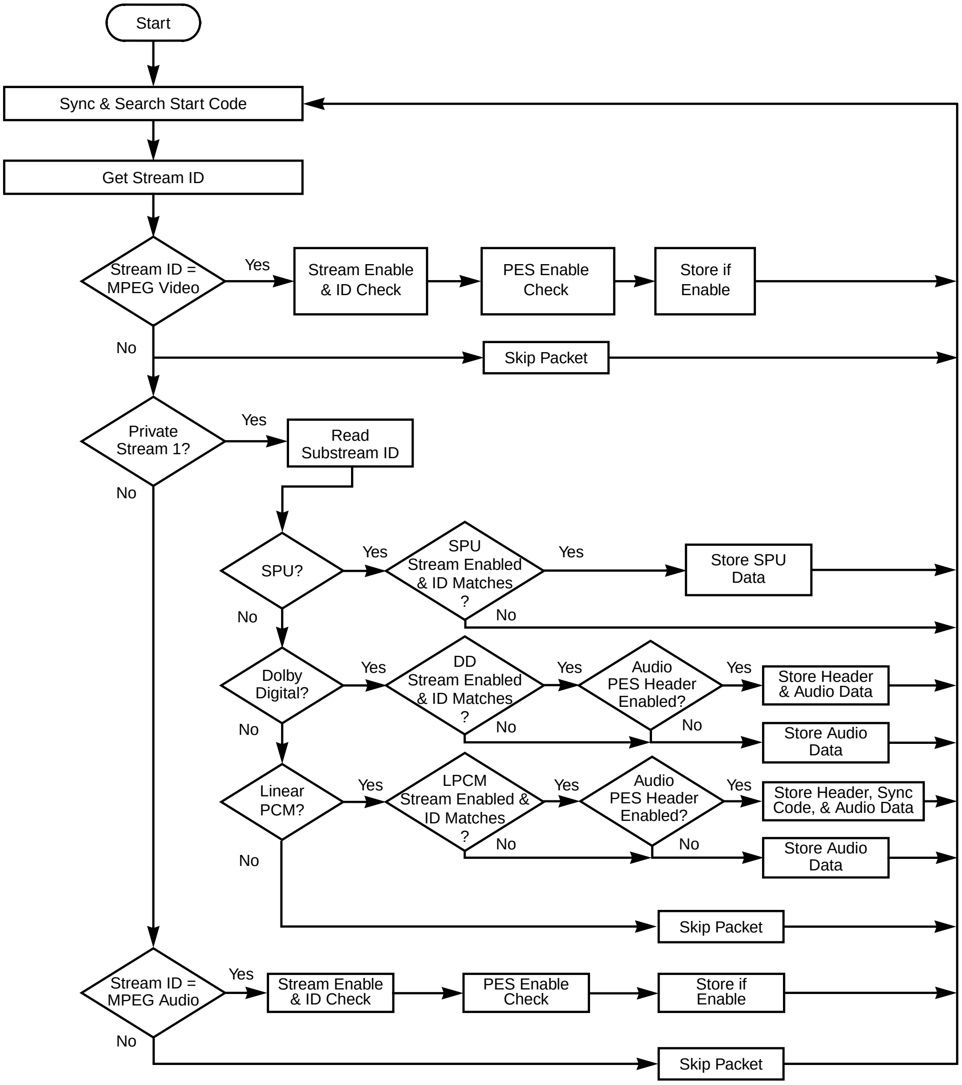
The operation of the Channel Interface is summarized in the flowchart in Figure 6.24 for MPEG-1 streams and MPEG-2 program streams, and in Figure 6.25 for A/V PES streams from transport demultiplexers.

Figure 6.24 MPEG-1/MPEG-2 Channel Interface Operation



1. Not in MPEG-1 streams.
2. In program streams with DVD.

Figure 6.25 A/V PES Mode Channel Interface Operation



Chapter 7
Memory Interface

This chapter describes the memory interface block of the L64020 Decoder. It contains the following sections:

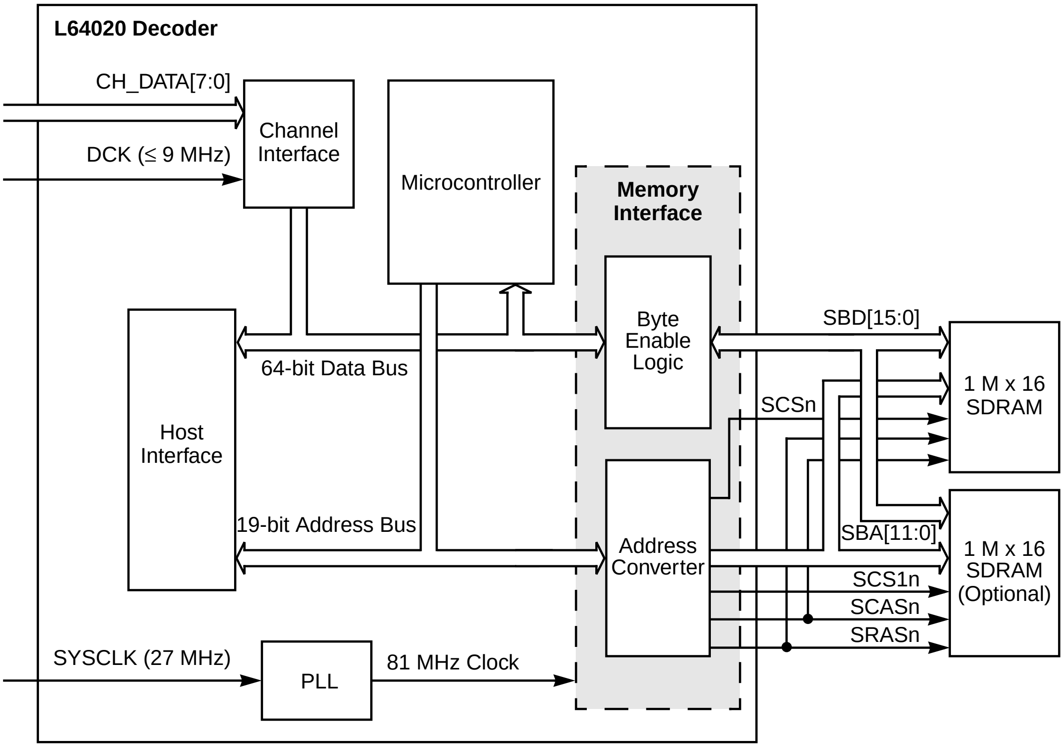
7.1 Overview

The L64020 DVD MPEG Decoder has a dedicated memory interface which is used for video frame storage during MPEG decode, channel buffering of elementary data, storing OSD graphics information, and storing private stream information in a DVD system. The interface includes a 16-bit data bus and a 12-bit multiplexed row/column address bus operating at 81 MHz to commodity synchronous SDRAMs. The L64020 SDRAM interface utilizes an on-chip Phase-Locked Loop (PLL) to generate the 81-MHz clocking signal from the 27-MHz system clock. Since the L64020 has a 64-bit wide internal bus, all SDRAM operations are bursts of four 16-bit accesses. All internal addressing and internal references are relative to 64-bit SDRAM bursts.

The block diagram for the Memory Interface is shown in Figure 7.1. It interfaces the internal address and 64-bit data bus of the L64020 to the 12-bit address bus and 16-bit data bus of the SDRAM(s). Addresses on the internal bus of the L64020 are in the form of simple RAM addresses for 2M x 16-bit RAM. Since the SDRAM is set up for a four-word burst at each access, the internal address bus of the L64020 is only 19 bits wide.

The Memory Interface contains Byte Enable Logic and an Address Converter. The Byte Enable Logic converts the internal 8-byte words to 2-byte SDRAM words and vice versa. The Address Converter converts the 19-bit internal addresses to chip selects SCSn and SCS1n, and multiplexed, 12-bit, row/column addresses. All transfers are in minimum bursts of four SDRAM words. Once a read or write cycle is initiated, however, the Address Converter continually increments the SDRAM address until the host or internal microcontroller terminates the transfer.

Figure 7.1 Memory Interface Block Diagram



7.2 SDRAM Configurations

The SDRAM interface uses commodity SDRAMs in the following configurations:

- 512 x 16-bit page size
- 81-MHz SDRAM clock (162 MBytes/sec max.)
- Page Break Penalty = 6 to 7 cycles (81 MHz)
- Memory capacity: 16 or 32 Mbit using one or two 1M x 16 bit chips

Typical SDRAM devices are the Samsung KM416S1120A or NEC uPD4516161.

The SDRAM interface uses a CAS latency of 3 and a burst length of 4. The 4-word burst provides high bandwidth transfer from the SDRAM 16-bit bus to the internal 64-bit bus. The mode register in the SDRAM is programmed to have CAS LATENCY = 3, and BURST LENGTH = 4.

For systems with 16 Mbit of external SDRAM, the SCSn signal of the L64020 is used as the only chip select (CS). The SCS1n signal is left unconnected. For systems with 32 Mbit of SDRAM, the SCSn signal is the chip select for the lower-address SDRAM and the SCS1n signal is the chip select for the higher-address SDRAM since they share the same data bus. Note that both SDRAM devices must have a 512 x 16 bit page size to match the interface’s column and row addressing.

7.3 SDRAM Timing and Modes

The timing of the SDRAM is very critical and requires careful layout of the PC board traces between the L64020 and the SDRAM device. The SDRAM power and ground lines must be noise-free with sufficient bypass capacitors. The traces connecting the SDRAM to the L64020 must be short and direct. The pinout on the L64020 has been optimized for a clean, single-layer layout to standard, TSOP (II), 50-pin SDRAM packages. The L64020 PLLVDD and PLLVSS pins supply power to the on-chip PLL which generates the 81-MHz clock. These pins must be isolated from the digital power and ground pins and have sufficient bypass coupling near the L64020 to ensure a noise-free PLL power and ground connection. Table 7.1, Figure 7.2, Figure 7.3, and Figure 7.4 show typical timing seen for standard SDRAMs during read, write, and refresh modes. For exact timing, refer to the SDRAM vendor’s data sheet.

Table 7.1 NEC’s 16 Mbit Synchronous SDRAM (Burst Length = 2)

Parameter	T_{RRD} act0 - act1	T_{RCD} act - r/w	T_{RP} pre - act	T_{RAS} act - pre	T_{RC} ref - ref/act	CAS Latency
Time (ns)	36	29	36	84	120	–
No. of Cycles	3	3	3	7	10	3

Figure 7.2 SDRAM Timing Requirements for Reads

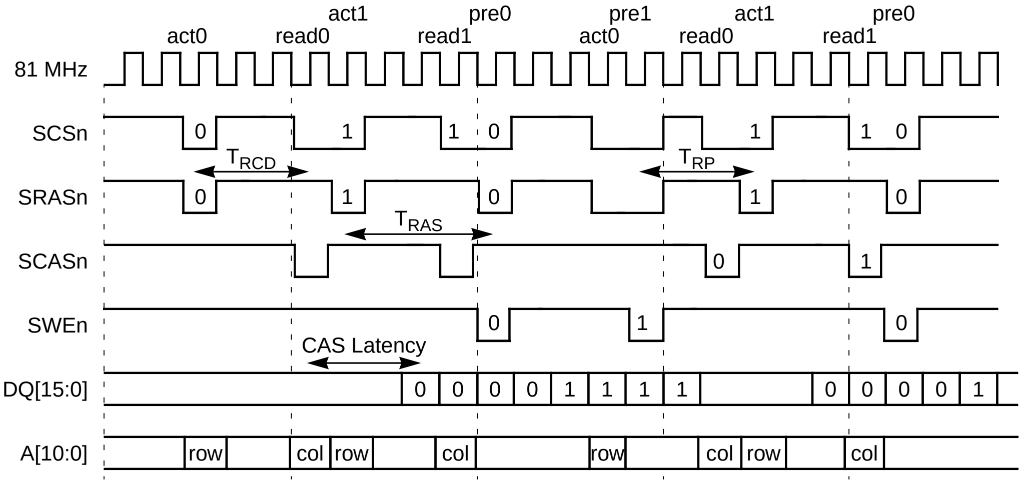


Figure 7.3 SDRAM Timing Requirements for Writes

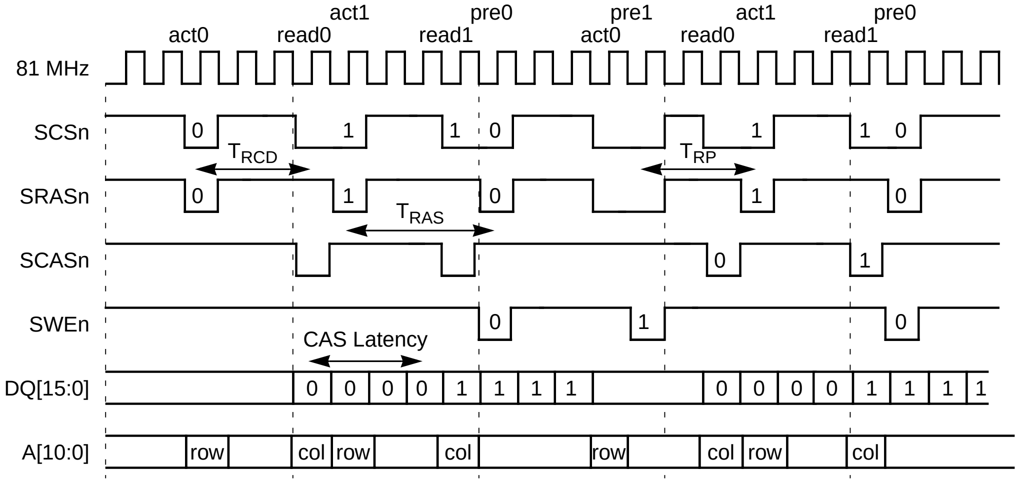
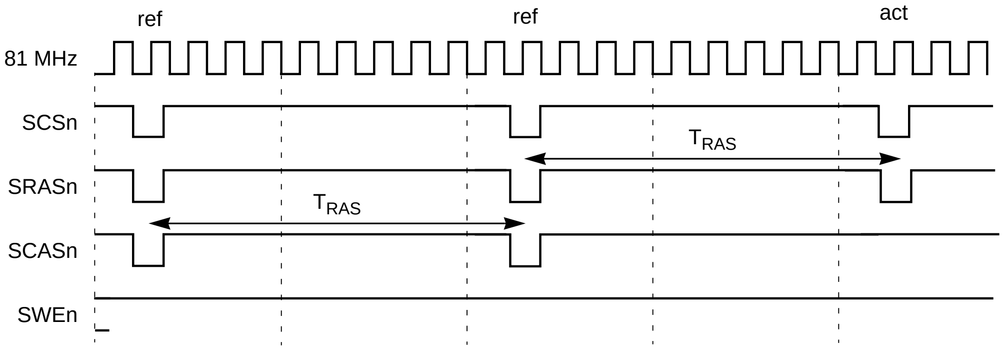


Figure 7.4 SDRAM Timing Requirements for Refresh



7.4 SDRAM Refresh and Arbitration

The refresh rate of the SDRAM is sufficient to maintain 2048 refresh cycles/32 ms. The number of refreshes per macroblock is set by the Refresh Extend bits in Register 193 (page 4-49). The default setting of 2 refreshes per macroblock is sufficient. More refreshes are excessive and the setting of 1 is for LSI Logic internal use only.

SDRAM arbitration is controlled by the internal microcontroller of the L64020. This microcontroller controls the functional units needed to decode MPEG video syntax. It is critical for the decoder to carefully control SDRAM access in order to ensure that the picture can be decoded in the available processing time. The arbitration priority is:

1. MPEG Video Decoder and Channel Interface
2. Display Interface and SPU Decoder
3. Host Interface, block move, and DMA
4. Refresh

7.5 Memory Channel Buffer Allocation

You must control SDRAM space allocation carefully to fit within a low-cost memory solution. Many items must be placed within the SDRAM address map including:

- Audio channel buffers
- Video channel buffers
- System header channel buffers
- Video frame stores (usually 3)
- OSD graphics objects
- SPU channel buffers
- Navi Pack or private stream channel buffers

The last two items are optional for DVD streams. Refer to Chapter 6 for the operation of the channel buffers. Table 7.2 shows typical sizes of buffers for an NTSC output.

Table 7.2 Example NTSC SDRAM Allocation

Item	Size (bytes)
Video channel	229,376
Video Real-Time decode overflow	62,500 (33 ms at 15 Mbps)
Audio channel	4,096
Audio Real-Time decode overflow	4,096
System Header channel	512
3 Video Frame stores	1,555,200
OSD storage area	optional

The area consumed by the channel buffering is defined in *ISO/IEC 13818* and in the *DVD Specification for Read-Only Disc* for private stream items, such as SPU channel and DSI channel. There are a number of items that affect the size of the channel buffering needed in a system.

The MPEG model is based upon an ideal decoder which can instantaneously decode an image at the decode time. Real implementations may take up to one frame time to decode an image. This results in bits backing up in the channel buffer for one frame time until the picture is decoded and removed from the channel. This phenomenon is known as real-time decode and it requires additional space in the video channel buffer. The additional space can be calculated as the frame time x bit rate. A PAL system at 15 Mbps with a 40-ms frame time requires an additional 600,000 bits (75 Kbytes) in the video channel buffer.

A similar calculation can be done for the audio frame time and bit rate. For the audio real-time decode overflow, if the A/V sync provides accuracy to within one frame time, the maximum size of the audio real-time decode overflow is upper bounded by the size of the audio channel. This restriction exists because, in one frame time of audio decode, the system cannot input more than one channel buffer size of audio bits.

The second item that contributes to additional channel buffering requirements is related to A/V synchronization. In most DVD systems, the A/V sync can be maintained very precisely and generally does not add to the channel buffer requirements. However, in a transport system, the A/V sync error can accumulate and require additional buffer space. The space needed is calculated in a manner similar to the real-time decode calculation. The additional bits are determined by the maximum error time provided by the A/V synchronization mechanism. For example, if the maximum A/V sync error is 10 ms, then 10 ms x 15 Mbps = 150,000 additional bits (18,750 bytes) are required. Additional space is similarly needed for audio data.

The third item requiring additional channel buffering is caused by the use of a slave mode pixel interface to the NTSC/PAL encoder. In this system configuration, the decoder is locked to the external VSYNC and cannot start decoding at a channel start until the next VSYNC arrives. This results in a decode start delay of up to one field time or 20 ms in a PAL system. The additional space required is then 20 ms x 15 Mbps = 300,000 bits (37,500 bytes). Although audio decoding starts immediately, the audio must be delayed 20 ms to maintain A/V synchronization. A well-controlled DVD system can avoid this problem by starting the disk play at VSYNC time so it does not need the additional buffering.

In most cases, the size of the SPU channel must be larger than one SPU unit since conditions can occur where one SPU unit is being decoded while another is being received. This generally requires twice the channel buffer size because the L64020 does not have instantaneous SPU decode.

It is the host’s responsibility to program the start and end SDRAM address for all the channel buffers, the video frame stores, and the OSD regions. The registers listed in Table 7.3 are used by the host to program the channel space in the L64020.

Table 7.3 Channel Buffer Architectures

Channel Buffer	Address Bits	Start Address Registers	End Address Registers
Video ES Channel Buffer	[7:0]	72 (page 4-24)	74 (page 4-24)
	[13:8]	73	75
Audio ES Channel Buffer	[7:0]	76 (page 4-25)	78 (page 4-25)
	[13:8]	77	79
Video PES Header/SPU Channel Buffer	[7:0]	80 (page 4-26)	82 (page 4-26)
	[13:8]	81	83
Data Dump Channel Buffer	[7:0]	84 (page 4-27)	86 (page 4-27)
	[13:8]	85	87
Audio PES Header/ System Channel Buffer	[7:0]	88 (page 4-28)	90 (page 4-28)
	[13:8]	89	91
Navi Pack Channel Buffer	[7:0]	92 (page 4-29)	94 (page 4-29)
	[13:8]	93	95

Note that all channel buffer start and end addresses are 14 bits. The SDRAM is addressed by the host and the L64020’s internal microcontroller as if it were simple RAM. The start and end addresses are the upper 14 bits. Therefore, buffer sizes are specified in blocks of 128, 16-bit words or 256 bytes.

7.6 Memory Frame Store Allocation

The SDRAM space allocated for video frame stores is dependent upon the operating mode of the device and the largest picture size expected in the bitstream. The size of the frame store cannot be altered while the video decoder is running. These values must be programmed at power-up time or at the channel-start time when the sequence header arrives. The pixel data in the frame store is arranged in a Luma (L) frame store and a Chroma (C) frame store.

7.6.1 Luma Store

If the reconstructed image is 720 pixels wide, then each line of luminance occupies 720 bytes. Since there are 8 pixels in a 64-bit word, one line of luminance requires 90 64-bit words or 90 bursts to SDRAM. Each frame store starts with the upper left pixel in the reconstruction space and increments in address as the frame store progresses across the pixel line. At the end of a reconstruction line, the next line starts immediately at the next 64-bit word address.

Figure 7.5 Luma Frame Store Organization

	0							63	
Word 0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Frame Start Address
Word 1	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	
	⋮							⋮	
Word 89	Y712	Y713	Y714	Y715	Y716	Y717	Y718	Y719	
Word 90	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
	⋮							⋮	

7.6.2 Chroma Store

The chroma data is stored slightly differently. The L64020 interleaves chroma pixels (Cr, Cb, Cr, Cb) within the same 64-bit word to increase the word fetch efficiency of the SDRAM interface. Each CrCb pair is stored in consecutive bytes in the 64-bit word. There is one CrCb pair for every two pixels. This results in a frame store with the same number of addresses per line in chroma as in luma. However, there are half the number of chroma lines in 4:2:0 source data. Hence, the chroma frame store requires only half the area of the luma frame store. The three operating modes and the area consumed is described below.

Figure 7.6 Chroma Frame Store Organization

	0							63	
Word 0	Cb0	Cr0	Cb1	Cr1	Cb2	Cr2	Cb3	Cr3	Line 1
Word 1	Cb4	Cr4	Cb5	Cr5	Cb6	Cr6	Cb7	Cr7	
	⋮							⋮	
Word 89	Cb356	Cr356	Cb357	Cr357	Cb358	Cr358	Cb359	Cr359	
Word 90	Cb0	Cr0	Cb1	Cr1	Cb2	Cr2	Cb3	Cr3	Line 3
	⋮							⋮	

7.6.3 Normal Mode

The normal operating mode of the decoder requires two frame stores for decoding two anchor frames (I and P pictures) and one frame store for decoding and storing B pictures. The frame store size is computed using the following equation:

Equation 7.1 Frame Store Size Calculation (Bytes)

$$\text{Luma Frame Store Memory} = \text{Pixel Width} \times \text{Line Height}$$

$$\text{Chroma Frame Store Memory} = \text{Pixel Width} \times (\text{Line Height} / 2)$$

Using NTSC images as an example yields the following:

Equation 7.2 Frame Store Size Example for NTSC:

$$\text{NTSC Luna} = 720 \times 480 = 345{,}600 \text{ bytes}$$

$$\text{NTSC Chroma} = 720 \times (480 / 2) = 172{,}800 \text{ bytes}$$

$$\text{Total Frame Store Size} = 3 \times 720 \times 480 \times 1.5 = 1{,}555{,}200 \text{ bytes}$$

7.6.4 Reduced Memory Mode (RMM)

This mode of operation is used in PAL systems and allows a partial frame store for decoding and displaying B frames. A complete description of this mode is given in Section 10.7, “Reduced Memory Mode.” This mode has some restrictions defined in MPEG. Primarily, it uses the Repeat

Last Field instead of Repeat First Field in 3:2 pulldowns and freeze modes. This is necessary because the RMM decoding operation makes use of less than one frame store for B frames. Part of the frame store is reused during the decode operation.

The frame store is broken into *segments.* Each segment represents eight lines of the decoded image and is used as the basis for decoding and displaying the B frames. As the first field of the B frame is being displayed, some of the segments are no longer needed for display and they are reused during the decode of the remainder of the B frame. This saves memory but only one field of the B frame is stored at any one time.

SDRAM space allocated for the frame store is similar to the anchor frame stores in the normal mode. However, space must be allocated for the B frame store based on the number of segments allowed for the B frame reconstruction. Segments must be allocated in pairs. The minimum number of segments is half a frame size. For a PAL image with 576 lines, the picture is divided into 576/8 = 72 segments. The minimum number of used segments is 72/2 = 36 segments. For most applications, 40 to 44 segments are recommended. More segments allow the decoder to decode ahead of the display and improve the bandwidth constraints on the decoder. This is important in a display mode with letterbox filtering, since the decoder is constrained in decode time. In letterbox display modes, 44 segments are recommend for PAL systems. Note that the chroma frame store size may be the full normal size if display modes using chroma field repeat are needed. This results in frame store memory space in the following sizes.

Equation 7.3 Reduced Memory Frame Store Size Calculation

$$\text{RMM Luma Frame Store Memory} = \text{Num Segments} \times \text{Pixel Width} \times 8$$

Equation 7.4 With Chroma Line Repeat Display Mode

$$\text{RMM Chroma Frame Store Memory} = \text{Num Segments} \times \text{Pixel Width} \times 4$$

Equation 7.5 With Chroma Field Repeat Display Mode

$$\text{Chroma Frame Store Memory} = \text{Pixel Width} \times \text{Line Height} \times 2$$

The following example shows the space consumed by the display B frame store in a PAL system. The example uses 44 segments with a Chroma Line Repeat Display mode.

Equation 7.6 Reduced Memory Frame Store Size Example for PAL

$$\text{PAL B Frame Luma} = 44 \times 720 \times 8 = 253{,}440 \text{ bytes}$$

$$\text{PAL B Frame Chroma} = 44 \times 720 \times 4 = 126{,}720 \text{ bytes}$$

$$\begin{aligned}\text{Total Frame Store Size} &= (2 \text{ Anchor Frames} \times 720 \times 576 \times 1.5) \\ &\quad + 253440 + 126720 = 1{,}624{,}320 \text{ bytes}\end{aligned}$$

Note: This is approximately 2.61 frame stores.

7.7 Summary

Table 7.4 shows an example of buffer and frame store SDRAM allocation for an MPEG-2 stream displayed in NTSC format.

Table 7.4 Example NTSC SDRAM Allocation with Frame Store (720 x 480)

Item	Size (bytes)
Video ES Channel Buffer	229,376
Audio ES Channel Buffer	4,096
Video PES Header/SPU Channel Buffer	512
Audio PES Header/System Channel Buffer	512
Navi Pack Channel Buffer	2,048
OSD Storage	Optional
Anchor Luma Frame Store 1	345,600
Anchor Chroma Frame Store 1	172,800
Anchor Luma Frame Store 2	345,600
Anchor Chroma Frame Store 2	172,800
B Luma Frame Store	345,600
B Chroma Frame Store	172,800
Decode Overflow + Other Usage	Optional

Chapter 8
Video Decoder Module

This chapter describes the operation of the Video Decoder Module in the L64020. The chapter contains the following sections:

8.1 Overview

The Video Decoder Module of the L64020 supports the following:

- MPEG-2 Main Profile @ Main Level decoding.
- Simple Profile @ Main Level is also supported. As such, it also decodes MPEG-1 video bitstreams.
- Picture resolutions up to 720 x 576. See Section 10.6, “Display Modes and Vertical Filtering,” regarding restrictions on PAL full-size resolutions.
- A whole host of trick modes, including skip, repeat, rip forward, etc. See Section 8.5, “Trick Modes,” for details.

For a complete description of the MPEG-1 syntax and grammar, see *ISO/IEC 11172-2*. Complete MPEG-2 descriptions can be found in *ISO/IEC 13818-2*.

A block diagram of the Video Decoder Module is shown in Figure 8.1. The module includes a Channel Read FIFO, Postparser, IDCT Pipeline, and the Auxiliary and User Data FIFOs and their controller. The microcontroller is also included since it decodes for the Postparser and controls most of the data transfers.

The Channel Interface, described in Chapter 6, parses pack, system, and packet headers from the bitstream and stores video packet payloads in the Video ES Channel Buffer in SDRAM. The preparsed video data is read from the Video ES Channel Buffer into the Channel Read FIFO.

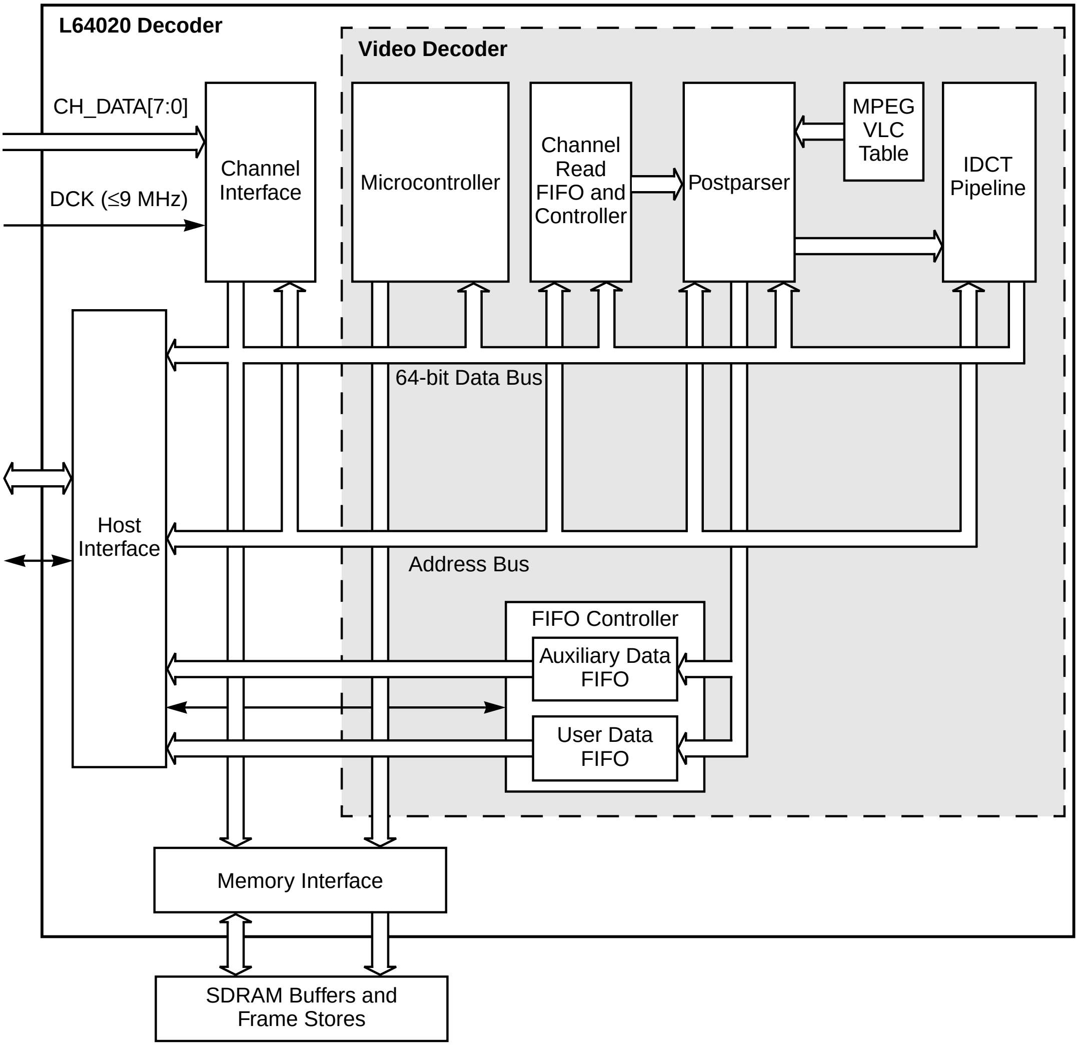
The Postparser, along with the microcontroller, strips the bitstream apart, and passes the appropriate bits and fields in the stream to the microcontroller for use in picture decoding, to the Auxiliary Data FIFO and User Data FIFO for processing by the host, and to the IDCT (Inverse Discrete Cosine Transform) Pipeline for picture data decoding and reconstruction. The Postparser decodes layers of syntax starting from the sequence layer and going through all the lower layers including the group of pictures layer, picture layer, slice layer, macroblock layer, and block layer. Table 8.1 through Table 8.12 define the postparsing operation.

The IDCT Pipeline decodes the block layer bytes and the results are placed in the frame stores in SDRAM. The Video Interface, described in Chapter 10, reads the picture data from SDRAM, mixes it with SPU and OSD video and sends the mix to the external NTSC/PAL Encoder.

The Auxiliary Data FIFO is used to store certain parameters from each of the layers of syntax. The data in the FIFO is available through a register for the host to read. In general, this data is useful in controlling the decoder. The User Data FIFO is used to store data that follows the user data start code in the MPEG-1/2 bitstream. User data also is available to the host through a register.

Some limited error detection is possible in the syntax and grammar parsing of a MPEG bitstream. Illegal transitions out of variable length decode trees and illegal grammars are detected. Usually, it is not appropriate to continue decoding once the first error in a given layer has been detected. Resync codes are used to try to establish synchronization as soon as possible after an error and to limit the propagation of errors. Resync is achieved by searching for the next start code of the appropriate layer in the bitstream. This approach of resynching is also useful in the situation where channels are switched. The channel switch time can be decreased by increasing the number of sequence start codes in the bitstream.

Figure 8.1 Video Decoder Block Diagram



8.2 Postparser Operation

As mentioned, the Postparser separates the bitstream into its individual bits, fields, and picture blocks and steers them to other modules in the Video Decoder. Table 8.1 through Table 8.10 list all of the header parameters in a sequence, shows their format, and indicates their disposition by the Postparser.

8.2.1 Sequence Header

Table 8.1 shows the actions the decoder takes for each of the parameters present in the Sequence Header**.**

Table 8.1 Sequence Header Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
sequence_header_ code (PscB3)	A[31:0]	32	bslbf	n	y								
horizontal_size_value	B[11:0]	12	uimsbf	y	y	0	0	0	0	B11	B10	B9	B8
						B7	B6	B5	B4	B3	B2	B1	B0
vertical_size_value	C[11:0]	12	uimsbf	y	y	0	0	0	0	C11	C10	C9	C8
						C7	C6	C5	C4	C3	C2	C1	C0
aspect_ratio_ information	D[3:0]	4	uimsbf	y	n	0	0	0	0	D3	D2	D1	D0
frame_rate_code	E[3:0]	4	uimsbf	y	n	0	0	0	0	E3	E2	E1	E0
bit_rate_value	F[17:0]	18	uimsbf	y	n	0	0	0	0	0	0	F17	F16
						F15	F14	F13	F12	F11	F10	F9	F8
						F7	F6	F5	F4	F3	F2	F1	F0
marker_bit	G0	1	bslbf	n	n								G0
vbv_buffer_size_ value	H[9:0]	10	uimsbf	y	n	0	0	0	0	0	0	H9	H8
						H7	H6	H5	H4	H3	H2	H1	H0
(Sheet 1 of 2)													

Table 8.1 Sequence Header Processing (Cont.)

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
constrained_ parameters_flag	I0	1	bslbf	y	n	0	0	0	0	0	0	0	I0
load_intra_ quantizer_matrix	J0	1	uimsbf	n	y								J0
if (load_intra_ quantizer_matrix)													
intra_quantizer_ matrix[64][2]	K	8*64	uimsbf	n	y								
load_non_intra_ quantizer_matrix	L0	1	uimsbf	n	y								L0
if (load_non_intra_ quantizer_matrix)													
non_intra_quantizer_ matrix[64][2]	M	8*64	uimsbf	n	y								
(Sheet 2 of 2)													

1. bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first
2. If present, custom quant matrix values are written into on-chip storage. These may be read by the host. See the Q table registers on page 4-65 and page 4-66 for details.

8.2.2 Sequence Extension

Table 8.2 shows the actions the decoder takes for each of the parameters present in the Sequence Extension.

Table 8.2 Sequence Extension Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension_start_code (PscB5)	A[31:0]	32	bslbf	n	y								
extension_start_code_ identifier	B[3:0]	4	uimsbf	y	y	0	0	0	0	0	0	0	1
profile_and_level_ indication	C[7:0]	8	uimsbf	y	n	C7	C6	C5	C4	C3	C2	C1	C0
progressive_sequence	D0	1	uimsbf	y	y	0	0	0	0	0	0	0	D0
chroma_format	E[1:0]	2	uimsbf	y	n							E1	E0
horizontal_size_ extension	F[1:0]	2	uimsbf	y	n							F1	F0
vertical_size_extension	G[1:0]	2	uimsbf	y	n	0	0	E1	E0	F1	F0	G1	G0
bit_rate_extension	H[11:0]	12	uimsbf	y	n	0	0	0	0	H11	H10	H9	H8
						H7	H6	H5	H4	H3	H2	H1	H0
marker_bit	I0	1	bslbf	n	n								I0
vbv_buffer_size_ extension	J[7:0]	8	uimsbf	y	n	J7	J6	J5	J4	J3	J2	J1	J0
low_delay[2]	K0	1	uimsbf	y	n								K0
frame_rate_extension_n	L[1:0]	2	uimsbf	y	n							L1	L0
frame_rate_extension_d	M[4:0]	5	uimsbf	y	n	K0	L1	L0	M4	M3	M2	M1	M0

1. bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first
2. Although the low_delay bit is not used in the decoding process, low_delay bitstreams (I and P pictures only) are supported by the decoder.

8.2.3 Sequence Display Extension

Table 8.3 shows the actions the decoder takes for each of the parameters present in the Sequence Display Extension.

Table 8.3 Sequence Display Extension Processing

Parameter	Parameter Id	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension_start_code_ identifier (0x2)	A[3:0]	4	uimsbf	y	y	0	0	0	0	0	0	1	0
video_format	B[2:0]	3	uimsbf	y	n						B2	B1	B0
color_description	C[0]	1	uimsbf	y	n	0	0	0	0				C0
if (color_description) {													
color_primaries	D[7:0]	8	uimsbf	y	n	D7	D6	D5	D4	D3	D2	D1	D0
transfer_characteristics	E[7:0]	8	uimsbf	y	n	E7	E6	E5	E4	E3	E2	E1	E0
matrix_coefficients	F[7:0]	8	uimsbf	y	n	F7	F6	F5	F4	F3	F2	F1	F0
}													
display_horizontal_size	G[13:0]	14	uimsbf	y	y	0	0	G13	G12	G11	G10	G9	G8
						G7	G6	G5	G4	G3	G2	G1	G0
marker_bit	H[0]	1	bslbf	n	n								H0
display_vertical_size	I[13:0]	14	uimsbf	y	n	0	0	I13	I12	I11	I10	I9	I8
						I7	I6	I5	I4	I3	I2	I1	I0

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

8.2.4 Group of Pictures Header

Table 8.4 shows the actions the decoder takes for each of the parameters present in the Group of Pictures Header.

Table 8.4 Group Of Pictures Header Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
group_start_code (PscB8)		32	bslbf	n	y								
time_code	A[24:0]	25	bslbf	y	n	0	0	0	0	0	0	0	A24
						A23	A22	A21	A20	A19	A18	A17	A16
						A15	A14	A13	A12	A11	A10	A9	A8
						A7	A6	A5	A4	A3	A2	A1	A0
closed_gop	B0	1	uimsbf	y	y	0	0	0	0	0	0	0	B0
broken_link	C0	1	uimsbf	y	y	0	0	0	0	0	0	0	C0

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

8.2.5 Picture Header

Table 8.5 shows the actions the decoder takes for each of the parameters present in the Picture Header.

Table 8.5 Picture Header Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
picture_start_code (Psc00)		32	bslbf	n	y								
temporal_reference	A[9:0]	10	uimsbf	y	y	0	0	0	0	0	0	A9	A8
						A7	A6	A5	A4	A3	A2	A1	A0
picture_coding_type	B[2:0]	3	uimsbf	y	y	0	0	0	0	0	B2	B1	B0
vbv_delay	C[15:0]	16	uimsbf	y	n	C15	C14	C13	C12	C11	C10	C9	C8
						C7	C6	C5	C4	C3	C2	C1	C0
if (picture_coding_ type==2 || picture_coding_type= =3) {													
full_pel_forward_ vector	D0	1	bslbf	n	y								D0
forward_f_code	E[2:0]	3	bslbf	n	y						E2	E1	E0
}													
if (picture_coding_type ==3) {													
full_pel_backward_ vector	F0	1	bslbf	n	y								F0
backward_f_code	G[2:0]	3	bslbf	n	y						G2	G1	G0
}													
while (nextbits() =='1') {													
(Sheet 1 of 2)													

Table 8.5 Picture Header Processing (Cont.)

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extra_bit_picture /* with the value 1 */	H0	1*n	uimsbf	n	n	0	0	0	0	0	0	0	1
extra_information_ picture	I[7:0]	8*n	uimsbf	n	n	I7	I6	I5	I4	I3	I2	I1	I0
}													
extra_bit_picture /* with the value 0 */	J0	1	uimsbf	n	n	0	0	0	0	0	0	0	0
(Sheet 2 of 2)													

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

8.2.6 Picture Coding Extension

Table 8.6 shows the actions the decoder takes for each of the parameters present in the Picture Coding Extension.

Table 8.6 Picture Coding Extension Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension_start_code (PscB5)		32	bslbf	n	y								
extension_start_code_ identifier	A[3:0]	4	uimsbf	y	y	0	0	0	0	1	0	0	0
f_code[0][0]	B[3:0]	4	uimsbf	y	y	0	0	0	0	B3	B2	B1	B0
f_code[0][1]	C[3:0]	4	uimsbf	y	y	0	0	0	0	C3	C2	C1	C0
f_code[1][0]	D[3:0]	4	uimsbf	y	y	0	0	0	0	D3	D2	D1	D0
f_code[1][1]	E[3:0]	4	uimsbf	y	y	0	0	0	0	E3	E2	E1	E0
intra_dc_precision	F[1:0]	2	uimsbf	y	y	0	0	0	0	0	0	F1	F0
(Sheet 1 of 2)													

Table 8.6 Picture Coding Extension Processing (Cont.)

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
picture_structure	G[1:0]	2	uimsbf	y	y	0	0	0	0	0	0	G1	G0
top_field_first	H0	1	uimsbf	y	y	0	0	0	0	0	0	0	H0
frame_pred_frame_dct	I0	1	uimsbf	y	y	0	0	0	0	0	0	0	I0
concealment_motion_vectors	J0	1	uimsbf	y	y	0	0	0	0	0	0	0	J0
q_scale_type	K0	1	uimsbf	y	y	0	0	0	0	0	0	0	K0
intra_vlc_format	L0	1	uimsbf	y	y	0	0	0	0	0	0	0	L0
alternate_scan	M0	1	uimsbf	y	y	0	0	0	0	0	0	0	M0
repeat_first_field	N0	1	uimsbf	y	y	0	0	0	0	0	0	0	N0
chroma_420_type	O0	1	uimsbf	y	n	0	0	0	0	0	0	0	O0
progressive_frame	P0	1	uimsbf	y	n	0	0	0	0	0	0	0	P0
composite_display_flag	Q0	1	uimsbf	y	n	0	0	0	0	0	0	0	Q0
if (composite_display_flag) {													
v_axis	R0	1	uimsbf	y	n								R0
field_sequence	S[2:0]	3	uimsbf	y	n						S2	S1	S0
sub_carrier	T0	1	uimsbf	y	n	0	0	0	0	0	0	0	T0
burst_amplitude	U[6:0]	7	uimsbf	y	n	0	U6	U5	U4	U3	U2	U1	U0
sub_carrier_phase	V[7:0]	8	uimsbf	y	n	V7	V6	V5	V4	V3	V2	V1	V0
}													
(Sheet 2 of 2)													

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

8.2.7 Quant Matrix Extension

Table 8.7 shows the actions the decoder takes for each of the parameters present in the Quant Matrix Extension.

Table 8.7 Quant Matrix Extension Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension_start_code_identifier (0x3)	A[3:0]	4	uimsbf	y	y	0	0	0	0	0	0	1	1
load_intra_quantizer_matrix		1	uimsbf	n	y								
if (load_intra_quantizer_matrix)													
intra_quantizer_matrix[64][2]		8*64	uimsbf	n	y								
load_non_intra_quantizer_matrix	B0	1	uimsbf	n	y								B0
if (load_non_intra_quantizer_matrix)													
non_intra_quantizer_matrix[64][1]		8*64	uimsbf	n	y								
load_chroma_intra_quantizer_ matrix	C0	1	uimsbf	n	n								C0
if (load_chroma_intra_quantizer_ matrix)													
chroma_intra_quantizer_matrix[64][3]		8*64	uimsbf	n	n								
load_chroma_non_intra_quantizer_ matrix	D0	1	uimsbf	n	n								D0
if (load_chroma_non_intra_quantizer_ matrix)													
chroma_non_intra_quantizer_ matrix[64]		8*64	uimsbf	n	n								

1. bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first
2. Although these values are not written to the auxiliary FIFO, they can still be read by the host microprocessor. See Table 3.2 on page 3-2 for details.
3. Chroma quant matrix extension values are not processed since they are not required for Main Profile @ Main Level.

8.2.8 Host Access of Q Table Entries

The host can read the intra and non-intra quant matrix values that are stored in the Q table in the L64020 for the current decode process. The quant matrix values may be the default values or they may have been provided by the bitstream in the sequence header or in the quant matrix extension.

The L64020 sets the Q Table Ready bit in Register 241 (page 4-65) when the quant matrix values are all stored. The host sets the Intra Q Table bit in the same register to select that table or clears it to select the non-intra Q table. Then the host writes the address (0 to 63) of the particular matrix entry into bits [7:2] of Register 241 (page 4-65) and reads the value at that entry from Register 242.

8.2.9 Picture Display Extension

Table 8.8 shows the actions the decoder takes for each of the parameters present in the Picture Display Extension.

Table 8.8 Picture Display Extension Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension_start_code_identifier (0x7)	A[3:0]	4	uimsbf	y	y	0	0	0	0	0	1	1	1
for (i=0; i<number_of_frame_center_offsets;i++) {[2]													
frame_center_horizontal_offset	B[15:0]	16*n	simsbf	y	y	0	0	B15	B14	B13	B12	B11	B10
						B9	B8	B7	B6	B5	B4	B3	B2
						0	0	0	0	0	0	B1	B0
marker_bit	C0	1*n	bslbf	n	n								C0
frame_center_vertical_offset[3]	D[15:0]	16*n	simsbf	y	n	0	0	0	0	D15	D14	D13	D12
						D11	D10	D9	D8	D7	D6	D5	D4
						0	0	0	0	D3	D2	D1	D0
marker_bit	E0	1*n	bslbf	n	n								E0
}													

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first; simsbf = signed integer, most significant bit first

[2] The value of the parameter Number_of_frame_center_offsets is a function of the following parameters received in the sequence coding extension and in the picture coding extension; Progressive_ sequence (`prog`), Picture_structure (`pic_str`), Top_field_first (`tff`), and Repeat_first_field (`rff`). See Table 8.9 for allowable values for these parameters.

[3] Vertical offsets are written to the Auxiliary FIFO but are not supported by the L64020.

Table 8.9 Number of Frame Center Offsets

Progressive Sequence Bit	Picture Structure	Top Field First Bit	Repeat First Field Bit	Display Order	Number of Frame Center Offsets
0	frame	0	0	bottom field, top field	2
0	frame	0	1	bottom, top, bottom field	3
0	frame	1	0	top, bottom field	2
0	frame	1	1	top, bottom, top field	3
0	top field	0	0	top field	1
0	bottom field	0	0	bottom field	1
1	frame	0	0	frame	1
1	frame	0	1	frame, frame	2
1	frame	1	1	frame, frame, frame	3

8.2.10 Copyright Extension

Table 8.10 shows the actions the decoder takes for each of the parameters present in the Copyright Extension.

Table 8.10 Copyright Extension Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to Auxiliary FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
extension_ start_code_ identifier (0x4)	A[3:0]	4	uimsbf	y	y	0	0	0	0	0	1	0	0
copyright_flag	B0	1	bslbf	y	n	0	0	0	0	0	0	0	B0
copyright_ identifier	C[7:0]	8	uimsbf	y	n	C7	C6	C5	C4	C3	C2	C1	C0
original_or_ copy	D0	1	bslbf	y	n	0	0	0	0	0	0	0	D0
reserved	E[6:0]	7	uimsbf	y	n	0	E6	E5	E4	E3	E2	E1	E0
marker_bit	F0	1	bslbf	n	n								F0
copyright_ number_1	G[19:0]	20	uimsbf	y	n	G19	G18	G17	G16	G15	G14	G13	G12
						G11	G10	G9	G8	G7	G6	G5	G4
						0	0	0	0	G3	G2	G1	G0
marker_bit	H0	1	bslbf	n	n								H0
copyright_ number_2	I[21:0]	22	uimsbf	y	n	I21	I20	I19	I18	I17	I16	I15	I14
						I13	I12	I11	I10	I9	I8	I7	I6
						0	0	I5	I4	I3	I2	I1	I0
marker_bit	J0	1	bslbf	n	n								J0
copyright_ number_3	K[21:0]	22	uimsbf	y	n	K21	K20	K19	K18	K17	K16	K15	K14
						K13	K12	K11	K10	K9	K8	K7	K6
						0	0	K5	K4	K3	K2	K1	K0

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

8.2.11 User Data

User data is written to the User Data FIFO, which is separate from the Auxiliary Data FIFO. The Group of Pictures (GOP) User Data Only bit in Register 239 (page 4-63) controls the user data processing. When this bit is cleared, user data of all layers is written to the User Data FIFO as shown in Table 8.11. When this bit is set, for each sequence, only the first GOP-layer (DVD-compliant user data) is put into the User FIFO as shown in Table 8.12. This is the first 120 bytes and is the line 21 closed-caption data. User data belonging to others layers is discarded. See also Section 8.2.15, "User Data FIFO Operation."

Table 8.11 All User Data Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to User FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
user_data_start_code (PscB2)		32	bslbf	n	n								
while (nextbits() != '0000 0000 0000 0000 0000 0001')													
user_data	A[7:0]	8*n	uimsbf	y	n	A7	A6	A5	A4	A3	A2	A1	A0

1. bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

Table 8.12 DVD-Compliant Closed Caption User Data Processing

Parameter	Parameter ID	No. of Bits	Bit Assignment[1]	Written to User FIFO	Used for Decoding	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
user_data_start_ code (PscB2)		32	bslbf	n	n								
reserved	A[15:0]	16	uimsbf	y	n	A15	A14	A13	A12	A11	A10	A9	A8
						A7	A6	A5	A4	A3	A2	A1	A0
top_field_flag_ of_gop	B	1	bslbf	y	n	0	0	0	0	0	0	0	B0
reserved	C0	1	bslbf	y	n	0	0	0	0	0	0	0	C0
number_of_ displayed_field_gop	D[5:0]	6	uimsbf	y	n	0	0	D5	D4	D3	D2	D1	D0
for(i=0;i <number_of_ displayed_ field_gop; i++) {													
marker bits, line21_switch	E[6:0] F0	8	uimsbf	y	n	E6	E5	E4	E3	E2	E1	E0	F0
linew21_data	G[15:0]	16	uimsbf	y	n	G15	G14	G13	G12	G11	G10	G9	G8
						G7	G6	G5	G4	G3	G2	G1	G0
}													

[1] bslbf = bit stream left bit first; uimsbf = unsigned integer, most significant bit first

8.2.12 Picture Data

All layers of syntax including and below picture data, i.e., slice, macroblock, and block are processed by the decoder to reconstruct the picture. None of the parameters in picture data are written to the Auxiliary Data FIFO or the User Data FIFO.

8.2.13 Unsupported Syntax

The decoder supports Main Profile @ Main Level. As such, there is no support for scalable extensions, i.e., sequence scalable extension, picture temporal scalable extension, and picture spatial scalable extension. Chroma formats 4:4:4 and 4:2:2 also are not supported.

8.2.14 Auxiliary Data FIFO Operation

The Auxiliary Data FIFO is used to store certain header parameters required by the system controller or host. The FIFO operates as a 128-byte-deep circular buffer. The various registers associated with the Auxiliary Data FIFO are listed in Table 8.13 and described in the text following. More complete descriptions can be found at the page references shown in the table.

Table 8.13 Aux Data FIFO Registers

Register	Bit(s)	R/W	Name	Page Ref.
0	1	R	Aux/User Data FIFO Ready Interrupt	4-2
		W	Aux/User Data FIFO Ready Interrupt Mask	
0	2	R	First Slice Start Code Detect Interrupt	4-3
		W	First Slice Start Code Detect Interrupt Mask	
64	0	W	Reset Aux Data FIFO	4-19
	[1:0]	R	Aux Data FIFO Status [1:0]	4-19
	[4:2]	R	Aux Data Layer ID [2:0]	4-19
67	[7:0]	R	Aux Data FIFO Output [7:0]	4-21

When the Postparser writes the first byte of auxiliary data into the Aux Data FIFO, the Aux/User Data FIFO Ready Interrupt is set. When the Postparser detects the First Slice Start Code in the bitstream, it sets the First Slice Start Code Detect Interrupt bit. When set and not masked, either bit causes INTRn to be asserted to the host. The host should respond by reading Registers 0 through 4 to determine the cause of the interrupt.

If the host detects that the AUX/User Data FIFO Interrupt is set, it should read the Aux Data FIFO Status bits and the User Data FIFO Status bits to determine which FIFO to read. The status code meanings are shown in Table 8.14.

Table 8.14 Aux Data FIFO Status

Bits 64[1:0]	Status
0b00	Empty
0b01	Data ready
0b10	Full
0b11	Overrun

The status changes from empty to data ready as soon as the first byte is written into the FIFO. Once overrun (0b11) occurs, the status remains at overrun until the host reads the register, and then changes to full until a byte is written in or read out. The Postparser must keep writing auxiliary data bytes into the FIFO as it encounters them in the bitstream. Bytes that overflow from the FIFO are lost. The host is not interrupted on overrun, so it must watch the status bits and empty the FIFO in a timely manner.

Once the Postparser is past the first slice start code of the picture, the remaining data in the picture belongs to the slice, macroblock, and block layers. Usually, all the auxiliary/user data pertaining to the current picture has already been written to the FIFOs when the first slice start code in the picture is encountered. Also at this point, the Video Decoder stalls to synchronize with the display process, giving the host ample time to read the FIFOs.

The three host options are then:

1. mask both interrupts and routinely read the FIFO status.
2. mask the First Slice Start Code Detect Interrupt and read the FIFO status when INTRn is asserted because of an Aux/User Data Ready Interrupt.

   Note: It is important for the host to respond to INTRn and read the interrupt registers. The interrupt bits are cleared when read. If one FIFO sets the data ready interrupt bit and the bit is not read, the other FIFOs cannot generate an INTRn interrupt.

3. mask the Aux/User Data Ready Interrupt and read the FIFO contents when the First Slice Start Code Detect Interrupt occurs.

When the first data byte is written to the FIFO, it is placed in the Aux Data FIFO Output register. At the same time, the Postparser writes the layer ID of the data byte into the Auxiliary Data Layer ID field of Register 64 (see Table 8.15). The host should read the ID first and then read the data. As soon as the data byte is read, the two registers are updated if another unread byte is available in the FIFO.

Table 8.15 Auxiliary Data Layer ID Assignments

Bits 64[4:2]	Layer
0b100	Packet
0b000	Sequence
0b001	Group of pictures
0b010	Picture
0b111	Extension layer (picture or sequence)

When the host writes a 1 to bit 0 of Register 64, the read and write pointers of the Aux Data FIFO are reset and the FIFO's status goes to empty. Any previously unread bytes in the FIFO will be overwritten and lost when new data is written into the FIFO.

8.2.15 User Data FIFO Operation

User Data FIFO operation is very much like Aux Data FIFO operation. In fact, they share some interrupt bits. The complete description, however, is given here for your convenience and not referenced back to the previous section.

The User Data FIFO is used to buffer user data parsed from the bitstream to the host. The User Data FIFO is 128 bytes deep and operates as a circular buffer. Since the decoder parses user data at 8-bits/cycle, the FIFO can fill up very quickly when large amounts of user data are in the channel. The various registers associated with the User Data FIFO are listed in Table 8.16 and described in the text following. More complete descriptions can be found at the page references shown in the table.

Table 8.16 User Data FIFO Registers

Register	Bit(s)	R/W	Name	Page Ref.
0	1	R	Aux/User Data FIFO Ready Interrupt	4-2
		W	Aux/User Data FIFO Ready Mask	
0	2	R	First Slice Start Code Detect Interrupt	4-3
		W	First Slice Start Code Detect Mask	
65	0	W	Reset User Data FIFO	4-20
	[1:0]	R	User Data FIFO Status	4-20
	[3:2]	R	User Data Layer ID	4-20
66	[7:0]	R	User Data FIFO Output	4-21
239	3	R/W	GOP User Data Only	4-63

When the Postparser writes the first byte of user data into the FIFO, the Aux/User Data FIFO Ready Interrupt is set. When the Postparser detects the First Slice Start Code in the bitstream, it sets the First Slice Start Code Detect Interrupt bit. When set and not masked, either bit causes INTRn to be asserted to the host. The host should respond by reading Registers 0 through 4 to determine the cause of the interrupt.

If the host detects that the AUX/User Data FIFO Interrupt is set, it should read the Aux Data FIFO Status bits and the User Data FIFO Status bits to determine which FIFO to read. The status code meanings are shown in Table 8.17.

Table 8.17 User Data FIFO Status

Bits 65[1:0]	Status
0b00	Empty
0b01	Data ready
0b10	Full
0b11	Overrun

The status changes from empty to data ready as soon as the first byte is written into the FIFO. Once overrun (0b11) occurs, the status remains at overrun until the host reads the register, and then changes to full until a byte is written in or read out. The Postparser must keep writing user data bytes into the FIFO as it encounters them in the bitstream. Bytes which overflow from the FIFO are lost. The host is not interrupted on overrun so it must watch the status bits and empty the FIFO in a timely manner.

Once the Postparser is past the first slice start code of the picture, the remaining data in the picture belongs to the slice, macroblock, and block layers. Usually, all the auxiliary/user data pertaining to the current picture has already been written to the FIFOs when the first slice start code in the picture is encountered. Also at this point, the Video Decoder stalls to synchronize with the display process, giving the host ample time to read the FIFOs.

The three host options are then:

1. mask both interrupts and routinely read the FIFO status.
2. mask the First Slice Start Code Detect Interrupt and read the FIFO status when INTRn is asserted because of an Aux/User Data Ready Interrupt.

   Note: It is important for the host to respond to INTRn and read the interrupt registers. The interrupt bits are cleared when read. If one FIFO sets the data ready interrupt bit and the bit is not read, the other FIFO cannot generate an interrupt.
3. mask the Aux/User Data Ready Interrupt and read the FIFO contents when the First Slice Start Code Detect Interrupt occurs.

When the first data byte is written to the FIFO, it is placed in the User Data FIFO Output register. At the same time, the Postparser writes the layer ID of the data byte into the User Data Layer ID field of Register 64 (see Table 8.18). The host should read the ID first and then read the data. As soon as the data byte is read, the two registers are updated if another unread byte is available in the FIFO.

Table 8.18 User Data Layer ID Assignments

Bits 65[3:2]	MPEG Layer
0b00	Sequence
0b01	Group of pictures
0b10	Picture
0b11	Slice

When the host writes a 1 to bit 0 of Register 65, the read and write pointers of the User Data FIFO are reset and the FIFO's status goes to empty. Any previously unread bytes in the FIFO will be overwritten and lost when new data is written into the FIFO.

When the GOP User Data Only bit is set, the decoder parses only DVD-compliant, closed-caption (line 21), user data at the GOP layer to the User Data FIFO. Other user data is discarded by the decoder, reducing the processing overhead of the host in a DVD system significantly. The default value of this bit at startup is 0; all user data of all layers are written to the User Data FIFO.

8.3 Video Decoder Pacing

The Video Decoder Module in the L64020 decodes preparsed data from the Video ES Channel Buffer. Decode should not be started until there is sufficient data in the Video ES Channel Buffer to decode a complete picture without the buffer underflowing. To decode with the minimum amount of frame store memory, picture reconstruction is controlled by the picture display rate (i.e., the vertical sync rate). The decode-to-display pacing is actually performed by comparing the Decode Time Stamp (DTS) and Presentation Time Stamp (PTS) of each picture to the System Clock Reference (SCR). The Video Decoder Module is controlled by the Decode Start/Stop Command bit (Register 246, bit 0). Setting this bit causes the decoder to start the process of reconstructing pictures from the input MPEG-1/MPEG-2 bitstream. The reconstruction proceeds in lock step with the display. This is required to decode with the minimum allowable amount of frame store memory. The rate of reconstruction is therefore controlled by the rate at which the picture is displayed, which again is controlled by the external sync signals (HS and VS signals).

The channel start command causes the Preparser to start accepting data from the external channel interface device. For details on how to control the location of the video channel in SDRAM memory and on selecting the proper stream ID, the reader is referred to Chapter 6, “Channel Interface,” and Chapter 7, “Memory Interface.” After the channel start command is issued and the Video ES Channel Buffer has filled to a sufficient level, the host decode start command is issued. This signals the decoder to begin decoding the bitstream and reconstructing the picture in the frame store memory, which is allocated in SDRAM. The start of video decode can also be achieved by using the Autostart Video function. The following sections explain the operation of the various host registers required to program the functions mentioned previously. The process is illustrated with the help of a time line that shows how these operations should be sequenced in time.

8.3.1 Channel Start/Stop and Status Bits

After powerup or chip reset, the channel is in the reset state. At this time, the various start and end addresses of the channel are assigned in SDRAM by writing to the appropriate registers (See Chapter 6, “Channel Interface,” for more details.). Writing a 1 to the Channel Start/Reset bit (bit 0 of Register 7, page 4-13) causes the channel to start. This results in the Video ES Channel Buffer receiving the MPEG elementary video bitstream from the external channel interface. The host can monitor the status of the channel by reading the Channel Start/Reset/Status bit in Register 7. The host can stop the channel by clearing the Channel Start/Reset/Status.

Note: The Channel Status bit is updated when the decoder acknowledges the channel stop command and not when the host writes a 0 to the bit. The Channel Start/Reset bit is checked regularly by the decoder.

8.3.2 Video Decoder Start/Stop

The actual start of decoding should be delayed from the start of the channel. This is done to allow the Video ES Channel Buffer to fill to a sufficient level so that there is no underflow/overflow of the buffer while actually reconstructing pictures. The host may choose one of the methods described in *ISO/IEC 11172* (MPEG-1) and *ISO/IEC 13818* (MPEG-2) to determine how long this delay should be. See also Section 6.4.2, “Detecting Potential Underflow Conditions in the Video Channel,” page 6-42.

The Postparser in the Video Decoder Module actually starts its parsing operation as soon as there is data in the Video ES Channel Buffer. The Postparser ignores bits from the buffer until it recognizes the first sequence start code. This is done so that the Video Decoder can resynchronize to the data in cases where a program has been changed (video stream ID changed) between sequence start codes. During this time, Picture Start Code Interrupts may occur for each skipped picture before the sequence start code is found.

After finding the first sequence start code, the Postparser then proceeds to read header data for the sequence layer, sequence extensions (if any), group of pictures layer, user data, picture layer, and picture layer extensions (if any). The Postparser stops parsing bits at the first picture data boundary (i.e., it reads the picture header) and waits for the Decode Start Command if it has not yet been issued.

No data is written to the Auxiliary Data FIFO while the Postparser is resynching to the first sequence start code.

The host can start the Video Decoder in one of two ways:

1. Setting the Decode Start/Stop Command bit in Register 246 (page 4-66).
2. Using the video autostart feature. This is done by writing an SCR Compare/Capture Value to Registers 13 through 16, setting the SCR Compare/Capture Mode bits in Register 17 to Compare mode (0b01), and setting the Video Start on Compare bit in Register 19. When the SCR counter catches up to and equals the value in the SCR Compare/Capture registers, the Decode Start Command is issued automatically. This feature can be used to synchronize the start of video decode with the Decode Time Stamps (DTSs) in the video PES headers preparsed from the bitstream.

As soon as the Video Decoder acknowledges the Decode Start Command, it starts parsing the payload data in the Video ES Channel Buffer and sets the Decode Status Interrupt bit in Register 0 (page 4-2).

This causes the INTRn signal to the host to be asserted if it is not masked for this interrupt. The host should then read the interrupt registers to determine the cause of the interrupt.

The host can stop the Video Decoder by issuing a Decode Stop Command (clearing bit 0 in Register 246). The Video Decoder, however, completes reconstruction of the current picture before acknowledging the command; i.e., it stops at the next picture boundary and generates the Decode Status Interrupt. When the Video Decoder stops, the Display Controller freezes on the last field of the currently displayed picture.

Note: A channel stop also causes a Video Decoder stop.

As mentioned previously, the decode/reconstruction process runs in lock step with the display. This ensures that the reconstruction of pictures happens at the same rate as the display (30 frames/second for NTSC and 25 frames/second for PAL) and results in the minimum amount of memory for frame stores.

Figure 8.2 and Figure 8.3 illustrate the process starting from channel start, searching for the first sequence start code, through start of decode, and then show the timing relationship between the reconstruction and the display of pictures.

The host can follow the sequence by reading the First Slice Start Code Detect Interrupt bit, Picture Start Code Detect Interrupt bit, and Begin Active Video (BAV) Interrupt bit as the interrupts occur. These bits are in Registers 0 and 1.

When a picture is encoded as two field pictures, there are two sets of picture start codes and first slice start codes as shown in Figure 8.3.

Figure 8.2 Time Line for Frame Picture

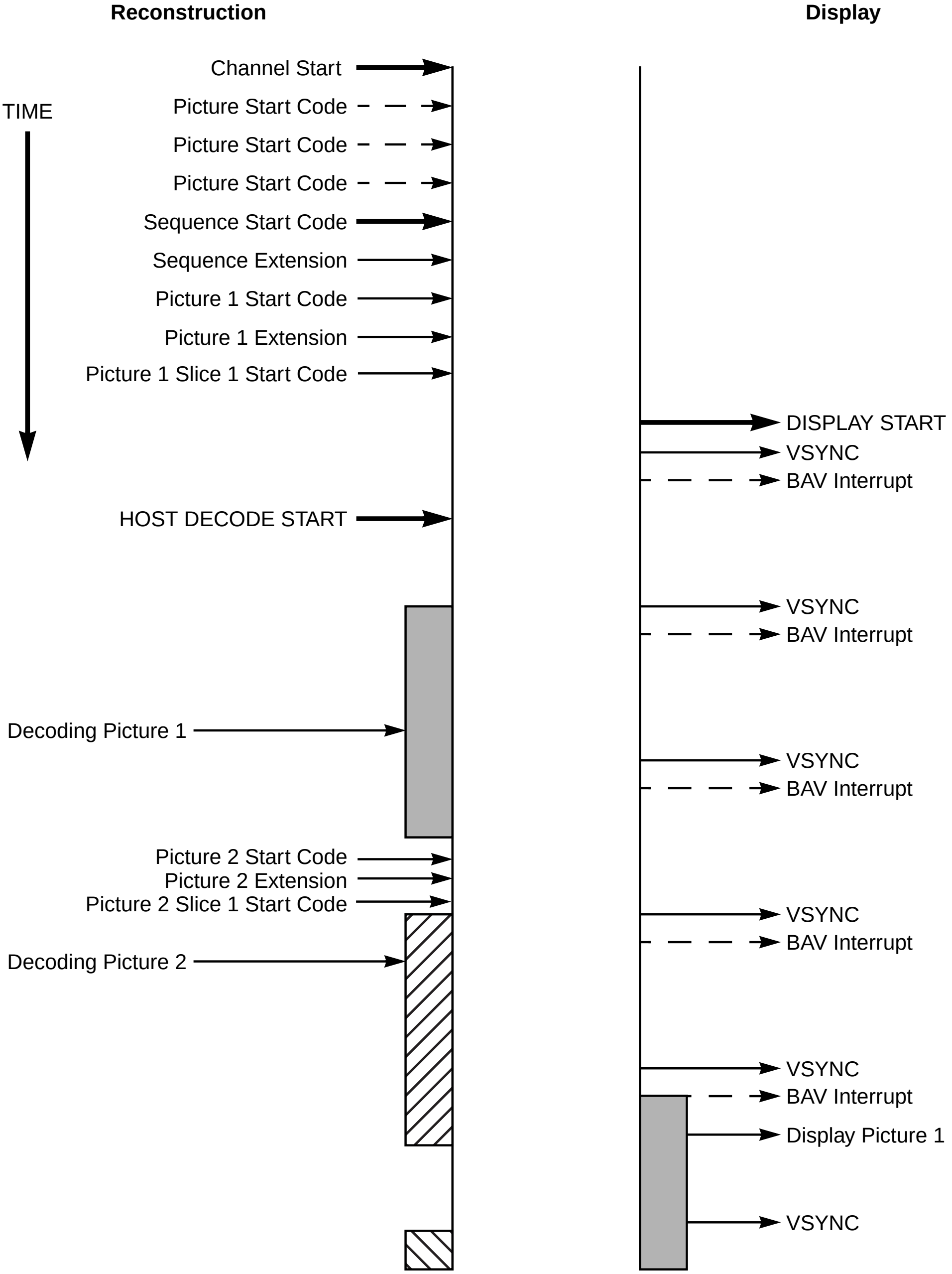
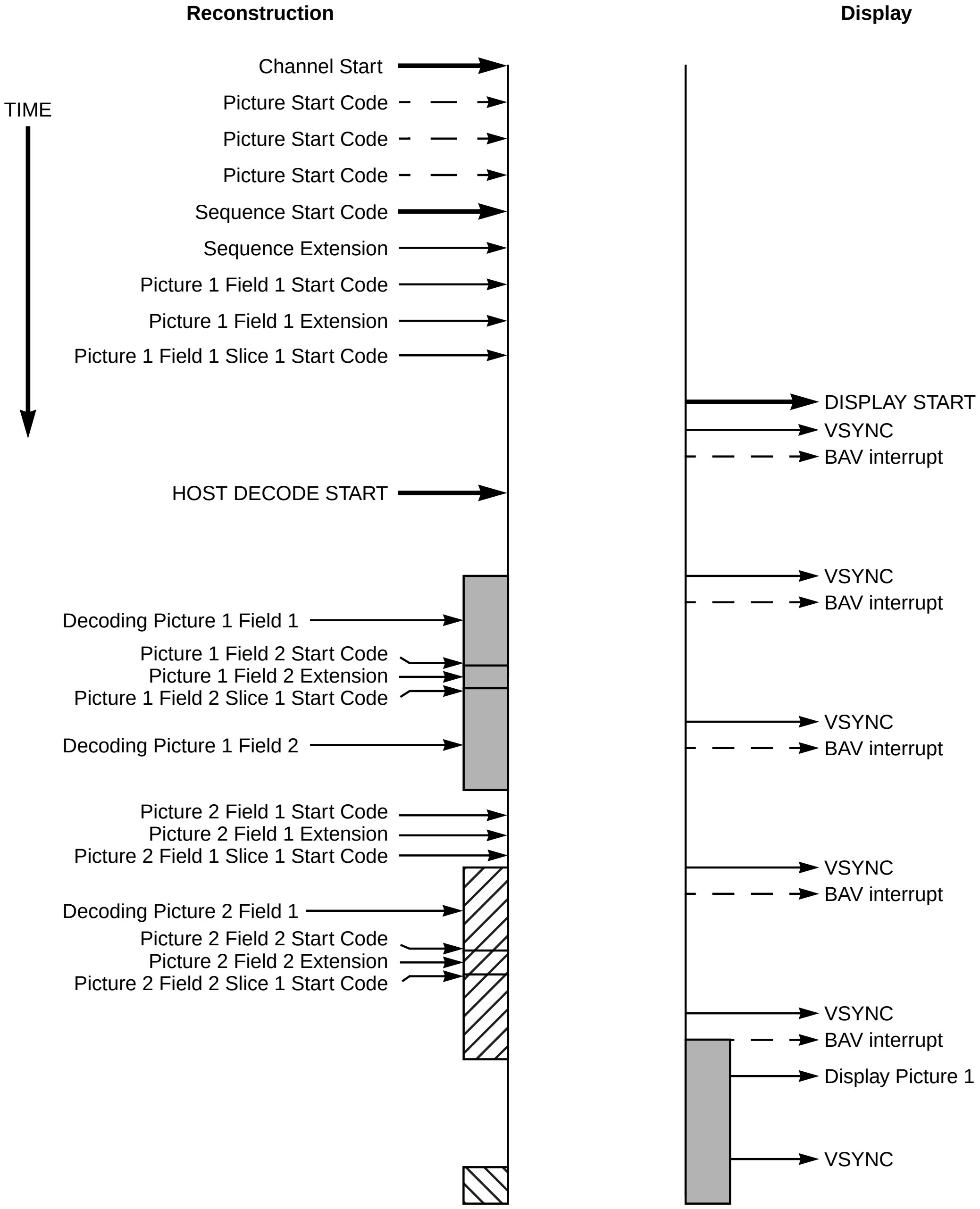


Figure 8.3 Time Line for Field Picture


Reconstruction
Display
TIME
Channel Start
Picture Start Code
Picture Start Code
Picture Start Code
Sequence Start Code
Sequence Extension
Picture 1 Field 1 Start Code
Picture 1 Field 1 Extension
Picture 1 Field 1 Slice 1 Start Code
DISPLAY START
VSYNC
BAV interrupt
HOST DECODE START
VSYNC
BAV interrupt
Decoding Picture 1 Field 1
Picture 1 Field 2 Start Code
Picture 1 Field 2 Extension
Picture 1 Field 2 Slice 1 Start Code
VSYNC
BAV interrupt
Decoding Picture 1 Field 2
Picture 2 Field 1 Start Code
Picture 2 Field 1 Extension
Picture 2 Field 1 Slice 1 Start Code
VSYNC
BAV interrupt
Decoding Picture 2 Field 1
Picture 2 Field 2 Start Code
Picture 2 Field 2 Extension
Picture 2 Field 2 Slice 1 Start Code
VSYNC
BAV interrupt
Display Picture 1
VSYNC

8.4 Frame Store Modes

This section describes how frame stores are organized in the available modes. Frame stores are maintained in the external SDRAM. The Video Decoder decodes macroblocks from the Video ES Channel Buffer and writes them to the frame stores as reconstructed pictures. Depending on the bitstream, there are three store modes:

- Normal or 3-Frame Store Mode for most MPEG streams
- Reduced Memory Mode (RMM) for high-resolution pictures like PAL (720 x 576)
- 2-Frame Store Mode for bitstreams without B pictures.

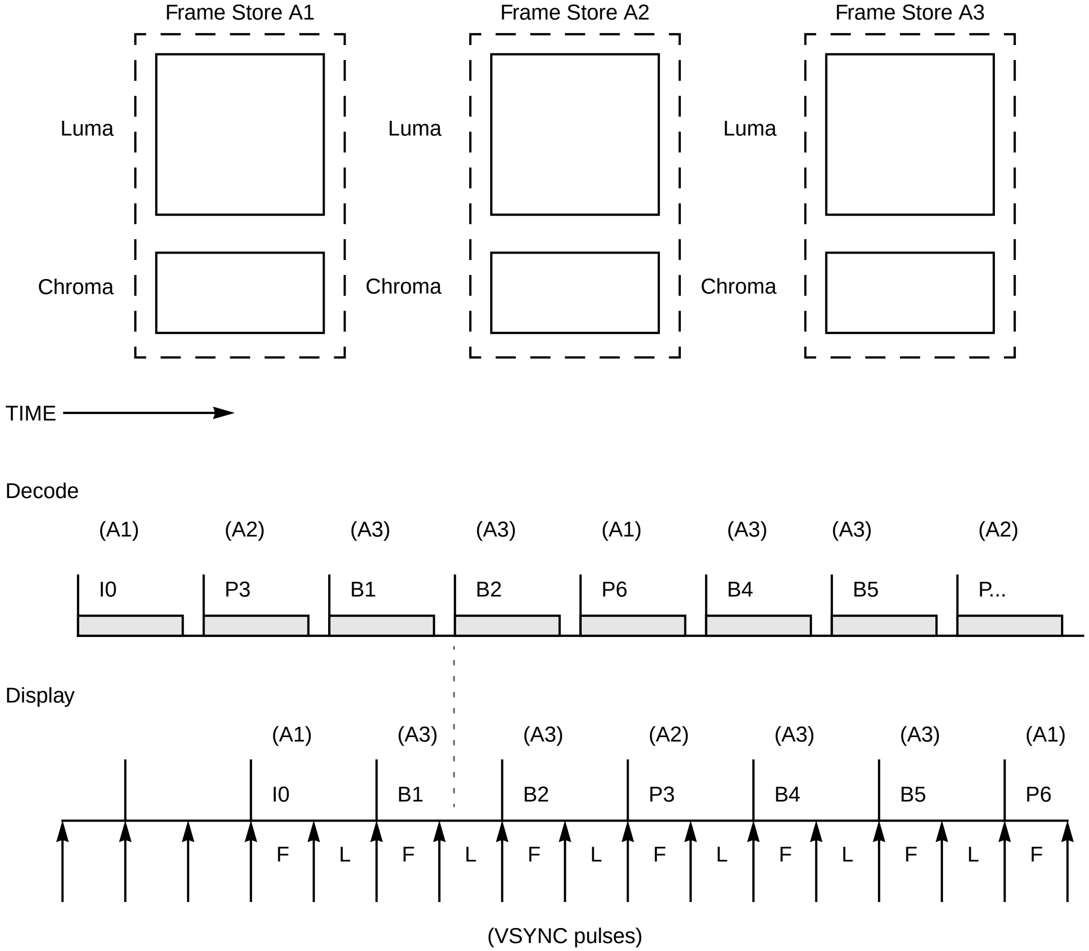
In the usual case, there are two frame stores used for decoding I (intracoded) and P (forward predictive coded) pictures. These are referred to as “anchor frame stores” in the description that follows. In addition, there is a third frame store used for decoding B frames. The B frames are decoded by performing motion compensation using the two anchor frames as references (previous picture and future picture). The sizes and locations of all three frame stores in the external SDRAM are programmable and are set by the host through registers. The sizes of the anchor frame stores are sufficient to hold the entire I or P frame. The size of the third frame store, which is used for decoding B pictures, varies depending on the decoding mode (i.e., normal mode or PAL Reduced Memory Mode). You should understand the various restrictions on the use of these modes. These restrictions are described in the following sections. To aid the host in implementing system control functions, status bits are provided that indicate which frame store is currently being used for picture reconstruction and which frame store is currently being used for display.

The on-chip Display Controller interfaces to the external SDRAM and displays the pictures that the Video Decoder reconstructs.

8.4.1 Normal (3-Frame Store) Mode

Figure 8.4 depicts the organization of the three frame stores. It also indicates the timing relationship between the reconstruction of frames from the incoming MPEG-1/MPEG-2 bitstream and the display of those frames by the Display Controller.

Figure 8.4 Frame Store Organization in Normal Mode



Note:

- Names inside parentheses indicate frame store being used for decode or display.
- F = First field.
- L = Last field.

Figure 8.4 assumes that encoded frame pictures are being decoded for display in interlaced mode. The reconstruction process is synchronized with the display. For example, reconstruction of frame B2 from the bitstream does not begin until the first field of frame B1 has been displayed and display of the last field of B1 has begun. This is done to process B frames using only one frame store (A3 in the example).

In trick modes, where it might be necessary to change the start addresses of the frame stores “on the fly,” the start address of the frame store being used for reconstruction is read in just before reconstruction of the frame store is about to begin. The host should ensure that the start address of the frame store is valid before the last field of the picture being displayed starts to display.

The start addresses of frame stores A1, A2, and A3 are programmed by the host using the registers listed in Table 8.19.

Table 8.19 Frame Store Base Address Registers

Frame Store	Address[1]	Register	Page Ref.
A1	Anchor Luma Frame Store 1 Base Address [7:0]	224	4-57
	Anchor Luma Frame Store 1 Luma Base Address [15:8][1]	225	
	Anchor Chroma Frame Store 1 Base Address [7:0]	226	4-57
	Anchor Chroma Frame Store 1 Base Address [15:8]	227	
A2	Anchor Luma Frame Store 2 Base Address [7:0]	228	4-57
	Anchor Luma Frame Store 2 Base Address [15:8]	229	
	Anchor Chroma Frame Store 2 Base Address [7:0]	230	4-58
	Anchor Chroma Frame Store 2 Base Address [15:8]	231	
A3	B Luma Frame Store Base Address [7:0]	232	4-58
	B Luma Frame Store Base Address [15:8]	233	
	B Chroma Frame Store Base Address [7:0]	234	4-58
	B Chroma Frame Store Base Address [15:8]	235	

[1] SDRAM addresses at 64-byte boundaries.

8.4.2 Reduced Memory Mode

In RMM, the anchor frames are reconstructed as described in the normal mode, but the B frame reconstruction uses less than a full frame store. This mode is used for decoding high-resolution pictures, such as for PAL (720 x 576), using only 1M x 16 bits of SDRAM.

The host enables RMM by setting the Reduced Memory Mode bit in Register 248 (page 4-67). This register is read by the Video Decoder only when it encounters an anchor frame (I or P picture). The host determines the amount of memory allocated to the B frame store by writing a value into the Number of Segments in RMM field in Register 289 (page 4-77). Each segment consists of a frame store for eight lines of the frame. The minimum and maximum number of segments can be calculated using the following formulas:

$$\text{Min NumSegments} = \frac{\text{Total Lines}}{8} + 4$$

$$\frac{\text{Total Lines}}{8} - 1 \leq \text{Max Num Segments} \leq 54$$

For a full-size PAL image, the minimum number of segments is 40. If sufficient SDRAM memory is available, the recommended number of segments for adequate performance is 44. The maximum number of segments for a PAL image is 54.

Allocating more segments than the minimum will always boost decoder performance by allowing “decode ahead” without waiting for the display process to free up segments. Allocating extra segments is especially recommended for applications, like letterbox display, which demand higher decoder performance.

There are certain restrictions which must be followed when RMM is used. SDRAM memory is dynamically allocated in this mode, and the SDRAM memory used to reconstruct the lines of the first field are reallocated to reconstruct other lines once the first field lines have been displayed. 3:2 pulldowns (Repeat_first_field parameter in picture coding extension) will be processed while in RMM. However, it should be noted that the actual display in this case will be; First Field, Last Field, Last Field. Thus, the Repeat First Field is actually implemented as a Freeze Last Field during RMM while displaying B-pictures.

Another restriction is the display modes that are allowed with RMM. Only the following modes may be used in conjunction with RMM:

- Display Mode 4: Interlaced Chroma Field Repeat and No Filter
- Display Mode 5: Interlaced Chroma Field Repeat and Filter
- Display Mode 6: Interlaced Chroma Line Repeat
- Display Mode 7: Interlaced Chroma Line Repeat and Filter
- Display Mode 8: Interlaced 0.75 Letterbox Filter

- Display Mode 10: Interlaced Repositioning
- Display Mode 11: Interlaced 0.5 Letterbox Filter

For more information on display modes, see Section 10.6, “Display Modes and Vertical Filtering.”

Display modes 4 and 5 above use Chroma Field Repeat. To achieve these modes, the chroma component of the B pictures should be allocated a full frame store even though the luma component uses less than a full frame store in RMM. Thus the chroma component effectively does not use reduced memory mode if the display mode is set to 4 or 5. Note that in this case, the number of segments programmed in Register 289 bits [6:1] indicate the number of segments used for luma.

8.4.3 Two-Frame Store Mode

If B pictures are not present in the bitstream, the decoder can be operated in a two-frame store mode, i.e., SDRAM memory needs to be allocated only to anchor frame stores A1 and A2. Note that, even if Low_delay is set in the bitstream, the delay between decoding a picture and displaying it is still three field display times.

8.4.4 Decode and Display Frame Store Status Indicators

The host has access to two registers that indicate which frame store is currently being used for reconstruction and which is currently being used for display. The coding for the Current Decode Frame bits [5:4] and the Current Display Frame bits [3:2] in Register 238 is identical and is shown in Table 8.20.

Table 8.20 Current Decode/Display Frame Bits Coding

Current Decode/Display Frame	Description
0b00	I/P Anchor 1 (A1)
0b01	I/P Anchor 2 (A2)
0b10	B (A3)
0b11	Reserved

The Current Decode Frame bits are updated after the last field of the currently displayed frame starts displaying. The Current Display Frame bits are updated at the first vertical sync pulse indicating the start of display of the first field in the frame.

Note: The Current Display Frame bits are not valid in the Display Override mode. Refer to page 4-68 for a description of Display Override Mode.

8.5 Trick Modes

The L64020 supports a variety of trick modes that are useful for implementing various functions required in a DVD system. These include skipping frames, repeating frames, avoiding video channel underflow by using a programmable “panic threshold,” Rip Forward mode, support for Broken Link/Open_GOP response, Search For Next GOP, Display Override Mode, Reconstruction Force Rate Control, and Single Still Picture display. The following sections describe each of these functions and the restrictions associated with them.

8.5.1 Skip Frame

Three bits in Register 236 control the skip frame feature. Bits [1:0] (page 4-59) let the host select skip frame mode and the type of frame to skip (see Table 8.21). When set, bit 2 in the register causes continuous skipping. When bit 2 is cleared, only one frame is skipped.

Table 8.21 Video Skip Frame Modes

Skip Frame Bits	Skip Frame Mode
0b00	none (normal play)
0b01	skip B frame
0b10	skip P or B frame
0b11	skip any frame

When the host selects a single frame skip, the internal microcontroller clears bits [1:0] after the frame is skipped to return to normal play. When the host selects continuous skip, the selected frames are skipped until the host clears either bits [1:0] or bit 2. If the host clears bit 2, one more frame is skipped and the internal microcontroller clears bits [1:0]. All three bits are read/write so the host can check the current skip mode status.

To skip a frame, the Postparser transfers the picture header information to the Auxiliary Data FIFO and reads the rest of the picture bytes out of the Video ES Channel Buffer at top speed without handing them off to the IDCT Pipeline. When two or more consecutive frames are skipped, the Postparser still searches for picture start codes, generates an interrupt at each, and transfers the header information of each to the Auxiliary Data FIFO.

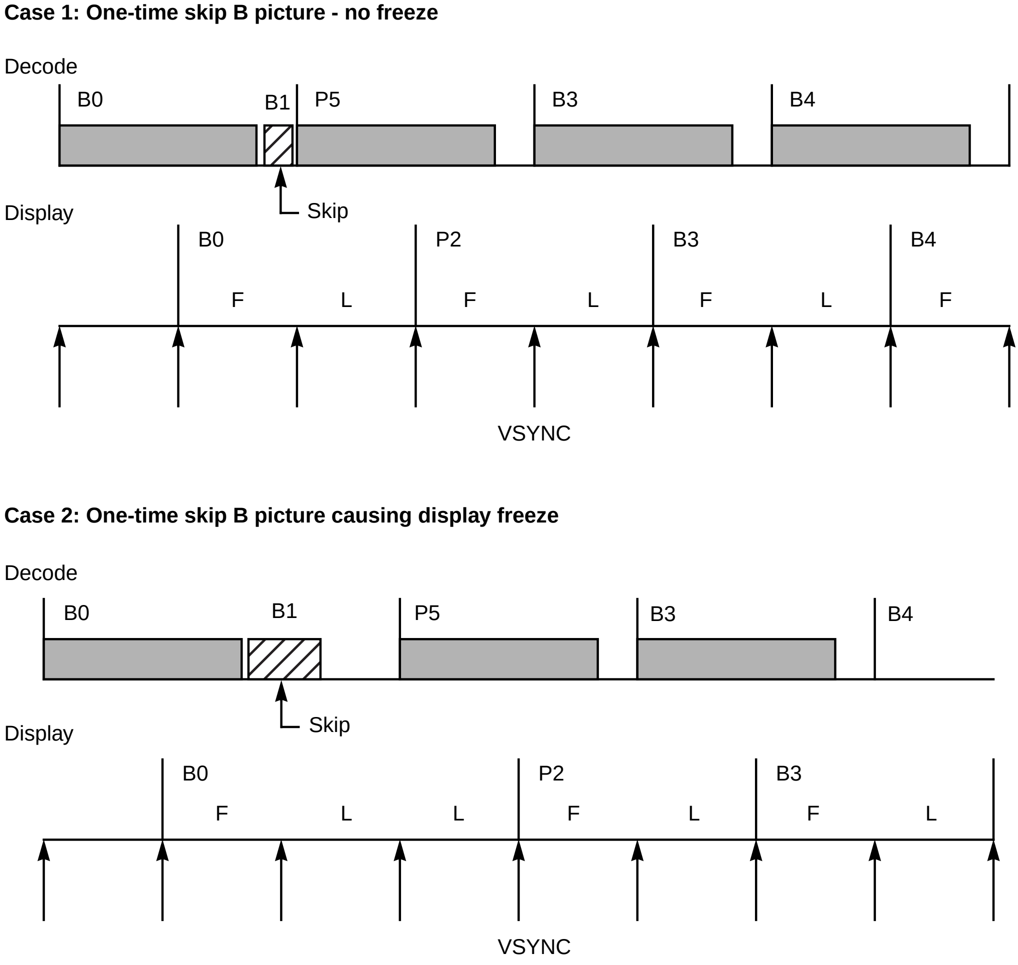
When two or more consecutive frames are skipped, the display is frozen on the last field of the picture before the skip until the next unskipped frame is displayed. Figure 8.5 shows two cases of decoder operation for a single frame skip.

In case 1, the host ordered a B frame skip while the first of two B frames (B0) was being decoded. The Postparser skips by frame B1 in time to decode frame P2. Since the B0 decode and B1 skip fit into one frame time, the display continues without freezing.

In case 2, the B1 frame skip time overlaps the next vertical sync and pushes the decode time for frame P2 to the next vertical sync. The display is automatically frozen on the last field of B0 for an extra field display time. This can occur when all of frame B1 is not in the Video ES Channel Buffer at the time of the skip, slowing down the Postparser. The host can avoid this situation by reading/managing the number of pictures in the Video ES Channel Buffer (refer to Section 6.4.2, “Detecting Potential Underflow Conditions in the Video Channel,” page 6-42 and Section 8.5.3, “Channel Buffer Underflow Panic Repeat,” page 8-39).

The host must issue the skip command before the picture start code interrupt for the picture that is to be skipped. Once the decoding of a frame starts, it cannot be skipped.

Figure 8.5 Single Skip with and without Display Freeze



Case 1: One-time skip B picture - no freeze

Case 2: One-time skip B picture causing display freeze

Note:

- F = First field.
- L = Last field.

8.5.2 Repeat Frame

The repeat frame feature is controlled by two bits in Register 237 (page 4-60). When the host clears the Video Continuous Repeat Frame Mode bit (bit 1) and sets the Video Repeat Frame Enable bit (bit 0), the Video Encoder repeats the last field of the frame currently being decoded twice. That is, its first field is displayed once and its last field is displayed three times in succession. This is shown in Figure 8.6. After the Video Encoder accepts the command, it automatically clears the Video Repeat Frame Enable bit.

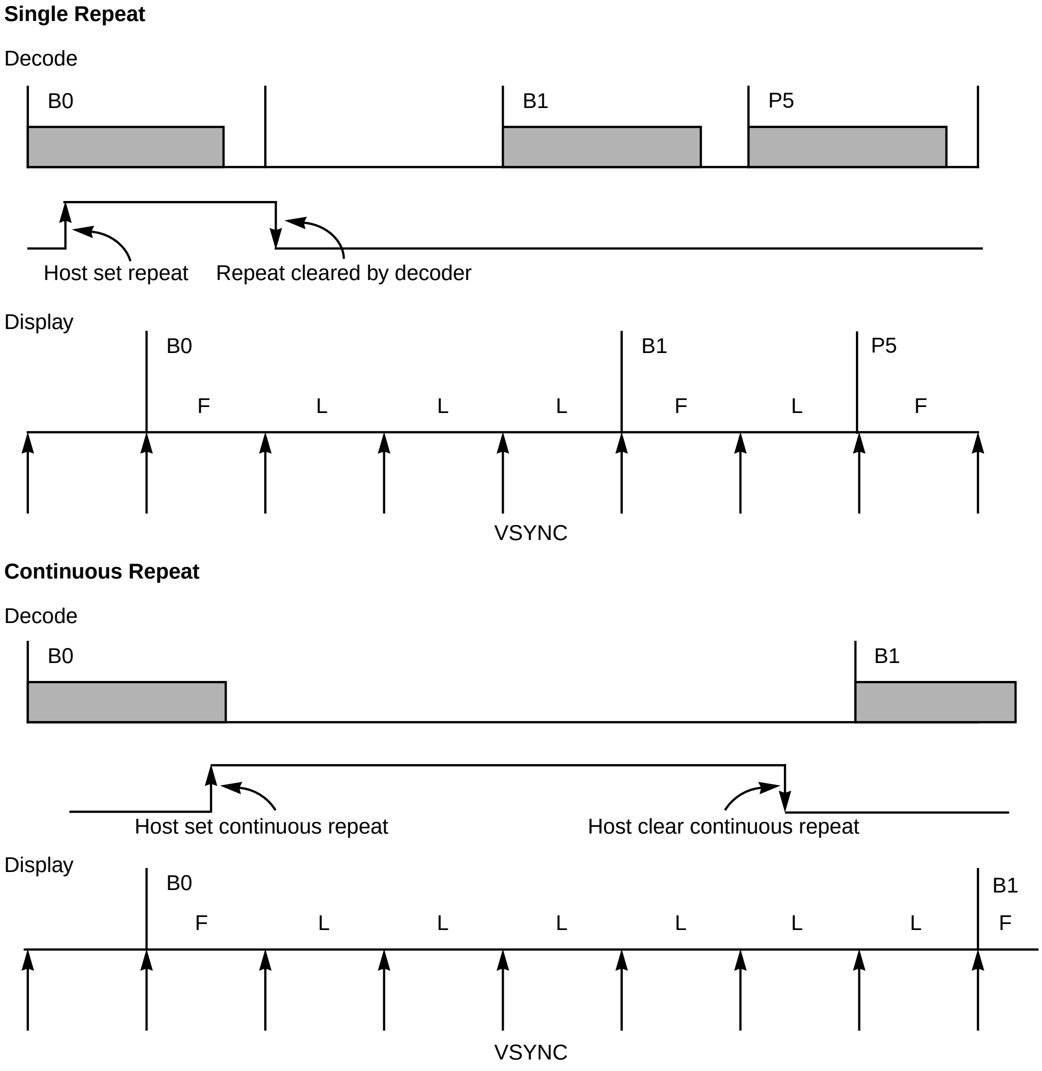
If the host sets both bits, the last field of the frame being decoded is continuously repeated and the Video Decoder is paused. If the repeat lasts over several frames, the Video ES Channel Buffer could overflow unless it also is paused or stopped by the host.

> Note: Since the Video Decoder is paused, picture start code interrupts are not generated and no data is read into the Auxiliary Data FIFO during the repeats.

The host can stop the repeat by clearing either or both bits. If it clears the continuous mode bit only, the field is repeated two more times and the Video Decoder clears the repeat mode bit. If the host clears the repeat enable bit only, the currently displayed frame is completed (by repeating the last field one more time, if necessary) and the next decoded frame is displayed.

If only one field is repeated, the fields are then out of sync with the even/odd interlacing. This condition is automatically corrected if the host sets the Automatic Field Inversion Correction bit in Register 279 (page 4-74).

Figure 8.6 Frame Repeat Modes



Note:

- F = First field.
- L = Last field.

8.5.3 Channel Buffer Underflow Panic Repeat

When this feature is enabled and the decoder detects that the Video ES Channel Buffer is in danger of underflowing, it automatically freezes the display on the last field of the currently displaying picture. The freeze is automatically removed when the channel buffer has filled to an adequate level. During the panic condition, the decoder pauses; it does not request any bytes from the channel for decoding.

To enable this feature, the host sets the Video Numitems/Pics Panic Mode Select bits in Register 69 (page 4-23) to either 0b01 to select number of items (64-bit words) or 0b10 to select number of pictures. The host must then enter an item or picture threshold value in Registers 134 through 136 (page 4-39). The Channel Buffer Controller compares the number of items or pictures in the Video ES Channel Buffer with the programmed threshold value. If the actual number falls below the threshold, a “panic” signal is sent to the Video Decoder. The Video Decoder responds by repeating a frame to let the Video ES Channel Buffer refill above the threshold. The panic signal is sampled by the decoder just before reconstruction of the picture is about to begin. Note that the decoder pauses for the panic signal to clear even if the host has commanded the decoder to skip a frame.

Note: This operation can violate the correct Video Buffering Verifier (VBV) model operation and is generally used in trick modes when the VBV is invalid.

8.5.4 Rip Forward Mode

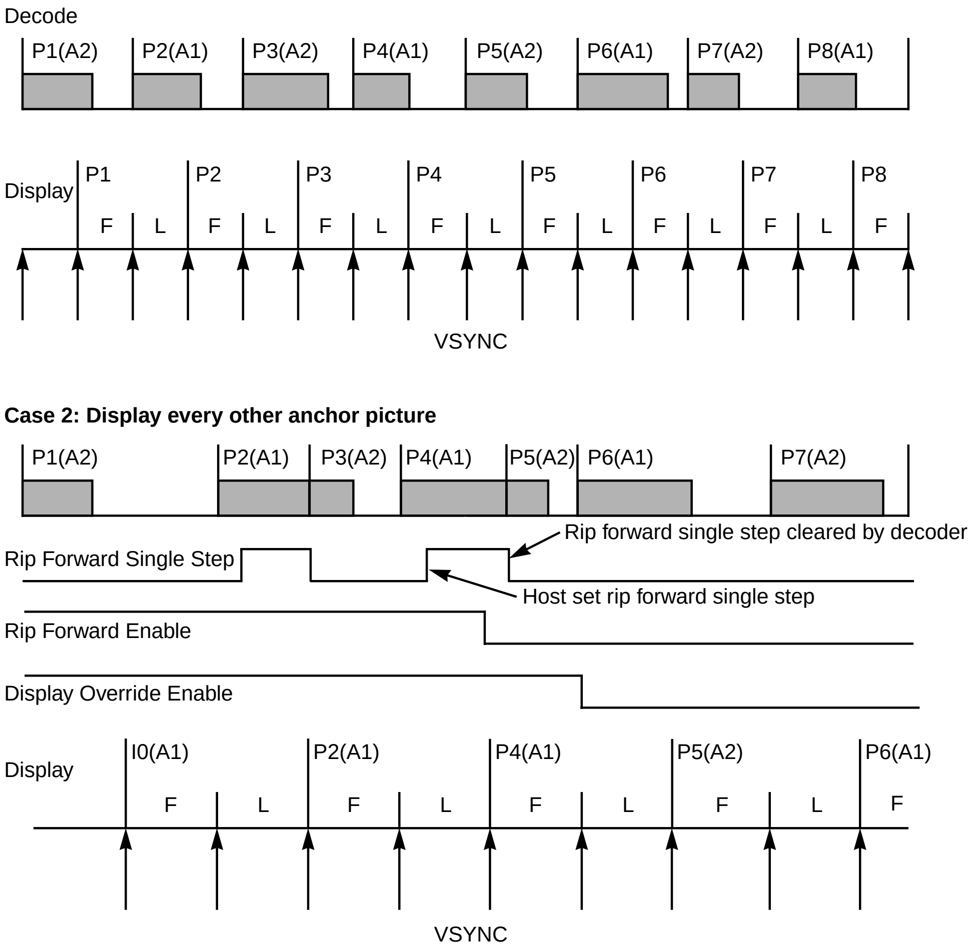
Setting the Rip Forward Mode Enable bit in Register 238 (page 4-61) enables the Rip Forward Mode. In this mode, the decoder processes pictures as fast as it can without regard to the status of the display, i.e., the rate control for the decode with respect to the Vertical Sync of the display is turned off. The rate control for the decode is governed by the Rip Forward Display Single Step Command bit in Register 238 (page 4-62). The on-chip microcontroller monitors the single step bit after it receives both a picture start code and the first slice start code, and has processed the picture header. The decode for that picture only proceeds when the single step bit is set. The single step bit is cleared on reading by the decoder.

The Rip Forward Mode is intended to be used in applications where not every picture that is decoded needs to be displayed. The picture to be displayed is specified in separate registers. These registers are read by the Video Interface. See the Display Override Mode bits in Register 265 (page 4-68) and the Override Display Start Address in Registers 285, 286, 287, and 288 (page 4-77).

Also, during the Display Override Mode, the host must specify the Override Picture Width in Register 283. The Pan and Scan from Bitstream bit in Register 279 must be cleared, and pan-and-scan values (if any) must be supplied by the host in Registers 276 through 281. The 3:2 Pulldown from Bitstream bit in Register 275 must be cleared, and the display must be specified completely by the host using the Host Repeat First Field and Host Top Field First bits in Register 275.

After the Rip Forward Mode is turned off, the reconstruction of the next picture begins at the boundary of the following even display field. This causes resynchronization between the reconstruction and the display process. Figure 8.7 shows an example of Rip Forward Mode with Display Override.

Figure 8.7 Setting Up Rip Forward/Display Override Command



Case 1: Normal Play

Case 2: Display every other anchor picture

Note:

- Names inside parentheses indicate frame store being used for decode or display.
- F = First field.
- L = Last field.

8.5.5 Broken Link/Open GOP

The L64020 automatically skips all B pictures before the first I picture in an open Group of Pictures (GOP) if the Broken Link bit in the bitstream is set. The host can force this feature in an open GOP regardless of the bitstream broken-link bit by setting the Host Force Broken Link Mode bit in Register 239 (page 4-63). This bit is automatically cleared by the Video Decoder when it encounters the next GOP after skipping.

The host must set the force bit before the next GOP header is encountered in the bitstream in order for this command to apply to the next GOP. The Video Decoder stays synchronized with the display because only one B picture is skipped per frame display period.

8.5.6 Search for Next GOP/Sequence Header

When the host sets the Host Search Next GOP/Seq Command bit in Register 240 (page 4-65), the Video Decoder stops decoding and skips all header and data bytes until it recognizes the next Sequence or GOP header, whichever comes first. When it finds the header, the Video Decoder clears the search bit.

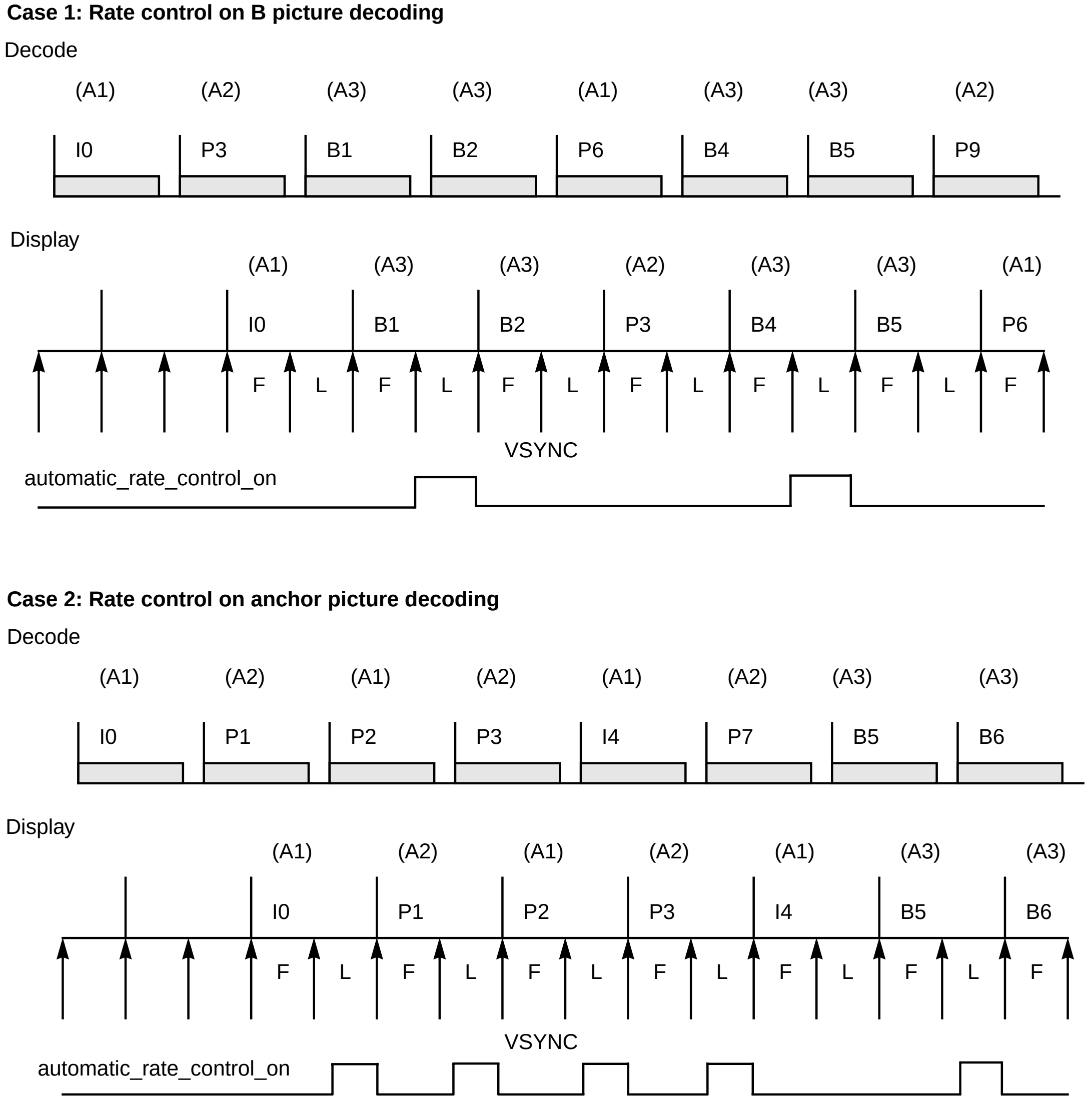
Since the Video Decoder reads this bit just before starting reconstruction of each frame, the display automatically freezes on the last field of the currently displayed or previous frame if the bit is set. Note that, unlike the skip feature, picture start code interrupts are generated but no header data is written to the Auxiliary Data FIFO for the pictures that are skipped.

8.5.7 Reconstruction Force Rate Control

With a 3-frame store SDRAM storage scheme, all B pictures are reconstructed to frame store A3. The display of these B pictures takes place approximately one field time after their reconstruction starts. When decoding and displaying a series of consecutive B pictures, steps need to be taken to make sure that the contents of A3 are not overwritten by reconstruction before they are displayed. The decoder is capable of automatically using its internal rate control mechanism to control the rate of reconstruction. The internal rate control stalls the decoder if reconstruction is about to overwrite SDRAM contents that have not been displayed. Specifically, the automatic rate control is turned on during the first field display time whenever the frame store for reconstruction and display happen to coincide. When the decoder is set for Rip Forward Mode, the internal automatic rate control is turned off since the intention is to reconstruct pictures as fast as possible. Figure 8.8 shows examples when rate control is applied on B picture and on anchor picture reconstruction.

The host can force rate control on for all pictures by setting the Force Rate Control bit in Register 239 (page 4-64). In this mode, the decoder always checks for reconstruction overrunning display based on the lines currently displayed. When this bit is cleared (normal mode), the reconstruction overrunning display check is only performed if the decoder is reconstructing over the top of the same physical frame store that is currently being displayed (i.e., the last field of the frame store is being displayed and the next picture is being reconstructed in the same frame store). It is recommended that this bit be set only during trick modes where the host software is changing the address of the frame stores in SDRAM dynamically. An example is Display Override Mode where the host is changing the frame store pointers (A1, A2, and A3) at the start of every picture reconstruction to point to different physical locations in SDRAM. In modes like this, the host should set the Force Rate Control bit since the Video Decoder may not automatically detect that rate control is needed. During the Rip Forward Mode, the internal rate control is turned off. If the host displays selective reconstructed pictures, it should force rate control on for certain fields. Figure 8.9 shows an example of how the Force Rate Control bit is used in Rip Forward Mode.

Figure 8.8 Automatic Rate Control


Case 1: Rate control on B picture decoding

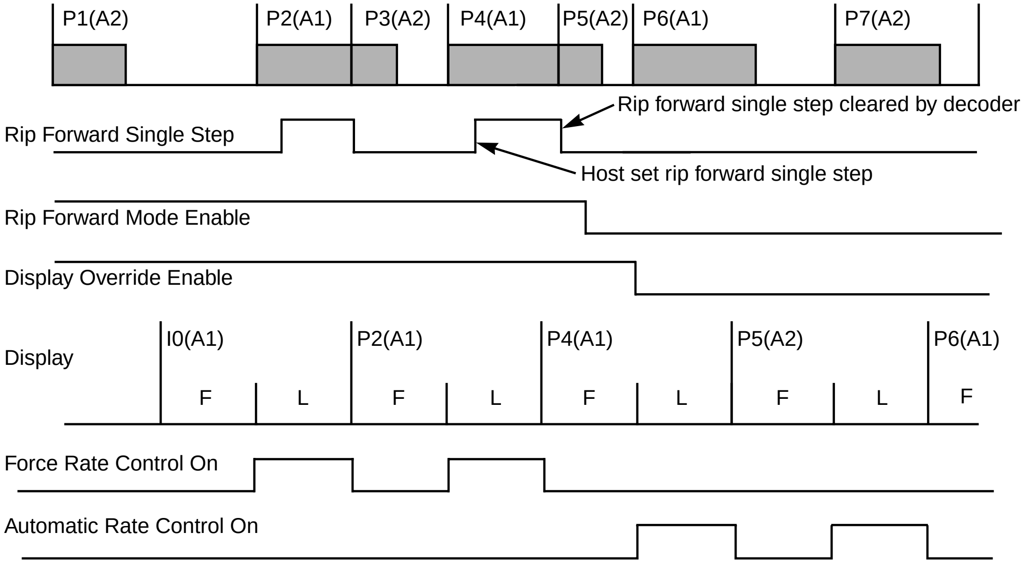
Case 2: Rate control on anchor picture decoding

Decode
(A1) (A2) (A3) (A3) (A1) (A3) (A3) (A2)
I0 P3 B1 B2 P6 B4 B5 P9
Display
(A1) (A3) (A3) (A2) (A3) (A3) (A1)
I0 B1 B2 P3 B4 B5 P6
F L F L F L F L F L F L F
VSYNC
automatic_rate_control_on
Decode
(A1) (A2) (A1) (A2) (A1) (A2) (A3) (A3)
I0 P1 P2 P3 I4 P7 B5 B6
Display
(A1) (A2) (A1) (A2) (A1) (A3) (A3)
I0 P1 P2 P3 I4 B5 B6
F L F L F L F L F L F L F
VSYNC
automatic_rate_control_on

Note:

- Names inside parentheses indicate frame store being used for decode or display.
- F = First field.
- L = Last field.

Figure 8.9 Using Force Rate Control in Rip Forward Mode



Note:

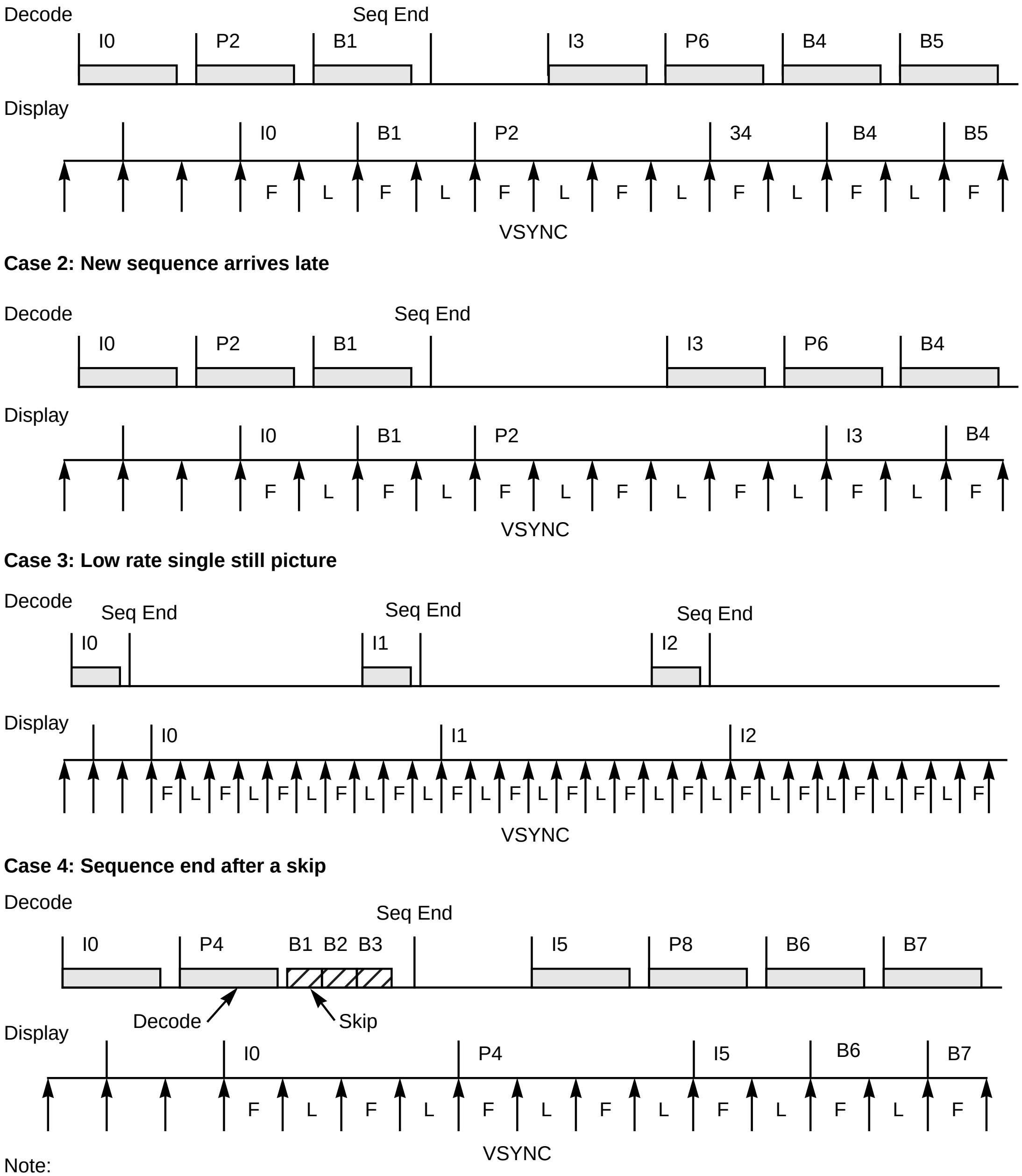
- Names inside parentheses indicate frame store being used for decode or display.
- F = First field.
- L = Last field.

8.5.8 Sequence End Processing

When the Video Decoder detects a sequence end code in the bitstream, it sets the Sequence End Code Detect Interrupt bit in Register 0 (page 4-3) and this asserts the INTRn signal to the host if the interrupt bit is not masked. After a sequence end code, the Video Decoder displays any decoded but undisplayed anchor pictures (I or P) and freezes the last frame on the display until the next sequence start code is detected. This may be valuable information to the host software in certain situations, such as displaying still images.

With the 3-frame store scheme, an anchor picture should not be displayed until the next anchor picture is encountered. This causes at least a 3-field display delay between an anchor picture’s reconstruction and its display. Case 1 in Figure 8.10 shows a new sequence starting right after a sequence end code. At the sequence end code, frame P2 is already decoded and waiting to be displayed. The Video Decoder displays it.

Figure 8.10 Example of Sequence End Processing



Case 1: New sequence arrives right away

Case 2: New sequence arrives late

Case 3: Low rate single still picture

Case 4: Sequence end after a skip

Note:

- F = First field.
- L = Last field.

Frame I3 of the new sequence gets decoded but has to be kept in the frame store until the first field of frame P6 is decoded. So, the Video Decoder repeats frame P2.

In case 2, the new sequence does not arrive until some time after the sequence end code, so frame P2 has to be repeated several times. Case 3 shows a bitstream with single pictures in the sequences and intentional delays between sequence ends and starts. The single pictures are continuously repeated between sequences.

Case 4 shows the situation where the host has ordered a continuous skip of B pictures and skips three of them immediately before a sequence end code.

Since there is likely to be a delay between a sequence end code and the next sequence start, it is practical to display the last anchor picture at the sequence end instead of waiting for the first anchor picture in the new sequence. In Rip Forward Mode, the decoder stalls at the sequence end code until the Rip Forward Single Step Command bit in Register 238 (page 4-62) is set.

As described, the last anchor picture in a sequence is displayed after the sequence end code is detected and is treated as a still picture until the next sequence start code. If the host sets the Ignore Sequence End bit in Register 239 (page 4-64), the last picture in the current sequence is not displayed until after the next sequence start code. This feature is useful when the delay between sequences is short and adding the extra display time could interfere with the synchronization of video and audio processing.

8.6 Error Handling and Concealment

The L64020 can detect a variety of errors in the bitstream. The decoder tries to conceal any errors found. This is usually done with the help of concealment motion vectors if they are present in the bitstream. Concealing errors helps minimize their effects and helps the decoder resynchronize to the bitstream as soon as possible.

8.6.1 Error Conditions Detected

The following error conditions can be detected by the Video Decoder:

1. Variable Length Code (VLC) in error.
2. Context error, i.e., a parameter in the bitstream that is not consistent with the context or an illegal value in the bitstream.
3. Unexpected start code. A start code in the MPEG syntax is defined as a string of 23 0s, a 1, and the Start_code_identifier. Start codes are used to separate and identify the various layers of syntax. If the decoder is expecting a certain parameter in the bitstream in a given layer of syntax and a transition to another layer is not expected, then the presence of a start code at that point in the parsing of the bitstream is treated as an error.
4. Run-level errors. Inconsistent run-level variable length codes in the block layer of the syntax (IDCT) are detected and flagged.

8.6.2 Recovery Mechanisms

Most of the error conditions listed previously occur inside the slice layer. The recovery mechanism consists of searching for the next slice start code or possibly a header at a higher level of syntax than the slice layer. This ensures that the decoder resynchronizes with the bitstream. For the portions of the picture that receive an erroneous bitstream or have missing data, the decoder performs motion compensation using concealment vectors (if they are present in the bitstream) to try to conceal the errors. When the MPEG-2 encoder keeps slices relatively small, the additional slice start codes provide a robust error recovery mechanism.

The host can command the Video Decoder to ignore any concealment vectors in the bitstream by setting the Concealment Copy Option bit in Register 239 (page 4-64). In this mode, the Video Decoder copies from the previously decoded valid picture. This bit is cleared at reset or power-up.

Chapter 9
SPU Decoder Module

This chapter describes the operation of the SPU Decoder module. This chapter contains the following sections:

9.1 Introduction

The SPU Decoder module decodes SPU streams as defined in the *DVD Specifications for Read-Only Disc*. The SPU Decoder, shown in Figure 9.1, consists of the following main submodules:

- SDRAM Address Generator and FIFO Controller
- SPU Controller
- PXD Run Length Decoder

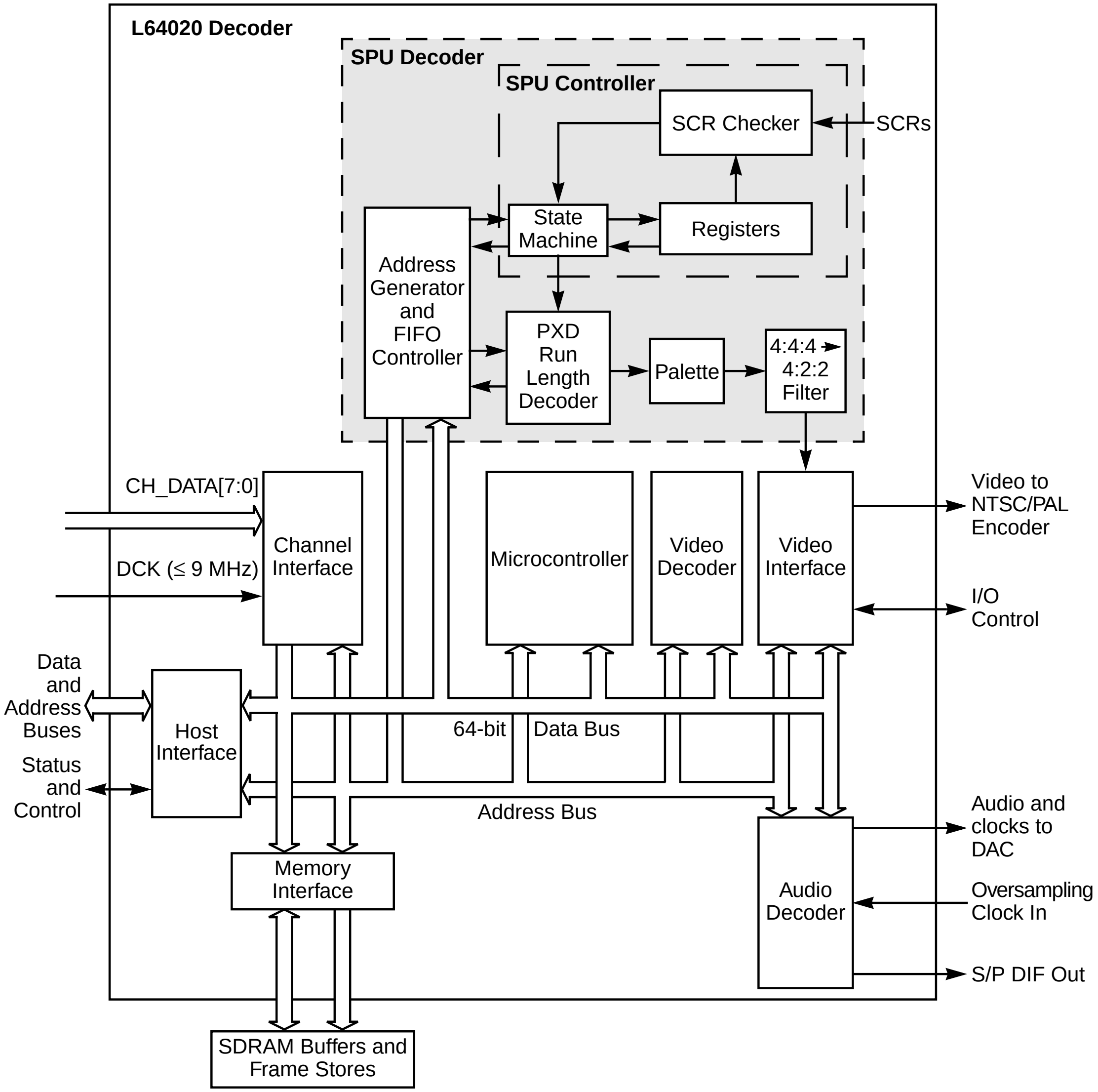
The SDRAM Address Generator and FIFO Controller controls both the SDRAM buffer pointers and the on-chip FIFO pointers. There are two FIFOs, the Pixel Data (PXD) FIFO and the Command FIFO. The SPU

Controller contains a State machine, SCR Checker, and Registers. The Sate Machine analyzes each SPU command and controls the entire SPU decoding schedule. The SCR Checker determines the command execution timing based on the SCR counter. The Pixel Data (PXD) Run-length Decoder decodes the pixel data compressed by run-length encoding. A palette table and some control registers are also included in the PDX Run-length Decoder.

The SPU Decoder has the following features:

- Full automatic decode and sync capability. The following operations require host intervention; decode start/stop, channel buffer control, SCR maintenance, and error handling. All other operations related to decoding and SCR synchronization are handled automatically by the SPU Decoder.
- Trick play, such as slow forward and pause, are controlled by the host through maintenance of the SCR counter.
- Automatic error handling and generation of hardware interrupts at error detection.
- Highlight function. The SPU Decoder allows the host to write highlight information into dedicated on-chip registers.

Figure 9.1 SPU Decoder Block Diagram



9.2 Normal Play Sequence

Normal Play is achieved by initializing the SPU registers and other registers in the L64020 and starting the Channel Interface. When an SPU start code is detected, the SCR counter is updated by the SCR value from the DVD stream, and the SPU's color palette is initialized by color information in the DVD PCI packet. Once the SPU Decoder start bit is set, the SPU waits for the first unit to fill the SPU channel buffer. The SPU then extracts the first PTS, waits until the SCR counter catches up to the PTS, and starts decoding while keeping synchronization with video display timing. In general, error handling and recovery is done automatically.

The typical start-up sequence is described in the following paragraphs.

Step 1. Initialize the L64020.

The host must initialize the SPU register bits and fields listed in Table 9.1.

Table 9.1 Host SPU Register Initialization

Name	Register/Bits	Page Ref.
Video PES Header/SPU Channel Buffer Start Address	80 and 81	4-26
Video PES Header/SPU Channel Buffer End Address	82 and 83	4-26
Reset Video PES Header/SPU Channel Buffer	68 bit 2	4-22
SPU Substream ID	146 bits [4:0]	4-43
SPU Stream Select Enable	146 bit 5	
SPU Chroma Filter Enable	274 bit 5	4-70
SPU Mix Enable	309 bit 1	4-81
Reset Autofill Counter for SPU Palette	416 bit 4	4-108
Frame-Based Execution	416 bit 6	
Command Time-Out for SPU	417 bits [3:0]	

The channel start and end addresses define the location and size of the SPU Channel Buffer in SDRAM. The channel reset bit sets the write pointer of the SPU Channel Buffer to the channel start address. The SPU Stream Select Enable must be set to decode SPU streams, and the SPU Substream ID for the stream to be parsed and decoded must be entered in the SPU Substream ID field in Register 146.

The Video Interface uses the Chroma Filter to enhance the edge conditions of the SPU. It detects the SPU edge, averages the two pixels adjacent to the edge, and adjusts the mix weights of the SPU edge pixels.

The SPU Mix Enable bit must be set for the Video Interface to mix the subpicture with the main display. The palette can be filled from the bitstream or by the host on a unit-by-unit basis. If the palette information is to be taken from the bitstream, the Reset Autofill Counter bit must be initialized to the beginning address of the palette table.

When the Frame-Based Execution bit is set, time-stamp analysis is performed only at the beginning of odd fields. When the bit is cleared, the analysis is performed at the beginning of every field. The Command Time-Out value is a host-controlled error margin for time-stamp analysis (see Section 9.4, “DCSQ Handling.”)

Step 2. Start the Channel Interface.

The host sets the Channel Start/Reset bit to read all bitstream bytes into the Channel Interface for preparsing and storage in the channel buffers.

Step 3. Wait for the first pack header, extract the System Clock Reference (SCR), and load it into the SCR counter.

The Channel Interface Preparser does this automatically when it detects the first pack header after channel start. This is done to update the SCR counter at the start of a program.

Step 4. Wait for the first Navi pack and initialize the palette table with color information in the packet.

The first SPU pack should be a Navi pack with color information in its PCI packet. If the palette needs to be changed between SPUs, a Navi pack is inserted between them in the bitstream. Unless the host intervenes, the address pointer for the palette table should wrap around to the start address each time the palette table is filled so it is ready for the next refill.

Step 5. Start the SPU Decoder.

The register bits and fields in Table 9.2 are used for starting the SPU Decoder.

Table 9.2 Host SPU Decoder Start Registers

Name	Register/Bit	Page Ref.
SPU PES Data Ready Interrupt	2 bit 3	4-6
SPU Decode Start	416 bit 0	4-107

The host must set the SPU Decode Start bit to start the decoder. It can set the bit immediately after starting the channel or wait until the preparser detects an SPU PES packet in the bitstream.

The host can set the SPU Pause bit in Register 416 (page 4-107) at any time during normal play to prevent the SPU Decoder from performing DCSQ analyses and decoding SPUs. Clearing the bit restarts the decoder.

9.3 PTS Handling

The preparser transcodes the PTS from the SPU PES header to the SPU header so that it is available for the SPU Decoder. The PTS is then read from the SPU Channel Buffer in SDRAM and written to the PTS register in the SCR checker just before the start of decode of the SPU. The SPU controller compares the SCR value with those in the PTS and Start Time (STM) registers at every field or every odd field as selected by the host.

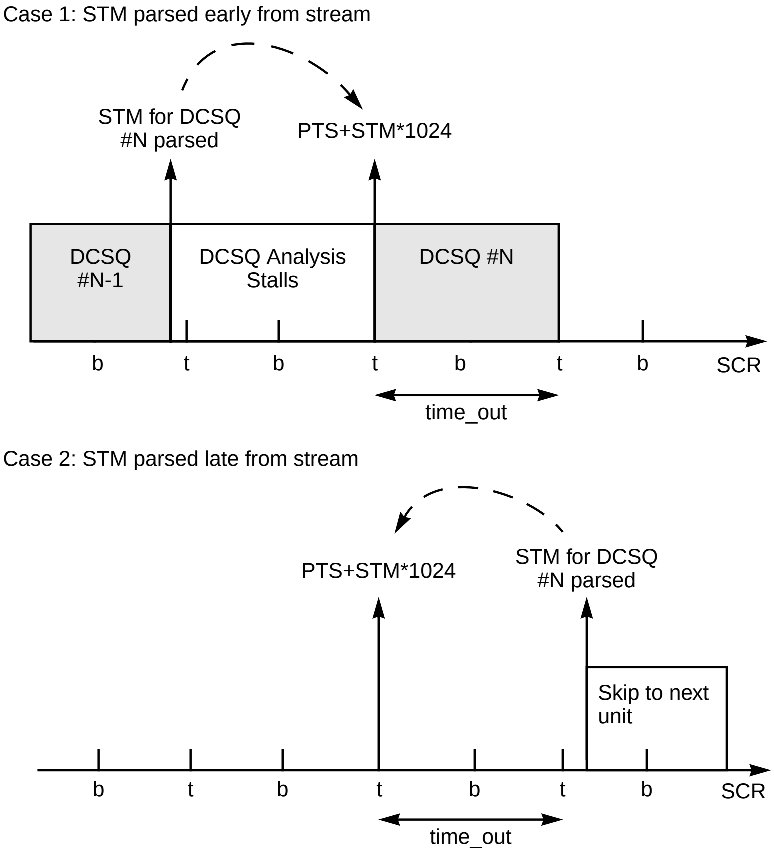
9.4 DCSQ Handling

The SPU controller performs Display Control Sequence (DCSQ) analysis using the PTS and STM time stamps stored in the bitstream to synchronize presentation of the SP data with the SCR. Analysis of the DCSQ is performed according to the following rules:

- If SCR < PTS + STM*1024, it is too early to analyze the DCSQ. The SPU Decoder stalls until the inequality becomes false.
- If PTS + (STM*1024) ≤ SCR < PTS + (STM + time_out)*1024, start DCSQ analysis.
- If SCR ≥ PTS + (STM + time_out)*1024, it is too late to perform analysis. The SPU Decoder skips to the next unit, clears the SPU display, and generates an error interrupt. If time_out is set to zero, then the error interrupt is not generated.

Figure 9.2 illustrates two examples of DSCQ analysis. The first shows an example of processing starting early, and the second shows an example of processing starting late. Setting time_out to zero is recommended only for trick play operation.

Figure 9.2 Examples of DCSQ Analysis



9.5 STM Handling

The controller extracts the STM from the SDRAM SPU Channel Buffer and uses it for synchronization of the SPU playout to the SCR. This data is parsed just after PTS extraction or just after execution of the previous DCSQ. A comparison with the SCR is made just after PTS extraction, after STM extraction, and after every new field.

An exception to STM handling is that the STM for the first DCSQ of each unit is treated as having 0 value.

9.6 SCR Handling

The host needs to update the SCR counter in the L64020 either when a new SCR is parsed from a pack header or when trick modes are activated. It is recommended that the SCR counter be updated after the Begin Active Video (BAV) interrupt. This ensures that updating the SCR does not interfere with the SPU functionality.

9.7 Highlight Information Setup

For the highlight feature, the host uses the registers listed in Table 9.3.

Table 9.3 Host SPU Highlight Registers

Name	Register/Bits	Page Ref.
Highlight Enable	447 bit 0	4-111
Highlight Color Info	448 and 449	4-111
Highlight Contrast Info	450 and 451	4-111
Highlight Area Info	452 through 457	4-112

The host must set the Highlight Enable bit for the SPU Decoder to use highlight data. The host must parse the highlight information from the PCI packet of the Navi pack ahead of the SPU pack in the bitstream by accessing the Navi Pack Channel Buffer. It must then write the highlight information to Registers 448 through 457 after the Beginning of Active Video (BAV) interrupt occurs. The SPU Decoder loads the information from these registers immediately after the next new field signal occurs.

9.8 Interrupts

The interrupt and error register bits associated with the SPU Decoder are listed in Table 9.4. The INTRn signal to the host is asserted for the interrupts in Registers 0 through 5 if the interrupt is not masked. The host should read all six registers to determine the cause of the interrupt. The interrupt bit is cleared when the host reads the registers.

Table 9.4 Host SPU Interrupt Registers

Name	Register/Bits	INTRn Asserted	Page Ref.
SPU SCR Compare Interrupt	0 bit 5	Yes	4-3
SPU Start Code Detect Interrupt	1 bit 3		4-5
SPU PES Data Ready Interrupt	2 bit 3		4-6
SPU Channel Buffer Overflow Interrupt	3 bit 2		4-8
SPU Channel Buffer Underflow Interrupt	3 bit 6		4-9
SPU Decode Error Interrupt	4 bit 4		4-10
SPU Packet Error Status	149 bit 4	No	4-46
Illegal Unit Error Flag	458 bit 7		4-113
Sync Word Error	458 bit 6		4-113
Size Error	458 bit 5		4-113
Unit Store Error	458 bit 4		4-113
Unit Error	458 bit 3		4-113
Syntax Error	458 bit 2		4-113
Time Stamp Error	458 bit 1		4-112
PXD FIFO Underflow	458 bit 0		4-112

The SPU SCR Compare Interrupt bit is set when the SCR catches up to and equals the PTS for the current SPU. This interrupt tells the host that the SPU is not being decoded. The Start Code Detect Interrupt is set when the SPU Decoder detects a start code. The SPU PES Data Ready Interrupt is set by the Preparser in the Channel Interface when it detects an SPU packet header in the incoming bitstream. The Preparser sets the SPU Packet Error Status bit when it detects an error in the packet header.

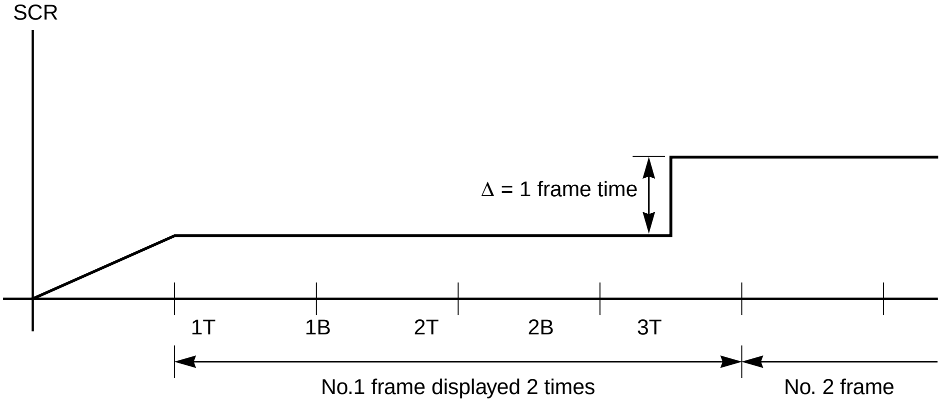
If the SPU Channel Overflow Interrupt occurs, the host can pause the channel until the SPU Decoder reads out some SPUs to relieve the condition. The SPU Channel Underflow Interrupt informs the host that the SPU Decoder is stalled due to a lack of SPUs in the buffer.

When the SPU Decode Error Interrupt occurs, the host should read Register 458 to determine the cause of the error and take the appropriate action, such as set the Jump to Next SPU bit in Register 416 (page 4-107), if necessary.

9.9 Trick Play

The SPU Decoder is capable of executing trick play with appropriate maintenance of the SCR counter. Before executing a trick play, the host should stop the SCR counter by setting the SCR Pause bit in Register 8 (page 4-13). This is shown in Figure 9.3 for slow forward. Frame 1 is repeated. To execute the DCSQ for frame 2, the host must update the SCR counter to the PTS + STM of the next new field. To resume normal play, the host must update the SCR counter to the PTS + STM of the next new field and clear the pause bit.

Figure 9.3 Slow Forward



9.10 Miscellaneous SPU Registers

The registers listed in Table 9.5 are also associated with the SPU Decoder.

Table 9.5 Miscellaneous SPU Registers

Name	Register/Bits	Page Ref.
Capture on SPU PES Ready	18 bit 0	4-16
SPU Display Off	416 bit 2	4-108
SPU Display Force Off	416 bit 3	4-108
Reset Autofill Counter for SPU Palette	416 bit 4	4-108
PTS in Current SPU	418 through 421	4-109
PTS in Next SPU	422 through 425	4-109
SP_DCSQ_STM in Next DCSQ	426 and 427	4-110
SPU Base Pointer	428 through 430	4-110
Color Palette Data for SPU	446	4-110
SPU State Machine Info	460	4-114

When the L64020 is in the Capture Mode and the Capture on SPU PES Ready bit is set, the SCR counter value is written to Registers 13 through 16 whenever the preparser in the Channel Interface detects an SPU PES header.

The host can set the SPU Display Off or SPU Display Force Off bit to command the SPU Decoder to clear the display window. Windows displayed by the FSTA_DSP command in the bitstream are not cleared by the Display Off but are cleared by the Display Force Off.

The remaining registers in Table 9.5, except for the Color Palette Data for SPU register, are read only and used primarily for diagnostics. The Color Palette Data for SPU register allows the host to write color information into the palette one byte at a time. The host should first set the Reset Autofill Counter for SPU Palette bit.

Chapter 10
Video Interface

This chapter describes the operation of the Video Interface of the L64020 Decoder. It includes a description of how to program it for proper operation, and an overview of the operation of the Vertical and Horizontal Post-processing Filters.

This chapter consists of the following sections:

10.1 Overview

The Video Interface is shown in the block diagram of Figure 10.1. It includes postprocessing filters, mixers, and display control timing.

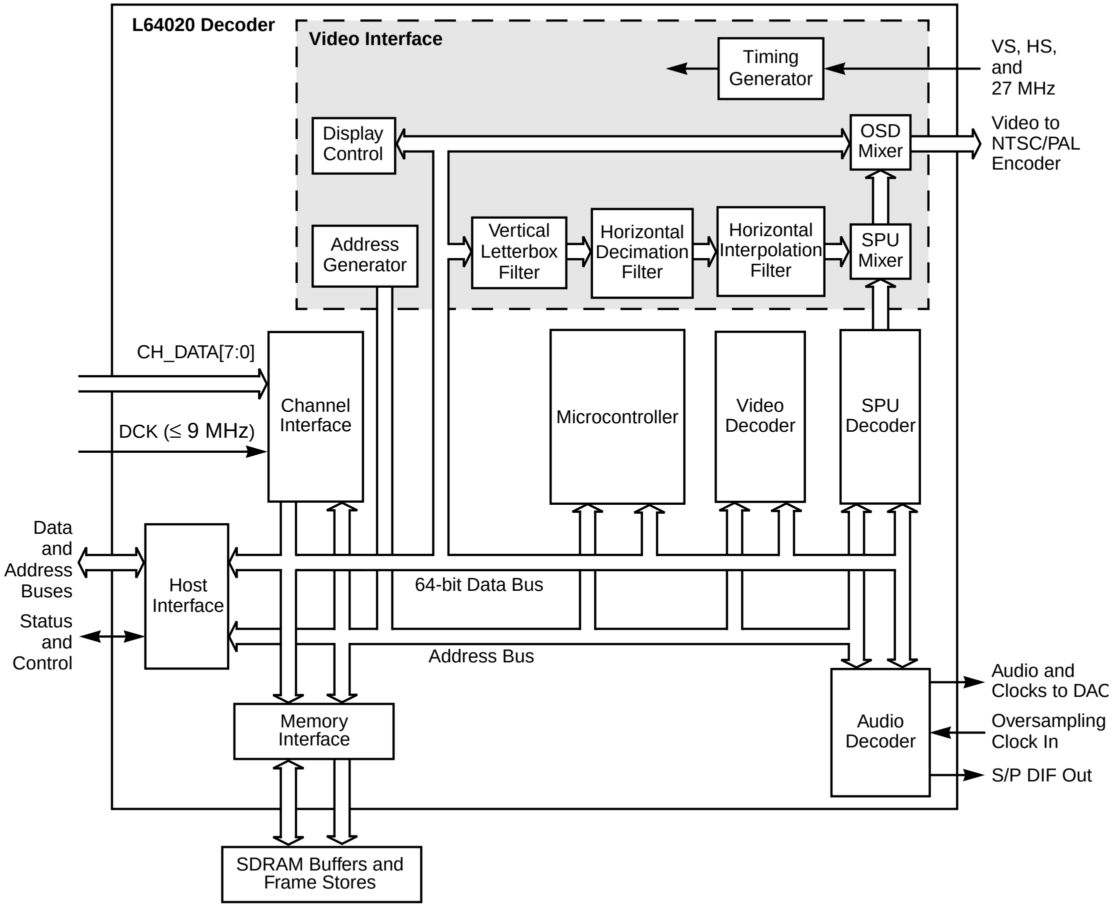
The Video Interface relies on a two-field display system operating with a 27-MHz pixel clock. The L64020 outputs 4:2:2 component video compatible with the ITU-R BT.601 format, allowing data to be time-division multiplexed onto an eight-bit bus. Eight-bit ITU-R BT.601 is the preferred interface for professional quality video equipment.

The Address Generator, under control of the Timing Generator, addresses the frame stores in SDRAM to read pixel data into the post-processing filters, reads display commands into the Display Controller, and reads On-Screen Display (OSD) bitmap data into the OSD Mixer. The postprocessing filters modify the pixel data on instructions from the Display Controller for letterboxing, 3:2 pulldown, and pan and scan.

The Display Controller also locates the video image with respect to the sync signals to account for the requirements of several different timing systems and display modes. The output of the filters passes through an SPU Mixer that adds in SPUs from the SPU Decoder and then through an OSD Mixer that adds in the OSD information. The OSD Controller times the OSD data and maintains the color palette.

The Display Controller provides a composite BLANK signal (horizontal and vertical blanking) and a CREF signal to the NTSC/PAL Encoder. CREF is high when a Cb byte is on the output bus.

Figure 10.1 Video Interface Block Diagram



10.2 Television Standard Select

To simplify programming, a Television Standard Select field in Register 290 (page 4-78) is provided. The field can be coded by the host for the modes shown in Table 10.1.

Table 10.1 Television Standard Select Field

TV Standard Select	Description
0b00	User programed (default)
0b01	NTSC (USA version)
0b10	PAL
0b11	Reserved

When the host enters either the NTSC or PAL code, the Display Controller initializes key display parameters to their defaults for L64020 operation. The default values are listed in Table 10.2.

> Note: When either the NTSC or PAL code is entered, the display parameter values in Table 10.2 overwrite any values previously programmed by the host.

The Television Standard Select code immediately returns to the user programmed mode (0b00) to allow the host to make any desired parameter modifications, such as for letterboxing or alternate display systems.

Table 10.2 Television Standard Select Default Values

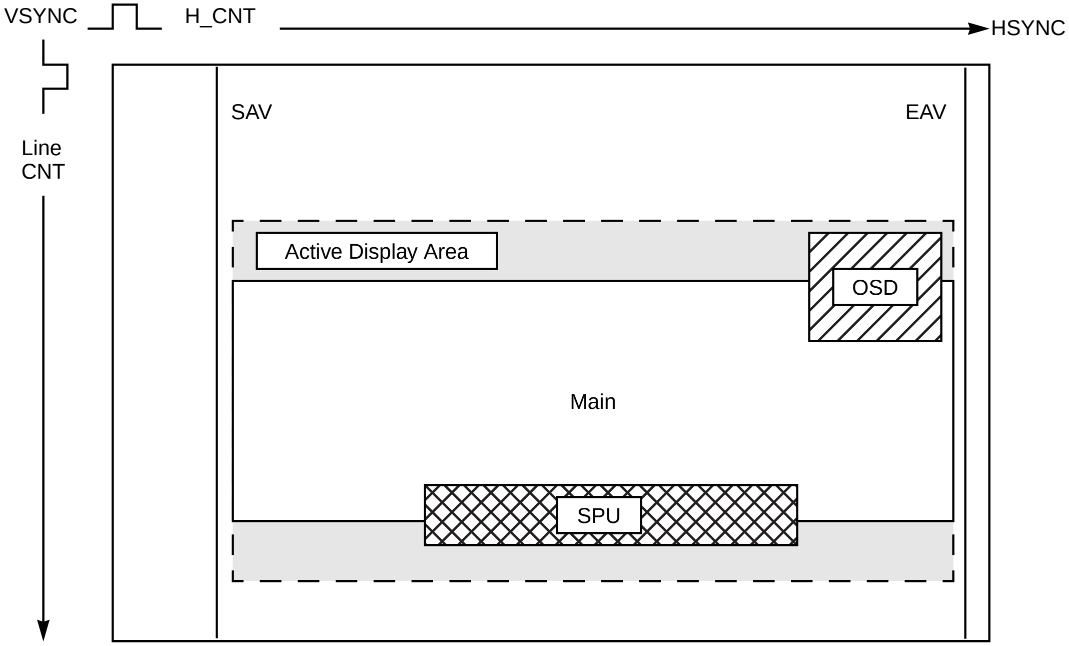
Parameter	Register[Field]	NTSC	PAL	Page Ref.
Main Reads per Line	278[6:0]	90	90	4-74
Vline Count Init	282[2:0]	4	1	4-75
Pixel State Reset Value [1:0]	284[4:3]	2	2	4-76
Main Start Row	299[2:0], 297[7:0]	23	23	4-78
Main End Row	299[6:4], 298[7:0]	262	310	
Main Start Column [10:0]	302[2:0], 300[7:0]	244	264	4-79
Main End Column [10:0]	302[6:4], 301[7:0]	1683	1703	
SAV Start Column [10:0]	308[2:0], 306[7:0]	240	260	4-80
EAV Start Column [10:0]	308[6:4], 307[7:0]	1684	1704	
Vcode Zero [4:0]	303[4:0]	21	21	4-79
Vcode Even [8:0]	303 bit 5, 304[7:0]	262	310	
Vcode Even Plus 1	303 bit 6	1	0	
Fcode [8:0]	305[7:0], 303 bit 7	265	312	4-80

10.3 Display Areas

From the Display Controller point of view, the entire display area can best be described as a blank area that is bounded vertically by the vertical sync (VS) input and horizontally by the horizontal sync (HS) input. The HS and VS input pulses determine field and line timing. For reliable operation, the sync inputs must be synchronous to the 27-MHz device clock. Additionally, VS must be received every field time and HS must be received every line time.

The Display Controller times and locates several display areas within the entire display area. Refer to Figure 10.2. The areas include the active display area, the main display area, the SPU display area, and the OSD area. The bottom-most layer is black. The active display area resides just above the black layer. The main display area is contained within the active display area. The SPU display is mixed on top of the main display. Finally, the OSD display is mixed on top of both the main and SPU areas.

Figure 10.2 Display Areas Example



The Display Controller includes counters for counting the horizontal offset and the vertical offset from the new field timing. The horizontal offset is measured in device clocks from the HS, while the vertical offset is measured as a line offset from the new field timing. The Display Controller uses these counters to determine the location of the various display areas. The host programs row and column start and end values in registers to define the location of the main display area. The start values determine the position of the upper left corners of the area while the end values set the position of the lower right corner. The location of the main display area is controlled by Registers 297–302 (page 4-78 through page 4-79).

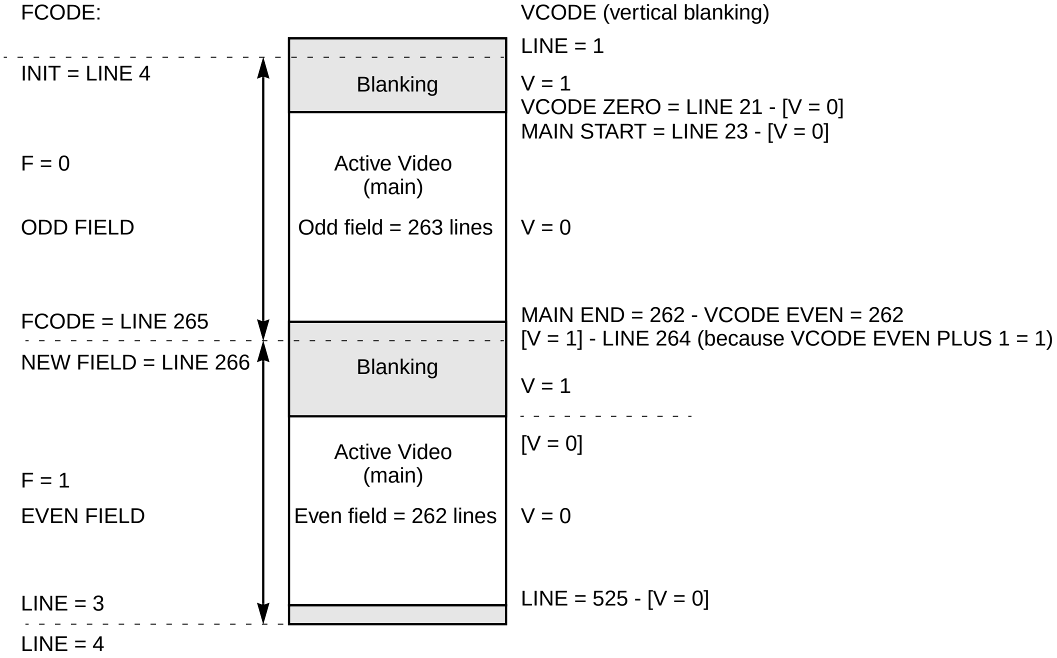
The main display area is intended for the display of either a decoded video sequence or a separate still image. This section focuses on the display of decoded video sequences; still image display features are covered later. As described earlier, the host must define the area by programming the start and end points of the area. The data for the main display area is both horizontally scaled and vertically filtered based on the programmed display mode. Since the image may be horizontally scaled, the number of pixels to be read from the frame store may not be the same as the number of pixels required for the displayed image. Therefore, the host must also program the required number of Main Reads per Line from the frame store in Register 278 (page 4-74). This value is the number of frame store pixels to be read divided by eight since there are eight luma bytes in an SDRAM burst. For example, if the source image is SIF resolution (352 pixels in width) and the target image is full resolution (720 pixels in width), the required main reads per line is equal to 352/8 = 44.

10.3.1 Vertical Timing

The active display area is bounded by the horizontal and vertical blanking intervals. The blanking intervals for the Display Controller are defined by the ITU-R BT.656 SAV/EAV timing codes. (Start of Active Video/End of Active Video). These codes include three signals for timing; a vertical blanking (V), a horizontal blanking (H), and an odd/even field (F). The Display Controller can optionally output the ITU-R BT.656 SAV/EAV timing codes on the pixel data bus by setting the ITU-R BT.656 Mode bit in Register 284 (page 4-76). Regardless of the setting of the ITU-R BT.656 Mode bit, the SAV/EAV control parameters *must* be programmed for predictable operation of the L64020. In addition to providing the SAV/EAV output codes and defining the active display area, these parameters are also used for generating the Display Controller interrupts. The ITU-R BT.656 control parameters are programmed in Registers 303–305 (page 4-79). The horizontal position of the SAV/EAV codes as an offset from the horizontal sync is programmable through the SAV Start Column and the EAV Start Column registers (306–308, page 4-80).

Figure 10.3 shows the vertical timing for an NTSC system and Figure 10.4 shows the timing for a PAL system.

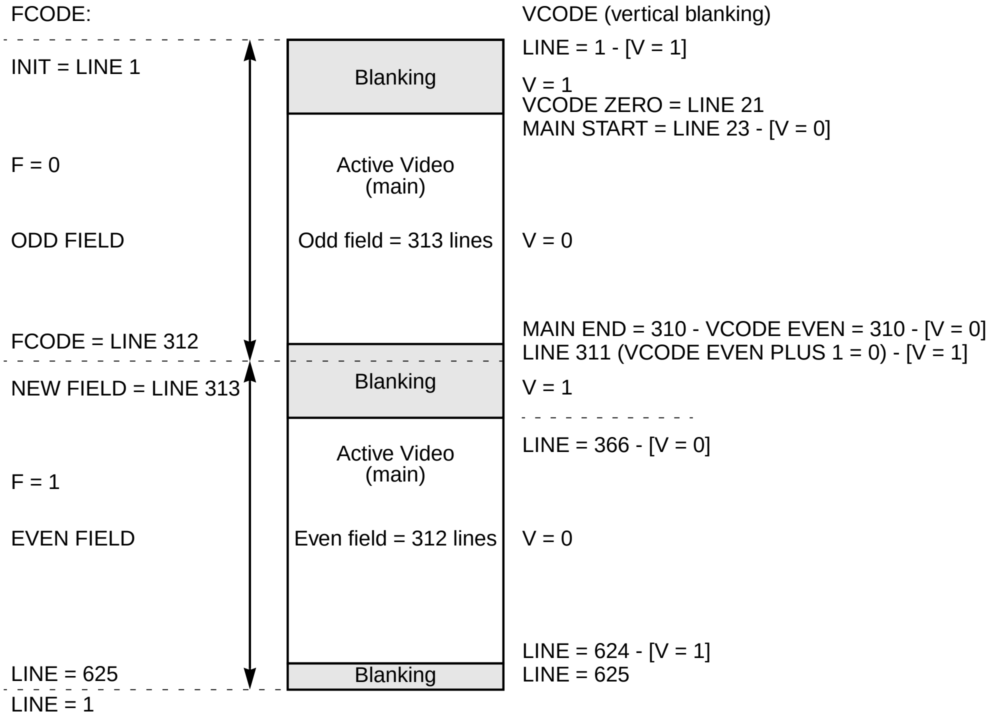
Figure 10.3 Vertical Timing Vcode and Fcode for NTSC



The Vcode Zero field of Register 303 (page 4-79) specifies the line number at which the vertical blanking bit in the EAV code should change from one to zero. This value is dependent on the particular timing system. It is typically set to 21 for NTSC and PAL.

The Vcode Even fields of Registers 303 and 304 (page 4-79) are combined to specify the line number at which the vertical blanking bit in the EAV code should change from zero to one during the even field time. In some timing systems (for example, 525-line NTSC systems), the odd field requires one additional line before changing the Vcode. Such systems are handled by programming the Vcode Even Plus 1 bit in Register 303. For an NTSC system, the Vcode Even value should be 262 and the Vcode Even Plus 1 should be set to 1. For a PAL system, the Vcode Even value should be 310 and the Vcode Even Plus 1 should be cleared to 0.

Figure 10.4 Vertical Timing Vcode and Fcode for PAL



The last required parameter relating to the ITU-R BT.656 timing is the Fcode fields of Registers 303 and 305 (page 4-79). These bits are combined to specify the line number at which the field code bit changes. In general, the Fcode changes one line prior to the new field. For NTSC and PAL, this value is 265 and 312, respectively.

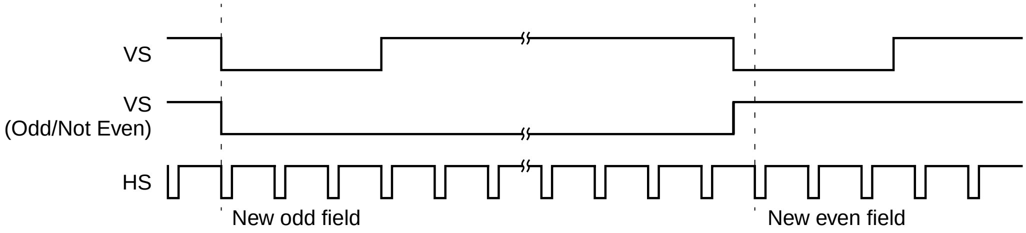
The SPU display area is defined in the SPU packet of the bitstream. The SPU decoder extracts positional location and data for the SPU and presents it to the SPU Mixer in the Video Interface for mixing with the decoded video at the proper time.

In the Internal OSD Mode, the OSD display area is obtained from the header of each OSD display list. Unlike the main display areas, the horizontal column positions represent pixel offsets from the horizontal sync, as opposed to device clocks. In the External OSD Mode, the entire display area is considered the OSD display area. When no mixing is desired, it is up to the external OSD controller to select a transparent color from the color palette.

10.3.2 Horizontal Timing

Figure 10.5 illustrates the timing of the horizontal and vertical sync inputs. The polarity of the VS and HS inputs on which the L64020 reacts is programmable with the Sync Active Low bit in Register 284 (page 4-76). Also when the VSYNC Input Type bit in Register 284 (page 4-76) is set, the VS input is used as an Even/Not Odd Field indicator. When the bit is cleared, VS is used as a sync pulse.

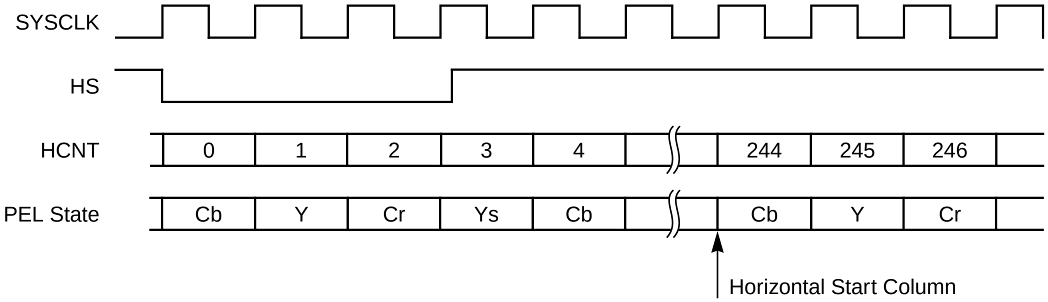
Figure 10.5 Sync Input Timing



Note: Active low mode shown. The even field is detected at the first HS after the VS.

The horizontal timing parameters are measured in terms of device clocks (81 MHz), with the leading edge of horizontal sync corresponding to a horizontal count of zero. The leading edge of the horizontal sync input initializes both the horizontal count and the pixel state (pel state) value as shown in Figure 10.6. Pel state is an internal control value for determining whether luma or chroma data is output onto the pixel data bus. Since there are four pel components (Cb, Y, Cr, Ys) of the video stream, the period of the horizontal sync signal should be modulo 4, thus preventing discontinuity in the pixel data output. For NTSC and PAL systems, this period is typically 1716 and 1728 device clocks, respectively.

Figure 10.6 Horizontal Input Timing



The horizontal start column refers to the left edge of the display area. A start column and end column must be programmed for main video. Only a start column needs to be programmed for SAV and EAV codes. The L64020 should be programmed such that each start column coincides with a Cb pel state. The Pixel State Reset Value, Register 284 (page 4-76), is programmable to ensure proper alignment. For most timing systems, the horizontal start columns are modulo 4, and an adjustment to the Pixel State Reset Value is not required. For NTSC and PAL systems, the Main Start Column values are typically 244 and 264, respectively. Use the following formula to determine the correct value for the Initial Pixel State.

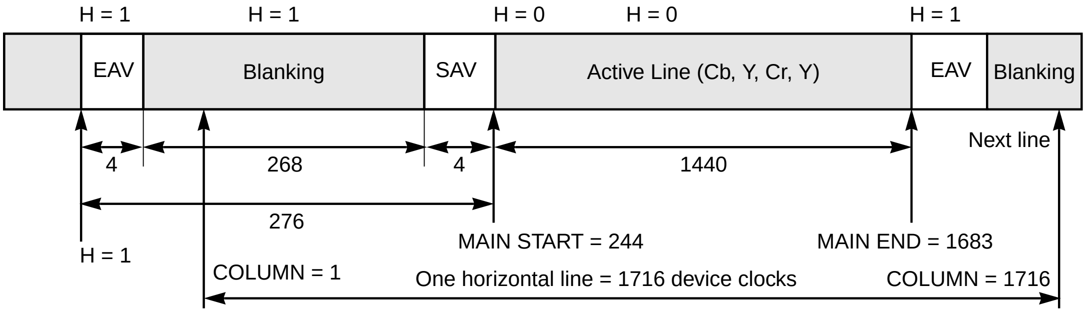
$$\text{Pixel State Reset Value} = (\text{Main Start Col} + 2)\text{mod}4$$

The general form for calculating the end column is:

$$\text{End Col} = \text{Start Col} + (\text{Picture Width} \times 2) - 1$$

For NTSC and PAL systems, the end column values are typically 1683 and 1703, respectively. Figure 10.7 shows the horizontal timing for an NTSC system.

Figure 10.7 Horizontal Timing for 8-Bit Digital Transmission for NTSC



The vertical line count is used for positioning the display areas vertically and it is initialized at the new field boundary. The vertical line count is used to compare with the start and end rows of each display area and increments with each horizontal sync. In some timing systems (i.e., NTSC), the vertical line count does not initialize to a value of one. For NTSC and PAL systems, the Vline Count Init value in Register 282 (page 4-75) should be programmed to four and one, respectively.

10.4 Video Background Modes

The host can set the display background by writing to the Force Video Background bits in Register 265 (page 4-68). The background selections are shown in Table 10.3.

Table 10.3 Force Video Background Selections

Force Video Background Bits	Description
0b00	No Background (default)
0b01	Video Black
0b10	Video Blue/User Programmable
0b11	Video on Blue

When the Force Video Background bits are set to No Background, the active display area that is not occupied by the main display area assumes the color black (Y=16, Cb=Cr=128). Usually, the main display covers the entire active display area, except when displaying small images or during letterbox filtering.

When set to Video Black, the active display area is set to black and the main display does not appear. SPU and OSD areas are mixed with the black background instead of the reconstructed images.

When set to Video Blue, the active display area assumes the color the host entered into the Programmable Background registers (Registers 266–268, page 4-69). The default value for these registers is saturated blue (Y=35, Cr=114, Cb=212), hence the name Video Blue. The reconstructed images do not appear, and the SPU and OSD areas are mixed with the programmed background color instead.

When set to Video on Blue, the active display area that is not occupied by the main display area assumes the programmable background color. This mode is ideal for the display of small images surrounded by a colored frame.

10.5 Still Image Display

The Display Controller can be programmed by the host to override the display of normal decoded video sequences to display a still image frame store instead. This is useful for Video CD still frame support and trick mode control. The registers available to the host for this feature are listed in Table 10.4.

Table 10.4 Override Display Registers

Function/Parameter	Register/Field	Page Ref.
DMA SDRAM Target Address [18:0]	213–215	4-55
DMA SDRAM Read Data [7:0]	219	4-56
DMA Mode [1:0]	193[2:1]	4-47
Display Override Luma Frame Store Start Address [15:0]	285 and 286	4-77
Display Override Chroma Frame Store Start Address [15:0]	287 and 288	
Host Top Field First bit	275 bit 4	4-71
Display Override Mode [1:0]	265[5:4]	4-68
Decode Start/Stop Command	246 bit 0	4-66
Anchor Luma Frame Store 1 Base Address [15:0]	224 and 225	4-57
Anchor Chroma Frame Store 1 Base Address [15:0]	226 and 227	4-57
Override Picture Width [6:0]	283[6:0]	4-75

First, the host must set up a display override frame store in SDRAM. This is normally accomplished using the DMA features of the Host Interface or by decoding a single frame sequence. The luma and chroma data must reside in separate frame stores and be formatted as shown in Figure 10.8.

Figure 10.8 Luma and Chroma Frame Store Format

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Y15	Y14	Y13	Y12	Y11	Y10	Y9	Y8

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Cr6	Cb6	Cr4	Cb4	Cr2	Cb2	Cr0	Cb0
Cr14	Cb14	Cr12	Cb12	Cr10	Cb10	Cr8	Cb8

Next, the host must set the Override B Luma and Chroma Frame Store Start Addresses. These are the DMA SDRAM Target addresses the host used for the two stores or the Luma and Chroma Base Addresses for a decoded picture.

In addition to the address pointers, the host must also program the width of the image using the Override Picture Width field. The picture width register has a resolution of 8 pixels, hence the frame store image width must be in 8-pixel increments. This picture width register is used by the Display Controller for accessing subsequent lines of the frame store.

To display a still picture it stored, the host adjusts the main display area if necessary, sets or clears the Host Top Field First bit as desired, enters the Decode Stop Command, and sets the Display Override Mode. The Video Decoder may be left running during the still display.

Display override has two modes, field and frame. Field Mode is provided for field structure pictures where motion between the fields may cause distortion of the image. In Field Mode, the first field is controlled by the Host Top Field First bit and is output during both field times. In Frame Mode, both fields are output to the display.

When enabled, the still image is processed through the horizontal and vertical filters of the Video Decoder. The override picture width and the main reads per line are separate registers and allows the flexibility of displaying a portion of the still image frame store (necessary for pan and scan scaling of a still frame).

When the Display Controller is programmed for still image display, the data is simply read from the Override Display Frame Store instead of the frame store indicated by the video decode engine. The parameters for the Override Display Frame Store are sampled internally at every field boundary, allowing the host to change the values in the middle of the field.

It is possible to display a still image and continue to decode video in the background. As long as the decoder is started and freeze is not active, the Display Controller continues to issue decode signals to the decoder. When a freeze is issued, the Display Controller temporarily suspends decoding while the freeze is active. This property can be exploited for various trick modes that require random access.

10.6 Display Modes and Vertical Filtering

To fully understand the display modes and their effects on picture quality, the following terminology should be understood.

- A *Progressive Frame* is a frame in which all of the data represents one instance in time. This is based on the encoded bitstream, not the display system.
- An *Interlaced Frame* contains two fields of data; one field is displayed before the other. This field is called the first field and may be either the top or bottom field. There may be motion between the fields.
- *Chroma Field Repeat* implies that the chroma data is field independent, and the entire chroma data can be repeated in both field times. Chroma Field Repeat is equivalent to Progressive Chroma.
- *Chroma Line Repeat* refers to repeating the chroma data on a line basis. This is one method of converting from 4:2:0 to 4:2:2 video format (chroma repositioning is the other method). Chroma Line Repeat is equivalent to Interlaced Chroma, assuming that the chroma data is not field repeated as well.

The Display Controller contains a 4-tap luma and a 2-tap chroma vertical filter. These filters are used to interpolate and reposition luma and chroma lines to improve picture quality. The interpolation display modes are provided to double the image size vertically, i.e., to interpolate SIF (Source Intermediate Format) resolution images to full resolution. The reposition display modes are provided to improve the quality of the picture based on the picture type. In addition, the Display Controller can perform letterbox filtering on both SIF and full-resolution images. The host can select the display mode by coding the Display Mode bits in Register 276 (page 4-72). Table 10.5 correlates the display modes to specific picture and memory parameters. The paragraphs following the table include further definitions.

Table 10.5 Display Mode Selection Table

Parameter	Display Mode [3:0]											
	0x0	**0x1**	**0x2**	**0x3**	**0x4**	**0x5**	**0x6**	**0x7**	**0x8**	**0x9**	**0xA**	**0xB**
Field Structure Picture							x	x	x		x	x
Frame Structure Picture	x	x	x	x	x	x	x	x	x	x	x	x
16:9 Aspect Ratio				x					x			
4:3 Aspect Ratio	x	x	x		x	x	x	x		x	x	x
SIF Resolution (240/288 lines)	x	x	x	x						x	x	
Full Resolution (480/576 lines)					x	x	x	x	x			x
RMM					x	x	x	x	x		x	x

The following display modes are provided for vertically interpolating SIF resolution images to full resolution:

- **Display Mode 0 - Progressive Luma/Chroma Line and Field Repeat.** In this display mode, the luma and chroma data remains unfiltered. The luma data is treated as progressive; each line of luma is displayed in both field times. The chroma data is both line and field repeated. Each line of chroma is displayed twice in both field times to achieve full-resolution 4:2:2.
- **Display Mode 1 - Progressive Luma Repositioning/Chroma Line and Field Repeat.** The luma data is treated as a progressive frame and is vertically filtered using a bilinear interpolation filter to improve luma positioning. The chroma data is both line and field repeated as in mode 0**.**
- **Display Mode 2 - Progressive Luma/Chroma Field Repeat Repositioning.** The luma data remains unfiltered and is treated as a progressive frame. The chroma data is field repeated but filtered using bilinear interpolation to improve chroma positioning.
- **Display Mode 9 - Progressive Luma Repositioning/Chroma Field Repeat Repositioning.** This display mode combines the luma component of mode 1 and the chroma component of mode 2 to

achieve improved luma and chroma positioning. It is best suited for frame based SIF images such as in MPEG-1.

- **Display Mode 10 (0xA) - Interlaced Luma Repositioning/Interlaced Chroma Repositioning.** Both the luma and chroma data is treated as interlaced. The odd lines of the frame store are used to interpolate the top field, while the even lines of the frame store are used to interpolate the bottom field. This display mode is best suited for field-structure, SIF-resolution images, or SIF format MPEG-2 images.
- **Display Mode 3 - Progressive Luma/Chroma Field Repeat Letterbox Filtering.** The progressive luma data is repeated each field time and decimated from four lines down to three using the on-chip, 4-tap, decimation filter. The chroma data is bilinearly interpolated to achieve the required decimation. This letterbox display mode is designed for frame-structure, SIF-resolution images with 16:9 aspect ratio displayed on 4:3 screens. The main start and end row positions must be adjusted to account for the 0.75 decimation. The total number of lines per field should be adjusted according to the following equation:

$$\text{Lines per Field} = \frac{\text{Main End Row} - \text{Main Start Row} + 3}{0.75}$$

The following display modes are provided for enhancing the display of full-resolution images:

- **Display Mode 4 - Interlaced Luma/Chroma Field Repeat.** The luma data is treated as interlaced while the chroma data is repeated in its entirety in both the odd and even field times. This display mode is best suited for frame-structure, full-resolution pictures.
- **Display Mode 5 - Interlaced Luma/Chroma Field Repeat with Repositioning.** The luma data is treated as interlaced. The chroma data is repeated in its entirety for both fields, but is filtered to improve its spatial positioning. This display mode is best suited for frame-structure pictures.
- **Display Mode 6 - Interlaced Luma/Chroma Line Repeat.** The luma data is treated as interlaced. The chroma data is treated as interlaced and is line repeated to achieve the 4:2:0 to 4:2:2 chroma conversion. This display mode is suitable for either field- or frame-structure pictures.

- **Display Mode 7 - Interlaced Luma/Interlaced Chroma with Repositioning.** Both the luma data and chroma data are treated as interlaced. The chroma data is filtered using the bilinear chroma filter to improve the chroma positioning. This display mode is suited for either field- or frame-structured pictures.
- **Display Mode 8 - Interlaced Luma/Interlaced Chroma 0.75 Letterbox Filtering.** Both the luma and chroma data is treated as interlaced and processed through the letterbox filter to achieve decimation of four lines down to three. The main start and end rows must be adjusted to account for this 0.75 decimation. The total number of lines per field should be adjusted according to the following equation:

  Main End Row - Main Start Row + 1 = (0.75*lines/field) - 2

- **Display Mode 11 (0xB) - Interlaced Luma/Interlaced Chroma 0.5 Letterbox Filtering.** Both the luma and chroma data is treated as interlaced and processed through the letterbox filter to achieve decimation of four lines down to two. The main start and end rows must be adjusted to account for this 0.5 decimation. The total number of lines per field should be adjusted according to the following equation:

  Main End Row - Main Start Row + 1 = (0.5*lines/field) - 2

10.7 Reduced Memory Mode

For applications where SDRAM space is limited, Reduced Memory Mode (RMM) may be enabled to reduce the memory required for B-frame reconstruction. RMM is intended for applications with PAL resolution images, or when large OSD display areas limit the availability of SDRAM.

Since RMM overwrites the memory used for decoding the first field of the B frame immediately after display, only a subset of the display modes and features are available for use in RMM. The available display modes include 4, 5, 6, 7, 8, and 10. However, since display modes 4 and 5 repeat the entire field of chroma during the second field time, only luma data can utilize the benefits of the reduced memory frame store. Care must be taken to allot enough memory for the entire frame of chroma.

Since the first field is not available after it has been displayed, RMM cannot fully support all of the freeze modes. In RMM, only Freeze Last Field is supported on B pictures because the first field of data is overwritten in memory. In addition, when performing pulldown the second field, instead of the first field, is repeated during the display of B-frames.

10.8 Horizontal Postprocessing Filters

The Display Controller integrates two separate horizontal postprocessing filters, a simple 2:1 horizontal decimation filter and an 8-tap interpolation filter. These filters are provided for scaling images horizontally along the scan line.

The decimation filter is a simple bilinear averaging filter that decimates two pixels down to one along the horizontal scan line. This filter may be used in conjunction with the 0.5 letterbox filter for displaying pictures at 1/4 resolution. Such an application may require a four-frame store system to account for the higher bandwidth requirements. The decimation filter is enabled by setting the Horizontal Decimation Filter Enable bit in Register 274 (page 4-70). When using the decimation filter, the main reads per line should be programmed to the number of frame-store reads required to reconstruct the picture, i.e., twice the picture width.

Regardless of the input picture resolution, the horizontal interpolation filter can provide up to 720 pixels on each line. In addition to its interpolation features, the filter also provides fine-scale horizontal pan and scan to within 1/8th of a pixel during pan and scan operation. This filter is used for both luminance and chrominance data.

The interpolation filter is implemented using an 8-tap polyphase filter. The filter is capable of generating up to eight, unique, subpixel values between two consecutive pixels on a scan line. The generation of pixels depends upon the ratio between the width of the source image and the target image. Typically, the target image width is 720 pixels.

Figure 10.9 through Figure 10.12 illustrate the characteristics of the horizontal interpolation filter. To activate the horizontal filter, the host must first enable the filter by setting the Horizontal Filter Enable bit in Register 276 (page 4-72). The desired filter response is selected by setting (response A) or clearing (response B) the Horizontal Filter Select bit in the same register.

Note that response A has a slightly higher cut-off frequency and provides a slightly sharper image. Response B has less ripple in the passband and provides more uniform brightness on complex patterns.

Figure 10.9 Frequency Response A

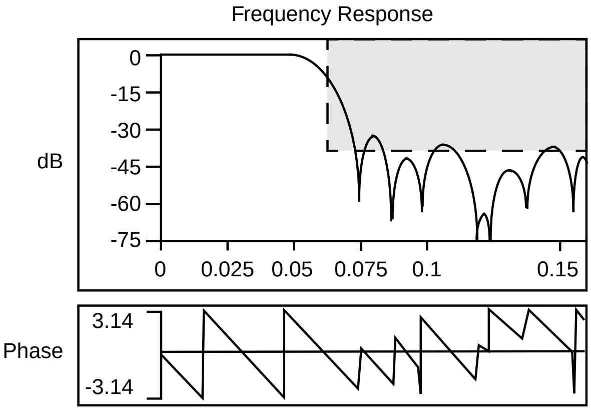


Figure 10.10 Impulse Response A

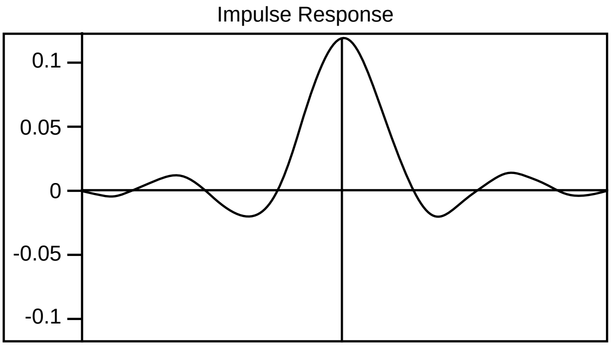


Figure 10.11 Frequency Response B

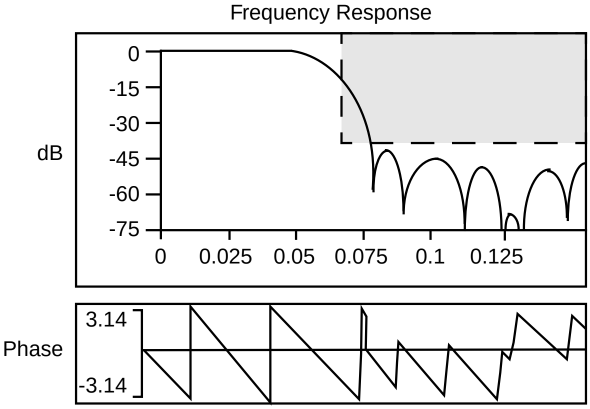
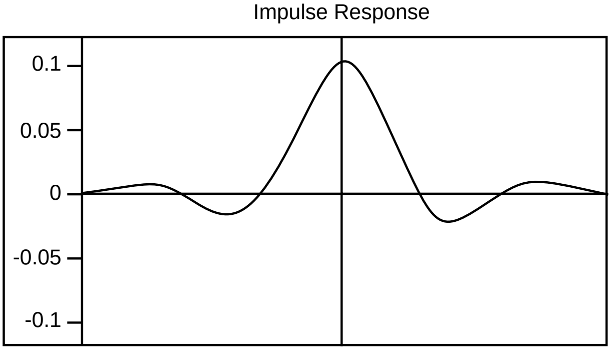


Figure 10.12 Impulse Response B



The scale factor of the interpolator is the ratio between the widths of the source image and the target image in 1/256th of a pixel increments. The interpolator calculates the subpixel position to within 1/256th of a pixel and chooses one of the eight filter banks closest to the calculated location. The filter integrates a raster mapper that increments by n/256 each output pixel.

Table 10.6 shows the raster mapper increment for each of a number of popular source image resolutions. The general form for deriving the raster mapper increment value is:

$$\text{Increment} = \left\lceil \frac{\text{Source Width}}{\text{Target Width}} \times 256 \right\rceil$$

where $\lceil x \rceil$ means to round the value x to the smallest integer larger than x. A value of zero in the Horizontal Filter Scale register (Register 277, page 4-73) is equivalent to an increment of 256. The raster mapper increment value is sampled at the new field boundary to allow the host to change the scale factor between fields.

Table 10.6 Raster Mapper Increment Value by Source Resolution

Target	Source	Ratio (Source:Target)	Increment	Main Reads/Line
720	352	0.488	126	44
720	480	0.667	171	60
720	544	0.755	194	68
720	640	0.888	228	80
704	352	0.500	128	44
704	480	0.682	175	60
704	544	0.755	198	68
704	640	0.909	233	80

Clearing the Horizontal Filter Enable bit disables the interpolation filter, leaving the data unfiltered. The filter is automatically disabled during the auxiliary data area of the display.

10.9 Subpicture Unit Display

The SPU Decoder extracts commands (mix weight information) and pixel data from the bitstream and presents the pixel data to the Video Interface at the appropriate time for mixing with the decoded video sequence. The SPU data is neither horizontally scaled nor vertically filtered and thus remains the same size regardless of the filtering of the video data.

In addition to the SPU mixing function, the host can set the SPU Chroma Filter Enable bit in Register 274 (page 4-70) to enhance the chrominance performance. This filter is designed to improve the edges of SPU areas.

10.10 On-Screen Display

The Video Interface integrates a flexible OSD Controller that allows the overlay of a bitmap image on top of the decoded video or background after SPU mixing. The OSD image may be written to SDRAM by the host (internal mode) or supplied to the L64020 from an external device (e.g., a character generator) on the EXT_OSD[3:0] input pins.

The OSD image is not horizontally filtered and is thus always the same size regardless of the resolution or mode of the video data. In addition, pan and scan of the video data does not affect the position of the OSD overlay. The OSD Controller also includes a chroma filter designed to enhance the edge conditions of OSD areas. This filter can be enabled by setting the OSD Chroma Filter Enable bit in Register 274 (page 4-70).

The OSD controller supports three basic image formats:

- Up to 720 x 576 pixels at 2 bits/pixel
- Up to 720 x 576 pixels at 4 bits/pixel
- Up to 720 x 576 pixels at 8 bits/pixel

10.10.1 OSD Modes

The host can select the OSD mode by setting the OSD Mode bits in Register 265 (page 4-67). Table 10.7 shows the coding of the bits.

Table 10.7 OSD Modes

OSD Mode Bits	Description
0b00	No OSD (default)
0b01	Internal OSD (Contiguous)
0b10	Internal OSD (Linked List)
0b11	External OSD

The internal OSD modes use the on-chip color palette and the OSD Controller. The external OSD mode simply uses the on-chip color palette and a color look-up value supplied on the L64020 EXT_OSD pins.

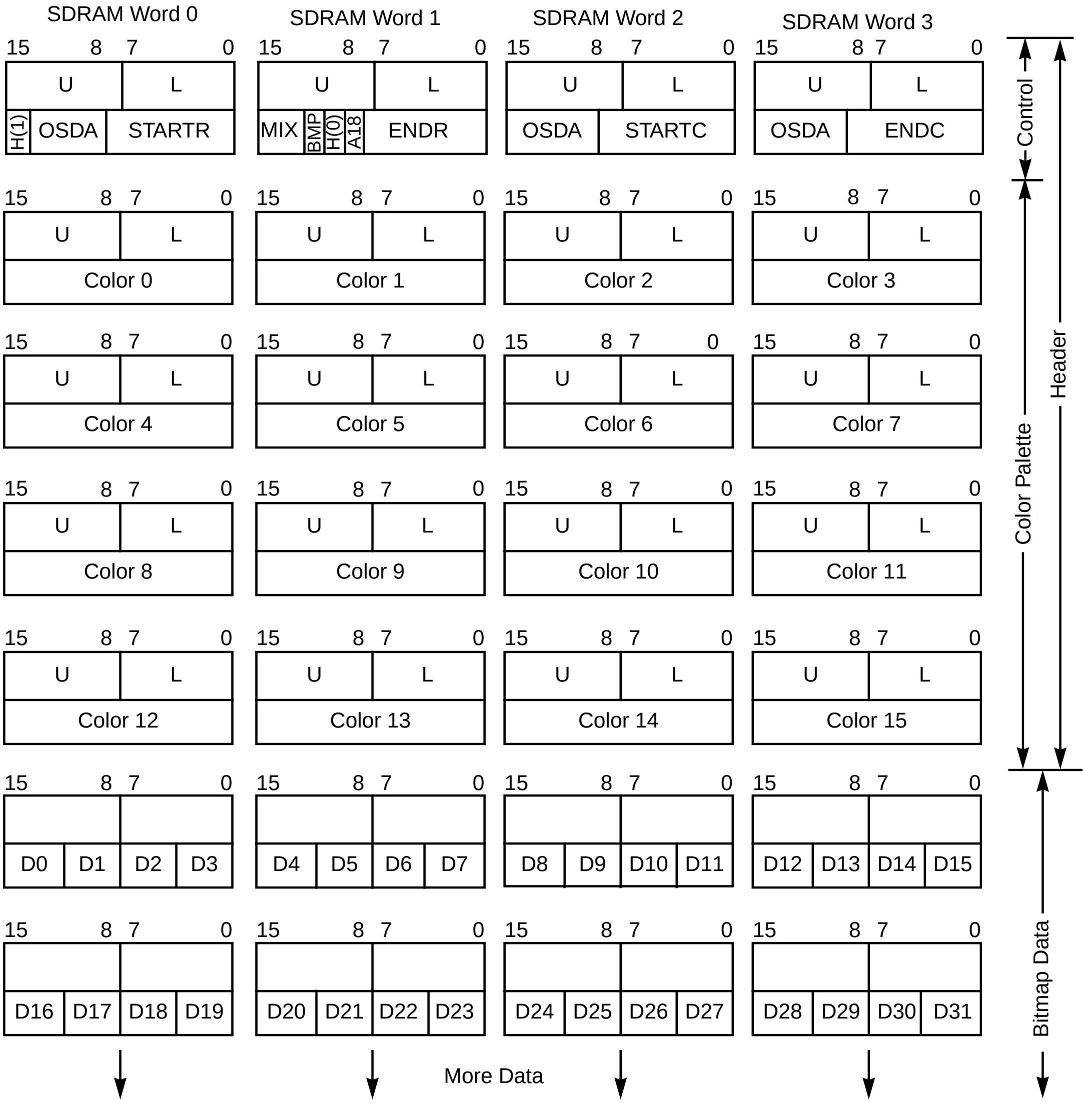
10.10.2 Internal OSD

For internal OSD mode, the host must write a header containing control information and an optional color palette, and an image bitmap to SDRAM for each OSD display area. This is best done using an external DMA Controller.

10.10.2.1 OSD Display Area Storage Layout

Figure 10.13 illustrates the SDRAM word organization of an OSD display area using 4 bits/pixel resolution. There are four words of control information followed by a 16-word palette. The palette is followed by the bitmap image. For 2 bits/pixel resolution, only a 4-color palette is required. For 8 bits/pixel resolution, a 256-color palette is required. The pixel data (2, 4, or 8 bits/pixel) is packed with pixel 0 (D0) in the upper bits of the first word of the bitmap. This layout is repeated for each OSD display area.

Figure 10.13 OSD Area Data Organization



10.10.2.2 Header Control Information

The layout of the control information in the OSD header is shown in Figure 10.14.

Figure 10.14 OSD Header Control Fields

SDRAM Word 0

15	14 9	8 0
HIC[1]	OSDA[5:0]	STARTR[8:0]

SDRAM Word 1

15 12	11	10	9	8 0
MIX[3:0]	HIC[0]	BMP ONLY	OSDA[18]	ENDR[8:0]

SDRAM Word 2

15 10	9 0
OSDA[11:6]	STARTC[9:0]

SDRAM Word 3

15 10	9 0
OSDA[17:12]	ENDC[9:0]

ENDR[8:0] **OSD End Row** **Word 1 [8:0]**
The line number as an offset from the new field time on which this OSD area ends is written into this field.

OSDA[18:0] **OSD Address**
Word 0 [14:9], Word 1 Bit 9, Word 2 [15:10], Word 3 [15:10]
When programmed, OSDA[18:0] contains the SDRAM address of the next OSD display area in a linked list. These fields are ignored in contiguous OSD mode.

BMP ONLY **Bitmap Only** **Word 1 Bit 10**
This bit is set to indicate that the OSD header does not contain a color palette and the existing color palette should be used.

HIC[1:0] **High Color Mode** **Word 0 bit 15 and Word 1 bit 11**
These two bits are used to set 2-, 4-, or 8-bit color mode according to the following table.

HIC[1:0]	Description
0b00	2 bits/pixel
0b01	8 bits/pixel
0b10	4 bits/pixel
0b11	Reserved

MIX[3:0] **Mix Weight** **Word 1 [15:12]**

The value written into MIX[3:0] sets the mix ratio between the foreground (overlay) pixels and the background (reconstructed picture) pixels in increments of 1/16 of the pixel value. If Mix Weight is zero, the output pixel is weighted 100% reconstructed picture and 0% OSD. If Mix Weight is 15, the output is 15/16 OSD and 1/16 reconstructed picture. This can be enabled or disabled for each color in the palette (see Mix Enable in the next section).

STARTR[8:0] **OSD Start Row** **Word 0 [8:0]**

The field line number (0 to 313) from the offset of the new field time where this OSD area begins is written into STARTR[8:0]. STARTR[8:0] is also used to end a linked-list OSD display by pointing to a line below the main display area.

ENDC[9:0] **OSD End Column** **Word 3 [9:0]**

The column number in pixels on which this OSD area ends is written into ENDC[9:0].

STARTC[9:0] **OSD Start Column** **Word 2 [9:0]**

The column number in pixels as an offset from the horizontal sync on which this OSD area begins is written into STARTC[8:0].

10.10.2.3 OSD Palette Color Fields

The 16-bit color fields, COL0 through COL15 or COL256, are formatted as shown in Figure 10.15.

Figure 10.15 OSD Header Color Fields

15 10	9	8	7 4	3 0
Y[5:0]	R	MIX	Cb[3:0]	Cr[3:0]

Cr[3:0] **Color Difference Y-R** **[3:0]**

Y-R color difference value with ITU-R BT.601 chromaticity. This value is multiplied by 16 before being used by the OSD controller.

Cb[3:0] **Color Difference Y-B** **[7:4]**
Y-B color difference value with ITU-R BT.601 chromaticity. This value is multiplied by 16 before being used by the OSD controller.

MIX **Mix Enable** **8**
If set, the Mix Weight field applies to this color; otherwise, the output data is 100% OSD.

R **Reserved** **9**
This bit should be cleared.

Y[5:0] **Luminance** **[15:10]**
Y contains the luminance value with ITU-R BT.601 chromaticity. This value is multiplied by four before being used by the OSD controller.

Note: If Y = 4 (scaled to 16) and both Cr and Cb = 8 (scaled to 128), then the output pixel is black. In the special case of a color palette entry Y = Cb = Cr = 0, the output pixel is transparent and the underlying decoded video is displayed.

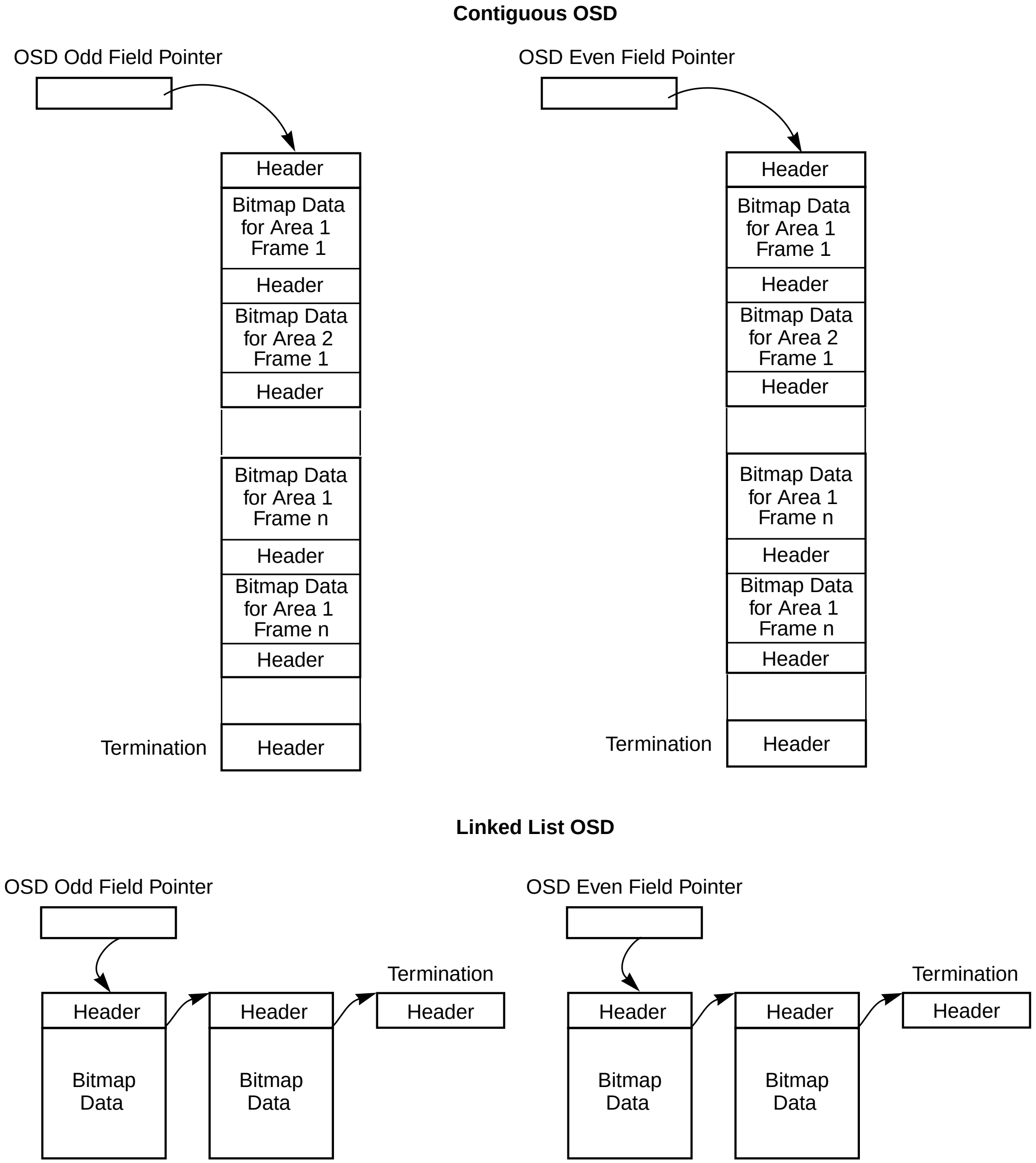
10.10.2.4 OSD Storage Formats

As mentioned, each OSD area requires the header control information and the data bitmap. If a color palette is not included in the header, the OSD Controller uses the palette information that was last entered. An OSD frame may contain two or more display areas. Each frame must be stored as two interlaced fields.

The OSD Controller accepts two formats for the SDRAM storage of OSD data, Contiguous OSD and Linked List OSD. The OSD Mode bits in Register 265 (Table 10.7) specify which format the host has selected. Figure 10.16 illustrates the two formats.

In Contiguous OSD Mode, the OSD areas must occupy contiguous locations in SDRAM, one for the odd fields and one for the even fields. The host must write the beginning addresses of the two memory blocks into the OSD Odd Field Pointer and OSD Even Field Pointer registers (Registers 270 through 273, page 4-70). The OSD Controller reads the pointers to start the OSD display and then updates them for every frame. It ignores the OSDA bits in the OSD headers. A termination header must be added after the data for the last OSD area. The termination header should specify a start row (STARTR) below the last line of the display.

Figure 10.16 OSD Storage Formats



In Linked List OSD Mode, the OSD areas do not need to reside consecutively in memory. The first OSD area for each field is addressed using the programmed OSD pointers. Subsequent OSD areas are addressed using the OSD address bits, OSDA[18:0], of the current OSD header. To terminate the Linked List OSD, the OSD address in the header for the last area should point to a termination header that has a start row (STARTR) below the last line of the display.

10.10.2.5 Bits/Pixel Modes

The OSD Controller reads the OSD data from the SDRAM frame stores and displays it at a rate of one pixel every two L64020 clocks. Using 6 bits for Y, and 4 bits each for Cb and Cr, the OSD color palette can contain any of 16,384 colors including black and transparent. The Cb and Cr values are decimated from 4:4:4 to 4:2:2 prior to mixing with the decoded video data. As indicated, the pixels can be coded using 2, 4, or 8 bits to produce 4, 16, or 256 YCbCr color values.

The color palette may be updated once per OSD area. The OSD controller automatically performs this process prior to displaying the next OSD area. This color expansion feature allows many more colors to be displayed on the overlay screen at one time.

Since the palette contains 256 colors in 8 bits/pixel mode, the palette should be loaded only once per field with the first OSD area of the frame. The data for subsequent OSD areas should include only the header control information and the area bitmap.

10.10.2.6 OSD Controller Operation

At the beginning of each field, the OSD Controller scans the display list stored in SDRAM and loads the first 64 bits (header control information) for the first OSD area into the internal OSD control registers. The OSD Controller then buffers the color palette information and bitmap data, and waits until the Display Controller output reaches the first pixel location of the OSD area (defined in the STARTR and STARTC fields of the header). It mixes the OSD data with the reconstructed video data according to the mix weight in the header and the mix enable in the color palette. When the OSD Controller outputs the last line of the OSD area, it immediately loads the next 64 bits of header information from the display list and the process repeats itself. Only one OSD area is active at a time and multiple areas cannot lie on the same horizontal line.

10.10.2.7 OSD Requirements

The following list of requirements must be met to program OSD areas:

- The OSD header MUST be word aligned in SDRAM.
- The OSD header should NOT include a row or column number that is out of range of the current display parameters, EXCEPT in the Termination Header.
- The row end address MUST be greater than the row start address, EXCEPT in the Termination Header.
- The column end address MUST be greater than the column start address.
- OSD areas MUST NOT overlap each other in any way or be programmed on the same line or lines.
- OSD areas should be ordered top to bottom in the display list.
- The OSDA pointers MUST point to valid OSD headers.
- In Linked-List OSD mode, the width of an OSD area must be a multiple of 32 pixels in 2 bits/pixel mode, 16 pixels in 4 bits/pixel mode, or 8 pixels in 8 bits/pixel mode.
- In Contiguous OSD mode, the total number of pixels in an OSD area must be a multiple of 32 pixels in 2 bits/pixel mode, 16 pixels in 4 bits/pixel mode, or 8 pixels in 8 bits/pixel mode.
- The number of bits in the bitmap should not exceed the available memory space.
- The colors programmed into the color palette should be legal ITU-R BT.601 colors.
- The OSD image width must not exceed 720 pixels.

10.10.3 External OSD

The Video Interface has provisions for interfacing with an external OSD Controller such as a character generator integrated circuit. To operate in External Mode, the host must first load the color palette information into the color look-up table (CLUT) in on-chip RAM. The bitmap data is fed in on the EXT_OSD[3:0] pins of the L64020.

Before switching to external OSD mode, the host loads the color palette information into the CLUT by first setting the Clear OSD Palette Counter bit in Register 265 (page 4-67) to reset the CLUT address pointer and then writing 32 consecutive bytes to the OSD Palette Write register (Register 269, page 4-69). Writes to this register automatically increment the CLUT address pointer. Since the OSD Palette Read-Write register is 8 bits wide and the CLUT is 16 bits wide, the first write loads the most significant byte and the second write loads the least significant byte of the CLUT data at each address. The last write to the register is the least significant byte of word 15 in the CLUT.

The EXT_OSD[3:0] inputs are sampled at a 13.5-MHz rate or every other 27-MHz system clock. For this reason, the recommended external OSD frequency is 13.5 MHz. Running at a faster frequency results in lost external pixels and running at a slower frequency results in replicated pixels.

The external OSD inputs are double buffered in the L64020 but should be supplied synchronous to the system clock. It is also important to keep in mind that there will be a delay of four to five 27-MHz clock cycles (or two pixels of video) between the time that the external OSD data is supplied and the time it appears at the video output port. This latency depends on the phase shift between the external dot clock and the internal sampling clock.

Finally, note that OSD data is always mixed with the reconstructed video in this mode. If any of the video is to be viewed, at least one of the 16 colors must be transparent (luma and chroma = 0). In general, eight of the 16 colors may be programmed as transparent to allow one of the four EXT_OSD inputs to act as an OSD blank function.

10.11 Pan and Scan Operation

The display control subsystem supports horizontal pan and scan of an image over the display area. The primary purpose for implementing pan and scan is for viewing pictures that are too wide to be displayed in the available screen area. An example of this situation is a wide-screen image (16:9 aspect ratio) that is displayed on a standard 4:3 aspect ratio screen without letterboxing.

The pan and scan offset can either be controlled by the host or automatically with values extracted from the bitstream. The host can set or clear the Pan and Scan from Bitstream bit in Register 279 (page 4-74) to specify the source of the pan and scan controls.

10.11.1 Host Controlled Pan and Scan

When the Pan and Scan from Bitstream bit is cleared, the host controls pan and scan and must program the registers listed in Table 10.8 based on bitstream parameters (horizontal size, vertical size, aspect ratio, etc.) written into Auxiliary Data FIFO.

Table 10.8 Host Controlled Pan and Scan Registers

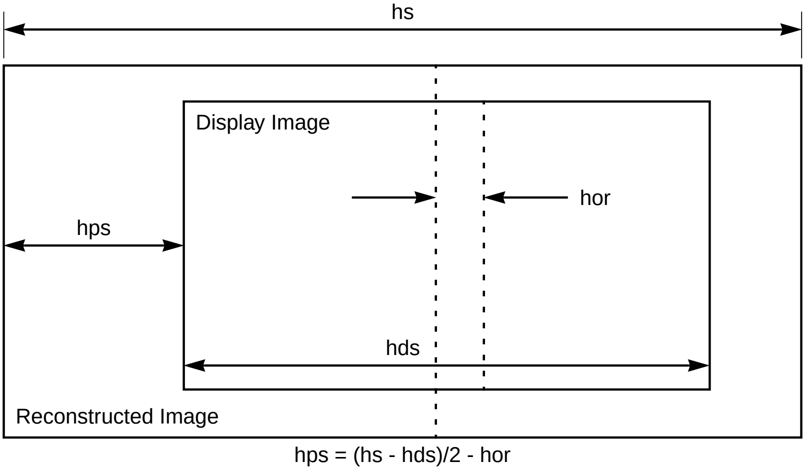
Parameter	Register/Field	Page Ref.
Horizontal Pan and Scan Luma/Chroma Word Offset [7:0]	280	4-75
Pan and Scan Byte Offset [2:0]	279 [5:3]	4-74
Pan and Scan 1/8 Pixel Offset [2:0]	279 [2:0]	
Horizontal Filter Enable bit	276 bit 1	4-72
Horizontal Filter Select bit	276 bit 2	
Horizontal Filter Scale [7:0]	277	4-73
Main Reads per Line [6:0]	278	4-74
Vertical Pan and Scan Line Offset [7:0]	281	4-75

The values in the three pan and scan offset register fields specify the horizontal offset of the displayed image from the stored image. The Pan and Scan Word Offset is the horizontal offset in 64-bit words (8 pixels). The Pan and Scan Byte Offset selects the byte within the selected word. The Pan and Scan 1/8 Pixel Offset changes the start phase of the horizontal interpolation filter to shift the display in 1/8-pixel increments. This requires the host to set the Horizontal Filter Enable bit and set or clear the Horizontal Filter Select bit. The right edge of the displayed image is set by the Main Reads per Line value. It tells the Display Control Subsystem the number of 64-bit words (8 pixels) to read from the stored image for each line starting at the offset.

The horizontal scale and main reads/line values are used for interpolating the horizontal display size up to the 720 pixels, and should be updated at the sequence boundary. The pan and scan offset values are used to display the desired portion of the reconstructed frame store. Unlike the pan and scan offset values embedded in the bitstream, an offset value of zero corresponds to the top-left pixel in the reconstructed frame store image, NOT the center of the image. When under host control, it is up to the host processor to convert the pan and scan value to an offset value. Refer to Figure 10.17 for calculating horizontal pan and scan offset values. The host can access the bitstream parameters necessary for calculating the pan and scan offset via the Auxiliary Data FIFO.

The pan and scan offset values shifts the display image to the right by (*wordOffset* x *8*) + *byteOffset* pixels. The pan and scan word offset, Register 280[7:0], corresponds to the horizontal offset in frame memory words, where one word is equivalent to 8 pixels. The pan and scan byte offset selects the pixel in the word after the selected word offset. The Horizontal Size (hs) is the width of the reconstructed frame store extracted from the sequence header and is used internally for accessing subsequent lines of the reconstructed image. To enable host-controlled horizontal pan and scan to 1/8 pixel boundaries, the host should program the Pan and Scan 1/8 Pixel Offset in Register 279 (page 4-74). This register changes the start phase of the horizontal interpolation filter and shifts the image in 1/8 pixel increments. The pan and scan offset values are sampled at the field boundary giving precise control over the pan and scan offset timing.

Figure 10.17 Horizontal Pan and Scan Calculation



Note:

- hs = horizontal size extracted from sequence header.
- hds = horizontal display size extracted from sequence display extension.
- hor = horizontal picture offset extracted from picture display extension.
- hps = horizontal pan and scan offset.

10.11.2 Bitstream Controlled Pan and Scan

When operating under bitstream control, the horizontal pan and scan offset values are automatically extracted from the bitstream and converted for the Display Controller. The pan and scan offset values are updated at the field boundary for precise control over the offset. The host is still responsible for deriving the Horizontal Filter Scale value and the Main Reads per Line value from the display information in the sequence header written to the Aux Data FIFO.

10.11.3 Vertical Pan and Scan

The Display Controller supports vertical panning via host control at a resolution of two lines/field or four lines/frame. The Vertical Pan and Scan Line Offset is host programmable in Register 281 (page 4-75) and is sampled at every new field time to allow for different offsets for each field. The value programmed into Register 281 must be a positive value representing the vertical pan and scan value in two-field line increments from the top of the image.

10.12 Display Freeze

The host can write to the Freeze Mode bits of Register 275 (page 4-71) to select one of the three freeze modes listed in Table 10.9.

Table 10.9 Freeze Modes

Freeze Mode Bits	Freeze Mode
0b00	Normal (no freeze)
0b01	Freeze Frame
0b10	Freeze Last Field
0b11	Freeze First Field and Hold

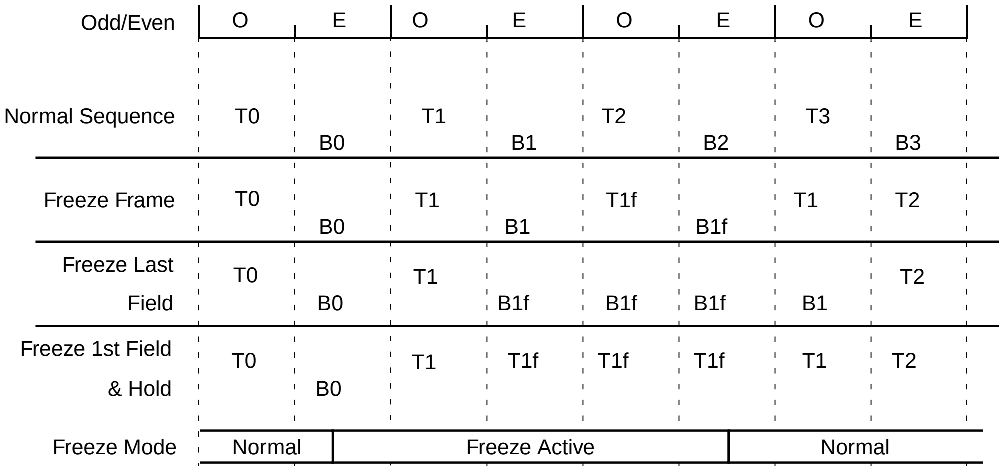
When the host issues a freeze, the display freezes in the requested mode and picture reconstruction is halted to prevent overwriting the stored image. When the host removes the freeze condition, the Display Controller displays one more field of the current frame and picture reconstruction is resumed.

Figure 10.18 shows the timing of the freeze modes. When the host issues a Freeze Frame request, the Display Controller repeats both the first and last field of the frame store. After the freeze is removed, the Display Controller displays one more field to restart the reconstruction process. This freeze mode is recommended only for frame-based pictures since there is no motion between the fields.

When executing a Freeze Last Field, the Display Controller displays the first field of the frame store once and then freezes on the last field of the frame store. After the freeze condition is removed, the Display Controller displays the last field one more time to restart the reconstruction process. It then displays the first field of the next frame.

During Freeze First Field and Hold, the Display Controller only displays the first field of the frame store. After the freeze is removed, the Display Controller displays the first field once more to restart the reconstruction process. It then displays the first field of the next frame.

Figure 10.18 Freeze Operation Timing



The Freeze Mode bits are sampled at the field boundaries. However, only freeze requests issued before the first field in the frame are applied to the frame. That is, a freeze request issued during the first field is applied to the next frame. The return to normal request, however, is honored at the next field.

Note: Freezing for an odd number of field times causes a field inversion. A field inversion is defined as displaying the top field of a frame during an even field time and the bottom field during an odd field time.

The host can detect the inversion condition by reading the First Field and Top/Not Bottom Field bits in Register 275 (page 4-72). If they are at opposite states, the fields are inverted. A single Freeze Last Field request can correct the inversion. If the host sets the Automatic Field Inversion Correction bit in Register 279 (page 4-74) and field inversion is detected by the Display Controller, it displays the next frame starting at display line two in the frame store.

The First Field bit in Register 275 (page 4-71) and Last Field bit in Register 276 (page 4-72) can be monitored by the host to determine which field in the frame is currently being displayed. When both bits are cleared, a middle field is being displayed as in pulldown or freeze modes.

10.13 Pulldown Operation

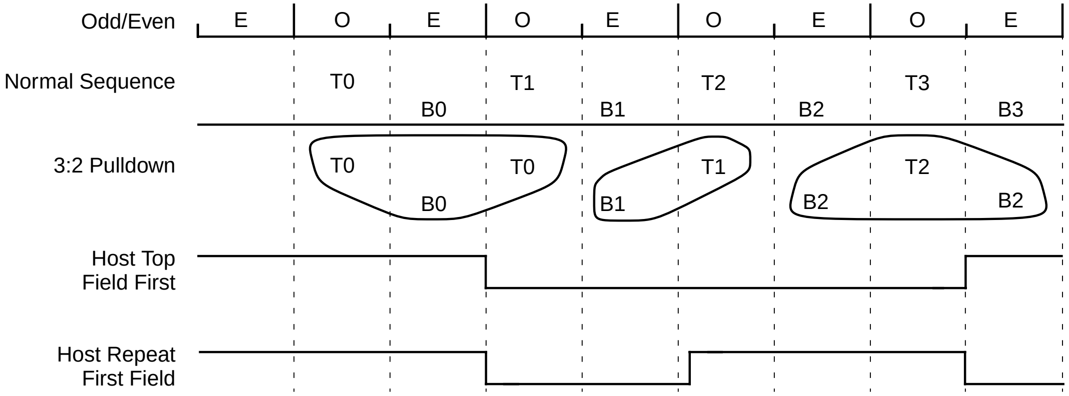
The 3:2 Pulldown from Bitstream bit in Register 275 (page 4-71) defaults to the set state at power-up or reset of the L64020. This causes the internal microcontroller to use the top field first and repeat first field bits in the picture coding extension of the bitstream. If both bits are set, the microcontroller commands the Display Control Subsystem to display the top field first in every frame and repeat it after the bottom field in alternate frames. This displays five fields for every four in the bitstream and is generally used to achieve frame rate conversion from 24 frames/second to 30 frames/second. Other frame rate conversions can also be achieved.

The host can control pulldown by first clearing the 3:2 Pulldown from Bitstream bit. This commands the microcontroller to ignore the pulldown bits in the bitstream. The host must then toggle the Host Top Field First and Host Repeat First Field bits in Register 275 on a frame-by-frame basis as shown in the timing of Figure 10.19.

During 3:2 pulldown, reconstruction is stalled to avoid overwriting the frame memory. Similar to the freeze operation, the pulldown control signals are sampled at the frame boundary.

The First Field bit in Register 275 (page 4-71) and the Last Field bit in Register 276 (page 4-72) can be monitored by the host to determine which field in the frame is currently being displayed. When both bits are cleared, a middle field is being displayed as in pulldown or freeze modes.

Figure 10.19 Pulldown Operation Timing

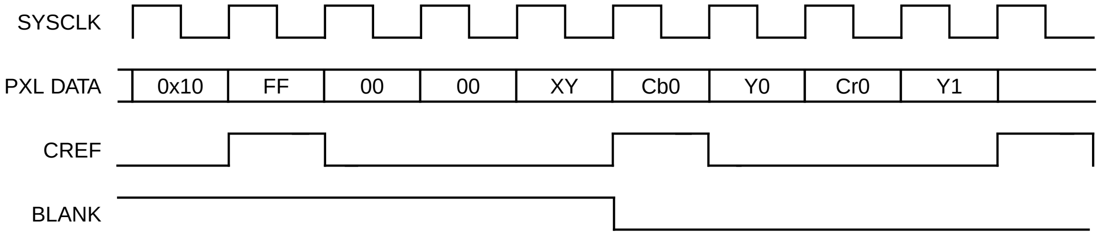


10.14 Video Output Format and Timing

Output timing of video and control signals is shown in Figure 10.20. The Video Interface outputs 8-bit video compatible with 4:2:2 ITU-R BT.601 format. The video is synchronous with the 27-MHz SYSCLK. During the blanking interval, luma data is set to 16 (black level), and Cb and Cr are both set to 128 (zero level). Output data is clipped to ITU-R BT.601 levels where luma has a range of 16 to 235 and chroma has a range of 16 to 240, giving exception to the SAV/EAV timing codes. To insert the ITU-R BT.656 SAV/EAV timing codes into the pixel data stream, the host must set the ITU-R BT.656 Mode bit in Register 284 (page 4-76). The host may optionally set the CrCb 2’s Complement bit in Register 284 to change the chroma outputs to two’s complement format with the center value equal to 0 instead of 128.

The Video Interface also outputs an active high BLANK signal that is based upon the programmed SAV/EAV values for h-blank and v-blank. When high, the CREF output indicates that the current byte on the output data bus is Cb data.

Figure 10.20 Video and Control Output Timing



10.15 Display Controller Interrupts

The Display Controller sets two interrupt bits in response to field timing, the Begin Active Video Interrupt bit and Begin Vertical Blank Interrupt bit, both in Register 1 (page 4-5). If the bits are not masked, INTRn is asserted to the host when either bit is set. The time at which these interrupts occur within each field time is based upon how the active display area is programmed by the host.

The host controls the location of the active display area by programming the SAV/EAV code parameters. Regardless of whether the target system requires the SAV/EAV tokens in the video stream, the SAV/EAV parameters must be programmed for proper operation of the Display Controller.

The Begin Active Video Interrupt occurs during the EAV when there is a transition in the Vcode from 1 to 0. The host processor controls this transition by programming the Vcode Zero bits in Register 303 (page 4-79).

The Begin Vertical Blank Interrupt occurs during the EAV when the Vcode transitions from 0 to 1. The host controls this transition by programming the Vcode Even bits in Registers 303 and 304.

Chapter 11
Audio Decoder Module

This chapter describes the operation of the L64020 Audio Decoder. The Audio Decoder processes three different audio input bitstreams; Dolby Digital, Linear PCM, and MPEG (MUSICAM). It also includes two output interfaces, a serial DAC interface, and a IEC958 S/P DIF interface.

This chapter consists of the following sections:

11.1 Features

- ♦ MPEG Decoder
  - – Decodes 1 or 2 main channels of audio
  - – Sampling Frequencies (Fs) =16, 22.05, 24, 32, 44.1, and 48 kHz
  - – Bit Rate = 8 to 448 Kbps
  - – Layer I includes 384 samples per frame, Layer II contains 1152 samples per frame.
  - – 16 bits per sample resolution
- ♦ Dolby Digital Decoder
  - – Decodes up to 5 channels but downmixes them to 2 channels. Accepts the following input channel combinations:
    - ◊ 1/0: mono (M)
    - ◊ 2/0: two channel stereo (L, R)
    - ◊ 2/1: left right stereo + one surround channel (L, R, S)
    - ◊ 2/2: two front stereo + two background surround sound (L, R, Ls, Rs)
    - ◊ 3/0: stereo + center (L, R, C)
    - ◊ 3/1: stereo + center + surround sound (L, R, C, S)
    - ◊ 3/2: complete 5 channels: stereo, center, left and right surround (L, R, C, Ls, Rs),
    - ◊ 1+1: dual channel (Ch1, Ch2)
  - – Sampling Frequencies (Fs) = 32, 44.1, and 48 kHz
  - – Bit Rate = 32 to 640 Kbps
  - – Reproduces 1536 audio samples for each channel per frame at 16 bits per sample resolution.
  - – Error handling: mute on CRC error or decoding error
  - – Karaoke capable: None, V1 only, V2 only, and V1+V2 karaoke output format selections
- ♦ Linear PCM Decoder
  - – Decodes 1 or 2 channels. Higher channel data is discarded.

  - Sampling Frequencies (Fs) = 48 kHz or 96 kHz (96 kHz is decimated to 48 kHz for the IEC958 S/P DIF interface output).
  - Quantization accuracy: 16, 20, or 24 bits. For S/P DIF, all the samples are truncated to the most significant 16 bits.

- MPEG and Dolby Digital Formatters
  - Sampling Frequencies (Fs) = 16, 22.05, 24, 32, 44.1, and 48 kHz for MPEG; 32, 44.1, and 48 kHz for Dolby Digital.
  - Compressed audio data is packed into 16-bit packet and sent to S/P DIF output.

- DAC Interface
  - Outputs two channels of decoded MPEG, Dolby Digital, or Linear PCM audio.
  - CD Bypass Mode to DAC and S/P DIP interfaces
  - 32 bits per sample per channel serial output

- S/P DIF Interface
  - Outputs two channels of decoded or formatted MPEG audio, decoded or formatted Dolby Digital audio, or decoded Linear PCM audio
  - 32 bits per sample per channel serial output; each bit represented by two binary states

11.2 Audio Decoder Overview

Figure 11.1 shows a block diagram of the Audio Decoder. The host can select one of seven modes of decoder operation by programming the Audio Decoder Mode Select [2:0] bits in Register 357 (page 4-93).

Important: The host must clear the Audio Formatter Start/Stop bit in Register 356 (page 4-92) before selecting Audio Decoder Mode 0b000, 0b001, 0b100, or 0b101 (see Table 11.1). That is, formatters must be stopped before selecting nonformatter modes and not started unless the mode is changed to include a formatter.

Table 11.1 Audio Decoder Modes

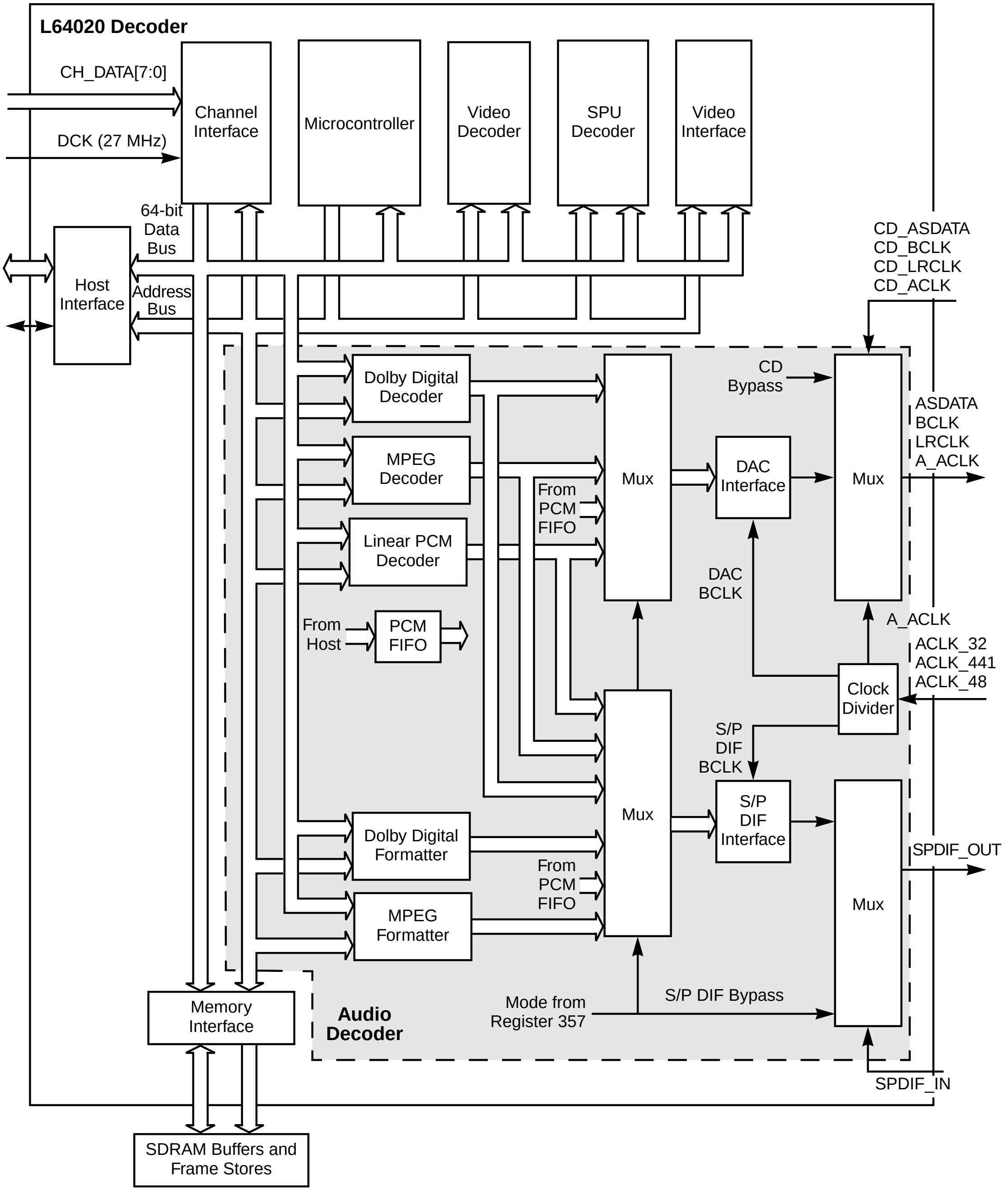
Mode Bits [2:0]	DAC Output	S/P DIF (IEC958) Output
0b000	MPEG decoder	MPEG decoder output PCM samples converted to IEC958 format
0b001	Dolby Digital Decoder	Dolby Digital Decoder output PCM samples converted to IEC958 format
0b010	MPEG Decoder	MPEG Formatter
0b011	Dolby Digital Decoder	Dolby Digital Formatter
0b100	Linear PCM Decoder	Linear PCM Decoder Note: If the sample frequency in the Linear PCM bitstream is 96 kHz, then the IEC958 output is derived from an on-chip filter that converts from 96-kHz to 48-kHz sample frequency.
0b101	Linear PCM Decoder output decimated through on-chip filter to convert from 96-kHz to 48-kHz sample rate. This mode should be selected if the output desired is through a DAC that supports a 48-kHz sample frequency only.	Same as DAC, converted to IEC958 format.
0b110	CD Bypass	S/P DIF bypass
0b111	PCM FIFO	PCM FIFO

When the host starts the selected audio decoder, audio frames/packets are retrieved from the Audio ES Channel Buffer in SDRAM and decoded and formatted. The three decoders parse most of the parameters from the bitstream and store them in registers in the Host Interface. The host reads these registers and writes decoder commands to other registers to modify the audio.

Both the MPEG and the Dolby Digital Decoders reproduce 16-bit audio samples from the bitstream with 24-bit internal processing precision. The packetized Linear PCM samples in the DVD stream can be 16, 20, or 24 bits in length. The host can override the bitstream sample resolution for all of the decoders by setting the Overwrite Quantization bit in

Register 366 (page 4-102) and programming the Host Quantization bits in the same register for 16, 20, or 24-bit samples. The decoders truncate or extend the samples accordingly.

Figure 11.1 L64020 Audio Decoder Block Diagram



Each audio frame in Dolby Digital, MPEG, and Linear PCM streams starts with a sync word and contains a fixed number of bytes; that is, every Dolby Digital frame has the same number of bytes, every MPEG Layer I frame has the same number of bytes, etc. Once instructed to start, the audio decoder looks for the first sync word and starts to decode immediately after detecting it. However, the decoder does not go into "in sync" state until it also detects the sync word in the following frame. Once synchronized, the decoder loses synchronization only when it fails to locate the sync word where it expects it to be in the next frame. When this occurs, the decoder continues searching and sets the Audio Sync Error Interrupt bit in Register 4 (page 4-10). If this bit is not masked, INTRn is asserted to the host. When the decoder successfully finds three consecutive sync words, it sets the Audio Sync Recovery Interrupt bit and asserts INTRn to the host. Also, each time a sync word is detected, the Audio Sync Code Detect Interrupt bit in Register 1 (page 4-4) is set and INTRn is asserted.

The decoders also detect CRC errors (corrupted audio data) and illegal bit errors (invalid bitstream parameters). When either is encountered, the decoders set the Audio CRC or Illegal Bit Error Interrupt bit in Register 4, reset their internal counters and state machines, and start searching for the next sync word.

If the host sets the Mute on Error bit in Register 358 (page 4-96), the audio output is muted during any of the previous errors to avoid sending out bad samples (noise) to the speaker(s). When the Audio Decoder Module is stopped, the decoders stay in the idle state and the read and write pointers of the Audio ES Channel Buffer are reset.

The two formatters take the encoded audio frames from the Audio ES Channel Buffer, add a preamble to them, and pad them out into S/P DIF bursts. Both formatters can run simultaneously with their decoder counterparts. The formatters detect out-of-sync conditions with the decoders and add pause bursts as necessary to resynchronize. The host can substitute zeros for the pause bursts.

The 16-, 20-, or 24-bit decoded audio samples that are input to the DAC interface are converted to 32-bit serial output (ASDATA). This format is obtained by sign extension of the input data.

The S/P DIF interface only supports 16-bit input data samples. The input to the S/P DIF interface comes either from the decoders or from the audio formatters. It produces a fixed-length, 32-bit packet per input sample and then represents each bit with two consecutive binary states (biphase mark) as a clock self-recovery technique.

In the PCM FIFO mode, the host writes decoded PCM audio bytes into a FIFO through a register in the Host Interface. According to the mode selected, the outputs of the appropriate decoder and formatter and the PCM FIFO are steered through the two multiplexers to the DAC and S/P DIF interfaces.

One of the ACLK inputs is selected by the host and divided into three clocks according to host divider selection. The two derived BCLKs convert the parallel inputs to the interfaces to serial outputs. The DAC BCLK is supplied to the external DAC as a bit clock. The divider also supplies the DAC clock, A_ACLK, to the output multiplexer. The host must select the proper divider to match the DAC.

The two output multiplexers can bypass the Audio Decoder entirely and turn the S/P DIF and CD inputs around to the output pins. This feature lets the L64020 Decoder share a DAC in the system with a CD decoder.

11.3 Decoding Flow Control

The first part of this section gives brief descriptions of the following register bits and fields; Audio Decoder Play Mode, Audio Decoder Start/Stop, Audio Formatter Play Mode, and Audio Formatter Start/Stop. The second part describes the procedures you should follow to start and stop the Audio Decoder properly.

11.3.1 Audio Decoder Play Mode

The Audio Decoder Play Mode bits in Register 355 (page 4-92) pause the decoder, set it to normal play, and increase or decrease the play speed by skipping or repeating samples. The decimation or interpolation factors are one out of every 16 samples for the MPEG and Dolby Digital Decoders and one out of every 8 samples for the Linear PCM Decoder. The mode selections are:

♦ 0b00 - Pause. The decoder is paused and the last pair of PCM and S/P DIF samples are repeated, effectively muting the output, until the play mode is changed. The Audio ES Channel Buffer keeps filling with new data. The decoder continues to decode and write to the output buffer until the output buffer overflows and stops the audio read pointer of the Audio ES Channel Buffer. Since there is no guarantee that the channel read pointer will stop at the end of a frame, decoder resynchronization is required when the mode is changed again.

A prolonged audio pause will cause the Audio ES Channel Buffer to overflow and the system parser to deassert the channel request signals, stopping video as well as audio.

- 0b01 - Normal Play. The decoder is playing at normal speed. When the PCM output buffer or S/P DIF output buffer empties and the next output signal is requested by the audio DAC or S/P DIF decoder, the last pair of output samples is repeated.
- 0b10 - Fast Play. The Dolby Digital and MPEG Decoder skip one out of every 16 pairs of samples and play at 16/15 normal speed. The Linear PCM Decoder performs fast play at 8/7 normal speed. When the Linear PCM bitstream is at 48 kHz, every eighth pair of PCM samples is skipped. When it is 96 kHz, every fifteenth and sixteenth pair of PCM samples and every eighth pair of S/P DIF samples are skipped.
- 0b11 - Slow Play. The Dolby Digital and MPEG Decoders repeat every sixteenth pair of samples and play at 16/17 normal speed. The Linear PCM Decoder runs at 8/9 normal speed in Slow Play Mode. When the Linear PCM bitstream is at 48 kHz, every eighth pair of PCM samples is played twice. When Fs is 96 kHz, every fifteenth and sixteenth pairs of PCM samples and every eighth of S/PDIF samples are played twice.

The host can determine the decoder mode at any time by reading the Audio Decoder Play Mode Status bits in Register 354 (page 4-91). The two bits are encoded identically to the play mode bits.

11.3.2 Audio Decoder Start/Stop

The Audio Decoder Start/Stop bit in Register 355 (page 4-92) is used to control both the selected audio decoder and its channel buffer read pointer. When the Audio Decoder Start/Stop bit is set, the Audio Decoder operates according to the play mode setting described in the previous section. When the start/stop bit is cleared, the Audio Decoder is stopped at the end of the current frame and the audio read pointer in the Audio

ES Channel Buffer is reset to the write pointer location. Any unread audio data is lost to the decoder. The S/P DIF read pointer is not affected, so the selected formatter can still run.

Once the decoder is stopped, the host should set the play mode bits to Pause Mode. On restart, the host should first set the start bit, wait for some unread audio to accumulate in the channel buffer, and then change the play mode bits from pause to play.

11.3.3 Audio Formatter Play Mode

The Audio Formatter bits in Register 356 (page 4-93) control the play mode of the selected formatter. There is no fast play or slow play mode for the formatters, so codes 0b10 and 0b11 are reserved.

- 0b00 - Formatter is paused.
- 0b01 - Formatter is in normal play.

11.3.4 Audio Formatter Start/Stop

When the Audio Formatter Start/Stop bit is set, the selected formatter is in the mode programmed into the play mode bits. When the start/stop bit is cleared, the formatter is stopped and the S/P DIF read pointer is reset to the Audio ES Channel Buffer write pointer. The audio read pointer is not affected, so the selected audio decoder can still run.

Once the formatter is stopped, the host should set the play mode bits to pause mode. On restart, the host should first set the start bit, wait for some unread audio to accumulate in the channel buffer, and then change the play mode bits from pause to play.

Important: The host must clear the Audio Formatter Start/Stop bit before selecting Audio Module Mode 0b000, 0b001, 0b100, or 0b101 (see Table 11.1). That is, formatters must be stopped before selecting non-formatter modes and not started unless the mode is changed to include a formatter.

11.3.5 Autostart

The selected audio decoder and formatter can be autostarted at a specified System Reference Clock (SCR) count. The registers associated with autostart are listed in Table 11.2.

Table 11.2 Audio Autostart Registers

Name	Register/Bits	Page Ref.
SCR Compare/Capture Mode	17 bits 0 and 1	4-15
SCR Compare Audio	20 through 23	4-17
Audio Start on Compare	19 bit 0	4-17
SCR Compare Audio Interrupt	1 bit 2	4-4

The host should use the following sequence for autostart:

1. Clear the Audio Decoder and Audio Formatter Start/Stop bits.
2. Change the Audio Decoder and Audio Formatter Play Mode to pause.
3. Program the SCR Compare/Capture Mode bits to compare.
4. Write the SCR value on which to start into the SCR Compare Audio registers.
5. Set the Audio Start on Compare bit.
6. Set the Audio Decoder and Audio Formatter Start/Stop bits.

When the SCR counter value equals that written into the SCR Compare Audio registers, an autostart pulse is generated to change the Play Mode of the decoder and formatter to normal play. The SCR Compare Audio Interrupt bit is set and INTRn is asserted to the host if the interrupt is not masked. Also, the SCR Compare/Capture Mode bits are reset to no compare or capture and the Audio Start on Compare bit is cleared.

11.4 MPEG Audio Decoder

The L64020 MPEG Decoder supports Layer I and Layer II of the MPEG-1 audio compression and MPEG-2 low bit rate decoding. For

MPEG-2 multichannel audio streams, the L64020 decodes the left and right channels and ignores the others.

11.4.1 MPEG Audio Syntax

The basic MPEG audio bitstream syntax is shown in Figure 11.2. The four bytes of frame header contain the 12-bit sync word and information on the characteristics of the audio, such as audio layer, sample frequency, bit rate, mode (mono, stereo, joint stereo, or dual channel), copyright, etc. If the protection bit in the header is set, the header is followed by a 2-byte CRC. In the Layer I stream, the CRC is for the audio data up to the scalefactors boundary. Similarly, in Layer II, the CRC is for the data up to the end of the scalefactor index.

The length of the scalefactors is fixed at six bits. The encoded subband samples can vary from 2 to 15 bits in length. The length of the samples per subband is indicated by the four bits of allocation information for each subband at the beginning of the data.

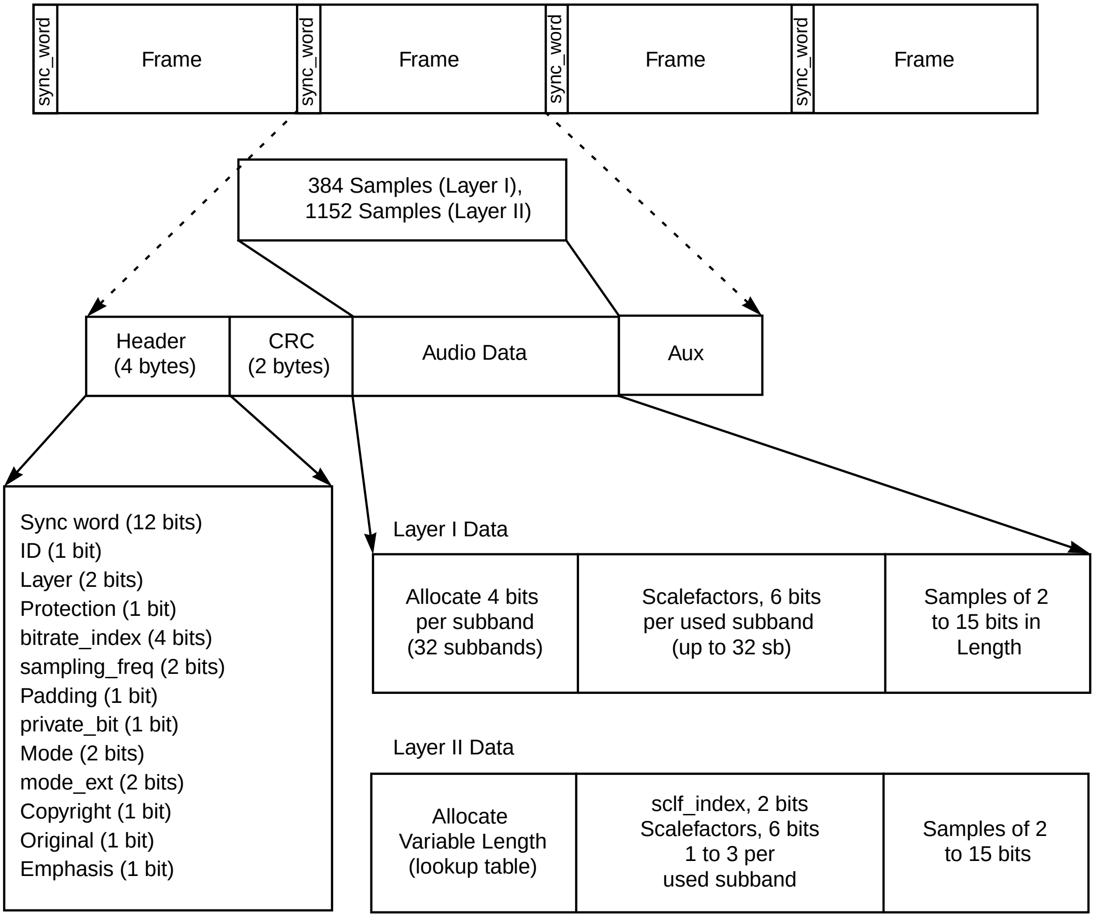
Layer I supports bit rates from 192 to 448 Kbits/sec. Layer II, which uses a more complex encoding model, provides CD quality audio at 128 Kbits/sec/channel and performs better compression of stereo signals. The highest supported bit rate for Layer II is 384 Kbits/sec.

Note: “Free format” bit rate is not supported in the L64020 DVD Decoder.

There are 384 PCM sample pairs encoded in each Layer I frame and 1152 PCM sample pairs in each Layer II frame. When a 48-kHz sampling frequency is assumed, the decoding time is 8 ms for a Layer 1 frame and 24 ms for a Layer II frame.

The MPEG-2 audio compression standard is compatible with MPEG-1. It provides a more sophisticated encoding scheme with low bit-rate support and encodes up to five audio channels. The L64020 Audio Decoder complies with the MPEG-2 low bit-rate standard and is able to decode a wide range of audio bit rates from 8 to 448 Kbps. For the MPEG-2 multichannel audio streams, the L64020 audio decoder processes only left and right channels.

Figure 11.2 MPEG Audio Bitstream Syntax



11.4.2 MPEG Audio Decoding

MPEG audio encoding is performed by transforming the input signals from the time domain to the frequency domain and dividing them into 32 frequency subband samples. The subband samples are then quantized, normalized, and encoded using a variable length encoding scheme. For decoding, this process is reversed.

Figure 11.3 shows the MPEG audio decoding flow.

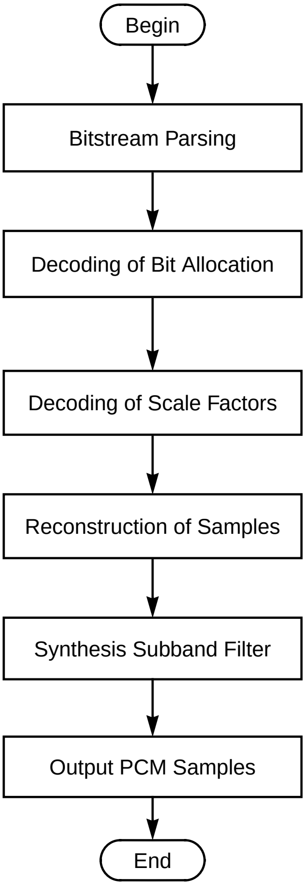
- *Input bitstream parsing:* The audio frame is unpacked and parsed, and the various pieces of coding information are demultiplexed.

- *Bit allocation decoding:* The bit allocation information is decoded first and is used to parse the scalefactors and audio subband samples later.
- *Scale factor decoding:* The scalefactor index (Layer II only) and scale factors are unpacked. The 32 audio subband samples are also parsed using the bit allocation information from the bitstream.
- *Reconstruction of samples:* The subband data samples are requantized and denormalized.
- *Subband synthesis:* Subband synthesis converts the frequency domain subband vectors back to time domain PCM samples by performing the inverse transformation.
- *Output PCM samples:* The decoder transfers 32, 16-bit PCM samples at a time to be played by the output interface modules.

The host can override the bitstream sample resolution for the decoder by setting the Overwrite Quantization bit in Register 366 (page 4-102) and programming the Host Quantization bits in the same register for 20- or 24-bit samples. The decoder extends the samples accordingly.

The host can also scale the output PCM samples down from their bitstream levels in increments of 1/256 of the levels by writing to Register 362, PCM Scale [7:0], page 4-97. A setting of 0x00 in this register mutes the audio output.

Figure 11.3 MPEG Audio Decoding Flow


Begin
Bitstream Parsing
Decoding of Bit Allocation
Decoding of Scale Factors
Reconstruction of Samples
Synthesis Subband Filter
Output PCM Samples
End

11.5 Dolby Digital Audio Decoder

This section includes a description of the Dolby Digital syntax, decoding by the L64020, the Karaoke Mode, and the Compression Mode.

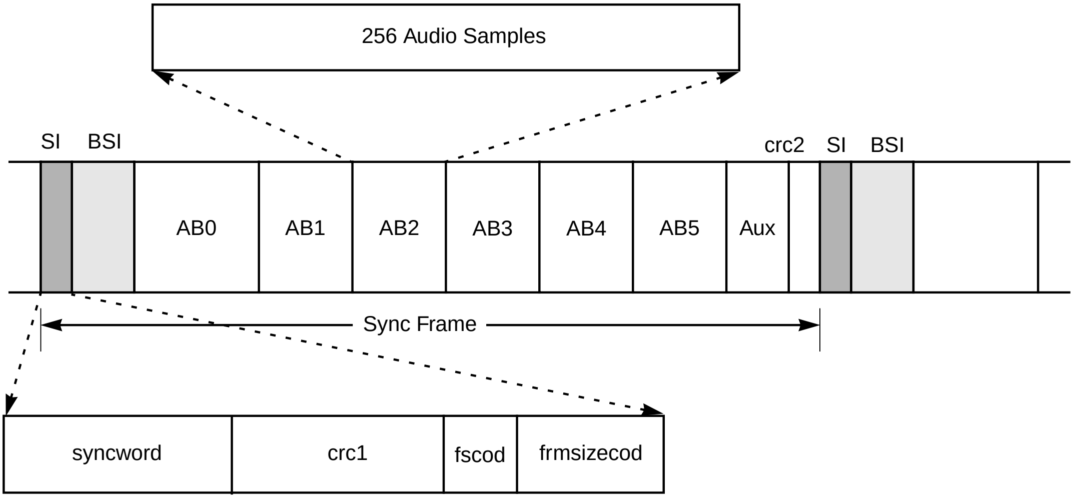
11.5.1 Dolby Digital Syntax

The basic Dolby Digital bitstream syntax is shown in Figure 11.4. An encoded Dolby Digital audio bitstream is made up of a sequence of synchronization frames. Each frame contains a Synchronization Information (SI) header, a Bitstream Side Information (BSI) field, six coded audio blocks (AB0 to AB5), and a CRC word. The SI header includes a sync word, an optional use CRC word, and the frame length fields (fscod and frmsizecod). The BSI field contains parameters describing the coded audio. Each audio block includes 256 audio samples.

The following constraints on the encoded bitstream to the Dolby Digital Decoder allow the L64020 to use smaller input memory buffers:

1. The size of AB0 and AB1 combined never exceeds 5/8 of the frame.
2. The sum of AB5 mantissa data and Aux data never exceeds the final 3/8 of the frame.
3. AB0 always contains all the necessary information to correctly begin decoding the bitstream.

Figure 11.4 Dolby Digital Syntax



11.5.2 Dolby Digital Decoding

The Dolby Digital decoding flow is shown in Figure 11.5 and described in the steps following the figure.

Step 1. Synchronization and Error Detection

The Dolby Digital bitstream format allows rapid synchronization by the decoder. The 16-bit sync word in the bitstream has a 2.5% probability of false detection.

When a synchronization pattern is detected, the decoder assumes that it is in sync and starts decoding the audio blocks. Since the CRC1 word is read at the beginning of the frame and it covers the first 5/8 of the frame, the result of a CRC1 check may be available after only 5/8 of the frame has been received. The decoder can check for CRC2 once the entire frame is read. If a CRC1 or CRC2 mismatch is detected, the decoder resets itself and starts to search for the next synchronization word. A CRC error sets the Audio CRC Error or Illegal Bit Error Interrupt bit in Register 4 (page 4-10), asserts INTRn to the host if the interrupt is not masked, and mutes the audio output.

Step 2. Unpack the Bitstream Information (BSI) Field

Some of the BSI data may be copied from the Audio ES Channel Buffer to dedicated registers for the host, some may be copied to specific working memory locations, and some may simply be left in the Audio ES Channel Buffer with pointers to them saved in another location for use when the information is required.

Step 3. Decode Exponent

The audio data in the Dolby Digital bitstream is the compressed frequency coefficients. These coefficients are represented in the form of an exponent and a mantissa. The exponent, which could range from 0 to 24 bits, indicates the number of leading zeroes in the binary representation of a frequency coefficient. The exponent must be decoded first to collect enough information to decode the mantissa. In the Dolby Digital bitstream, there are exponents for all independent channels, all coupled channels, and for the coupling and low frequency effects channels.

Step 4. Bit Allocation

The number of bits allocated for each mantissa is determined by the bit allocation operation. The inputs to the bit allocation computation are the decoded exponents and bit allocation side information. The outputs of the bit allocation computation are a set of bit allocation pointers (baps), one for each coded mantissa. The bap indicates the quantizer used for the mantissa and how many bits in the bitstream were used to encode each mantissa.

Step 5. Process Mantissa

The coarsely quantized mantissas make up the bulk of the Dolby Digital data stream. Each mantissa is quantized to a level of precision indicated by the corresponding bap. The mantissa data is unpacked by peeling off groups of bits as indicated by the baps. Grouped mantissas must be ungrouped. The individual coded mantissa values are converted into dequantized values. Mantissas having a zero-valued bap may be reproduced as either zero or by a random dither value (under the control of the dither flag).

Step 6. Decoupling

When coupling is in use, the channels that are coupled must be decoupled. Decoupling involves reconstructing the high frequency section (exponent and mantissa) of each coupled channel from the common coupling channel and the coupling coordinates for the individual channel. Within each coupling band, the coupling channel coefficients are multiplied by the individual channel coupling coordinates.

Step 7. Rematrixing

In the stereo 2/0 mode, rematrixing may be employed. When the flag indicates a band is rematrixed, the coefficients encoded in the bitstream are sum and difference values instead of the original left and right channel samples.

$$Lr = 1/2\ (L + R)$$

$$Rr = 1/2\ (L - R)$$

If the original left and right channels are highly correlated, or identical, this coding technique results in an Lr only signal since the Rr value is close to zero. This saves about half of the transmitted data and increases the accuracy of the Lr data.

Step 8. Dynamic Range Control

A dynamic range control value (dynrange_value) may be included in the bitstream for each audio block. The decoder, by default, uses this value to alter the magnitude of the coefficients (exponent and mantissa). The host can also specify dynscale values in Registers 360 and 361 (page 4-96). The dynscale factors are in increments of 1/256. The final sample value is computed using the following equation:

final sample value = dynrange_value * dynscale(high/low) *
decoded Dolby Digital samples

Step 9. Inverse Transform and Window Overlapping

The decoding steps described above result in a set of frequency coefficients for each encoded channel. The inverse transform converts the blocks of frequency coefficients into blocks of time samples. After that, the individual block of time samples must be windowed, and adjacent blocks must be overlapped and added together to reconstruct the final continuous time output PCM audio signal.

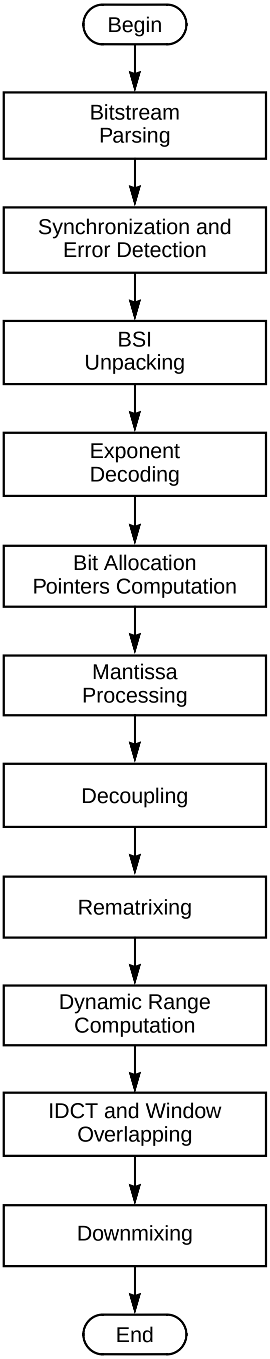
Step 10. Downmixing

When the number of output channels is less than the number of transmitted channels in the audio stream, downmixing is required to reproduce the complete audio program. The host can set the Dolby Digital Downmix Mode bit in Register 358 (page 4-95) to downmix into a matrix surround stereo pair LtRt (also called Dolby surround stereo) or clear it to downmix to a conventional stereo signal LoRo. The L64020 defaults to the latter mode at reset and power on.

The downmixed stereo signal (LoRo or LtRt) may be further mixed into mono, M, by a simple summation of the 2 channels. The conventional LoRo downmixing is preferred in this case since the surround information is lost in the LtRt channels when combined.

The host can also scale the output PCM samples down from their decoded levels in increments of 1/256 of the levels by writing to Register 362, PCM Scale [7:0], page 4-97. A setting of 0x00 in this register mutes the audio output.

Figure 11.5 Dolby Digital Decoding Flow


Begin
Bitstream Parsing
Synchronization and Error Detection
BSI Unpacking
Exponent Decoding
Bit Allocation Pointers Computation
Mantissa Processing
Decoupling
Rematrixing
Dynamic Range Computation
IDCT and Window Overlapping
Downmixing
End

The general downmixing equations for conventional LoRo and Mo in 3/2 mode (three front channels, two background surround) and in single surround channel mode (three front, one back) are shown in Table 11.3.

Table 11.3 Conventional LoRo Downmixing Equations

	3/2 Mode (Left, Right, Center, Ls, Rs)	3/1 Mode (Left, Right, Center, S)
Lo	1.0 * L + clevel * C + slevel * Ls	1.0 * L + clevel * C + 0.7 * slevel * S
Ro	1.0 * R + clevel * C + slevel * Rs	1.0 * R + clevel * C + 0.7 * slevel * S
Mo	1.0 * L + 2.0 * clevel * C + 1.0 * R + slevel * (Ls + Rs)	1.0 * R + 2 * clevel * C + 1.4 * slevel * S

The general downmixing equations for Dolby Digital surround downmixing, LtRt, with 3/2 mode (three front channels, two background surround) and with single surround channel mode (three front, one back surround) are shown in Table 11.4.

Table 11.4 Dolby Surround Stereo Downmixing Equations

	3/2 Mode (Left, Right, Center, Ls, Rs)	3/1 Mode (Left, Right, Center, S)
Lt	1.0 * L + 0.707 * C - 0.707 * Ls - 0.707 * Rs	1.0 * L + 0.707 * C - 0.707 * S
Rt	1.0 * L + 0.707 * C+ 0.707 * Ls + 0.707 * Rs	1.0 * R + 0.707 * C - 0.707 * S
Mt [1]	2.0 * L + 1.414 * clevel * C	1.0 * R + 1.414 * C

[1] Note that the surrounding channels are lost in Mono transmission.

The dynamic range control value transmitted from the BSI is applied to scale the final downmixed coefficients up or down.

The host can override the bitstream sample resolution for the decoder by setting the Overwrite Quantization bit in Register 366 (page 4-102) and programming the Host Quantization bits in the same register for 20- or 24-bit samples. The decoder extends the samples accordingly.

11.5.3 Karaoke Mode

The L64020 Dolby Digital Decoder also includes Karaoke Mode. In Karaoke Mode, the Dolby Digital bitstream carries audio channels designated as L and R (2-channel stereo music), M (guide melody), and V1 and V2 (one or two vocal tracks). The decoder is *Karaoke Capable* meaning that it allows the listeners to optionally reproduce the V1 and/or

V2 channels, and to adjust the relative levels (mixing balance) of the M, V1, and V2 channels.

The implementation of the karaoke feature relies on a technique similar to downmixing. The equations for mixing Karaoke channels are:

$$L_{ko} = (1.0 \times L) + (\text{clevel} \times M) + (\alpha_1 \times V1) + (\beta_1 \times V2)$$

$$R_{ko} = (1.0 \times R) + (\text{clevel} \times M) + (\alpha_2 \times V1) + (\beta_2 \times V2)$$

where: L is the left channel music
R is the right channel music
M is the guide melody
clevel is 0 or 1
V1 is vocal 1
V2 is vocal 2
α*1*, α*2*, β*1*, and β*2* are set by the host

The host can set the Karaoke Center Level On bit in Register 364 (page 4-100) to set clevel at 1 and turn on the guide melody or clear the bit to turn it off. Also, the host can program the Karaoke Mode bits in the same register to adjust the values of α*1,* α*2,* β*1*, and β*2* as shown in Table 11.5.

Table 11.5 Karaoke Modes

Karaoke Mode	Karaoke Output	α***1***	α***2***	β***1***	β***2***
00	No vocal output	0	0	0	0
01	Vocal 1 only	0.7	0.7	0	0
10	Vocal 2 only	0	0	0.7	0.7
11	Vocal 1 + Vocal 2	1.0	0	0	1.0

11.5.4 Dynamic Range Compression Mode

The L64020 defaults to the Dolby Digital Line-out mode at reset and power on. The host can program the Compression Mode [1:0] bits in Register 358 (page 4-94) for other compression modes. Refer to the Dolby Digital specifications for descriptions of the compression modes.

11.6 Linear PCM Audio Decoder

Linear PCM is a high-fidelity audio coding technique. Unlike Dolby Digital and Musicam, Linear PCM is not lossy compressed so that decoders can achieve high-quality audio reproduction. As shown in Figure 11.6, a Linear PCM bitstream is divided into Groups of Audio Frames (GOF). Each GOF consists of several audio packs. Each audio pack contains a header followed by Linear PCM data packets.

Figure 11.6 Linear PCM Group of Frames Syntax

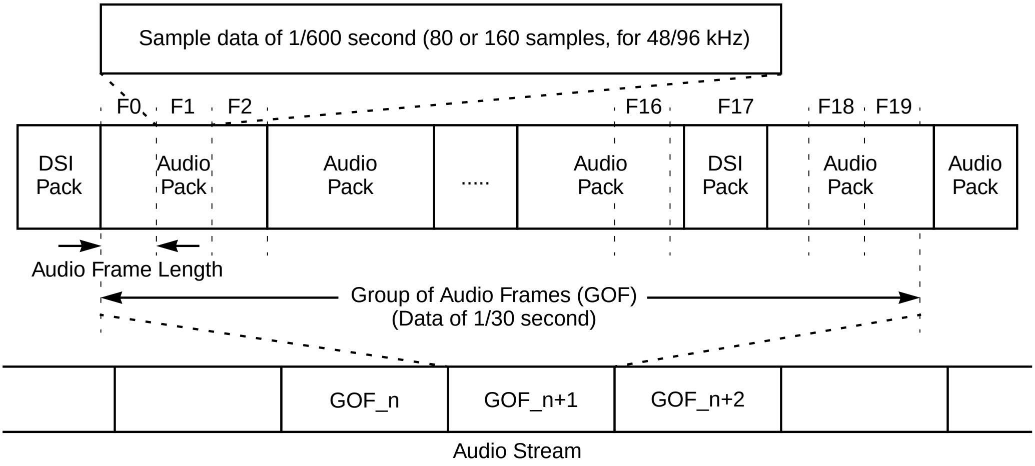
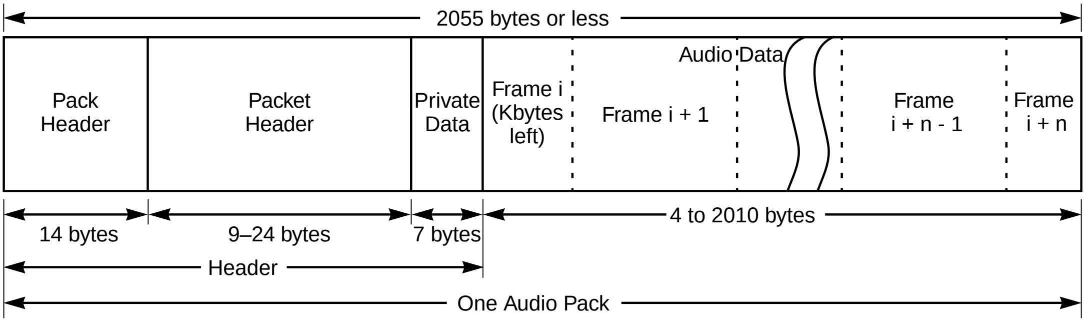


Figure 11.7 Linear PCM Packet Syntax



11.6.1 Packet Header Syntax

As shown in Figure 11.7, every Linear PCM pack contains a pack header, a packet header, a private data section, and audio data. The parameters inside the private data section include: a substream ID; a packet length code; and audio frame information such as frame number, mute, emphasis, first access pointer, number of channels, quantization, sampling frequency, and dynamic range control value. The packet length, quantization, sampling frequency, and number of channels are defined in Table 11.6.

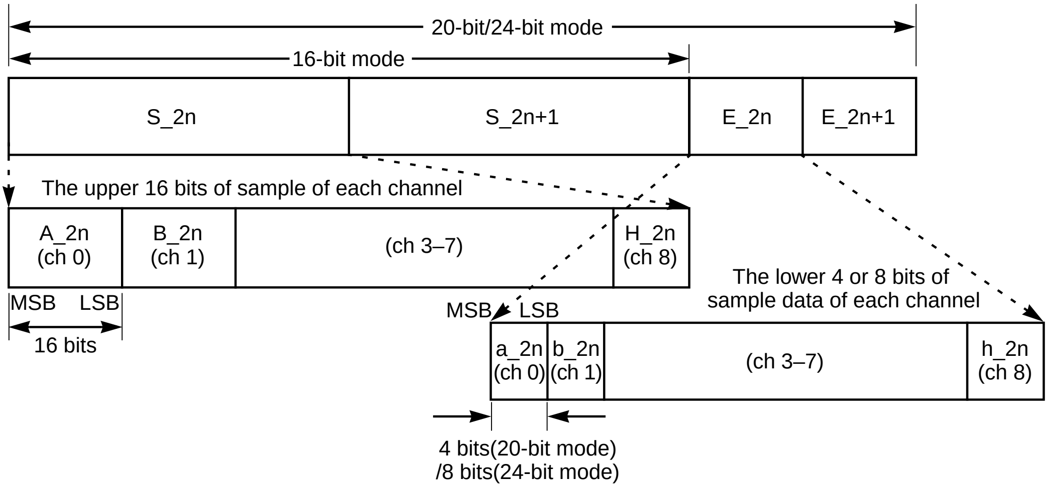
Table 11.6 Valid Linear PCM Stream Permutations

Number of Channels	Sampling Frequency (kHz)	Quantization (Bits)	Maximum Number of Samples in One Pack	Data Size (Bytes)
1 (mono)	48 / 96	16	1004	2008
	48 / 96	20	804	2010
	48 / 96	24	670	2010
2 (stereo)	48 / 96	16	502	2008
	48 / 96	20	402	2010
	48 / 96	24	334	2004
3	48 / 96	16	334	2004
	48 / 96	20	268	2010
	48	24	222	1998
4	48 / 96	16	250	2000
	48	20	200	2000
	48	24	166	1992
5	48	16	200	2000
	48	20	160	2000
	48	24	134	2010
6	48	16	166	1992
	48	20	134	2010
7	48	16	142	1988
8	48	16	124	1984

11.6.2 Synchronization

The Preparser in the Channel Interface substitutes the original substream Linear PCM ID with an 8-byte sync word to mark the beginning of each Linear PCM packet. The Linear PCM Decoder searches for and synchronizes to the sync word. If the decoder loses synchronization, it sets the Audio Sync Error Interrupt bit in Register 4 (page 4-10), asserts INTRn to the host if the interrupt is not masked, mutes the audio output, and searches for the next sync word.

Figure 11.8 Linear PCM Audio Sample Syntax



Linear PCM bitstream samples can be 16, 20 or 24 bits as shown in Figure 11.8. Twenty or 24-bit samples are divided into the upper 16 bits and the lower 4 or 8 bits. The output PCM samples to the DAC interface can be 16, 20, or 24 bits in length. On the other hand, the S/P DIF interface accepts only 16-bit samples. The last 4 or 8 bits of 20- or 24-bit samples are truncated for the S/P DIF interface.

The host can override the bitstream sample resolution for the decoder by setting the Overwrite Quantization bit in Register 366 (page 4-102) and programming the Host Quantization bits in the same register for 16-, 20-, or 24-bit samples. The decoder truncates or extends the samples accordingly.

If the data in the Linear PCM bitstream does not agree with the options available, the decoder sets the Context Error Interrupt bit in Register 4 (page 4-9), asserts INTRn to the host if the interrupt is not masked, mutes the audio output, and searches for the next sync word.

Dynamic Range Control is a compressed gain value that should be applied to all the samples in an audio frame. One audio frame has 80 or 160 samples when the sampling frequency is 48 or 96 kHz, respectively. One audio frame can extend across an audio pack boundary. The first byte location of an audio frame is defined by the first access pointer. All Linear PCM samples in one audio frame have a unique dynamic range control gain value, even if the audio frame is separated by an audio pack boundary. The host can turn Dynamic Range Control on or off by setting or clearing the Dynamic Range On bit in Register 364 (page 4-100). When the Dynamic Range Control is off, the default gain value is one. When Dynamic Range Control is on, the decoder uses the dynamic range control value (dynrange_value) included in the bitstream for each audio frame. The host can also specify dynscale values in Registers 360 and 361 (page 4-96) to scale the bitstream dynrange_value. The dynscale factors are in increments of 1/256.

The host can also program a direct scale factor into the PCM Scale [7:0] bits in Register 362 (page 4-97). Settings here are also in 1/256 increments. The final sample value is computed using the following equation:

final sample value = [|dynrange_value - 1|* dynscale(high/low) + 1]
*PCM Scale * decoded sample

The Linear PCM Decoder produces stereo PCM and S/P DIF outputs. Up to two channels of Linear PCM samples can be decoded; additional channel samples are dropped. When the Linear PCM bitstream contains only one channel, PCM and S/P DIF output samples in the first channel are duplicated in the second channel.

11.6.3 Other Host Controls and Status

The bitstream mute, emphasis, quantization, and sampling frequency information is written to Register 352 (page 4-90) for the host. The audio_frm_num and num_of_audio_ch are written to Register 351. When the mute bit in the audio packet is 1, PCM samples are muted by the output DAC and S/P DIF interface.

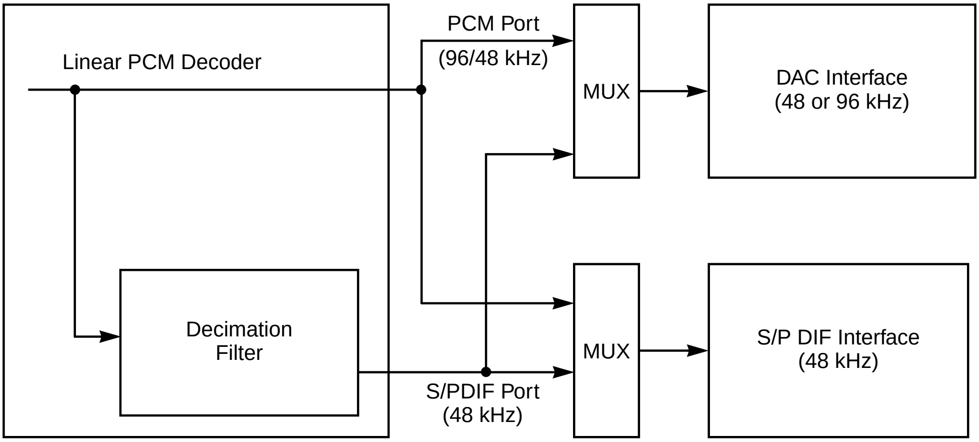
The host can program the Audio Decoder Play Mode bits in Register 355 (page 4-92) to place the Linear PCM Decoder in normal play, pause, fast play, or slow play mode. The current play mode is reported to the host with the Audio Decoder Play Mode Status bits in Register 354 (page 4-91).

11.6.4 Sample Decimation for S/P DIF

The sampling frequency of input Linear PCM bitstream can be either 48 kHz or 96 kHz. Decoded Linear PCM samples are passed to the audio DAC Interface which handles 48 kHz or 96 kHz, and the S/P DIF Interface which can only support up to 48 kHz. The Linear PCM module has two output ports as shown in Figure 11.9, a PCM port for nondecimated samples (48 or 96 kHz) and an S/P DIF port for decimated samples (48 kHz).

When the Linear PCM input bitstream sample rate is at 48 kHz, the PCM output is at 48 kHz and can be used for both the DAC interface and the S/P DIF interface. The S/P DIF port of the Linear PCM module has no output. When the Linear PCM input bitstream runs at 96 kHz, the PCM output is at 96 kHz and the S/P DIF port is at 48 kHz. Either PCM samples or S/P DIF samples can be used as inputs to DAC interface depending upon which frequency is desired. The S/P DIF interface can only use S/P DIF port samples.

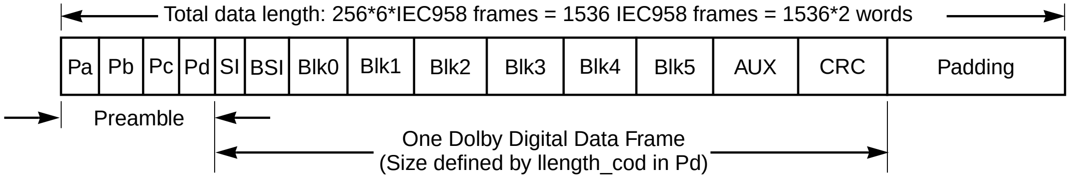
Figure 11.9 Linear PCM Output Ports



11.7 Dolby Digital Formatter

The Dolby Digital Formatter formats Dolby Digital data into S/P DIF (IEC958) format. The formatter gets Dolby Digital data directly from the Audio ES Channel Buffer and forms it into data bursts. As shown in Figure 11.10, each data burst contains a preamble (Pa, Pb, Pc, and Pd) followed by the burst payload and stuffing bits. The formatted data is passed to the S/P DIF Interface in 16-bit samples.

Figure 11.10 S/P DIF Burst Syntax


Total data length: 256*6*IEC958 frames = 1536 IEC958 frames = 1536*2 words
Pa
Pb
Pc
Pd
SI
BSI
Blk0
Blk1
Blk2
Blk3
Blk4
Blk5
AUX
CRC
Padding
Preamble
One Dolby Digital Data Frame
(Size defined by llength_cod in Pd)

Note:

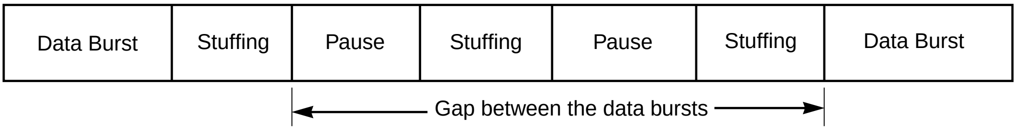
- Pa: 16 bits sync word (0xF872)
- Pb: 16 bits sync word (0x4E1F)
- Pc: 16 bits burst information

0–4 data-type	Set to 00001 for Dolby Digital Data.
5–6	Reserved 00.
7 error flag	0.
8–10 (bsmod)	Data-type-dependent parsed from BSI field (bits 5-7).
11–12 data-type-dependent	Reserved for Dolby Digital Data 00.
13–15 data-stream-number	Fixed, set to 0.

- Pd: 16 bit length-cod representing 0 to 65535 bits. Decided by fscod and frmsizecod using the frame-length lookup table.
- fscod is obtained from bits 32–33 of the Dolby Digital bitstream.
- frmsizecod is parsed from bits 34–39 of the Dolby Digital bitstream.
- Padding: all 0’s to the end of each S/P DIF frame.

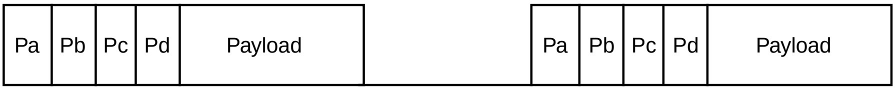
In case that there is an irregularity or discontinuity in the data bursts causing a gap between bursts, the *DVD Specifications for Read-Only Disc* dictates that the gaps should be filled with Pause bursts as shown in Figure 11.11.

Figure 11.11 Bridging Gaps with Pause Bursts



The fields of the Pause bursts are shown in Figure 11.12. The burst contains a preamble and a 32-bit payload.

Figure 11.12 Pause Burst Fields



The first 16 bits of the payload are an optional gap-length parameter. The remaining bits are reserved and set to 0. Since gap-length is optional, the L64020 Dolby Digital Formatter sets the first 16 bits of the payload to zeroes. For a complete description of the Pause frame syntax, refer to the *DVD Specifications for Read-Only Disc*.

The L64020 Dolby Digital Formatter uses the Pause burst to fill in the audio bitstream when:

- the Audio ES Channel Buffer is empty,
- there is a host pause,
- the Dolby Digital Formatter is synchronizing to the Dolby Digital Decoder and is in either wait or skip mode, or
- there are errors in the bitstream.

Table 11.7 lists the preamble and payload values of the Pause burst recognized by the L64020 Dolby Digital Formatter.

Table 11.7 Pause Burst Syntax

Field	Bits	Value	Content	Comments
Pa	0–15	0xF872	Preamble	
Pb	0–15	0x4E1F	Preamble	
Pc	0–4	0b00011	Data type	
Pc	5–6	0b00	Reserved	
	7	0b0	Error flag	
	8–12	0b0	Continuation of data	Used when Audio ES Channel Buffer is empty, wait for Dolby Digital Decoder, or user pause.
		0b1	Data discontinued	Used when user stop, skip to be in sync with Dolby Digital Decoder, or error conditions.
	13–15	0b000	Reserved	
Pd	0–15	0x0020	Payload = 32 zero bits	
Payload	0–31	0x0000.0000	32 zeroes sent as two 16-bit packets	

If Pause bursts are inserted due to discontinuance of data, the next Dolby Digital data burst starts at a new frame boundary.

The host can set the Formatter ES1 Compliant bit in Register 366 (page 4-103) to command the Dolby Digital Formatter to fill gaps in the bitstream with all zeroes instead of Pause bursts.

11.7.1 Synchronization

Both the Dolby Digital Decoder and Dolby Digital Formatter can run simultaneously. The formatter automatically detects when it is not in synchronization with the decoder and resynchronizes by either waiting for the decoder or by skipping ahead of the currently processed data. The wait or skip operation is performed only at data frame boundaries. The formatter outputs Pause bursts when it is either in wait or skip mode. Table 11.8 shows the coding of the Pc field in the Pause burst when the formatter is in wait or skip mode.

The threshold for the formatter to decide whether it should start skip or wait operation is set at one Digital Dolby frame. This is the default value for the Formatter Skip Frame Size bits in Register 366 (page 4-103) and should not be changed by the host.

11.7.2 Error Handling

Table 11.8 describes the error handling procedures implemented in the Dolby Digital Formatter.

Table 11.8 Dolby Digital Formatter Error Handling

Error	Action	Output
Incorrect sync word	Search for the next sync word.	a_sync_error = 1 Pause with Pc bits 8-12 = 0x0.0001
Illegal table entry	Search for the next sync word.	a_illegal_bit = 1 Pause with Pc bits 8-12 = 0x0.0001
User pause	Start Pause bursts and wait until user pause is deasserted.	Pause with Pc bits 8-12 = 0x0.0000
Audio ES Channel Buffer empty	Start Pause bursts and wait until Audio ES Channel Buffer is not empty.	Pause with Pc bits 8-12 = 0x0.0000
User stop	Start Pause bursts and wait until user stop is deasserted.	Pause with Pc bits 8-12 = 0x0.0001
Wait for Dolby Digital Decoder	Start Pause bursts and wait until synchronization.	Pause with Pc bits 8-12 = 0x0.0000
Skip	Start Pause bursts and skip until synchronization.	Pause with Pc bits 8-12 = 0x0.0001
Any other error	Start Pause bursts and search for the next sync word.	Pause with Pc bits 8-12 = 0x0.0001

11.8 MPEG Formatter

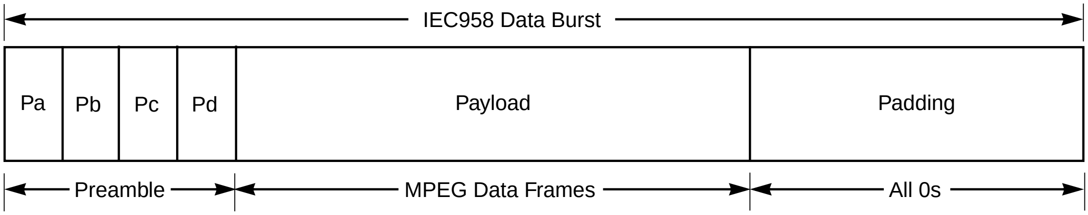
The L64020 MPEG Formatter formats the following audio bitstreams into IEC958 format:

- MPEG-1 Layer I data
- MPEG-1 Layer II data or MPEG-2 data without extension

- MPEG-2 data with extension
- MPEG-2 Layer I low sample rate
- MPEG-2 Layer II low sample rate

The MPEG Formatter accepts MPEG-compliant bitstreams from the Audio ES Channel Buffer and converts them to IEC958 format by arranging the data into bursts. Each burst contains a preamble (Pa, Pb, Pc, and Pd) followed by the burst payload and padding bits as shown in Figure 11.13.

Figure 11.13 Syntax of the MPEG Data in IEC958 Format



Note: Padding is all zeros until the end of each IEC958 frame.

The preamble values for the MPEG formatter supported bitstream are given in the Table 11.9.

Table 11.9 MPEG Formatter Data Burst Preamble Syntax

Field	Bits	Value	Content
Pa	0–15	0xF872	Sync Word 1
Pb	0–15	0x4E1F	Sync Word 2
Pc	0–4	0x04	MPEG-1 Layer I data
		0x05	MPEG-1 Layer II or MPEG-2 without extension
		0x06	MPEG-2 data with extension
		0x08	MPEG-2 Layer I low sample rate
		0x09	MPEG-2 Layer II low sample rate
(Sheet 1 of 2)			

Table 11.9 MPEG Formatter Data Burst Preamble Syntax (Cont.)

Field	Bits	Value	Content
Pc	5, 6	0b00	Reserved
Pc	7	0b0	Error flag - always set to 0
Pc	8–10	Host Pc Info	Host programmed into Host Pc Info bits in Register 368 (page 4-104)
Pc	11–12	0b00	Data-type dependent information
Pc	13–15	0b000	Bitstream number
Pd	0–15	Variable	Length of burst payload in bits (see Table 11.10)[1]
(Sheet 2 of 2)			

[1] Does not include the preamble.

11.8.1 Number of IEC958 Frames when Formatting MPEG Data

The size of the IEC958 data burst differs according to the type of the incoming MPEG bitstream being fed to the MPEG Formatter. This is explained in detail in Table 11.10.

Table 11.10 IEC958 Frame Sizes Supported in MPEG Audio Formatter

Data Type	Number of IEC958 Frames[1]
MPEG-1 Layer I data	384
MPEG-1 Layer II data or MPEG-2 data without extension	1152
MPEG-2 data with extension	1152
MPEG-2 Layer I low sample rate	384
MPEG-2 Layer II low sample rate	1152

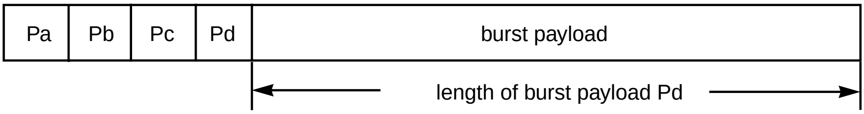
[1] One IEC958 frame contains a left and right sample and is 16 x 2 = 32 bits long.

As previously explained, the burst is headed with a burst preamble, followed by the burst payload, and stuffed with stuffing bits until it contains the number of IEC958 frames required for the data type being formatted.

11.8.2 Pd Field

The IEC958 specification dictates that the Pd field contain the length of the burst payload in bits (length code) from 0 to 65535 as shown in Figure 11.14. The size of the preamble is not counted in the value of the length code.

Figure 11.14 Length of Burst Payload



The host can program the value of the Pd field in the burst preamble by first programming the Pd Selection bits in Register 368 (page 4-104). Table 11.11 shows the Pd Selection bit codes for this mode and two other modes.

Table 11.11 Pd Selection

Bits [4:3]	Description
0b00	Previous multichannel extension packet
0b01	Base packet without extension
0b10	Host force
0b11	Reserved

If the host selects the Host Force mode, it can then write into the Pd field through Registers 369 and 370 (page 4-105). This automatically sets the Pd Data Valid bit in Register 368. When the MPEG Formatter reads the PD value in Registers 369 and 370, the Pd Data Valid bit is automatically cleared.

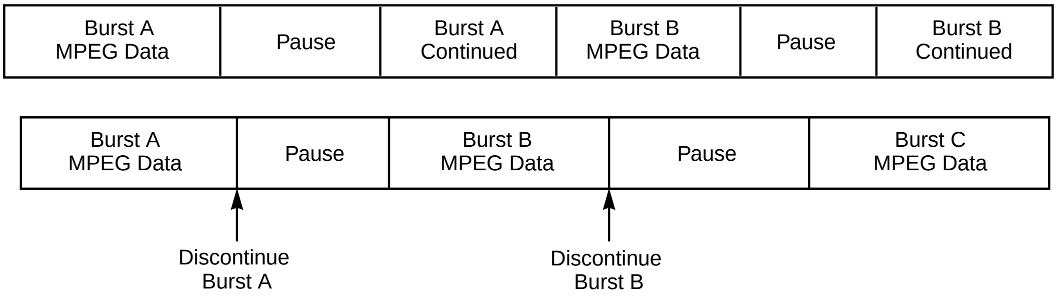
11.8.3 Pause Burst

When there is a gap in the bitstream due to an irregularity or discontinuity, the formatter inserts Pause bursts as shown in Figure 11.15. The gaps can be caused by conditions such as:

- the Audio ES Channel Buffer empty,

- a host pause,
- errors in the bitstream, or
- the MPEG Formatter is waiting or skipping to synchronize to the MPEG Decoder.

Figure 11.15 Inserting Pause Bursts in the MPEG Formatter Output



The syntax of the Pause bursts used in the MPEG Formatter is given in Table 11.12.

Table 11.12 MPEG Formatter Pause Burst Syntax

Field	Bits	Value	Content	Comments
Pa	0–15	0xF872	Preamble	
Pb	0–15	0x4E1F	Preamble	
Pc	0–4	0b00011	Data type	Pause data type
	5–6	0b00	Reserved	
	7	0b0	Error flag	
	8–12	0b0.0000	Continuation of data	Used for Audio ES Channel Buffer is empty, wait for the MPEG decoder, and user pause.
		0b0.0001	Data discontinued	Used for user stop, skip to be in sync with MPEG decoder, and error conditions.
Pd	0–15	0x03C0	Payload length in bits	
Payload	0–959	0	30 IEC958 zero frames	30 IEC958 frames = 2*16*30 bits = 960 zero bits

The host can command the formatter to change the Pause burst payload to all zero bits by setting the Formatter ES1 Compliant bit in Register 366 (page 4-103).

11.8.4 Synchronization

The MPEG Decoder and MPEG Formatter can run simultaneously. The formatter automatically detects when it is out of synchronization with the decoder and recovers by either waiting for the decoder or by skipping ahead of the currently processed data. The wait or skip operation is performed only at MPEG data frame boundaries. The Formatter outputs Pause bursts when it is in Wait or Skip Mode. The Pc field in the Pause burst is coded to indicate Wait or Skip Mode (see Table 11.12).

The out-of-sync threshold is set to two MPEG frames by the Formatter Skip Frame Size bits in Register 366 (page 4-103). If the formatter and decoder get out of sync by two MPEG frames, the formatter waits or skips to resynchronize. The Formatter Skip Frame Size bits default to 0b00 and should not be changed by the host.

The host can set the MPEG Formatter only bit in Register 366 (page 4-103) to run the formatter and not the decoder. This automatically disables the skip/wait synchronization feature of the formatter.

11.8.5 Error Conditions

Table 11.13 describes the error handling procedures implemented in the MPEG Formatter.

Table 11.13 MPEG Audio Formatter Error Handling

Error	Action	Output
Incorrect sync word	Search for the next sync word.	a_sync_error = 1 Pause with Pc bits 8-12 = 0x0.0001
Illegal table entry	Search for the next sync word.	a_illegal_bit = 1 Pause with Pc bits 8-12 = 0x0.0001
User pause	Start Pause bursts and wait until user pause is deasserted.	Pause with Pc bits 8-12 = 0x0.0000
(Sheet 1 of 2)		

Table 11.13 MPEG Audio Formatter Error Handling (Cont.)

Error	Action	Output
Audio ES Channel Buffer empty	Start Pause bursts and wait until Audio ES Channel Buffer is not empty.	Pause with Pc bits 8-12 = 0x0.0000
User stop	Start Pause bursts and wait until user start is asserted.	Pause with Pc bits 8-12 = 0x0.0001
Wait for MPEG decoder	Start Pause and wait until synchronization.	Pause with Pc bits 8-12 = 0x0.0000
Skip	Start Pause and skip until synchronization.	Pause with Pc bits 8-12 = 0x0.0001
Any other error	Start Pause and search for the next sync word.	Pause with Pc bits 8-12 = 0x0.0001
(Sheet 2 of 2)		

11.9 PCM FIFO Mode

The host can write four-byte, L-R PCM samples (two bytes for each channel) into the PCM FIFO and select these values to play through the Linear PCM Decoder. The registers associated with PCM FIFO mode are listed in Table 11.14. The host can read the FIFO full, near full, and empty status bits and monitor the near full signal (PREQn) for external DMA control.

Table 11.14 PCM FIFO Mode Registers

Register	Bits	Name	Page Ref.
246	0	Decode Start/Stop Command	4-66
357	[7:5]	Audio Decoder Mode Select [2:0]	4-93
359	[7:0]	PCM FIFO Data In [7:0]	4-96
353	7	PCM FIFO Full	4-90
	6	PCM FIFO Near Full	
	5	PCM FIFO Empty	

The host uses the following sequence for PCM FIFO mode:

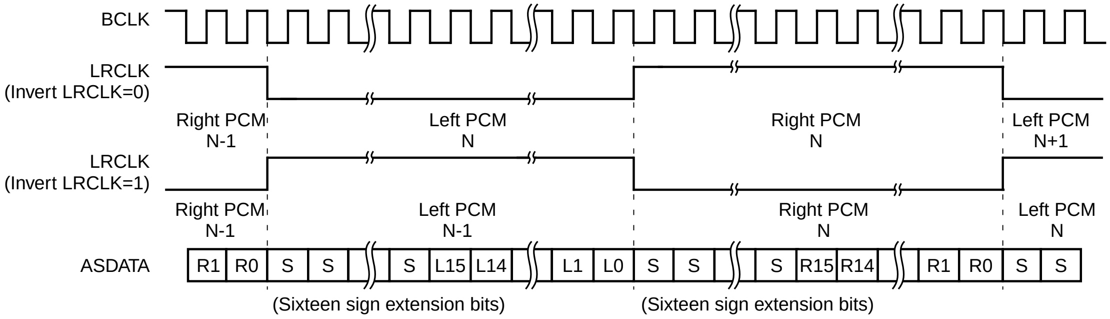
1. Clear the Decode Start/Stop Command bit to stop the Audio Decoder.
2. Program the Audio Module Mode Selection bits to 0b111 to select the PCM FIFO mode.
3. Using a host DMA controller, write PCM audio into the PCM FIFO Data In register to fill the PCM FIFO. The audio data is written in the following order; left channel LSB, left channel MSB, right channel LSB, and right channel MSB. The PCM FIFO is 16 words deep x 16 bits wide.
4. Set the Decode Start/Stop Command bit to start the Audio Decoder.
5. Monitor the PCM FIFO status bits in Register 353 and the PREQn output signal of the L64020. The PCM FIFO Near Full bit is cleared when the PCM FIFO contains less than 25 unread words. When the bit is cleared, PREQn is also asserted to the external DMA controller.

11.10 DAC Interface

The DAC Interface in the Audio Decoder converts the 16-, 20-, or 24-bit parallel PCM data received from the decoders into 32-bit, serial frames and transmits them to the external DAC. A demultiplexer controlled by the Audio Module Mode Select bits in Register 357 (page 4-93) selects the output of one of the three decoders or the PCM FIFO as the DAC Interface input.

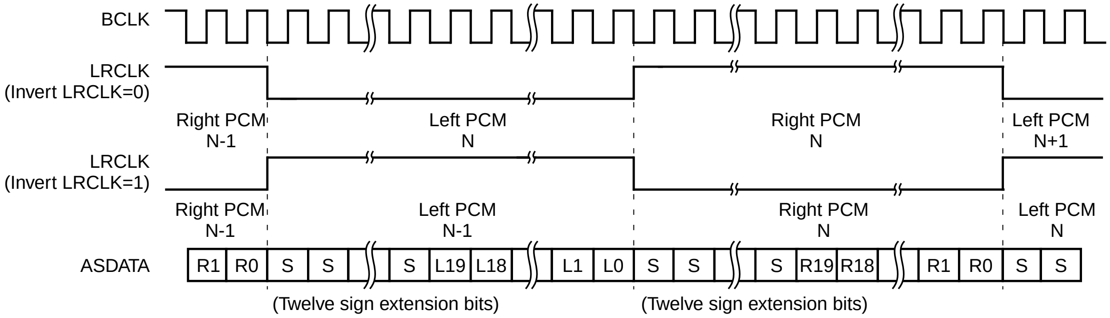
The audio samples are multiplied by a scale factor, PCM Scale, in the DAC Interface to control the output volume. At reset and power on, the PCM Scale [7:0] bits in Register 362 are set to 0xFF to pass the input samples through the interface with no change in level. The host can write to the register to scale the samples level down in increments of 1/256. Setting the PCM Scale bits to 0x00 mutes the audio output. The output samples are sign-extended to 32 bits as shown in Figure 11.16 through Figure 11.18.

Figure 11.16 DAC Output Mode: PCM Sample Precision = 16 Bit



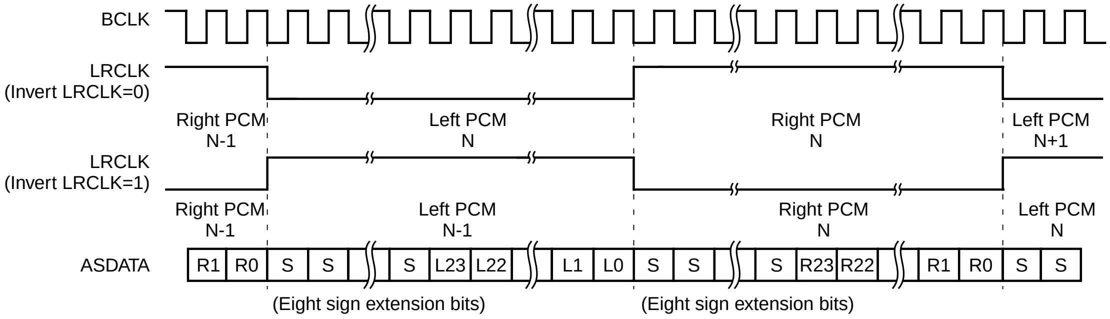
Note: S means sign-extension (0 for positive PCM values, 1 for negative PCM values).

Figure 11.17 DAC Output Mode: PCM Sample Precision = 20 Bit



Note: S means sign-extension (0 for positive PCM values, 1 for negative PCM values).

Figure 11.18 DAC Output Mode: PCM Sample Precision = 24 Bit



Note: S means sign-extension (0 for positive PCM values, 1 for negative PCM values).

The interface supplies four signals to the DAC:

- a sample clock, A_ACLK,
- the bit clock, BLCK,
- a left/right channel clock, LRCLK, and
- the serial audio data, ASDATA.

BCLK and A_ACLK are derived from an ACLK input in the Clock Divider (see Section 11.12, “Clock Divider.”) BCLK is at the output bit rate and expressed as:

$$\text{BCLK} = \text{Sample Freq} \times \text{Sample Resolution} \times 2(\text{channels})$$

The ASDATA bits are clocked out on every BCLK falling edge. The A_ACLK is the DAC clock and is at 256 or 384 times the sample frequency depending on the DAC used.

> Note: Some DACs have an on-chip Phase-Locked Loop (PLL) to derive their operating clock from the incoming bit clock. The A_ACLK output of the L64020 is not used in this case.

LRCLK specifies which PCM channel, left or right is currently being transferred. The Invert LRCLK bit in Register 363 (page 4-98) determines the LRCLK state channel assignment. The bit defaults to the clear state at reset and power on. This sets LRCLK high for left channel sample outputs and low for right channel outputs. The host can invert this sense (high for right; low for left) by setting the Invert LRCLK bit.

The DAC Interface also uses a special, soft-muting scheme to avoid a click on the speakers when the output is turned off. The host can set the Mute on Error bit in Register 358 (page 4-96) to force a soft-mute of the audio output when certain errors occur in the bitstream or the decoder. The host can also mute the audio outputs by setting the User Mute Bit in Register 358. An Audio Decoder Soft Mute Status bit is available in Register 354 (page 4-91) for the host to read.

When the host programs the Audio Decoder Mode Select bits in Register 357 (page 4-93) to 0b110 to select the CD Bypass Mode, the output demultiplexer substitutes the CD_ASDATA, CD_BCLK, CD_LRCLK, and CD_ACLK for the normal outputs of the DAC Interface.

11.11 S/P DIF Interface

The S/P DIF (IEC958) Interface is a serial, unidirectional, self-clocking interface for the interconnection of digital equipment for consumer and professional applications. The L64020 supports the consumer output mode only, which carries stereophonic digital programs with a resolution of up to 16 bits per sample. Twenty and 24-bit samples are clipped by dropping the least significant 4 or 8 bits.

The demultiplexer at the interface input is controlled by the Audio Module Mode Select bits in Register 357 (page 4-93) to select the output of any one of the three decoders or the output of one of the formatters. The interface serializes the selected samples and formats them as described in Section 11.11.2, “IEC958 Syntax.” The output demultiplexer selects either the output of the interface or the SPDIF_IN when the host selects the S/P DIF Bypass Mode.

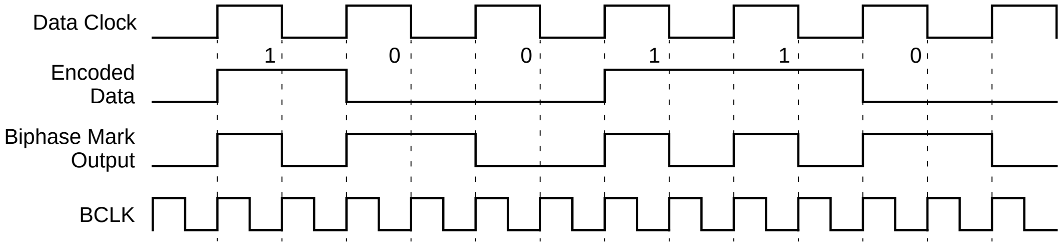
11.11.1 Biphase Mark Coding

To minimize the DC component on the transmission line, facilitate clock recovery from the bitstream, and make the interface insensitive to the polarity of connection, the bitstream is encoded in biphase marks.

Refer to Figure 11.19. BCLK, derived from an ACLK input in the Clock Divider (see Section 11.12, “Clock Divider”), divides each data bit into biphase marks. The S/P DIF BCLK rate is sample frequency x sample resolution x 2 channels x 2 biphase marks/bit.

Each bit is represented by a symbol with two consecutive binary states, 1 bits by two opposite states and 0 bits by two equal states. The first state of a symbol is always different from the second state of the previous symbol. This forces the data stream to transition at least once for every bit.

Figure 11.19 IEC958 Biphase Mark Representation



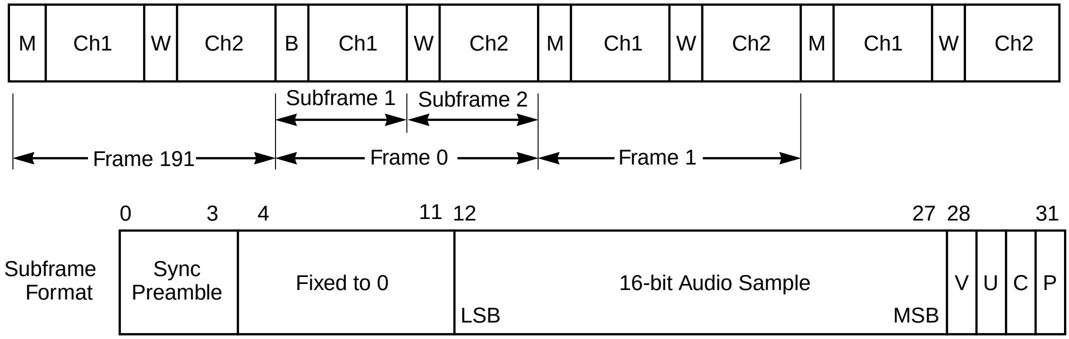
11.11.2 IEC958 Syntax

The IEC958 stream is organized into blocks of 192 frames with each frame containing two subframes, one for each audio channel as shown in Figure 11.20. The frame transmission rate is equal to the source sampling frequency when the input audio source is sampled at 32, 44.1, or 48 kHz. When the input data is 96-kHz Linear PCM samples, the samples are decimated down to 48 kHz in the decoder for the S/P DIF output.

The layout of the subframes is also shown in Figure 11.20. Each subframe starts off with a 4-bit (8-state) preamble. The preamble is coded to mark the first frame in a block, to differentiate between subframes in a frame, and to violate the biphase mark rule twice. This latter feature prevents other data in the stream from mimicking a preamble. The preamble states are listed in Table 11.15. Since the biphase mark violations do not occur between data and the preamble, two codes are used for each of the three subframe applications depending on the last state of the previous data bit.

Bits 4 through 11 of the subframes are fixed at zero by the S/P DIF Interface. The 16-bit audio sample is packed in bits 12 through 27 of the subframe with the LSB in bit 12. The interface defaults to setting the V bits and clearing the U bits. The host can change their values each subframe by writing to the User and Valid bits in Register 363 (page 4-98). The P bit is set for even parity across the subframe.

Figure 11.20 IEC958 Syntax



Note:
- V = Validity bit (set to 1).
- U = User data (set to 0).
- C = Channel status.
- P = Parity bit.

Table 11.15 IEC958 Subframe Preambles

Preamble	Preceding Preamble State = 0	Preceding Preamble State = 1	Subframe
B	11101000	00010111	Subframe 1 at the start of blocks
M	11100010	00011101	Subframe 1 except at the start of blocks
W	11100100	00011011	Subframe 2

11.11.3 IEC958 Channel Status

The L64020 uses the first 32 C bits of each channel in each block to carry the four bytes of channel status information shown in Figure 11.21. The remaining C bits in the blocks are cleared to 0. The Copyright and Emphasis bits are from the incoming bitstream. The S/P DIF Interface inserts one of the two Category Codes shown in the following table:

Data Format	Default Category Code
PCM Samples	0b0000.0000
Digital Data	0b1001.1000

Figure 11.21 IEC958 Channel Status

	0	1	2	3	4	5	6	7
Byte 0	0 for Consumer Mode Only	1 = Formatter Output, 0 = Decoder Output	Copyright	Emphasis from Bitstream			Mode = 0b00	
Byte 1	Category Code (User Programmable)							
Byte 2	Source Number = 0b0000				Channel (L = 0b1000, R = 0b0100, Don’t Care = 0b0000)			
Byte 3	Sampling Frequency (44.1/48/32)				Clock Accuracy = 0b01			

The host can overwrite the Copyright bit, the Emphasis bits, and Category Code by setting the associated overwrite bit in Registers 365 (page 4-101) and 366 (page 4-102), and writing to the Emphasis bit, Copyright field, or Category field in Register 367. The remaining bits and fields of the channel status bytes are fixed or filled in by the S/P DIF Interface as shown in Figure 11.21.

11.12 Clock Divider

As mentioned in the output interface descriptions, the Clock Divider in the Audio Decoder derives a BCLK for each interface and an LRCLK and A_ACLK for the external DACs from an input audio clock. The L64020 has three audio clock input pins, ACLK_32, ACLK_441, and ACLK_48 for 32, 44.1, and 48 kHz sampling rates. The inputs to these must be the sampling rate times N, where N can be 256, 384, 512, or 768. The N value must be an integral multiple of the sample resolution (16, 20, or 24). Any or all of the inputs can be connected depending on the audio sampling rates and resolutions expected.

At reset and power on, the L64020 defaults to using the clock supplied on the ACLK_ 48 pin. The host selects the ACLK_ input pin by programming the ACLK Select [1:0] bits in Register 363 (page 4-97). The host also selects the divisor values used in the Clock Divider by programming the ACLK Divider Select [3:0] bits in Register 364.

The divisor values depend on ACLK_ availability, the input audio sampling frequency (Fs), the sample resolution (16/24/32 bits per sample), and the external DAC capabilities. The equations for the derived clocks are:

S/P DIF BCLK = Fs*sample resolution*2 channels*2 marks
= Fs*32*4
= Fs*128

DAC BCLK = Fs*sample resolution*2 channels
= Fs*32*2
= Fs*64

A_ACLK = Fs*sample resolution*K
= Fs*256 or Fs*384

The ACLK Divider Select bit selections and the resulting clocks are listed in Table 11.16. Use the following cases as selection criteria:

- Case I: All of the ACLK_ inputs are available. Select the ACLK_ which is a multiple of the input sampling frequency using bits 0 and 1 in Register 363. Then use the 0x0 through 0x4 ACLK Divider Select code that matches the Fs multiple of the ACLK_. For example, if the input sampling frequency is 32 kHz and ACLK_32 = 512 * 32 kHz, use the 0x2 ACLK Divider Select code.
- Case IIA: The Linear PCM bitstream with a sampling frequency of 96 kHz is selected and the external DAC supports 96-kHz sampling frequency. ACLK_48 at a multiple of 512 or 768 must be available and it must be selected. Use divider code 0x5 for ACLK = 768 * 48 and code 0x6 for ACLK = 512 * 48.
- Case IIB: The Linear PCM bitstream with a sampling frequency of 96 kHz is selected but the external DAC does not support 96-kHz sampling frequency. ACLK_48 must be available and it must be selected. Set the Audio Decoder Mode Select field (Register 357, bit [7:5], page 4-93) to 0b101 to decimate the output samples to 48 kHz. Use the 0x0 through 0x4 divider code that matches the ACLK_48 multiple.
- Case III: The input sampling rate is 32 kHz but ACLK_32 is not available. Select ACLK_48 and the 0xC through 0xF divider code that matches the ACLK_48 multiple to derive the 32-kHz clocks from ACLK_48.

Note: The CD bypass mode has a dedicated ACLK input pin called CD_ACLK.

Table 11.16 ACLK Divider Select [3:0] Code Definitions

ACLK Divider Select [3:0]	ACLK Input	S/P DIF Interface BCLK	DAC Interface BCLK	DAC A_ACLK
0x0	768 x Fs	128 x Fs = ACLK ÷ 6	64 x Fs = ACLK ÷ 12	256 x Fs = ACLK ÷ 3
0x1	768 x Fs	128 x Fs = ACLK ÷ 6	64 x Fs = ACLK ÷ 12	384 x Fs = ACLK ÷ 2
0x2	512 x Fs	128 x Fs = ACLK ÷ 4	64 x Fs = ACLK ÷ 8	256 x Fs = ACLK ÷ 2
0x3	384 x Fs	128 x Fs = ACLK ÷ 3	64 x Fs = ACLK ÷ 6	384 x Fs = ACLK ÷ 1
0x4	256* x Fs	128 x Fs = ACLK ÷ 2	64 x Fs = ACLK ÷ 4	256 x Fs= ACLK ÷ 1
0x5	768 x 48	128 x 48 = ACLK ÷ 6	64 x 96 = ACLK ÷ 6	384 x 96 = ACLK ÷ 1
0x6	512 x 48	128 x 48 = ACLK ÷ 4	64 x 96 = ACLK ÷ 4	256 x 96 = ACLK ÷ 1
0x7–0xB	Not Used			
0xC	768 x 48	128 x 32 = ACLK ÷ 9	64 x 32 = ACLK ÷ 18	384 x 32 = ACLK ÷ 3
0xD	512 x 48	128 x 32 = ACLK ÷ 6	64 x 32 = ACLK ÷ 12	256 x 32 = ACLK ÷ 3
0xE	512 x 48	128 x 32 = ACLK ÷ 6	64 x 32 = ACLK ÷ 12	384 x 32 = ACLK ÷ 2
0xF	256 x 48	128 x 32 = ACLK ÷ 3	64 x 32 = ACLK ÷ 6	256 x 32 = ACLK ÷ 1

Chapter 12 Specifications

This chapter specifies the L64020 electrical and mechanical characteristics. It is divided into the following sections:

- Section 12.1, “Electrical Requirements,” page 12-1
- Section 12.2, “AC Timing,” page 12-4
- Section 12.3, “Pinouts and Packaging,” page 12-18

Note: All specifications are for the L64020 in LSI Logic’s 3.3-V, 0.25-micron G10®-p process technology and are subject to change. AC timing has been simulated and not characterized.

12.1 Electrical Requirements

This section specifies the electrical requirements for the L64020. Four tables list electrical data in the following categories:

- Absolute Maximum Ratings (Table 12.1)
- Recommended Operating Conditions (Table 12.2)
- Capacitance (Table 12.3)
- DC Characteristics (Table 12.4)

Table 12.1 Absolute Maximum Ratings

Symbol	Parameter	Limits	Unit
V_{DD}	DC Supply	-0.3 to + 3.9[1]	V
V_{IN}[2]	5 V Compatible Input Voltage	-1.0 to + 6.5[1]	V
I_{IN}	DC Input Current	±10	mA
T_{STGM}	Storage Temperature Range	-40 to + 125	°C

1. Referenced to V_{SS}.
2. All signal inputs are TTL compatible and can withstand this range.

Table 12.2 Recommended Operating Conditions

Symbol	Parameter	Limits	Unit
V_{DD}	DC Supply	+3.14 to + 3.46	V
T_A	Ambient Temperature	0 to + 70	°C

Table 12.3 Capacitance

Symbol	Parameter[1]	Min	Units
C_{IN}	Input Capacitance	2.5	pF
C_{OUT}	Output Capacitance	2.5	pF
C_{IO}	I/O Bus Capacitance	2.5	pF

1. Measurement conditions are V_{IN} = 3.3 V, T_A = 25 °C, and clock frequency = 1 MHz.

Table 12.4 DC Characteristics

Symbol	Parameter	Condition[1]	Min	Typ	Max	Units
V_{IL}	Voltage Input Low TTL CMOS		 – –	 – –	 0.8 0.2 V_{DD}	 V V
V_{IH}	Voltage Input High TTL CMOS 5 V Compatible		 2.0 0.7 V_{DD} 2.0	 – – –	 – – 5.5	 V V V
V_{OL}	Voltage Output Low 4-mA Output Buffers 6-mA Output Buffers	 I_{OL} = 4.0 mA I_{OL} = 6.0 mA	 – –	 0.2 0.2	 0.4 0.4	 V V
V_{OH}	Voltage Output High 4-mA Output Buffers 6-mA Output Buffers	 I_{OH} = -4.0 mA I_{OH} = -6.0 mA	 2.4 2.4	 – –	 – –	 V V
I_{IL}	Current Input Leakage[2] with Pulldown with Pullup	$V_{IN} = V_{DD}$ or V_{SS} $V_{IN} = V_{DD}$ $V_{IN} = V_{SS}$	-10 35 -214	±10 115 -115	+10 222 -35	µA µA µA
I_{OZ}	Current 3-State Output Leakage	V_{DD} = Max, $V_{OUT} = V_{SS}$ or V_{DD}	-10	±1	+10	µA
I_{OSP4}	Current P-Channel Output Short Circuit (4-mA Output Buffers)[3, 4]	V_{DD} = Max, $V_{OUT} = V_{SS}$	-117	-75	-40	mA
I_{OSN4}	Current N-Channel Output Short Circuit (4-mA Output Buffers)[3, 4]	V_{DD} = Max, $V_{OUT} = V_{DD}$	37	90	140	mA
I_{DD}	Quiescent Supply Current	$V_{IN} = V_{DD}$ or V_{SS}	–	10	–	mA
I_{CC}	Dynamic Supply Current	V_{DD} = Max, f = 27 MHz	–	210	–	mA

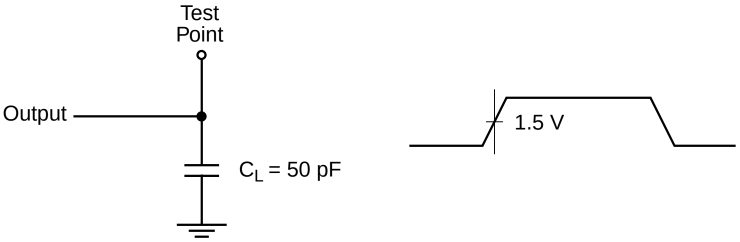
1. Specified at V_{DD} equals 3.3 V ±5% at ambient temperature over the specified range.
2. For CMOS and TLL inputs.
3. Not more than one output may be shorted at a time for a maximum duration of one second.
4. These values scale proportionally for output buffers with different drive strengths.

12.2 AC Timing

This section presents AC timing information for the L64020 MPEG-2 Audio/Video Decoder. The timing diagrams in this section illustrate the clock edges and specific signal edges from which the timing parameters are measured. These diagrams do not imply any other timing relationships. For specific information on functional relationships between the signals, refer to the appropriate functional and signal definition sections.

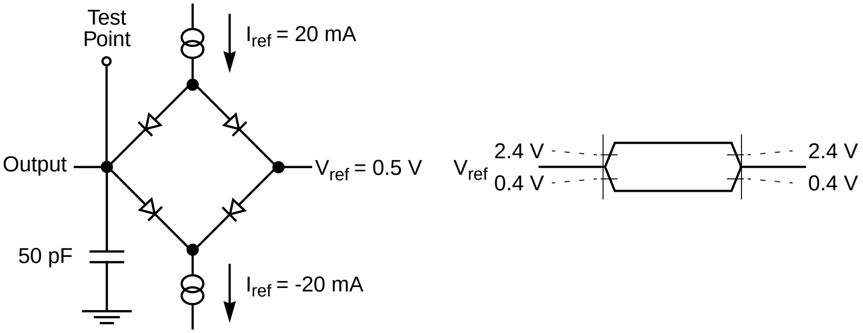
During AC testing, HIGH inputs are driven at V_{DD} = Min and LOW inputs are driven at 0 V. For transitions between HIGH, LOW, and invalid states, timing measurements are made at 1.5 V, as shown in Figure 12.1. The test load, C_L, for each output signal is 50 pF.

Figure 12.1 AC Test Load and Waveform for Standard Outputs



For 3-state outputs (see Figure 12.2), timing measurements are made from the point at which the output turns ON or OFF. An output is ON when its voltage is greater than 2.4 V or less than 0.4 V. An output is OFF when its voltage is less than 2.4 V or greater than 0.4 V.

Figure 12.2 AC Test Load and Waveform for 3-State Outputs



AC Timing is organized by interface and shown in the following tables and figures:

- SDRAM Interface Timing - Table 12.5, Figure 12.3 and Figure 12.4
- Host Interface Timing (Motorola Mode) - Table 12.6, Figure 12.5 and Figure 12.6
- Host Interface Timing (Intel Mode) - Table 12.7, Figure 12.7 and Figure 12.8
- Asynchronous Channel Write Timing - Table 12.8 and Figure 12.9
- Synchronous VALID Signals Timing - Table 12.9 and Figure 12.10
- Reset Timing - Figure 12.11
- Video Interface Timing - Table 12.10 and Figure 12.12
- Audio Interface Timing - Table 12.11, Figure 12.13, Figure 12.14 and Figure 12.15

The numbers in timing diagrams refer to the timing parameters listed in the first column of Table 12.5, which lists the AC timing values for the signals. The test conditions for the AC timing values are V_{DD} ± 5% within an ambient temperature range from 0 ˚C to 70 ˚C.

Table 12.5 SDRAM Interface AC Timing

Parameter	Description	Min	Max	Units
1	SCSn Setup	2	6	ns
2	SCSn Hold	2	6	ns
3	SRASn Setup	2	6	ns
4	SRASn Hold	2	6	ns
5	SCASn Setup	2	6	ns
6	SCASn Hold	2	6	ns
7	SWEn Setup	2	6	ns
8	SWEn Hold	2	6	ns
9	Read Data Setup	2	6	ns
10	Read Data Hold	2	6	ns
11	Address Valid	2	6	ns
12	Address Hold	2	6	ns
13	SCLK Cycle Tc	37	–	ns
13b	SCLK Duty Cycle	0.45Tc	0.55Tc	ns
14	Write Data Setup	2	6	ns
15	Write Data Hold	2	6	ns

Figure 12.3 SDRAM Read Cycle

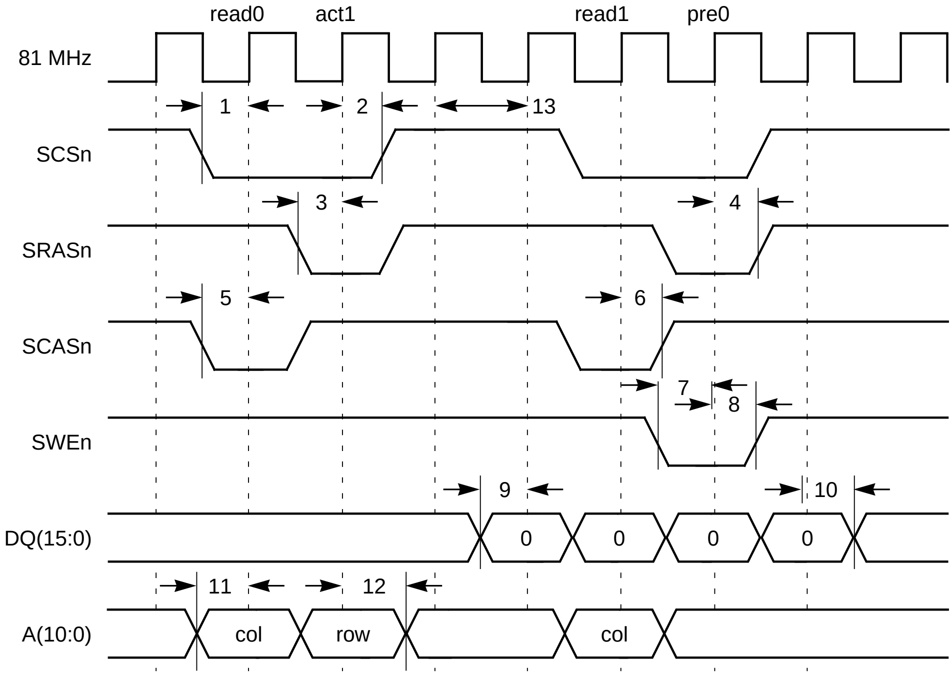


Figure 12.4 SDRAM Write Cycle

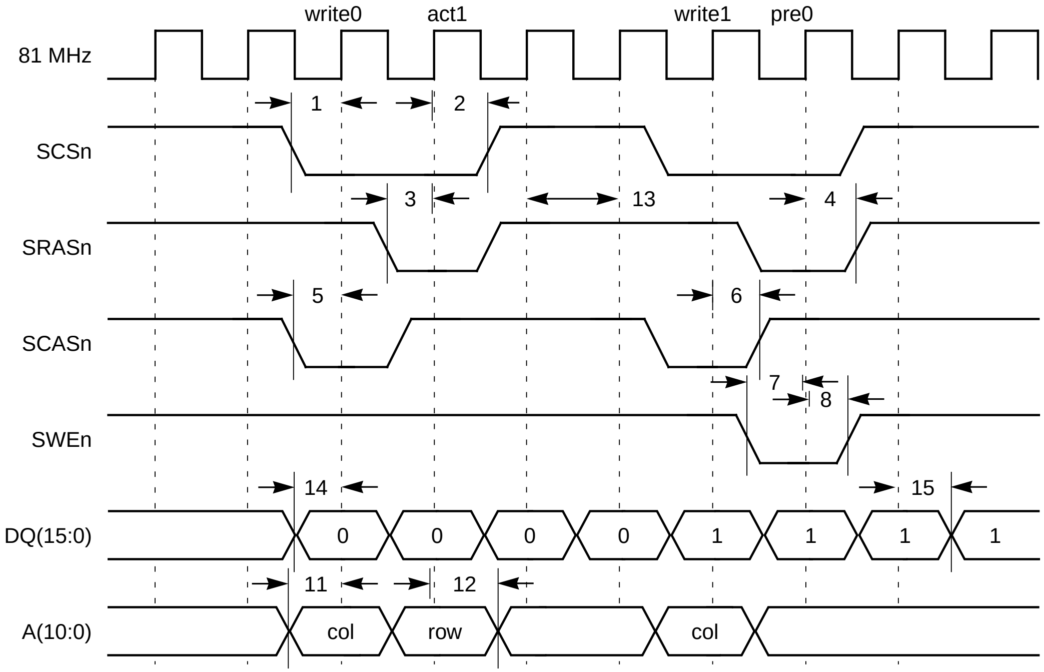


Table 12.6 Host Interface AC Timing (Motorola Mode)

Parameter	Description	Min	Max	Units
1	Addr setup to ASn falling	7	–	ns
2	Addr hold from ASn falling	7	–	ns
3	ASn low pulse width	0.5Tc[1]	–	ns
4	Data setup to DSn rising (Wr cycle)	10	–	ns
5	Data hold from DSn rising (Wr cycle)	0	–	ns
6	READ setup to DSn falling	7	–	ns
7	READ hold from DSn falling	7	–	ns
8	CSn setup to DSn falling	7	–	ns
9	CSn hold from DSn rising	0	–	ns
10	DSn low pulse width (Write Cycle)	3Tc[1]	–	ns
11	DSn rising to DSn rising (Write Cycle)	3.5Tc[1]	–	ns
12	CSn falling to WAITn/DTACKn active	–	12	ns
13	DSn falling to WAITn low/DTACKn high	–	12	ns
14	CSn rising to WAITn/DTACKn 3-state	2	15	ns
15	DSn falling to WAITn high/DTACKn low (Write cycle)	–	2.5Tc[1] + 15	ns
16	DSn rising to WAITn low/DTACKn high	–	15	ns
17	DSn low pulse width (Read Cycle)	4Tc[1]	–	ns
18	DSn falling to DSn falling (Read Cycle)	4.5Tc[1]	–	ns
19	DSn falling to WAITn high/DTACKn low (Read cycle)	–	3.5Tc[1] + 15	ns
20	Data setup BEFORE WAITn high/DTACKn low (Read Cycle)	10	–	ns
21	DSn falling to Data 3-state (Read Cycle)	2	–	ns
22	READ falling to Data 3-state (Read Cycle)	2	–	ns

[1] Tc = 1/27 MHz = 37 ns.

Figure 12.5 Host Write Timing (Motorola Mode)

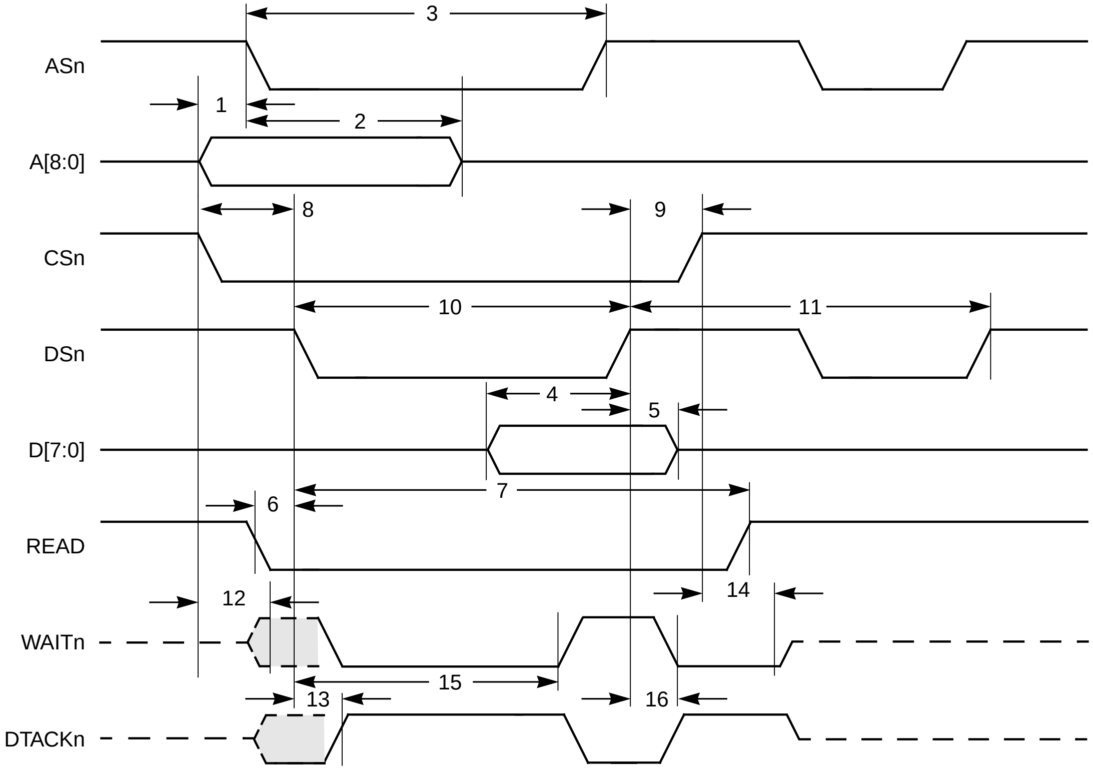


Figure 12.6 Host Read Timing (Motorola Mode)

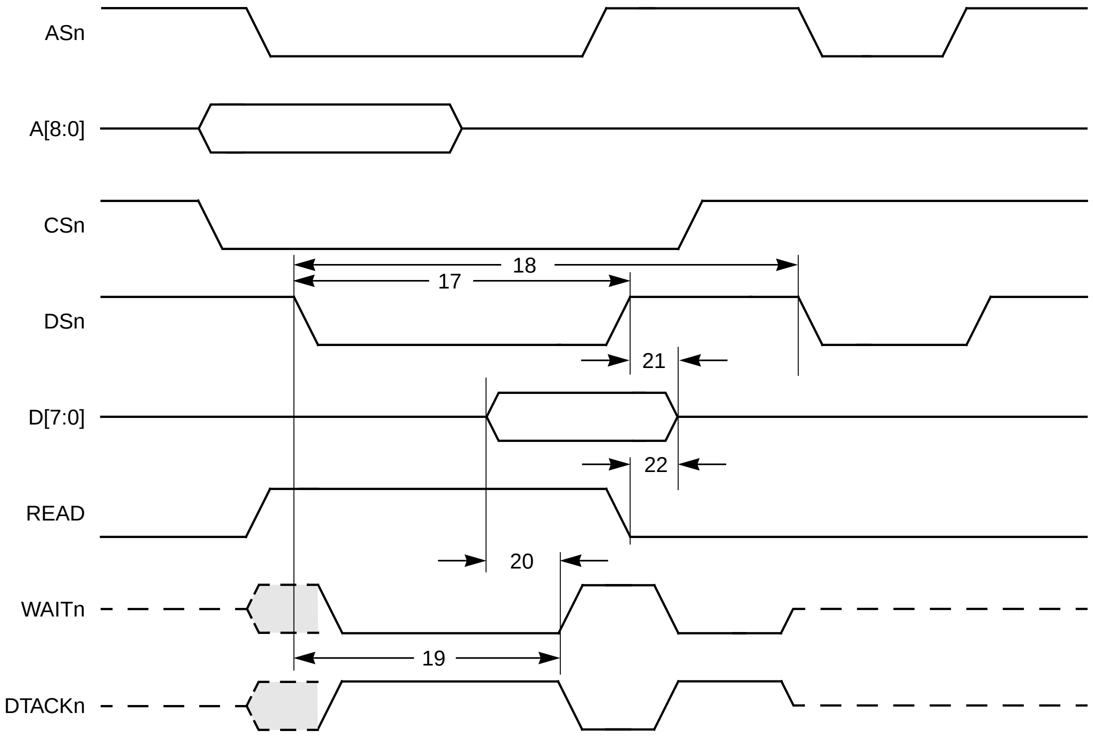


Table 12.7 Host Interface AC Timing (Intel Mode)

Parameter	Description	Min	Max	Units
1	Addr setup to Write/Read falling	7	–	ns
2	Addr hold from Write/Read falling	7	–	ns
3	CSn setup to Write/Read falling	7	–	ns
4	CSn hold from Write/Read rising	0	–	ns
5	Write low pulse width (Write Cycle)	3Tc[1]	–	
6	Write rising to Write rising (Write Cycle)	3.5Tc[1]	–	ns
7	Data setup to Write rising	10	–	ns
8	Data hold from Write rising	0	–	ns
9	CSn falling to WAITn/DTACKn active	–	12	ns
10	Write/Read falling to WAITn low/DTACKn high	–	12	ns
11	CSn rising to WAITn /DTACKn 3-state	2	15	ns
12	Write falling to WAITn high/DTACKn low (Write cycle)	–	2.5Tc[1] + 15	ns
13	Write/Read rising to WAITn low/DTACKn high	–	15	ns
14	Read low pulse width (Read Cycle)	4Tc[1]	–	ns
15	Read falling to Read falling (Read Cycle)	4.5Tc[1]	–	ns
16	Read falling to WAITn high/DTACKn low (Read Cycle)	–	3.5Tc[1] + 15	ns
17	Data setup BEFORE WAITn high/ DTACKn low (Read Cycle)	10	–	ns
18	Read falling to Data 3-state (Read Cycle)	2	–	ns

1. Tc = 1/27 MHz = 37 ns.

Figure 12.7 Host Write Timing (Intel Mode)

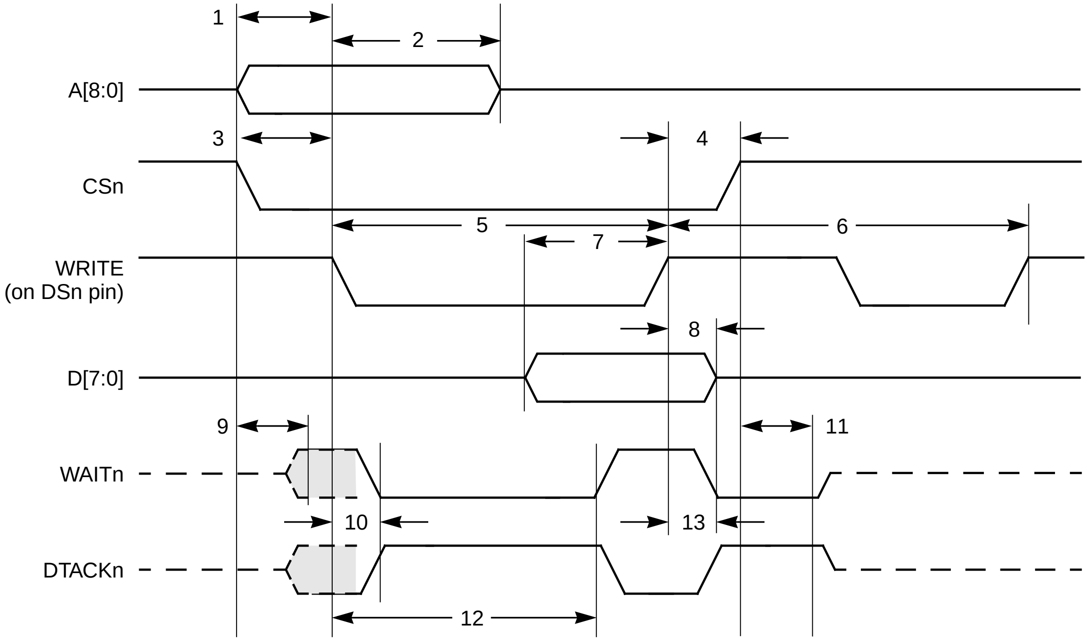


Figure 12.8 Host Read Timing (Intel Mode)

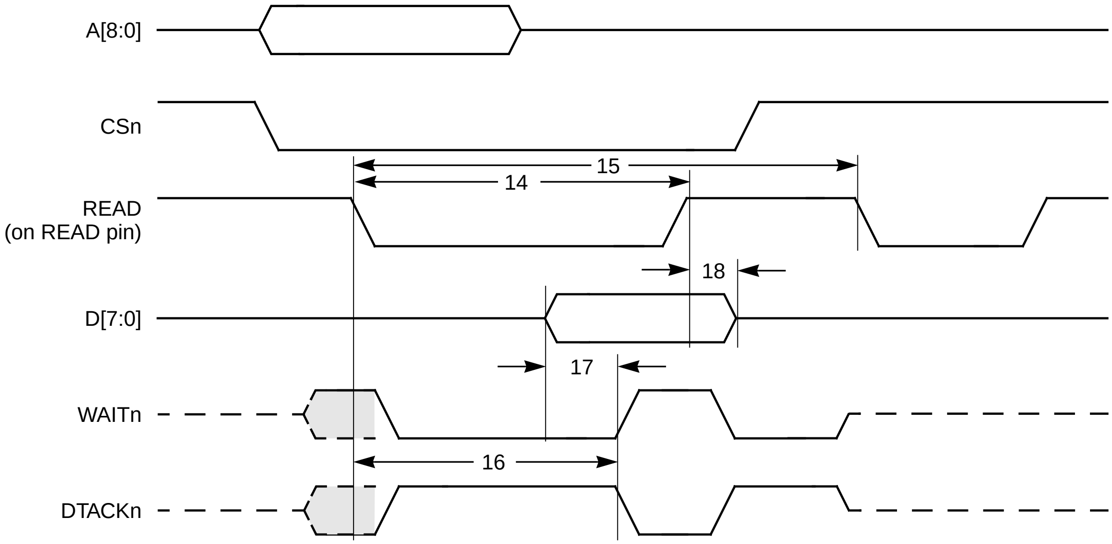
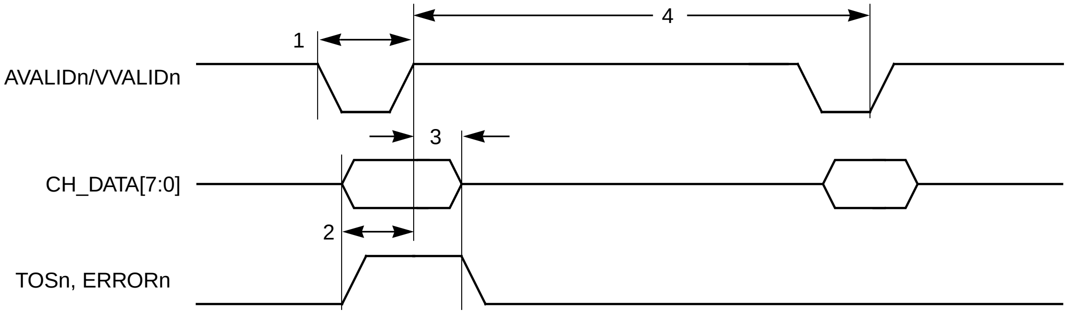


Table 12.8 Asynchronous Channel Write AC Timing

Parameter	Description	Min	Max	Units
1	AVALIDn/VVALIDn low pulse width	0.5Tc[1]	–	ns
2	Data setup to AVALIDn/VVALIDn rising	12	–	ns
3	Data hold from AVALIDn/VVALIDn rising	1	–	ns
4	AVALIDn/VVALIDn rise to AVALIDn/VVALIDn rise	3Tc[1]	–	ns

[1] Tc = 1/27 MHz = 37 ns.

Figure 12.9 Asynchronous Channel Write Timing



Note: During asynchronous usage of the channel input, the DCK signal must be tied to VSS.

Table 12.9 Synchronous AVALIDn/VVALIDn Signals AC Timing

Parameter	Description	Min	Max	Units
1	AVALIDn/VVALIDn hold from DCK rising	1	–	ns
2	AVALIDn/VVALIDn setup to DCK falling	7	–	ns
3	DCK cycle time	3Tc[1]	–	ns
4	CH_DATA setup to DCK rising	15	–	ns
5	CH_DATA hold from DCK rising	1	–	ns

[1] Tc = 1/27 MHz = 37 ns.

Figure 12.10 Synchronous AVALIDn/VVALIDn Signals Timing

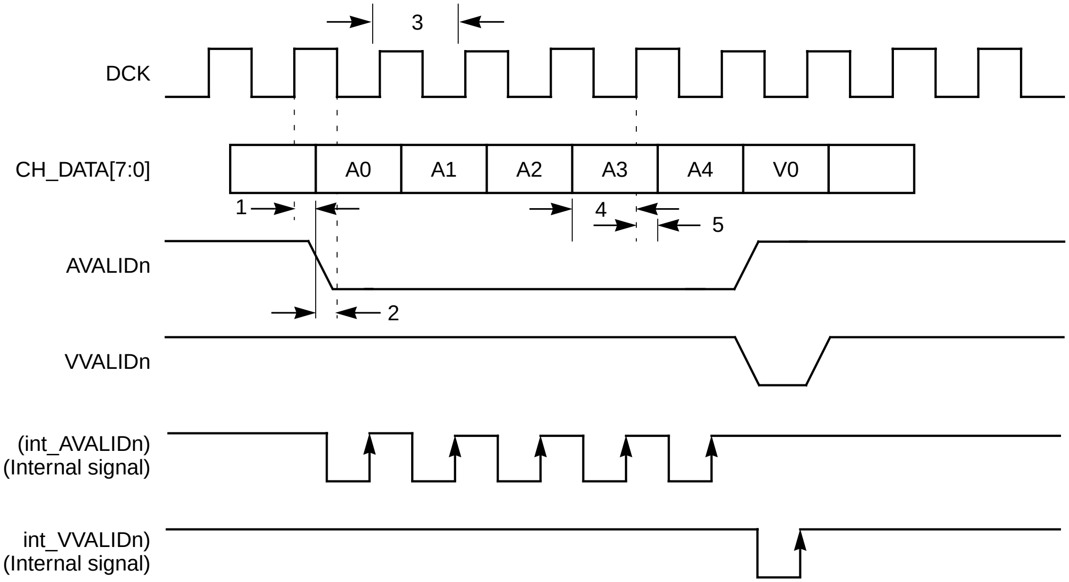
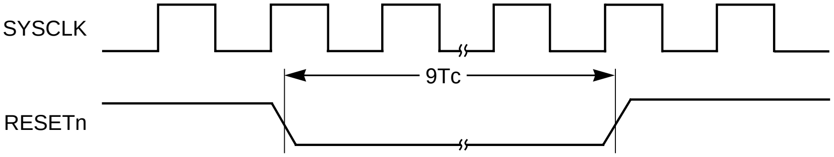


Figure 12.11 Reset Timing



Note: Tc = 1/27 MHz = 37 ns.

Table 12.10 Video Interface AC Timing

Parameter	Description	Min	Max	Units
1	HS/VS Hold Time	0	–	ns
2	HS/VS Setup Time	8	–	ns
3	HS/VS Minimum Pulse Width	1	–	SYSCLK
4	HS/VS Maximum Cycle Time	–	2047	SYSCLK
5	SYSCLK to Pixel Data Out	–	15	ns
6	SYSCLK to CREF Out	–	15	ns
7	SYSCLK to OSDA Out	–	15	ns

Figure 12.12 Video Interface Timing

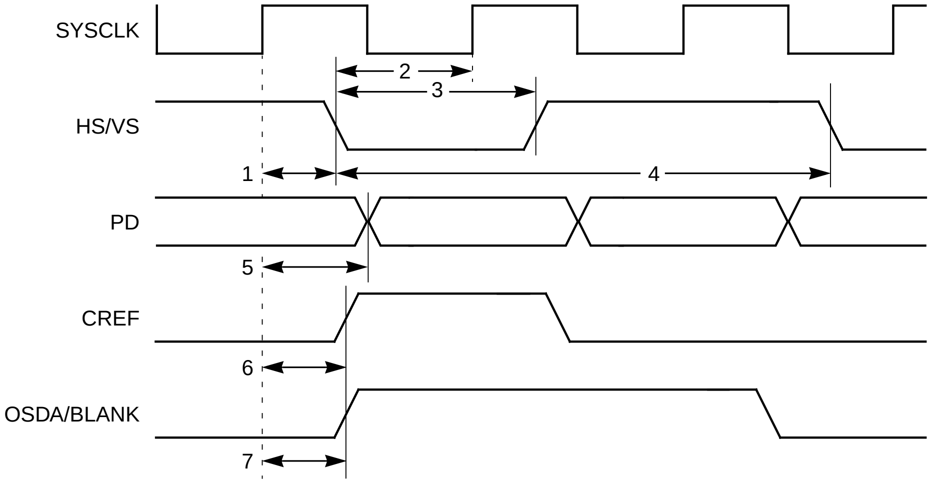


Table 12.11 Audio Interface AC Timing

Parameter	Description	Min	Max	Units
1	ASDATA change before BCLK rising	18	–	ns
2	ASDATA change after BCLK rising	18	–	ns
3	LRCLK change before BCLK rising	18	–	ns
4	A_ACLK change after ACLK input	4	13	ns
5	PREQn change after SYSCLK	6	20	ns

Figure 12.13 Serial PCM Data Out Timing

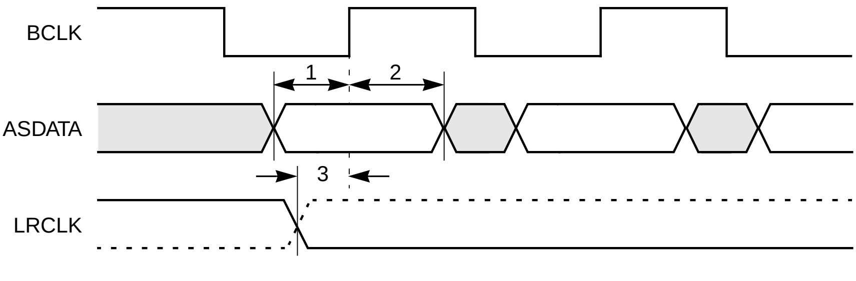


Figure 12.14 A_ACLK Timing

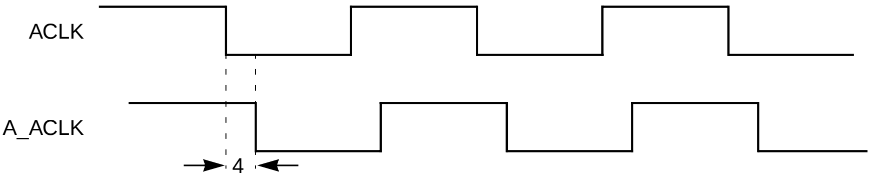
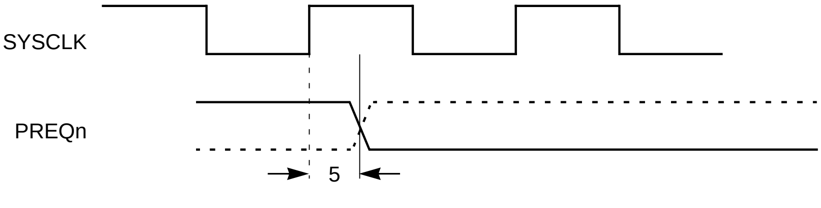


Figure 12.15 PREQn Timing



12.3 Pinouts and Packaging

The L64020 DVD Audio/Video Decoder is available in a 160-pin Plastic Quad Flat Package (PQFP), a 176-pin Thin Quad Flat Package (TQFP), and a 208-pin mini-BGA (HG) Package Array Layout. Table 12.12 lists the L64020's input/output signals in alphabetical order and includes:

- pin numbers for each package
- a signal description
- the signal type, direction, and whether it is pulled up or down in the chip
- the current drive capacity for output and bidirectional signals

Following the table, Figure 12.16 and Figure 12.17 show the pinout and outline drawing for the 160-pin package, Figure 12.18 and Figure 12.19 show the pinout and outline drawing for the 176-pin package, and Figure 12.20, Figure 12.21, and Table 12.13 show the pinout and outline drawing for the 208-pin package.

Table 12.12 Alphabetical Pin Summary

Mnemonic	160-Pin	176-Pin	208-Pin	Description	Type	Drive (mA)
A0	45	51	59 (P4)	Register Address Bus	TTL Input	–
A1	44	50	58 (R4)			
A2	43	49	57 (T3)			
A3	39	41	48 (P2)			
A4	38	40	47 (P1)			
A5	37	39	46 (N2)			
A6	36	38	45 (N1)			
A7	35	37	44 (N3)			
A8	34	36	43 (M2)			
A_ACLK	103	113	134 (H16)	DAC Clock	TTL Output	4
(Sheet 1 of 6)						

Table 12.12 Alphabetical Pin Summary (Cont.)

Mnemonic	160-Pin	176-Pin	208-Pin	Description	Type	Drive (mA)
ACLK_32	96	106	124 (K14)	Audio Reference Clock	TTL Input, pulldown	–
ACLK_48	95	105	123 (K16)	Audio Reference Clock	TTL Input, pulldown	–
ACLK_441	94	104	122 (K15)	Audio Reference Clock	TTL Input, pulldown	–
AREQn	30	32	38 (L1)	Audio Transfer Request	TTL Output	4
ASDATA	107	117	139 (G15)	Audio Serial Data	TTL Output	4
ASn	67	73	87 (R10)	Address Strobe	TTL Input, pullup	–
AUDIO_SYNC	112	122	145 (E14)	Audio Sync Strobe	TTL Output	4
AVALIDn	26	28	34 (K1)	Audio Data Valid	TTL Input, pullup	–
BCLK	105	115	137 (G14)	DAC Bit Clock	TTL Output	4
BLANK	86	96	113 (N13)	Blank	TTL Output	4
BUSMODE	60	66	76 (P8)	Host Controller Select	TTL Input, pullup	–
CD_ACLK	100	110	129 (J13)	CD Mode DAC Clock	TTL Input, pulldown	–
CD_ASDATA	101	111	132 (H13)	CD Mode Audio Data	TTL Input, pulldown	–
CD_BCLK	98	108	127 (J16)	CD Mode DAC Bit Clock	TTL Input, pulldown	–
CD_LRCLK	99	109	128 (J14)	CD Mode DAC Left/Right Clock	TTL Input, pulldown	–
CH_DATA 0	12	14	16 (F3)	Channel Data Bus	TTL Input	–
CH_DATA 1	13	15	17 (F4)			
CH_DATA 2	14	16	18 (G2)			
CH_DATA 3	15	17	19 (G1)			
CH_DATA 4	16	18	20 (G3)			
CH_DATA 5	17	19	21 (G4)			
CH_DATA 6	18	20	22 (H2)			
CH_DATA 7	19	21	23 (H1)			
(Sheet 2 of 6)						

Table 12.12 Alphabetical Pin Summary (Cont.)

Mnemonic	160-Pin	176-Pin	208-Pin	Description	Type	Drive (mA)
CREF	87	97	114 (M15)	Chroma Reference	TTL Output	4
CSn	68	74	88 (N11)	L64021 Chip Select	TTL Input, pullup	–
D0	54	60	69 (N6)	Host Interface Data Bus	TTL Bidirectional	6
D1	53	59	68 (P6)			
D2	52	58	67 (T6)			
D3	51	57	66 (R6)			
D4	50	56	65 (N5)			
D5	49	55	64 (P5)			
D6	48	54	63 (T5)			
D7	47	53	62 (R5)			
DCK	28	30	36 (L4)	External Channel Clock	TTL Input, pullup	–
DREQn	58	64	74 (R8)	DMA Transfer Request	TTL Output	4
DSn/WRITEn	64	70	83 (R9)	Data Strobe/Write Indicator	TTL Input, pullup	–
DTACKn/RDYn	62	68	81 (P9)	Data Acknowledge/ Ready	3-State Output	6
ERRORn	24	26	31 (J2)	Bitstream Error	TTL Input, pullup	–
EXT_OSD 0	88	98	115 (M16)	Palette Selection Bus	TTL Input, pulldown	–
EXT_OSD 1	89	99	116 (M14)			
EXT_OSD 2	90	100	117 (M13)			
EXT_OSD 3	91	101	118 (L15)			
HS	70	76	90 (T11)	Horizontal Sync	TTL Input	–
INTRn	59	65	75 (T8)	Interrupt	TTL Open-drain Output	6
LRCLK	106	116	138 (G16)	DAC Left/Right Sample Clock	TTL Output	4
(Sheet 3 of 6)						

Table 12.12 Alphabetical Pin Summary (Cont.)

Mnemonic	160-Pin	176-Pin	208-Pin	Description	Type	Drive (mA)
OSD_ACTIVE	72	78	93 (P12)	On-Screen Display	TTL Output	4
PD 0	73	79	94 (T12)	Pixel Data Output Bus	TTL Output	4
PD 1	74	80	95 (R12)			
PD 2	76	82	98 (T13)			
PD 3	77	83	99 (T14)			
PD 4	78	84	100 (R14)			
PD 5	82	92	108 (P15)			
PD 6	83	93	109 (P16)			
PD 7	84	94	110 (N15)			
PLLVDD	156	170	203 (A3)	PLL Power Supply	3.3 V Input	–
PLLVSS	158	172	205 (A2)	PLL Ground		–
PREQn	117	127	150 (D16)	PCM FIFO Request	TTL Output	4
READ/READn	63	69	82 (T9)	Read/Write Strobe - Read Indicator	TTL Input	–
RESETn	57	63	73 (N7)	Reset	TTL Input, pullup	–
SBA 0	144	158	188 (D7)	SDRAM Address Bus	TTL Output	6
SBA 1	142	156	185 (C8)			
SBA 2	141	155	184 (D8)			
SBA 3	140	154	181 (D9)			
SBA 4	138	152	178 (A9)			
SBA 5	137	151	177 (D10)			
SBA 6	136	150	176 (C10)			
SBA 7	134	148	173 (D11)			
SBA 8	133	147	172 (C11)			
(Sheet 4 of 6)						

Table 12.12 Alphabetical Pin Summary (Cont.)

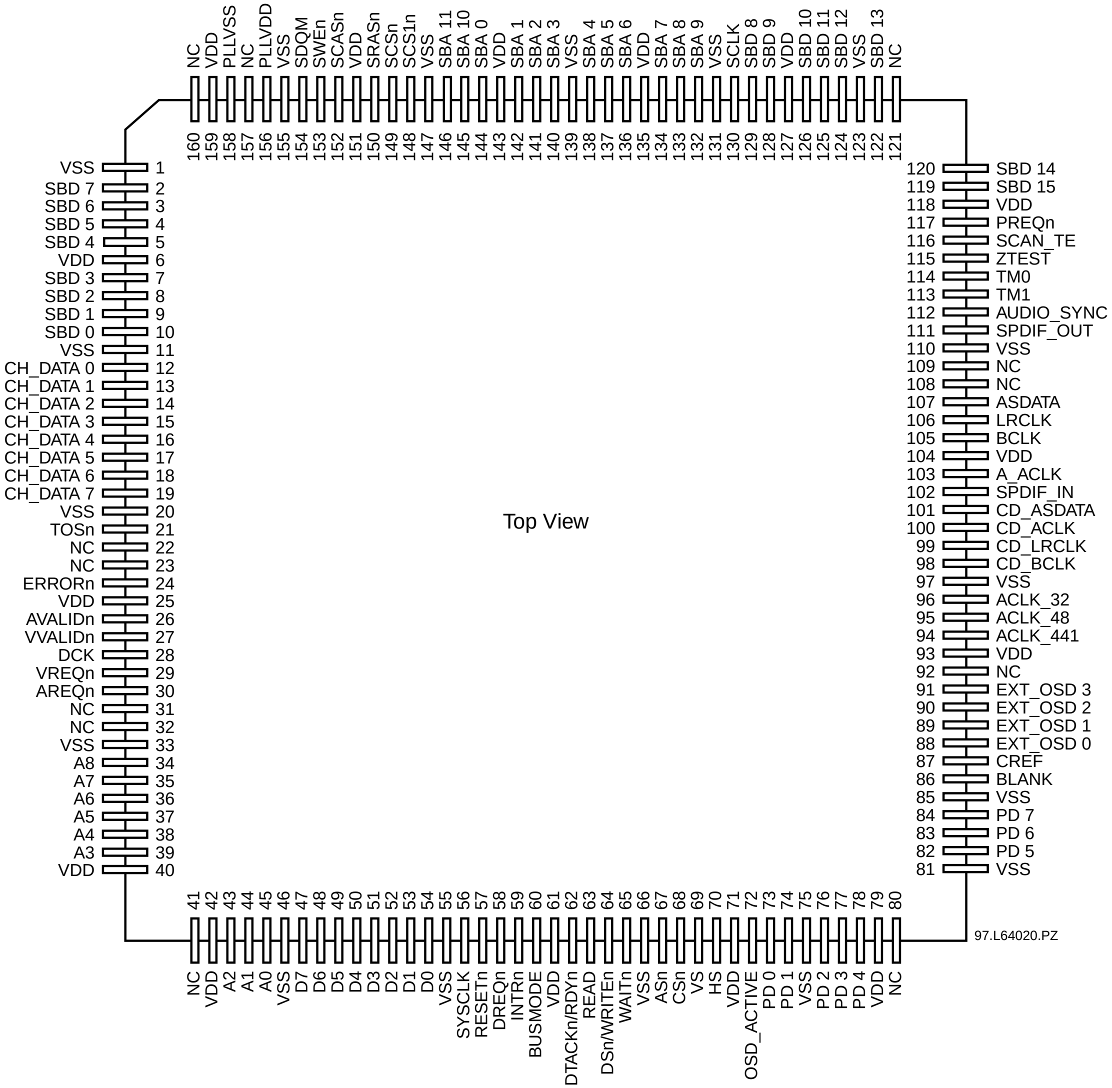
Mnemonic	160-Pin	176-Pin	208-Pin	Description	Type	Drive (mA)
SBA 9	132	146	171 (A11)	SDRAM Address Bus	TTL Output	6
SBA 10	145	159	189 (C7)			
SBA 11	146	160	190 (A7)			
SBD 0	10	12	13 (E4)	SDRAM Data Bus	TTL Bidirectional	4
SBD 1	9	11	12 (E3)			
SBD 2	8	10	11 (E1)			
SBD 3	7	9	10 (E2)			
SBD 4	5	7	7 (D3)	SDRAM Data Bus	TTL Bidirectional	4
SBD 5	4	6	6 (D2)			
SBD 6	3	5	5 (C2)			
SBD 7	2	4	4 (C1)			
SBD 8	129	143	167 (A12)			
SBD 9	128	142	166 (B12)	SDRAM Data Bus	TTL Bidirectional	4
SBD 10	126	140	163 (C13)			
SBD 11	125	139	162 (B13)			
SBD 12	124	138	161 (A14)			
SBD 13	122	136	158 (C14)			
SBD 14	120	130	154 (A16)			
SBD 15	119	129	153 (B16)			
SCAN_TE	116	126	149 (D15)	Scan Test Mode	TTL Input, pulldown	–
SCASn	152	166	198 (A5)	SDRAM Column Address Select	TTL Output	6
SCLK	130	144	168 (C12)	SDRAM 81-MHz Clock	TTL Bidirectional	4
SCSn	149	163	194 (A6)	SDRAM Chip Select	TTL Output	6
(Sheet 5 of 6)						

Table 12.12 Alphabetical Pin Summary (Cont.)

Mnemonic	160-Pin	176-Pin	208-Pin	Description	Type	Drive (mA)
SCS1n	148	162	193 (C6)	Second SDRAM Chip Select	TTL Output	6
SDQM	154	168	200 (C4)	SDRAM Control Pin	TTL Output	6
SPDIF_IN	102	112	133 (H14)	External S/P DIF	TTL Input, pulldown	–
SPDIF_OUT	111	121	144 (E13)	S/P DIF Output	TTL Output	4
SRASn	150	164	195 (B6)	SDRAM Row Address Select	TTL Output	6
SWEn	153	167	199 (B5)	SDRAM Write Enable	TTL Output	6
SYSCLK	56	62	72 (P7)	Device Clock	TTL Input	–
TM0	114	124	147 (E15)	Test Mode 0	TTL Input	–
TM1	113	123	146 (E16)	Test Mode 1	TTL Input	–
TOSn	21	23	28 (J4)	Top Of Sector	TTL Input, pulldown	–
VDD	multiple[1]	multiple[2]	multiple[3]			
VDD2			multiple[3]			
VREQn	29	31	37 (L3)	Video Transfer Request	TTL Output	4
VS	69	75	89 (P11)	Vertical Sync	TTL Input	–
VSS	multiple[1]	multiple[2]	multiple[3]			
VSS2			multiple[3]			
VVALIDn	27	29	35 (K2)	Video Data Valid	TTL Input, pullup	–
WAITn	65	71	84 (N10)	Wait - Chip Busy	3-State Output	6
ZTEST	115	125	148 (D14)	Test Mode	TTL Input	–
(Sheet 6 of 6)						

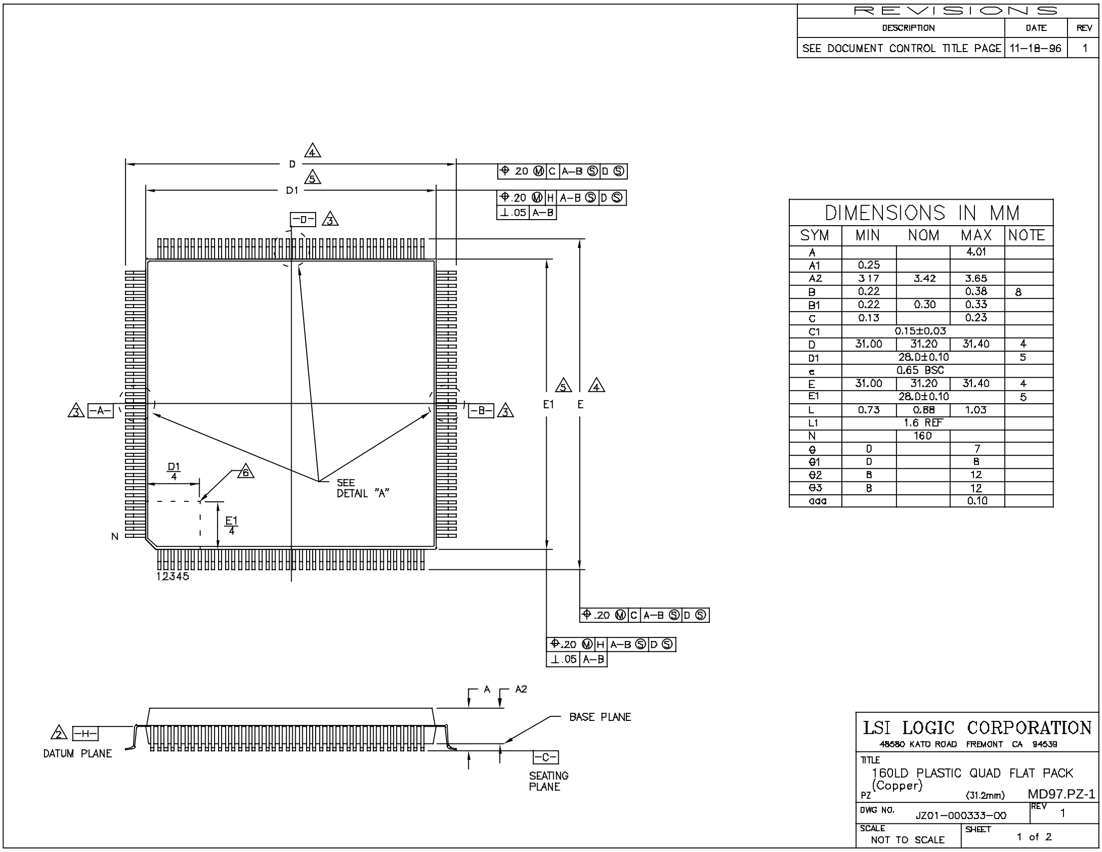
1. Refer to Figure 12.16.
2. Refer to Figure 12.18.
3. Refer to Figure 12.20 and Table 12.13.

Figure 12.16 160-Pin Package Pinout



1. *NC* pins are not connected.

Figure 12.17 160-Pin PQFP (PZ) Mechanical Drawing (Sheet 1 of 2)



REVISIONS		
DESCRIPTION	DATE	REV
SEE DOCUMENT CONTROL TITLE PAGE	11–18–96	1

DIMENSIONS IN MM				
SYM	MIN	NOM	MAX	NOTE
A			4.01	
A1	0.25			
A2	3 17	3.42	3.65	
B	0.22		0.38	8
B1	0.22	0.30	0.33	
C	0.13		0.23	
C1	0.15±0.03			
D	31.00	31.20	31.40	4
D1	28.0±0.10			5
e	0.65 BSC			
E	31.00	31.20	31.40	4
E1	28.0±0.10			5
L	0.73	0.BB	1.03	
L1	1.6 REF			
N		160		
θ	D		7	
θ1	D		B	
θ2	B		12	
θ3	B		12	
aaa			0.10	

LSI LOGIC CORPORATION
48580 KATO ROAD FREMONT CA 94539

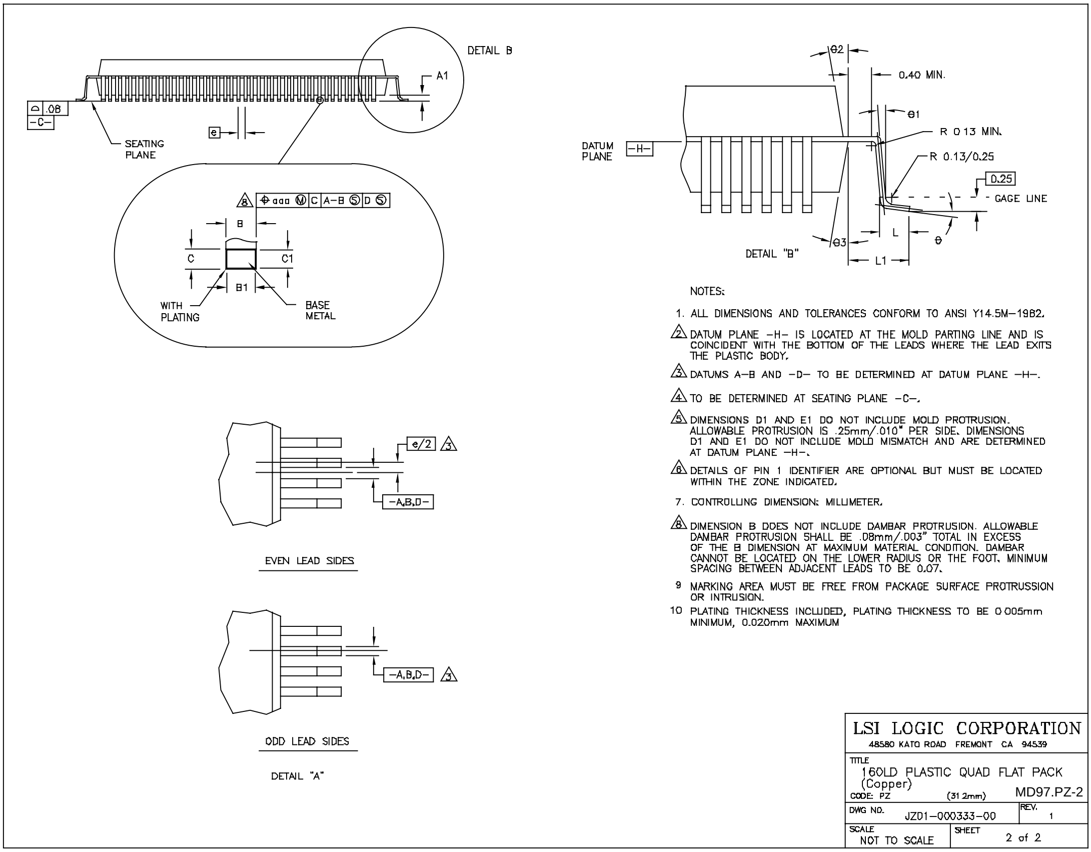
TITLE 160LD PLASTIC QUAD FLAT PACK (Copper)
PZ (31.2mm) MD97.PZ-1

DWG NO. JZ01–000333–00 REV 1

SCALE NOT TO SCALE SHEET 1 of 2

Important: **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code PZ.**

Figure 12.17 160-Pin PQFP (PZ) Mechanical Drawing (Sheet 2 of 2)



NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.

△2 DATUM PLANE -H- IS LOCATED AT THE MOLD PARTING LINE AND IS COINCIDENT WITH THE BOTTOM OF THE LEADS WHERE THE LEAD EXITS THE PLASTIC BODY.

△3 DATUMS A-B AND -D- TO BE DETERMINED AT DATUM PLANE -H-.

△4 TO BE DETERMINED AT SEATING PLANE -C-.

△5 DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS .25mm/.010" PER SIDE. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.

△6 DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL BUT MUST BE LOCATED WITHIN THE ZONE INDICATED.

7. CONTROLLING DIMENSION: MILLIMETER.

△8 DIMENSION B DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE .08mm/.003" TOTAL IN EXCESS OF THE B DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACING BETWEEN ADJACENT LEADS TO BE 0.07.

9 MARKING AREA MUST BE FREE FROM PACKAGE SURFACE PROTRUSSION OR INTRUSION.

10 PLATING THICKNESS INCLUDED, PLATING THICKNESS TO BE 0.005mm MINIMUM, 0.020mm MAXIMUM

LSI LOGIC CORPORATION
48580 KATO ROAD FREMONT CA 94539

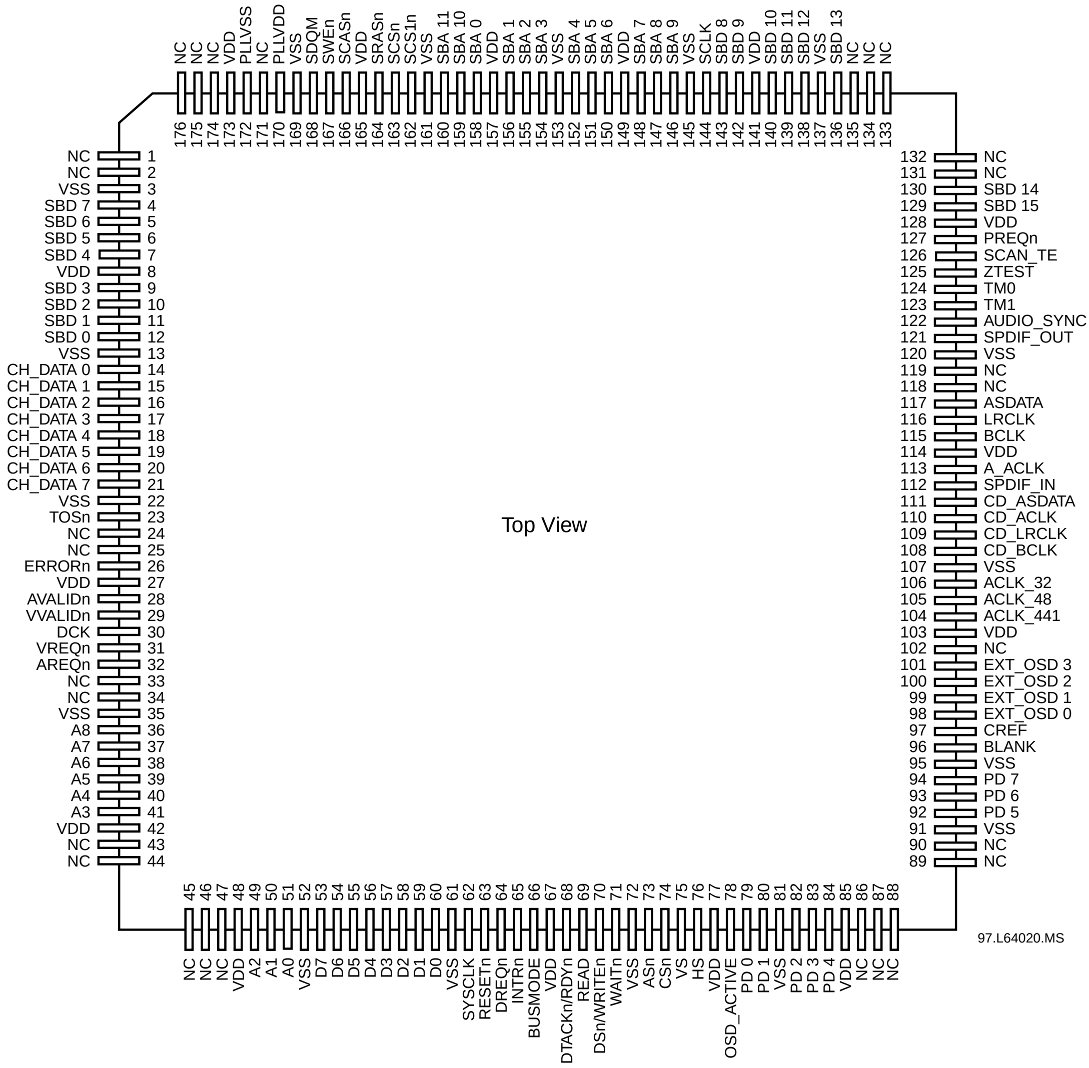
TITLE
160LD PLASTIC QUAD FLAT PACK
(Copper)
CODE: PZ (31.2mm) MD97.PZ-2

DWG NO. JZD1-000333-00
REV. 1

SCALE NOT TO SCALE
SHEET 2 of 2

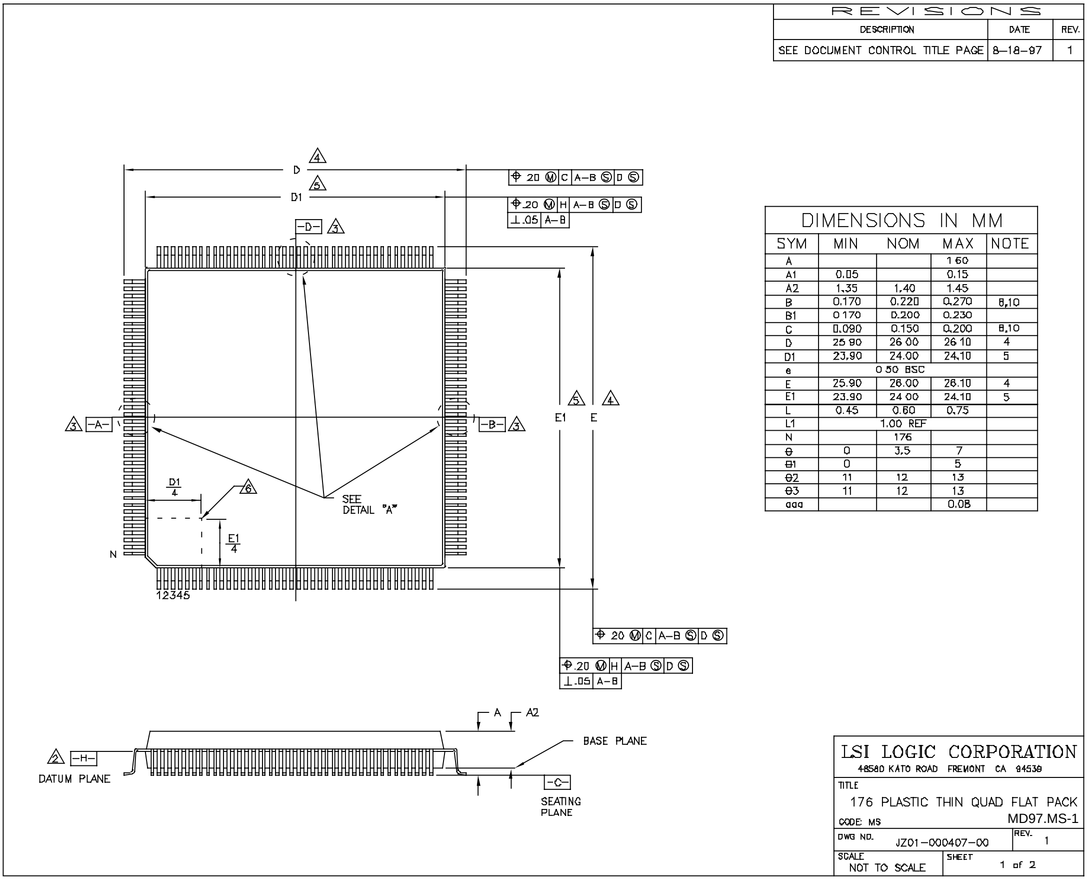
Important: **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code PZ.**

Figure 12.18 176-Pin Package Pinout



1. *NC* pins are not connected.

Figure 12.19 176-Pin TQFP (MS) Mechanical Drawing (Sheet 1 of 2)



REVISIONS		
DESCRIPTION	DATE	REV.
SEE DOCUMENT CONTROL TITLE PAGE	8–18–97	1

DIMENSIONS IN MM

SYM	MIN	NOM	MAX	NOTE
A			1 60	
A1	0.05		0.15	
A2	1.35	1.40	1.45	
B	0.170	0.220	0.270	8,10
B1	0 170	0.200	0.230	
C	0.090	0.150	0.200	8,10
D	25 90	26 00	26 10	4
D1	23.90	24.00	24.10	5
e	0 50 BSC			
E	25.90	26.00	26.10	4
E1	23.90	24 00	24.10	5
L	0.45	0.60	0.75	
L1	1.00 REF			
N		176		
θ	0	3.5	7	
θ1	0		5	
θ2	11	12	13	
θ3	11	12	13	
aaa			0.08	

LSI LOGIC CORPORATION
48580 KATO ROAD FREMONT CA 94539

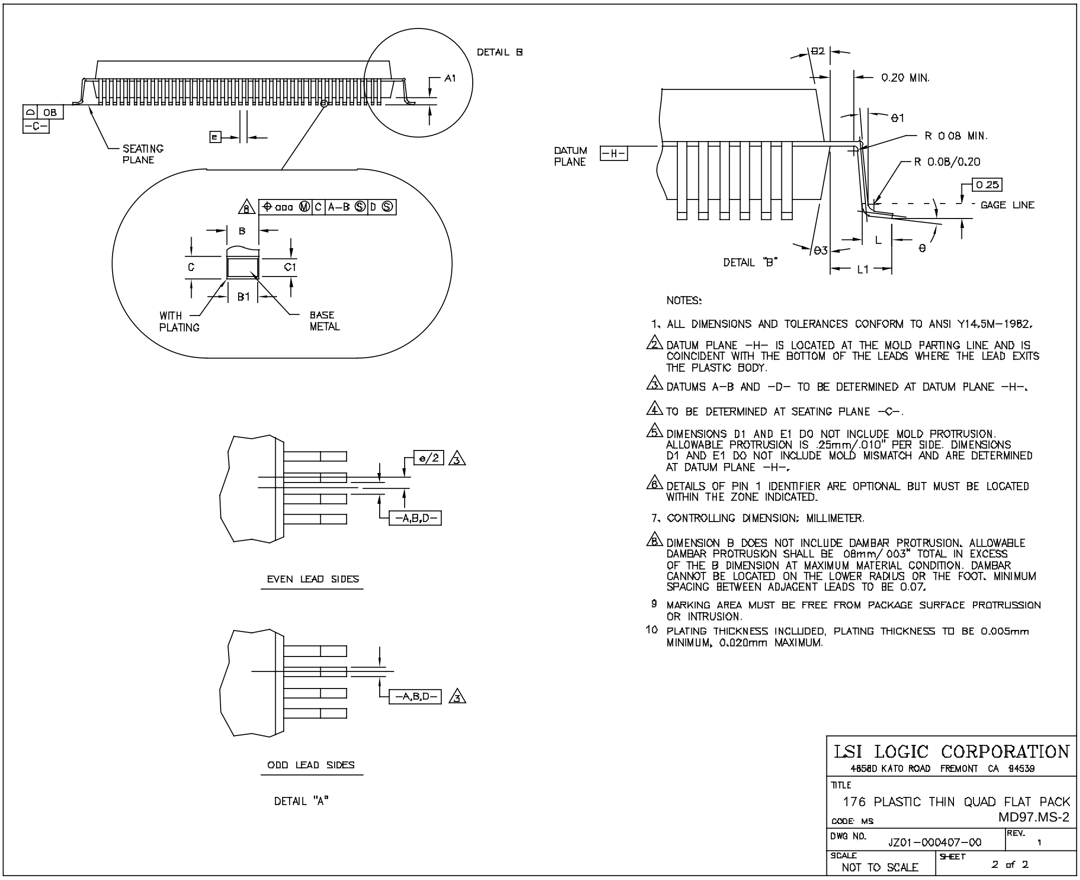
TITLE
176 PLASTIC THIN QUAD FLAT PACK
CODE: MS MD97.MS-1

DWG NO. JZ01–000407–00 REV. 1

SCALE NOT TO SCALE SHEET 1 of 2

Important: **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code MS.**

Figure 12.19 176-Pin TQFP (MS) Mechanical Drawing (Sheet 2 of 2)



NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.

△2 DATUM PLANE -H- IS LOCATED AT THE MOLD PARTING LINE AND IS COINCIDENT WITH THE BOTTOM OF THE LEADS WHERE THE LEAD EXITS THE PLASTIC BODY.

△3 DATUMS A-B AND -D- TO BE DETERMINED AT DATUM PLANE -H-.

△4 TO BE DETERMINED AT SEATING PLANE -C-.

△5 DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS .25mm/.010" PER SIDE. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.

△6 DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL BUT MUST BE LOCATED WITHIN THE ZONE INDICATED.

7. CONTROLLING DIMENSION; MILLIMETER.

△8 DIMENSION B DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 08mm/003" TOTAL IN EXCESS OF THE B DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACING BETWEEN ADJACENT LEADS TO BE 0.07.

9 MARKING AREA MUST BE FREE FROM PACKAGE SURFACE PROTRUSSION OR INTRUSION.

10 PLATING THICKNESS INCLUDED, PLATING THICKNESS TO BE 0.005mm MINIMUM, 0.020mm MAXIMUM.

LSI LOGIC CORPORATION 4858D KATO ROAD FREMONT CA 94539	
TITLE 176 PLASTIC THIN QUAD FLAT PACK CODE: MS MD97.MS-2	
DWG NO. JZ01-000407-00	REV. 1
SCALE NOT TO SCALE	SHEET 2 of 2

Important: **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code MS.**

Figure 12.20 208-Pin mini-BGA (HG) Package Array Layout

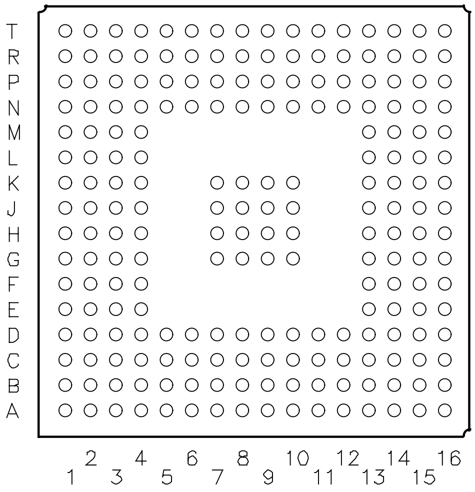


Table 12.13 208-Pin mini-BGA Array Signal Assignments

Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
A1	VDD	B10	VDD	D3	SBD 4
A2	PLLVSS	B11	VSS2	D4	VDD2
A3	PLLVDD	B12	SBD 9	D5	VDD2
A4	VSS2	B13	SBD 11	D6	VSS
A5	SCASn	B14	VSS	D7	SBA 0
A6	SCSn	B15	NC	D8	SBA 2
A7	SBA 11	B16	SBD 15	D9	SBA 3
A8	VDD2	C1	SBD 7	D10	SBA 5
A9	SBA 4	C2	SBD 6	D11	SBA 7
A10	VDD2	C3	VSS	D12	VSS
A11	SBA 9	C4	SDQM	D13	VDD2
A12	SBD 8	C5	VDD	D14	ZTEST
A13	VDD	C6	SCS1n	D15	SCAN_TE
A14	SBD 12	C7	SBA 10	D16	PREQn
A15	VSS2	C8	SBA 1	E1	SBD 2
A16	SBD 14	C9	VSS	E2	SBD 3
B1	VSS2	C10	SBA 6	E3	SBD 1
B2	VDD2	C11	SBA 8	E4	SBD 0
B3	NC[1]	C12	SCLK	E13	SPDIF_OUT
B4	VSS	C13	SBD 10	E14	AUDIO_SYNC
B5	SWEn	C14	SBD 13	E15	TM0
B6	SRASn	C15	VDD2	E16	TM1
B7	VSS2	C16	VDD	F1	VSS2
B8	VDD	D1	VDD	F2	VSS
B9	VSS2	D2	SBD 5	F3	CH_DATA 0

(Sheet 1 of 2)

Table 12.13 208-Pin mini-BGA Array Signal Assignments (Cont.)

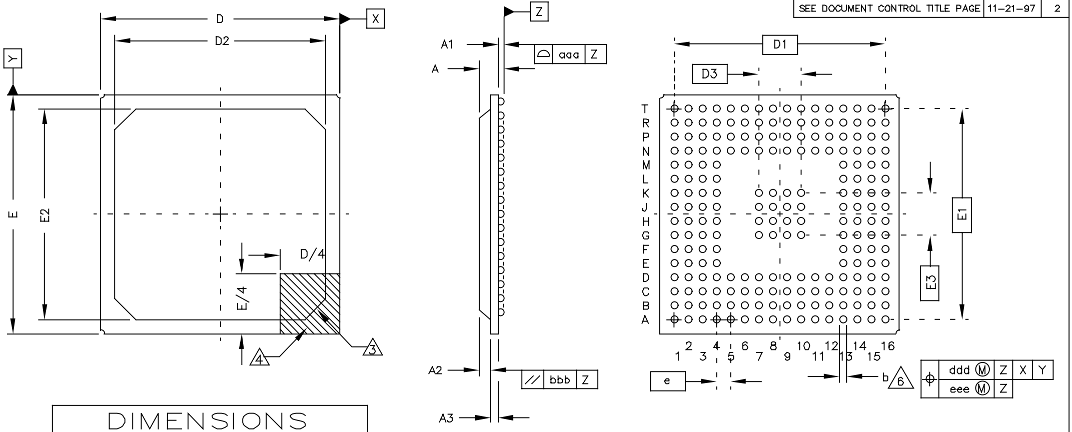
Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name
F4	CH_DATA 1	K7	NC	P6	D1
F13	NC	K8	NC	P7	SYSCLK
F14	NC	K9	NC	P8	BUSMODE
F15	VSS	K10	NC	P9	DTACKn/RDYn
F16	VSS2	K13	VSS2	P10	VSS2
G1	CH_DATA 3	K14	ACLK_32	P11	VS
G2	CH_DATA 2	K15	ACLK_441	P12	OSD_ACTIVE
G3	CH_DATA 4	K16	ACLK_48	P13	VSS2
G4	CH_DATA 5	L1	AREQn	P14	VSS2
G7	NC	L2	NC	P15	PD 5
G8	NC	L3	VREQn	P16	PD 6
G9	NC	L4	DCK	R1	VDD
G10	NC	L13	VDD2	R2	NC
G13	VDD	L14	VDD	R3	VDD2
G14	BCLK	L15	EXT_OSD 3	R4	A1
G15	ASDATA	L16	NC	R5	D7
G16	LRCLK	M1	VSS	R6	D3
H1	CH_DATA 7	M2	A8	R7	VSS2
H2	CH_DATA 6	M3	VSS2	R8	DREQn
H3	VSS	M4	NC	R9	DSn/WRITEn
H4	VSS2	M13	EXT_OSD 2	R10	ASn
H7	NC	M14	EXT_OSD 1	R11	VDD2
H8	NC	M15	CREF	R12	PD 1
H9	NC	M16	EXT_OSD 0	R13	VSS
H10	NC	N1	A6	R14	PD 4
H13	CD_ASDATA	N2	A5	R15	NC
H14	SPDIF_IN	N3	A7	R16	VSS
H15	VDD2	N4	VSS2	T1	VDD2
H16	A_ACLK	N5	D4	T2	VDD
J1	NC	N6	D0	T3	A2
J2	ERRORn	N7	RESETn	T4	VSS
J3	NC	N8	VDD	T5	D6
J4	TOSn	N9	VDD2	T6	D2
J7	NC	N10	WAITn	T7	VSS
J8	NC	N11	CSn	T8	INTRn
J9	NC	N12	VDD	T9	READ
J10	NC	N13	BLANK	T10	VSS
J13	CD_ACLK	N14	VSS2	T11	HS
J14	CD_LRCLK	N15	PD 7	T12	PD 0
J15	VSS	N16	VSS	T13	PD 2
J16	CD_BCLK	P1	A4	T14	PD 3
K1	AVALIDn	P2	A3	T15	VDD2
K2	VVALIDn	P3	NC	T16	VDD
K3	VDD	P4	A0		
K4	VDD2	P5	D5		

(Sheet 2 of 2)

Note: *NC* pins are not connected.

Figure 12.21 208-Pin mini-BGA (HG) Mechanical Drawing

REVISIONS		
DESCRIPTION	DATE	REV.
SEE DOCUMENT CONTROL TITLE PAGE	11–21–97	2



DIMENSIONS				
SYM	MIN	NOM	MAX	NOTES
A	1.35	1.56	1.76	
A1	0.30	0.40	0.50	
A2	0.73	0.80	0.88	
A3	0.29	0.36	0.40	
b	0.35	0.50	0.65	
D	16.80	17.00	17.20	
D1		15.00 BSC		
D2	14.00	15.00	16.00	
D3		3.00 BSC	–	
E	16.80	17.00	17.20	
E1		15.00 BSC		
E2	14.00	15.00	16.00	
E3		3.00 REF		
e		1.00 BSC		
aaa	–	–	0.15	
bbb			0.25	
ddd	–	–	0.30	
eee	–	–	0.10	
TOTAL NO. OF BALLS		208		

NOTES:

1 DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.

2 ALL DIMENSIONS IN MM.

3 CORNER DETAIL IS LSI LOGIC CORPORATION OPTION.

4 DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL, AND MAY CONSIST OF INK, LASER MARK OR METALLIZED MARKING, BUT MUST BE LOCATED WITHIN THE ZONE INDICATED

5 SOLDER BALLS ARE TYPICALLY 63/37 TIN LEAD.

6 JEDEC CODE MO–151.

7 PRIMARY DATUM Z AND SEATING PLANE ARE DEFINE BY THE SPHERIACAL CROWNS OF THE SOLDER BALLS.

LSI LOGIC CORPORATION
1551 McCARTHY BLVD. MILPITAS, CA 95035

TITLE 208 mBGA 17X17mm 1.0mm Ball Pitch CODE HG MD98.HG	
DWG NO. JZ01–000499–00	REV. 2
SCALE NOT TO SCALE	SHEET 1 of 1

Important: **This drawing may not be the latest version. For board layout and manufacturing, obtain the most recent engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code HG.**

Appendix A
Video/Audio Compression and Decompression Concepts

This appendix provides an overview of MPEG compression and decompression for audio and video. It contains the following sections:

- Section A.1, “Video Compression and Decompression Concepts,” page 1-1
- Section A.2, “Audio Compression and Decompression Concepts,” page 1-7

A.1 Video Compression and Decompression Concepts

The MPEG standard defines a format for compressed digital video. Encoders designed to work within the confines of the standard compress video information, and decoders decompress it.

The MPEG algorithms for video compression and decompression are flexible, but generally fit the following criteria:

- Data rates are about 1 to 1.5 Mbit/s for MPEG-1 and up to 15 Mbit/s for MPEG-2. The L64020 MPEG-2 decoder is capable of supporting sustained data rates up to 20 Mbit/s for either MPEG-1 or MPEG-2.
- Resolutions are about 352 pixels horizontally up to about 288 lines vertically for MPEG-1 and 720 x 576 for MPEG-2 (main profile). The L64020 is capable of resolutions up to 720 x 576 for either MPEG-1 or MPEG-2.
- Picture rates are between 24 to 30 pictures per second.

A.1.1 Video Encoding

For a video signal to be compressed, it must be sampled, digitized, and converted to luminance and chrominance signals (Y, Cr, Cb). The MPEG standard stipulates that the brightness or luminance component (Y) be sampled with respect to the color difference or chrominance signals (Cr and Cb) by a ratio of 4:1. That is, for every four samples of Y, there is to be one subsample each of Cr and Cb. This is because the human eye is much more sensitive to luminance than to chrominance. Video sampling takes place in both the vertical and horizontal directions. Once video is sampled, it is reformatted, if necessary, into a non-interlaced signal. An interlaced signal contains two fields for each frame, one with all of the odd lines and the other with all of the even lines.

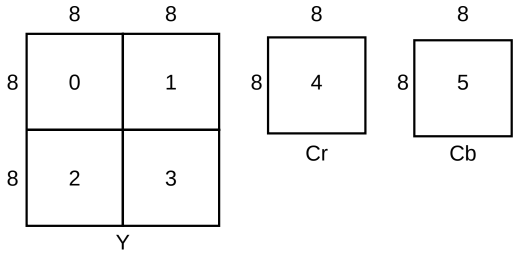
The encoder must also choose which picture type to use. A picture corresponds to a single frame of motion video, or to a movie frame. There are three picture types:

- Intracoded pictures (*I pictures*) are coded without reference to any other pictures.
- Predictive-coded pictures (*P pictures*) are coded using motion-compensated prediction from the past I or P reference pictures.
- Bidirectionally predictive-coded pictures (*B pictures*) are coded using motion-compensation predictions from a previous and a future I or P picture.

A typical coding scheme contains a mixture of I, P, and B pictures. An I picture may occur every half of a second, with two B pictures inserted between each pair of I or P pictures.

Once the picture types have been defined, the encoder must estimate motion vectors for each *macroblock* in the picture. A macroblock (see Figure A.1) consists of a 16-pixel by 16-line section of luminance and two spatially corresponding 8-pixel by 8-line sections, one for each chrominance component. Each macroblock then consists of a total of six 8 x 8 blocks (four 8 x 8 luminance blocks, one 8 x 8 Cr block, and one 8 x 8 Cb block). The spatial picture area covered by the four 8 x 8 blocks of luminance is the same area covered by each of the 8 x 8 chrominance blocks.

Figure A.1 MPEG Macroblock Structure



Motion vectors define the displacement of the image in the current macroblock from its position in the previous picture. P pictures use motion compensation to exploit temporal redundancy in the video.

When an encoder provides B pictures, it must reorder the picture sequence so that the decoder operates properly. Because B pictures use motion compensation based on previously sent I or P pictures, they can only be decoded after the referenced pictures have been sent.

For a given macroblock, the encoder must choose a coding mode. The coding mode depends on the picture type, the effectiveness of motion compensation in the particular region of the picture, and the nature of the signal within the macroblock. In addition for MPEG-2, the encoder must code the macroblock as either a field or frame. After it selects the coding method, the encoder performs a motion-compensated prediction of the block contents based on past and/or future reference pictures. The encoder then produces an error signal by subtracting the prediction from the actual data in the current macroblock. The error signal is a macroblock and a discrete cosine transform (DCT) is performed on each 8 x 8 block.

The DCT operation converts an 8 x 8 block of pixel values to an 8 x 8 matrix of horizontal and vertical spatial frequency coefficients. An 8 x 8 block of pixel values can be reconstructed by performing the inverse discrete cosine transform (IDCT) on the spatial frequency coefficients. In general, most of the energy is concentrated in the low frequency coefficients, which are located in the upper left corner of the transformed matrix. A quantization step achieves compression—where an index identifies the quantization intervals. Because the encoder identifies the interval and not the exact value within the interval, the pixel values of the block reconstructed by the IDCT have reduced accuracy.

The DCT coefficient in the upper left location (0, 0) of the block represents the zero horizontal and zero vertical frequencies and is known as the *DC coefficient.* The DC coefficient is proportional to the average pixel value of the 8 x 8 block, and additional compression is provided through predictive coding because the difference in the average value of neighboring 8 x 8 blocks tends to be relatively small.

The other coefficients represent one or more nonzero horizontal or nonzero vertical spatial frequencies, and are called *AC coefficients.* The quantization level of the coefficients corresponding to the higher spatial frequencies favors the creation of an AC coefficient of zero by choosing a quantization step size such that the human visual system is unlikely to perceive the loss of the particular spatial frequency, unless the coefficient value lies above the particular quantization level. The statistical encoding of the expected runs of consecutive zero-valued coefficients of higher-order coefficients accounts for some coding gain.

To cluster nonzero coefficients early in the series and to encode as many zero coefficients as possible following the last nonzero coefficient in the ordering, the coefficient sequence is specified to be a *zigzag ordering*. Zigzag ordering concentrates the highest spatial frequencies at the end of the series. The MPEG-2 standard includes additional block scanning orders.

After block scanning has been performed, the encoder performs *run-length coding* on the AC coefficients. This process reduces each 8 x 8 block of DCT coefficients to a number of events represented by a nonzero coefficient and the number of preceding zero coefficients. Because many coefficients are likely to be zero after quantization, run-length coding increases the overall compression ratio.

The encoder then performs *variable-length coding* (VLC) on the resulting data. VLC is a reversible procedure for coding that assigns shorter codewords to frequent events and longer codewords to less frequent events, thereby reducing the number of bits necessary to represent a data set without losing any information. Huffman encoding is a particularly well known form of VLC.

The final compressed video data is now ready for transmission to either a local storage device from which a video decoder may later retrieve and decompress the data, or to a remote video decoder via cable, or direct satellite broadcast, for example.

A.1.2 Bitstream Syntax

The MPEG standard specifies the syntax for a compressed bitstream. The video syntax contains six layers, each of which supports either a signal processing or a system function. The layers and their functions are described in Table A.1.

Table A.1 MPEG Compressed Bitstream Syntax

Syntax Layers	**Function**
Sequence Layer	Random Access Unit: Context
Group of Pictures Layer	Random Access Unit: Video
Picture Layer	Primary Coding Unit
Slice Layer	Resynchronization Unit
Macroblock Layer	Motion Compensation Unit
Block Layer	DCT Unit

The MPEG syntax layers correspond to a hierarchical structure. A *sequence* is the top layer of the video coding hierarchy and consists of a header and some number of *groups-of-pictures (GOPs).* The sequence header initializes the state of the decoder so that it is not affected by past decoding history.

A GOP is a random access point, that is, it is the smallest coding unit that can be independently decoded within a sequence. The GOP consists of a header and some number of pictures. The GOP header contains time and editing information.

The three types of pictures as explained earlier are:

- I pictures
- P pictures
- B pictures

Because of the picture dependencies, the bitstream order (the order in which pictures are transmitted, stored, or retrieved) is not the display order but rather the order in which the decoder requires the pictures for decoding the bitstream. For example, a typical sequence of pictures in display order might be as shown in Figure A.2.

Figure A.2 Typical Sequence of Pictures in Display Order

I	B	B	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

In contrast, the bitstream order corresponding to the given display order would be as shown in Figure A.3.

Figure A.3 Typical Sequence of Pictures in Bitstream Order

I	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P	B	B
0	3	1	2	6	4	5	9	7	8	12	10	11	15	13	14	18	16	17

Because the B pictures depend on the subsequent I or P picture in the display order, the I or P picture must be transmitted and decoded before the dependent B pictures.

Pictures consist of a header and one or more *slices.* The picture header contains time, picture type, and coding information. Slices consist of a header and one or more macroblocks. The slice header contains position and quantizer scale information. A slice provides some immunity to data errors. Should the bitstream become unreadable within a picture, the decoder should be able to recover by waiting for the next slice without having to drop an entire picture.

A macroblock is the basic unit for motion compensation and quantizer scale changes. In MPEG-2 the macroblock can be either field or frame coded. Each macroblock consists of a header and the six 8 x 8 blocks. The macroblock header contains quantizer scale and motion compensation information. A skipped macroblock is one for which no DCT information is encoded.

Blocks are the basic coding unit, and the DCT is applied at the block level. Each block is transformed into a set of frequency coefficients which are quantized and encoded to reduce the number of bytes needed to represent the block.

A.1.3 Video Decoding

Video decoding is the reverse of video encoding and is intended to reconstruct a moving picture sequence from a compressed, encoded bitstream. Decoding is simpler than encoding because there is no motion estimation performed and there are far fewer options.

The data in the bitstream is decoded according to the syntax defined in the MPEG-2 standard. The decoder must first identify the beginning of a coded picture, identify the type of picture, then decode each individual macroblock of the picture. Motion vectors and macroblock types (each of the picture types I, P, and B have their own macroblock types) present in the bitstream are used to construct a prediction of the current macroblock based on past and future reference pictures that the encoder has already decoded and stored. Coefficient data is then inverse quantized and operated on by an inverse DCT process that changes data from the frequency domain to the time and space domain.

After the decoder processes all of the macroblocks, the picture reconstruction is complete. If the picture just reconstructed is a reference picture (I or P picture), it replaces the oldest stored reference picture and is used as the new reference for subsequent pictures. The pictures may need to be reordered before they are displayed.

A.2 Audio Compression and Decompression Concepts

Given an *elementary stream* of audio data, an MPEG encoder first digitally compresses and codes the data. The MPEG algorithm offers a choice of levels of complexity and performance for this process.

To prepare a stream of compressed audio data for transmission, it is formatted into *audio frames*. Each audio frame contains audio data, error-correction data, and optional user-defined *ancillary data*. The audio frames are then sent in *packets* grouped within *packs* in an ISO MPEG *System Stream*.

The packs in system streams may contain a mix of audio packets and video packets for one or more channels. Packs may contain packets from separate elementary streams. Thus, MPEG can easily support multiple channels of program material, and a decoder given access to a system stream may access large numbers of channels.

A.2.1 MPEG Audio Encoding

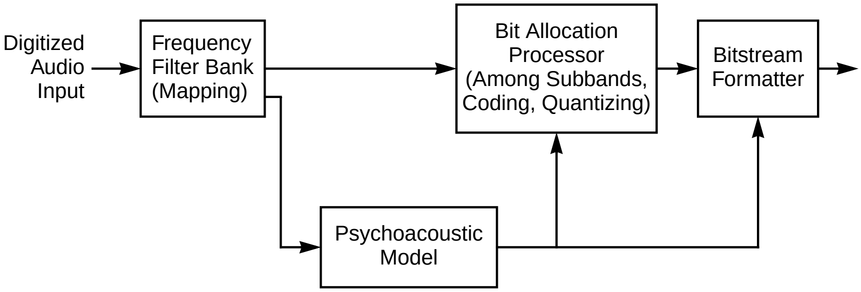
MPEG audio encoding is intended to efficiently represent a digitized audio stream by removing redundant information. Because different applications have different performance goals, MPEG uses different encoding techniques. These techniques, called *Layers*, provide different trade-offs between compression and signal quality. The MPEG algorithm uses the two following processes for removing redundant audio information:

- Coding and quantization
- Psychoacoustic modeling

Coding and quantization are techniques that are applied to data that has been mapped into the frequency domain and filtered into subbands. Psychoacoustic modeling is a technique that determines the best allocation of data within the available data channel bandwidth based on human perception.

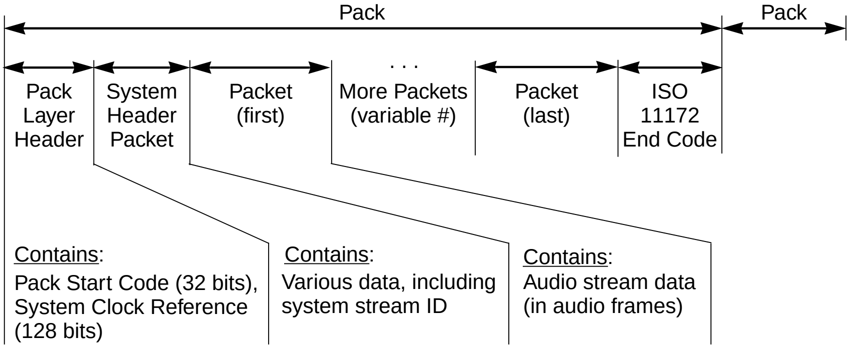
The general structure of an MPEG audio encoder is shown in Figure A.4.

Figure A.4 Audio Encoding Process (Simplified)


Digitized Audio Input
Frequency Filter Bank (Mapping)
Bit Allocation Processor (Among Subbands, Coding, Quantizing)
Bitstream Formatter
Psychoacoustic Model

Once audio data has been coded, it may be stored or transmitted digitally. MPEG provides a framework for use of packet-oriented transmission of compressed data. In particular, ISO CD 11172 defines formats for digital data streams for both video and audio. The ISO System Stream format is designed to accommodate both audio packets and video packets within the same framework for transmission. The data may be physically delivered in parallel form or serial form. The System Stream is composed of a sequence of packs, as shown in Figure A.5.

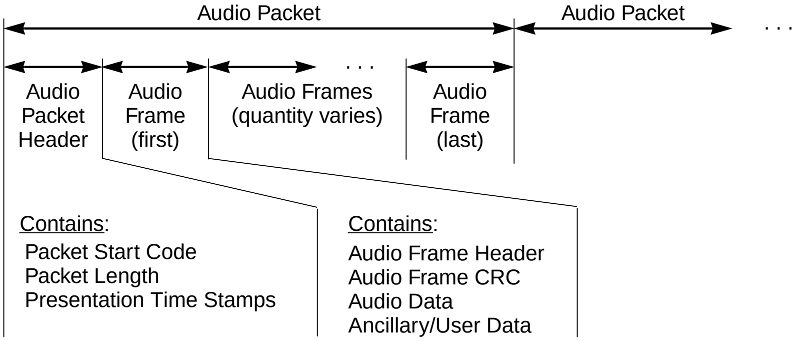
Figure A.5 ISO System Stream


Pack
Pack
Pack Layer Header
System Header Packet
Packet (first)
. . .
More Packets (variable #)
Packet (last)
ISO 11172 End Code
Contains:
Pack Start Code (32 bits), System Clock Reference (128 bits)
Contains:
Various data, including system stream ID
Contains:
Audio stream data (in audio frames)

An MPEG pack is composed of a *pack layer header*, a *system header packet*, a sequence of *packets*, and ends with an ISO 11172 *end code.* The pack layer header contains a pack start code used for synchronization purposes, and a system clock value. The system header packet contains a variety of housekeeping data, in particular, a system stream ID used to differentiate among multiple system streams. A sequence of one or more packets contains either encoded audio or encoded video stream data. The ISO 11172 end code is the final element in an MPEG pack. For detailed definition of pack headers, refer to the ISO CD 11172-1 system stream descriptions.

Any one MPEG packet carries either audio or video data, but not both simultaneously. An MPEG Audio Packet contains an audio packet header and one or more Audio Frames. Figure A.6 shows the packet structure.

Figure A.6 MPEG Audio Packet Structure



A.2.1.1 Audio Packet Header

An audio packet header contains the following:

- **Packet Start Code**

  Identifies a packet as an audio packet. The Packet Start Code also contains a five-bit audio stream identifier that may be set by you to identify the audio channel.

- **Packet Length**

  Indicates the number of bytes remaining in the audio packet.

- **Presentation Time Stamps (PTS**)

  The PTS indicates when audio data should be presented.

A.2.1.2 Audio Frame

An audio frame contains a slice of the audio data stream together with some supplementary data. Audio frames have the following elements:

- **Audio Frame Header**

  Data in the audio frame header set the parameters that describe the format and mode of the audio data.

- **Audio Frame Cyclic Redundancy Code (CRC)**

  This field contains a 16-bit checksum, which can be used to detect errors in the audio frame header.

- **Audio Data**

  The decoder uses the audio data to reconstruct the sampled audio data. Its format is beyond the scope of this document. The data structures for Layer I dual channel/stereo, intensity stereo, and for the more complex Layer II audio data fields are described in Sections 2.4.1.5 and 2.4.1.6 of the ISO/IEC 11172-3.

- **Ancillary Data**

  The final field in an audio frame containing user-defined data (ancillary data) is discarded by the L64020.

A.2.2 Audio Decoding

Audio decoding is the reverse of audio encoding and is intended to reconstruct the compressed audio data. MPEG audio decoding involves:

- Identifying and removing a channel’s audio frames from the audio packets in the System Stream.
- Managing the temporary storage of frames.
- Applying appropriate algorithms for decoding the audio frames.
- Merging decoded audio frames back into continuous audio.
- Synchronizing the audio with the video.
- Limiting the effect of transmission errors.

Appendix B
Glossary of Terms and Abbreviations

Numerics

3:2 Pulldown

Film material digitized at 24 pictures per second forms an excellent source for the MPEG video bitstream. To display 24 frame-per-second video at the television frame rate of 30 frames-per-second, 3:2 pulldown is necessary. A single frame of 24 frames per second video is displayed three times at the television field rate of 60 fields-per-second, followed by the next frame displayed two times. This pattern of three and then two repeated frames continues. The net result is that a total of two frames of 24 frame-per-second video is displayed over a period of five television field times, or 5/60ths (1/12th) of a second. This result is exactly the same amount of time occupied by two frames of 24 frame per second video (2/24 = 1/12).

B

B Picture **Bidirectionally Predictive-Coded Picture**

B pictures in an MPEG bitstream are pictures that are predicted from an I or P (anchor) picture and are compressed by coding the differences between the B picture and the referenced I or P picture.

bap **bit allocation pointer**

The outputs of the bit allocation computation in the Dolby Digital Decoder are a set of bit allocation pointers (baps), one for each coded mantissa. The bap indicates the quantizer used for the mantissa and how many bits in the bitstream were used to encode each mantissa.

C

CRC **Cyclic Redundancy Check**
Bitstream error detection scheme. A check performed on data to see if an error has occurred in transmitting, reading, or writing the data. The result of a CRC is typically stored or transmitted with the checked data. The stored or transmitted result is then compared to a CRC calculated for the data to determine if an error has occurred.

Chroma **Chrominance**
The color information portion of a signal; UV portion of YUV color system or CbCr portion of the YCbCr color system. Both systems are mathematically derived from the RGB (red, blue, green) system used in television. See YUV and YCbCr.

CMOS **Complementary Metal-Oxide Semiconductor**
An electronic circuit fabricated on a silicon chip that uses charge-based switching to provide high integration and low power dissipation.

D

DAC **Digital-to-Analog Converter**
An integrated circuit that converts a serial PCM bitstream into a continuous analog signal. Typically used to convert the digital audio in MPEG bitstream to analog to drive the system speakers.

DCSQ **Display Control Sequence**
DCSQ analysis uses the PTS (Presentation Time Stamps) and STM (Start Times) stored in DVD bitstreams to synchronize the presentation of SPUs (Subpicture Units) with the SCR (System Clock Reference).

DMA **Direct Memory Access**
Direct communication between an I/O device and memory without CPU intervention. Used for high-speed transfers and with busy CPUs.

DRAM **Dynamic Random Access Memory**
RAM that does not require continuous power but periodic power refreshes for data retention.

DSI **Data Search Information**
Present in DVD navigation packets. Not used by the L64020 but made available in an SDRAM buffer for the host.

DTS **Decode Time Stamp**
Decoding times for presentation units extracted from the PES (Packetized Elementary Stream) headers. Used by the L64020 to start reading data from the video channel buffer and decoding it into frame stores.

DVD **Digital Versatile Disk**
Previously Digital Video Disk. Now used generically to indicate any component, bitstream, or system that supports DVD elements.

E

EAV **End of Active Video**
A CCIR656 timing code programmed into the output bitstream by the host to mark the end of the active display area. See CCIR656 and SAV.

Endian **Byte Order**
The endian of a component specifies the order of the bytes in multiple-byte formats. Little endian indicates that the most significant byte is in the highest bit positions and the least significant byte is in the lowest bit positions. Big endian signifies that the most significant byte is in the lowest bit positions and the least significant byte is in the highest bit positions.

ES **Elementary Stream**
In MPEG, elementary streams are bitstreams containing compressed data from a single source, such as video, audio, etc. Elementary streams are combined in a single stream by packetizing them. See PES.

F

Field

In television, a single frame consists of two fields containing the odd and even scan lines, respectively.

FIFO **First In, First Out**
FIFOs are often referred to as buffers or elastic memory. They are contiguous memory locations specified by width and depth such as 8 bits x 8 locations. In one type, parallel data is strobed into the input stage and it propagates to the first empty stage at the output. When the data at the output is strobed out, the rest of the data moves toward the output stage to fill the empty locations. In another type, referred to as a circular buffer, input and output pointers rotate through the memory locations as data is written in and read out.

Frame
In motion video, a single image. Frames can be presented at 25 frames per second (PAL standard) or at 30 frames per second (NTSC standard).

G

GOF **Group of Frames**
A Linear PCM audio bitstream is divided in groups of audio frames. Each GOF includes several audio packs which, in turn, contain header data and audio frames.

GOP **Group of Pictures**
MPEG bitstreams are divided into sequences, groups of pictures, picture slices, macroblocks, and blocks in that order. Each GOP includes at least one I picture and one or more P and B pictures. See I, P, and B pictures.

I

I Picture **Intraframe Picture**
An I picture is a video frame that is individually compressed and encoded without reference to another frame. It is referred to as an anchor frame and is used to predict successive frames using motion estimation.

IDCT **Inverse Discrete Cosine Transform**
An IDCT converts digital data from the frequency domain into the time (spatial) domain.

IEC **International Electrotechnical Commission**

IEC958
The S/P DIF specification adopted by the IEC. See S/P DIF.

ITU **International Telecommunications Union**

ITU-R BT.601

Recommendation for digital video (4:2:2, 720 samples per line). Also recommends chromaticity for YCbCr color space.

ITU-R BT.656

Recommendation for the generation of SAV/EAV (Start of Active Video/End of Active Video) timing codes in the bitstream. The SAV/EAV timing codes determine the vertical blanking interval and the location of the active display area.

L

LPCM **Linear PCM**

See PCM. A linear PCM encoder codes analog samples on a straight-line basis, for example, sample amplitude 1 is coded as a binary 1, 2 as a binary 2, 3 as a binary 3, etc. In A/V systems, Linear PCM refers to audio that has been encoded in this manner. Non-linear PCM encoders are used in systems such as telephone carrier systems to reduce low-level noise.

LSB **Least Significant Bit/Byte**

The bit or byte in a larger binary element, such as a word, that has the smallest value.

Luma **Luminance**

Brightness of an image (Y portion of a YUV signal or a YCbCr signal). See YUV and YCbCr.

M

MPEG **Motion Picture Expert Group**

An IEEE (Institute of Electrical and Electronic Engineers) committee that sets standards for compression and format of motion picture (video) bitstreams and accompanying audio information.

MSB **Most Significant Bit/Byte**

The bit or byte in a larger binary element, such as a word, that has the largest value.

MUSICAM **MPEG Audio**

See MPEG.

N

NTSC **National Television Standards Committee**
A committee which set the television standard used today in United States and Japan. The standard dictates a 720-pixel wide by 525-line high display in a 4:3 aspect ratio produced by two interlaced scans alternating at 60 scans per second; one for odd lines and one for even lines. The standard also defines a luminance/chrominance color system that allows black and white TVs to receive a color picture in black and white.

O

OSD **On-Screen Display**
Graphics overlay on video background. For the L64020, OSD data is supplied either by the host or from an external device, such as a character generator.

P

P Picture
Predictive coded picture (from past reference I picture). See I Picture.

PAL
PAL refers to the TV standard in much of Europe except France (which uses SECAM).

PCM **Pulse Code Modulation**
The process of encoding an analog signal into digital format. The analog signal is sampled and the amplitude of each sample is converted to a set of binary digits or digital bits. The bits are then formed into a serial or parallel stream.

PES **Packetized Elementary Stream**
A Packetized Elementary Stream is an Elementary Stream such as an encoded audio or video stream arranged in packets containing a header with control information and a payload of data (audio, video, subpictures, etc.). Packetizing allows Elementary Streams to be combined by time division multiplexing to form complete program streams.

PQFP **Plastic Quad Flat Pack**

PS **Program Stream/Private Stream**
A Program Stream is a bitstream containing multiple streams related to a single audio/video programs. A Private Stream is a stream not further defined by MPEG-2 but accommodated in an MPEG-2 stream, such as an SPU stream.

PTS **Presentation Time Stamp**
Presentation times of presentation units extracted from the PES (Packetized Elementary Stream) headers. Used by the L64020 to display fields from the frame stores.

PXD **Pixel Data**
A pixel is a display position on a horizontal scan line of a TV picture. Pixel data is color to be displayed and is usually encoded into 8 bits.

R

RAM **Random Access Memory**
A memory that requires continuous power and that can have any location addressed randomly.

RMM **Reduced Memory Mode**
The L64020 normally uses three frame stores in SDRAM for reconstruction and display of video frames. I and P frames require two full frames stores. A third frame store is used exclusively for B pictures. When SDRAM space is limited because of the large space required for PAL images or because large OSD areas are being used, the B frame store space can be reduced. This means that the decoder overwrites the part of the B frame that has been displayed and does limit some trick mode features.

S

S/P DIF **Sony/Phillips Digital Interface**
A specification for forming encoded or unencoded audio into bursts and coding the bits to reduce the DC component of the bitstream, facilitate clock recovery, and make the interface insensitive to the polarity of the connection.

SAV **Start of Active Video**
The end of vertical blanking and the start of the active area of video. See CCIR656.

SCR **System Clock Reference**
In MPEG systems, a 90-kHz Systems Time Clock (STC) is used as the reference time base. The current STC time is included in MPEG bitstreams in SCR fields spaced no further apart than 700 ms. Audio and video PTSs (Presentation Time Stamps in the PES packets are compared with the SCR for display and audio synchronization.

SDRAM **Synchronous Dynamic Random Access Memory**
DRAM that requires a clock interface signal for data transfers.

SIF **Source Intermediate Format**

SPU **Subpicture Unit**
A Subpicture is typically a small picture inset in the main TV picture showing a different scene. An SPU is a group of bits in the bitstream with information and data to display one frame of a Subpicture. It includes a PTS (Presentation Time Stamp) which is used in the Display Control Sequence (DCSQ) analysis to synchronize the SPU with the main picture.

STM **Start Time**
STM is included in the SPU bitstream and also used in the DCSQ analysis. See SPU.

T

TQFP **Thin Quad Flat Package**

Trick Modes
Trick modes are those modes of display, such as slow play, pause, etc., that require skipping or repeating fields.

TTL **Transistor-Transistor Logic**
A digital signal interface convention that defines particular levels between 0 and ±5 Vdc that are the LOW-to-HIGH and HIGH-to-LOW switching points between logic 0 and logic 1.

V

VBV **Video Buffering Verifier**
An idealized model of a decoder defined by MPEG. It is used to further define parameters of a fixed-bit rate stream to be sent to a decoder, such as bit rate, picture rate, video buffer size, and picture delay in the buffer.

VCD **Video Compact Disk**

VCO **Voltage-Controlled Oscillator**
An oscillator whose output frequency is directly proportional to its input control voltage.

VLC **Variable-Length Coding**
A data compression technique in which variable-length bit patterns are assigned to source data values with the most common values receiving the shortest patterns.

Y

YCbCr
Color space used in MPEG. Y is luminance and CbCr are chrominance scaled from the UV components of the YUV system so that they are always positive and have about the same range.

YUV
Color space used in PAL. Y is luminance; U and V are the 1.3-MHz color difference (U = Y – R and V = Y – B) chrominance components.

Index

Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

Reader's Comments

Fax your comments to:

LSI Logic Corporation
Technical Publications
M/S E-198
Fax: 408.433.4333

Please tell us how you rate this document: *L64020 DVD Audio/Video Decoder Technical Manual.* Place a check mark in the appropriate blank for each category.

	Excellent	Good	Average	Fair	Poor
Completeness of information	____	____	____	____	____
Clarity of information	____	____	____	____	____
Ease of finding information	____	____	____	____	____
Technical content	____	____	____	____	____
Usefulness of examples and illustrations	____	____	____	____	____
Overall manual	____	____	____	____	____

What could we do to improve this document?

____

____

____

____

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

____

____

____

Please complete the information below so that we may contact you directly for clarification or additional information.

Name ______________________________ Date ____________

Telephone ____________________ Fax ____________________

Title ____

Department _________________________ Mail Stop __________

Company Name ____

Street ____

City, State, Zip ____

U.S. Distributors by State

H. H. Hamilton Hallmark
W. E. Wyle Electronics

Alabama
Huntsville
H. H. Tel: 205.837.8700
W. E. Tel: 800.964.9953

Alaska
H. H. Tel: 800.332.8638

Arizona
Phoenix
H. H. Tel: 602.736.7000
W. E. Tel: 800.528.4040
Tucson
H. H. Tel: 520.742.0515

Arkansas
H. H. Tel: 800.327.9989

California
Irvine
H. H. Tel: 714.789.4100
W. E. Tel: 800.626.9953
Los Angeles
H. H. Tel: 818.594.0404
W. E. Tel: 800.288.9953
Sacramento
H. H. Tel: 916.632.4500
W. E. Tel: 800.627.9953
San Diego
H. H. Tel: 619.571.7540
W. E. Tel: 800.829.9953
San Jose
H. H. Tel: 408.435.3500
Santa Clara
W. E. Tel: 800.866.9953
Woodland Hills
H. H. Tel: 818.594.0404

Colorado
Denver
H. H. Tel: 303.790.1662
W. E. Tel: 800.933.9953

Connecticut
Chesire
H. H. Tel: 203.271.5700
Wallingford
W. E. Tel: 800.605.9953

Delaware
North/South
H. H. Tel: 800.526.4812
Tel: 800.638.5988

Florida
Fort Lauderdale
H. H. Tel: 305.484.5482
W. E. Tel: 800.568.9953
Orlando
H. H. Tel: 407.657.3300
W. E. Tel: 407.740.7450
Tampa
W. E. Tel: 800.395.9953
St. Petersburg
H. H. Tel: 813.507.5000

Georgia
Atlanta
H. H. Tel: 770.623.4400
W. E. Tel: 800.876.9953

Hawaii
H. H. Tel: 800.851.2282

Idaho
H. H. Tel: 801.266.2022

Illinois
North/South
H. H. Tel: 847.797.7300
Tel: 314.291.5350
Chicago
W. E. Tel: 800.853.9953

Indiana
Indianapolis
H. H. Tel: 317.575.3500
W. E. Tel: 317.581.6152

Iowa
Cedar Rapids
H. H. Tel: 319.393.0033

Kansas
Kansas City
H. H. Tel: 913.663.7900

Kentucky
Central/Northern/ Western
H. H. Tel: 800.984.9503
Tel: 800.767.0329
Tel: 800.829.0146

Louisiana
North/South
H. H. Tel: 800.231.0253
Tel: 800.231.5575

Maine
H. H. Tel: 800.272.9255

Maryland
Baltimore
H. H. Tel: 410.720.3400
W. E. Tel: 800.863.9953

Massachusetts
Boston
H. H. Tel: 508.532.9808
W. E. Tel: 800.444.9953
Marlborough
W. E. Tel: 508.480.9900

Michigan
Detroit
H. H. Tel: 313.416.5800
W. E. Tel: 888.318.9953
Grandville
H. H. Tel: 616.531.0345

Minnesota
Minneapolis
H. H. Tel: 612.881.2600
W. E. Tel: 800.860.9953

Mississippi
H. H. Tel: 800.633.2918

Missouri
St. Louis
H. H. Tel: 314.291.5350

Montana
H. H. Tel: 800.526.1741

Nebraska
H. H. Tel: 800.332.4375

Nevada
Las Vegas
H. H. Tel: 800.528.8471
W. E. Tel: 702.765.7117

New Hampshire
H. H. Tel: 800.272.9255

New Jersey
North/South
H. H. Tel: 201.515.1641
Tel: 609.222.6400
Oradell
W. E. Tel: 201.261.3200
Pine Brook
W. E. Tel: 800.862.9953

New Mexico
Albuquerque
H. H. Tel: 505.293.5119

New York
Long Island
H. H. Tel: 516.434.7400
W. E. Tel: 800.861.9953
Rochester
H. H. Tel: 716.475.9130
W. E. Tel: 800.319.9953
Syracuse
H. H. Tel: 315.453.4000

North Carolina
Raleigh
H. H. Tel: 919.872.0712
W. E. Tel: 800.560.9953

North Dakota
H. H. Tel: 800.829.0116

Ohio
Cleveland
H. H. Tel: 216.498.1100
W. E. Tel: 800.763.9953
Dayton
H. H. Tel: 614.888.3313
W. E. Tel: 800.763.9953

Oklahoma
Tulsa
H. H. Tel: 918.459.6000

Oregon
Portland
H. H. Tel: 503.526.6200
W. E. Tel: 800.879.9953

Pennsylvania
Pittsburgh
H. H. Tel: 412.281.4150
Philadelphia
H. H. Tel: 800.526.4812
W. E. Tel: 800.871.9953

Rhode Island
H. H. 800.272.9255

South Carolina
H. H. Tel: 919.872.0712

South Dakota
H. H. Tel: 800.829.0116

Tennessee
East/West
H. H. Tel: 800.241.8182
Tel: 800.633.2918

Texas
Austin
H. H. Tel: 512.219.3700
W. E. Tel: 800.365.9953
Dallas
H. H. Tel: 214.553.4300
W. E. Tel: 800.955.9953
El Paso
H. H. Tel: 800.526.9238
Houston
H. H. Tel: 713.781.6100
W. E. Tel: 800.888.9953
Rio Grande Valley
H. H. Tel: 210.412.2047

Utah
Draper
W. E. Tel: 800.414.4144
Salt Lake City
H. H. Tel: 801.365.3800
W. E. Tel: 800.477.9953

Vermont
H. H. Tel: 800.272.9255

Virginia
H. H. Tel: 800.638.5988

Washington
Seattle
H. H. Tel: 206.882.7000
W. E. Tel: 800.248.9953

Wisconsin
Milwaukee
H. H. Tel: 414.513.1500
W. E. Tel: 800.867.9953

Wyoming
H. H. Tel: 800.332.9326

Sales Offices and Design Resource Centers

LSI Logic Corporation
Corporate Headquarters
Tel: 408.433.8000
Fax: 408.433.8989

NORTH AMERICA

California
Irvine
◆ Tel: 714.553.5600
Fax: 714.474.8101

San Diego
Tel: 619.613.8300
Fax: 619.613.8350

Silicon Valley
Sales Office
Tel: 408.433.8000
Fax: 408.954.3353
Design Center
◆ Tel: 408.433.8000
Fax: 408.433.7695

Colorado
Boulder
Tel: 303.447.3800
Fax: 303.541.0641

Florida
Boca Raton
Tel: 561.989.3236
Fax: 561.989.3237

Illinois
Schaumburg
◆ Tel: 847.995.1600
Fax: 847.995.1622

Kentucky
Bowling Green
Tel: 502.793.0010
Fax: 502.793.0040

Maryland
Bethesda
Tel: 301.897.5800
Fax: 301.897.8389

Massachusetts
Waltham
◆ Tel: 781.890.0180
Fax: 781.890.6158

Minnesota
Minneapolis
◆ Tel: 612.921.8300
Fax: 612.921.8399

New Jersey
Edison
◆ Tel: 732.549.4500
Fax: 732.549.4802

New York
New York
Tel: 716.223.8820
Fax: 716.223.8822

North Carolina
Raleigh
Tel: 919.785.4520
Fax: 919.783.8909

Oregon
Beaverton
Tel: 503.645.0589
Fax: 503.645.6612

Texas
Austin
Tel: 512.388.7294
Fax: 512.388.4171

Dallas
◆ Tel: 972.509.0350
Fax: 972.509.0349

Washington
Issaquah
Tel: 425.837.1733
Fax: 425.837.1734

Canada
Ontario
Ottawa
◆ Tel: 613.592.1263
Fax: 613.592.3253

Toronto
◆ Tel: 416.620.7400
Fax: 416.620.5005

Quebec
Montreal
◆ Tel: 514.694.2417
Fax: 514.694.2699

INTERNATIONAL

Australia
Reptechnic Pty Ltd
New South Wales
◆ Tel: 612.9953.9844
Fax: 612.9953.9683

Denmark
LSI Logic Development Centre
Ballerup
Tel: 45.44.86.55.55
Fax: 45.44.86.55.56

France
LSI Logic S.A.
Immeuble Europa
Paris
◆ Tel: 33.1.34.63.13.13
Fax: 33.1.34.63.13.19

Germany
LSI Logic GmbH
Munich
◆ Tel: 49.89.4.58.33.0
Fax: 49.89.4.58.33.108

Stuttgart
Tel: 49.711.13.96.90
Fax: 49.711.86.61.428

Hong Kong
AVT Industrial Ltd
Hong Kong
Tel: 852.2428.0008
Fax: 852.2401.2105

India
LogiCAD India Private Ltd
Bangalore
◆ Tel: 91.80.526.2500
Fax: 91.80.338.6591

Israel
LSI Logic
Ramat Hasharon
◆ Tel: 972.3.5.480480
Fax: 972.3.5.403747

VLSI Development Centre
Netanya
◆ Tel: 972.9.657190
Fax: 972.9.657194

Italy
LSI Logic S.P.A.
Milano
◆ Tel: 39.39.687371
Fax: 39.39.6057867

Japan
LSI Logic K.K.
Tokyo
◆ Tel: 81.3.5463.7821
Fax: 81.3.5463.7820

Osaka
◆ Tel: 81.6.947.5281
Fax: 81.6.947.5287

Korea
LSI Logic Corporation of Korea Ltd.
Seoul
◆ Tel: 82.2.528.3400
Fax: 82.2.528.2250

The Netherlands
LSI Logic Europe Ltd
Eindhoven
Tel: 31.40.265.3580
Fax: 31.40.296.2109

Singapore
LSI Logic Pte Ltd
Singapore
◆ Tel: 65.334.9061
Fax: 65.334.4749

Spain
LSI Logic S.A.
Madrid
◆ Tel: 34.1.556.07.09
Fax: 34.1.556.75.65

Sweden
LSI Logic AB
Stockholm
◆ Tel: 46.8.444.15.00
Fax: 46.8.750.66.47

Switzerland
LSI Logic Sulzer AG
Brugg/Biel
Tel: 41.32.536363
Fax: 41.32.536367

Taiwan
LSI Logic Asia-Pacific
Taipei
◆ Tel: 886.2.2718.7828
Fax: 886.2.2718.8869

Jeilin Technology Corporation, Ltd.
Taipei
Tel: 886.2.2248.4828
Fax: 886.2.2242.4397

Lumax International Corporation, Ltd
Taipei
Tel: 886.2.2788.3656
Fax: 886.2.2788.3568

United Kingdom
LSI Logic Europe Ltd
Bracknell
◆ Tel: 44.1344.426544
Fax: 44.1344.481039

◆ Sales Offices with Design Resource Centers

Notes

Notes