Ultralow Cost Video Codec

ADV601LC

FEATURES
100% Bitstream Compatible with the ADV601
Precise Compressed Bit Rate Control
Field Independent Compression
8-Bit Video Interface Supports CCIR-656 and Multiplexed Philips Formats
General Purpose 16- or 32-Bit Host Interface with 512 Deep 32-Bit FIFO

PERFORMANCE
Real-Time Compression or Decompression of CCIR-601 to Video:
720 × 288 @ 50 Fields/Sec — PAL
720 × 243 @ 60 Fields/Sec — NTSC
Compression Ratios from Visually Loss-Less to 350:1
Visually Loss-Less Compression At 4:1 on Natural Images (Typical)

APPLICATIONS
PC Video Editing
Remote CCTV Surveillance
Digital Camcorders
Digital Video Tape
Wireless Video Systems
TV Instant Replay

GENERAL DESCRIPTION

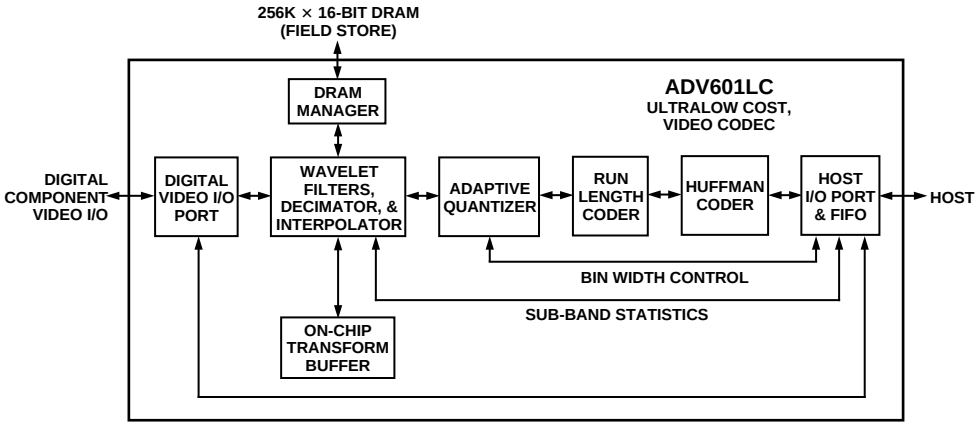
The ADV601LC is an ultralow cost, single chip, dedicated function, all digital CMOS VLSI device capable of supporting visually loss-less to 350:1 real-time compression and decompression of CCIR-601 digital video at very high image quality levels. The chip integrates glueless video and host interfaces with on-chip SRAM to permit low part count, system level implementations suitable for a broad range of applications. The ADV601LC is 100% bitstream compatible with the ADV601.

The ADV601LC is a video encoder/decoder optimized for real-time compression and decompression of interlaced digital video. All features of the ADV601LC are designed to yield high performance at a breakthrough systems-level cost. Additionally, the unique sub-band coding architecture of the ADV601LC offers you many application-specific advantages. A review of the General Theory of Operation and Applying the ADV601LC sections will help you get the most use out of the ADV601LC in any given application.

The ADV601LC accepts component digital video through the Video Interface and outputs a compressed bit stream though the Host Interface in Encode Mode. While in Decode Mode, the ADV601LC accepts a compressed bit stream through the Host Interface and outputs component digital video through the Video Interface. The host accesses all of the ADV601LC's control and status registers using the Host Interface. Figure 1 summarizes the basic function of the part.

(continued on page 2)

FUNCTIONAL BLOCK DIAGRAM



REV. 0

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: http://www.analog.com
Fax: 781/326-8703

TABLE OF CONTENTS

This data sheet gives an overview of the ADV601LC functionality and provides details on designing the part into a system. The text of the data sheet is written for an audience with a general knowledge of designing digital video systems. Where appropriate, additional sources of reference material are noted throughout the data sheet.

GENERAL DESCRIPTION (*Continued from page 1*)

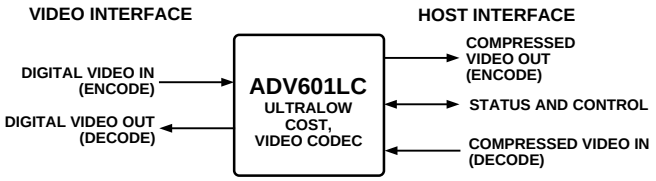


Figure 1. Functional Block Diagram

The ADV601LC adheres to international standard CCIR-601 for studio quality digital video. The codec also supports a range of field sizes and rates providing high performance in computer, PAL, NTSC, or still image environments. The ADV601LC is designed only for real-time interlaced video, full frames of video are formed and processed as two independent fields of data. The ADV601LC supports the field rates and sizes in Table I. Note that the maximum active field size is 768 by 288. The maximum pixel rate is 14.75 MHz.

The ADV601LC has a generic 16-/32-bit host interface, which includes a 512-position, 32-bit wide FIFO for compressed video. With additional external hardware, the ADV601LC's host interface is suitable (when interfaced to other devices) for moving compressed video over PCI, ISA, SCSI, SONET, 10 Base T, ARCnet, HDSL, ADSL, and a broad range of digital interfaces. For a full description of the Host Interface, see the Host Interface section.

The compressed data rate is determined by the input data rate and the selected compression ratio. The ADV601LC can achieve a near constant compressed bit rate by using the current field statistics in the off-chip bin width calculator on the external DSP or Host. The process of calculating bin widths on a DSP or Host can be "adaptive," optimizing the compressed bit rate in real time. This feature provides a near constant bit rate out of the host interface in spite of scene changes or other types of source material changes that would otherwise create bit rate burst conditions. For more information on the quantizer, see the Programmable Quantizer section.

The ADV601LC typically yields visually loss-less compression on natural images at a 4:1 compression ratio. Desired image quality levels can vary widely in different applications, so it is advisable to evaluate image quality of known source material at different compression ratios to find the best compression range for the application. The sub-band coding architecture of the ADV601LC provides a number of options to stretch compression performance. These options are outlined on in the Applying the ADV601LC section.

Table I. ADV601LC Field Rates and Sizes

Standard Name	Active Region Horizontal	Active Region Vertical[1]	Total Region Horizontal	Total Region Vertical	Field Rate (Hz)	Pixel Rate (MHz)[2]
CCIR-601/525	720	243	858	262.5	59.94	13.50
CCIR-601/625	720	288	864	312.5	50.00	13.50

NOTES
[1]The maximum active field size is 720 by 288.
[2]The maximum pixel rate is 13.5 MHz.

INTERNAL ARCHITECTURE

The ADV601LC is composed of eight blocks. Three of these blocks are interface blocks and five are processing blocks. The interface blocks are the Digital Video I/O Port, the Host I/O Port, and the external DRAM manager. The processing blocks are the Wavelet Kernel, the On-Chip Transform Buffer, the Programmable Quantizer, the Run Length Coder, and the Huffman Coder.

Digital Video I/O Port
Provides a real-time uncompressed video interface to support a broad range of component digital video formats, including "D1."

Host I/O Port and FIFO
Carries control, status, and compressed video to and from the host processor. A 512 position by 32-bit FIFO buffers the compressed video stream between the host and the Huffman Coder.

DRAM Manager
Performs all tasks related to writing, reading, and refreshing the external DRAM. The external host buffer DRAM is used for reordering and buffering quantizer input and output values.

Wavelet Kernel (Filters, Decimator, and Interpolator)
Gathers statistics on a per field basis and includes a block of filters, interpolators, and decimators. The kernel calculates forward and backward bi-orthogonal, two-dimensional, separable wavelet transforms on horizontal scanned video data. This block uses the internal transform buffer when performing wavelet transforms calculated on an entire image's data and so eliminates any need for extremely fast external memories in an ADV601LC-based design.

On-Chip Transform Buffer
Provides an internal set of SRAM for use by the wavelet transform kernel. Its function is to provide enough delay line storage to support calculation of separable two dimensional wavelet transforms for horizontally scanned images.

Programmable Quantizer
Quantizes wavelet coefficients. Quantize controls are calculated by the external DSP or host processor during encode operations and de-quantize controls are extracted from the compressed bit stream during decode. Each quantizer Bin Width is computed by the BW calculator software to maintain a constant compressed bit rate or constant quality bit rate. A Bin Width is a per block parameter the quantizer uses when determining the number of bits to allocate to each block (sub-band).

Run Length Coder
Performs run length coding on zero data and models nonzero data, encoding or decoding for more efficient Huffman coding. This data coding is optimized across the sub-bands and varies depending on the block being coded.

Huffman Coder
Performs Huffman coder and decoder functions on quantized run-length coded coefficient values. The Huffman coder/decoder uses three ROM-coded Huffman tables that provide excellent performance for wavelet transformed video.

GENERAL THEORY OF OPERATION

The ADV601LC processor's compression algorithm is based on the bi-orthogonal (7, 9) wavelet transform, and implements field independent sub-band coding. Sub-band coders transform two-dimensional spatial video data into spatial frequency filtered sub-bands. The quantization and entropy encoding processes provide the ADV601LC's data compression.

The wavelet theory, on which the ADV601LC is based, is a new mathematical apparatus first explicitly introduced by Morlet and Grossman in their works on geophysics during the mid 80s. This theory became very popular in theoretical physics and applied math. The late 80s and 90s have seen a dramatic growth in wavelet applications such as signal and image processing. For more on wavelet theory by Morlet and Grossman, see *Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape* (journal citation listed in References section).

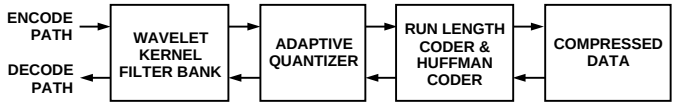


Figure 2. Encode and Decode Paths

References
For more information on the terms, techniques and underlying principles referred to in this data sheet, you may find the following reference texts useful. A reference text for general digital video principles is:

Jack, K., *Video Demystified: A Handbook for the Digital Engineer* (High Text Publications, 1993) ISBN 1-878707-09-4

Three reference texts for wavelet transform background information are:

Vetterli, M., Kovacevic, J., *Wavelets And Sub-band Coding* (Prentice Hall, 1995) ISBN 0-13-097080-8

Benedetto, J., Frazier, M., *Wavelets: Mathematics And Applications* (CRC Press, 1994) ISBN 0-8493-8271-8

Grossman, A., Morlet, J., *Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape*, Siam. J. Math. Anal., Vol. 15, No. 4, pp 723-736, 1984

THE WAVELET KERNEL

This block contains a set of filters and decimators that work on the image in both horizontal and vertical directions. Figure 6 illustrates the filter tree structure. The filters apply carefully chosen wavelet basis functions that better correlate to the broadband nature of images than the sinusoidal waves used in Discrete Cosine Transform (DCT) compression schemes (JPEG, MPEG, and H261).

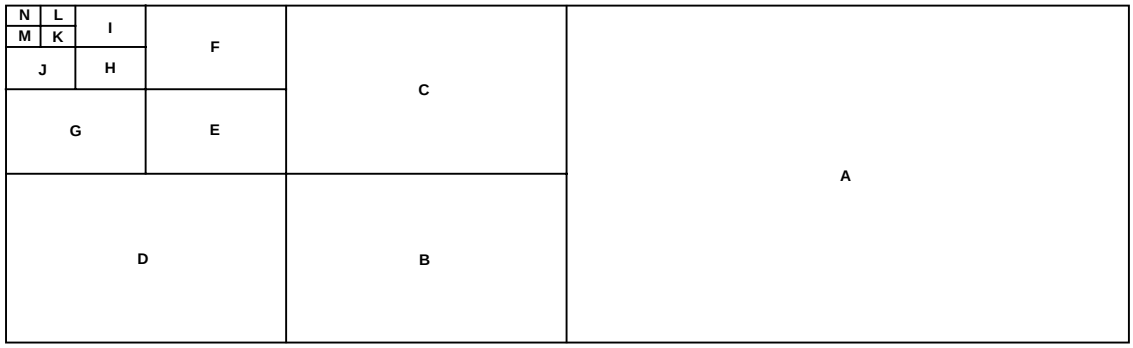
An advantage of wavelet-based compression is that the entire image can be filtered without being broken into sub-blocks as required in DCT compression schemes. This full image filtering eliminates the block artifacts seen in DCT compression and offers more graceful image degradation at high compression ratios. The availability of full image sub-band data also makes image processing, scaling, and a number of other system features possible with little or no computational overhead.

The resultant filtered image is made up of components of the original image as is shown in Figure 3 (a modified Mallat Tree). Note that Figure 3 shows how a component of video would be filtered, but in multiple component video luminance and color components are filtered separately. In Figure 4 and Figure 5 an actual image and the Mallat Tree (luminance only) equivalent is shown. It is important to note that while the image has been filtered or transformed into the frequency domain, no compression has occurred. With the image in its filtered state, it is now ready for processing in the second block, the quantizer.

Understanding the structure and function of the wavelet filters and resultant product is the key to obtaining the highest performance from the ADV601LC. Consider the following points:

- The data in all blocks (except N) for all components are high pass filtered. Therefore, the mean pixel value in those blocks is typically zero and a histogram of the pixel values in these blocks will contain a single “hump” (Laplacian distribution).
- The data in most blocks is more likely to contain zeros or strings of zeros than unfiltered image data.
- The human visual system is less sensitive to higher frequency blocks than low ones.
- Attenuation of the selected blocks in luminance or color components results in control over sharpness, brightness, contrast and saturation.
- High quality filtered/decimated images can be extracted/created without computational overhead.

Through leverage of these key points, the ADV601LC not only compresses video, but offers a host of application features. Please see the Applying the ADV601LC section for details on getting the most out of the ADV601LC’s sub-band coding architecture in different applications.



BLOCK A IS HIGH PASS IN X AND DECIMATED BY TWO.

BLOCK B IS HIGH PASS IN X, HIGH PASS IN Y, AND DECIMATED BY EIGHT.
BLOCK C IS HIGH PASS IN X, LOW PASS IN Y, AND DECIMATED BY EIGHT.
BLOCK D IS LOW PASS IN X, HIGH PASS IN Y, AND DECIMATED BY EIGHT.

BLOCK E IS HIGH PASS IN X, HIGH PASS IN Y, AND DECIMATED BY 32.
BLOCK F IS HIGH PASS IN X, LOW PASS IN Y, AND DECIMATED BY 32.
BLOCK G IS LOW PASS IN X, HIGH PASS IN Y, AND DECIMATED BY 32.

BLOCK H IS HIGH PASS IN X, HIGH PASS IN Y, AND DECIMATED BY 128.
BLOCK I IS HIGH PASS IN X, LOW PASS IN Y, AND DECIMATED BY 128.
BLOCK J IS LOW PASS IN X, HIGH PASS IN Y, AND DECIMATED BY 128.

BLOCK K IS HIGH PASS IN X, HIGH PASS IN Y, AND DECIMATED BY 512.
BLOCK L IS HIGH PASS IN X, LOW PASS IN Y, AND DECIMATED BY 512.
BLOCK M IS LOW PASS IN X, HIGH PASS IN Y, AND DECIMATED BY 512.
BLOCK N IS LOW PASS IN X, LOW PASS IN Y, AND DECIMATED BY 512.

Figure 3. Modified Mallat Diagram (Block Letters Correspond to Those in Filter Tree)

Figure 4. Unfiltered Original Image (Analog Devices Corporate Offices, Norwood, Massachusetts)

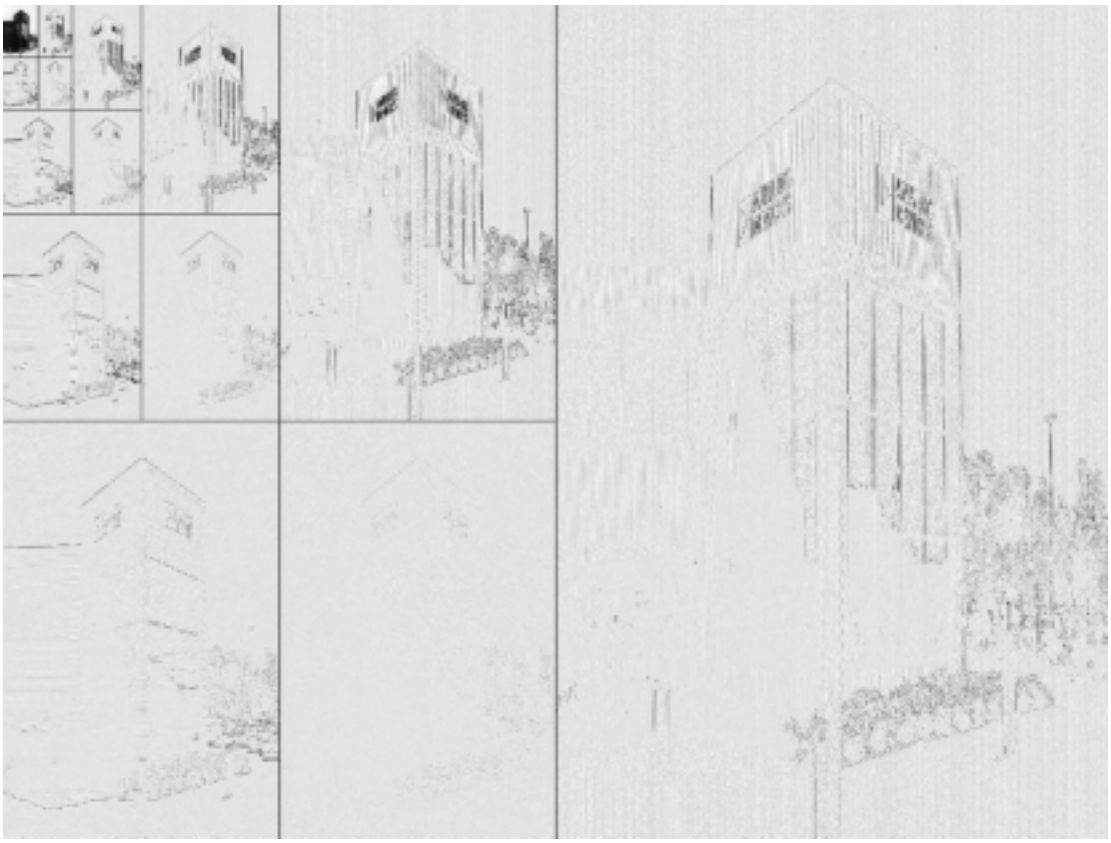


Figure 5. Modified Mallat Diagram of Image

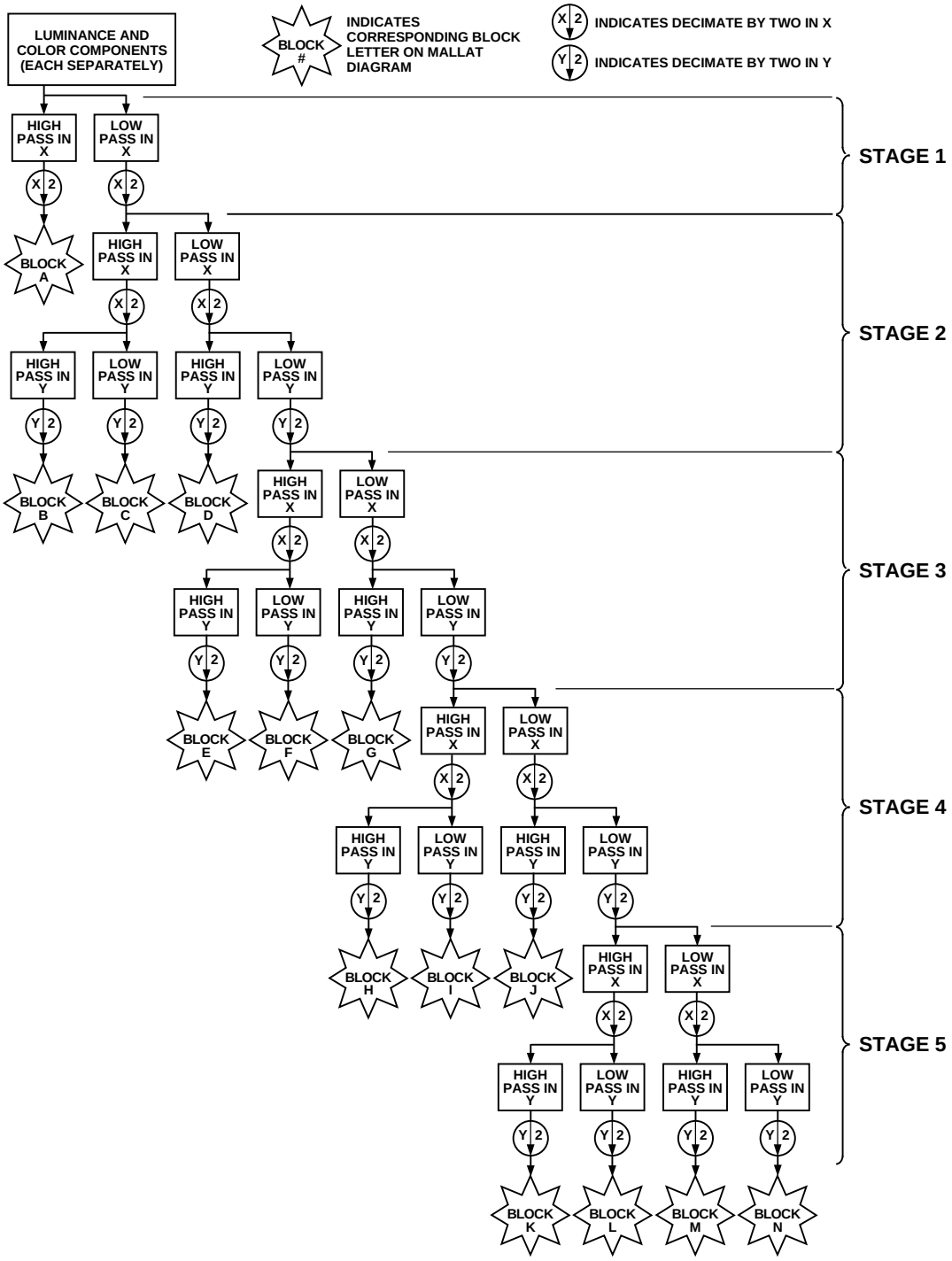


Figure 6. Wavelet Filter Tree Structure

THE PROGRAMMABLE QUANTIZER

This block quantizes the filtered image based on the response profile of the human visual system. In general, the human eye cannot resolve high frequencies in images to the same level of accuracy as lower frequencies. Through intelligent "quantization" of information contained within the filtered image, the ADV601LC achieves compression without compromising the visual quality of the image. Figure 7 shows the encode and decode data formats used by the quantizer.

Figure 8 shows how a typical quantization pattern applies over Mallat block data. The high frequency blocks receive much larger quantization (appear darker) than the low frequency blocks (appear lighter). Looking at this figure, one sees some key point concerning quantization: (1) quantization relates directly to frequency in Mallat block data and (2) levels of quantization range widely from high to low frequency block. (Note that the fill is based on a log formula.) The relation between actual ADV601LC bin width factors and the Mallat block fill pattern in Figure 8 appears in Table II.

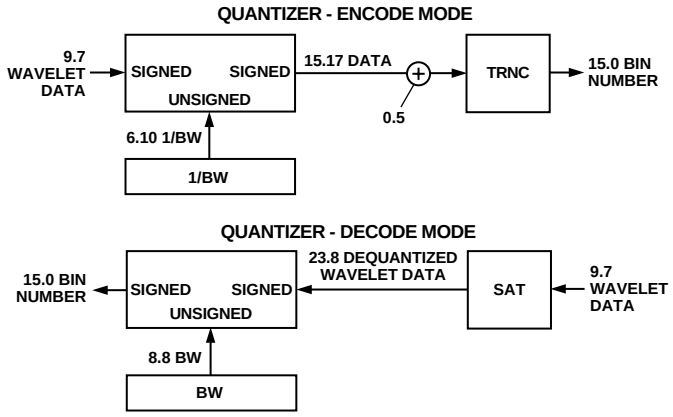


Figure 7. Programmable Quantizer Data Flow

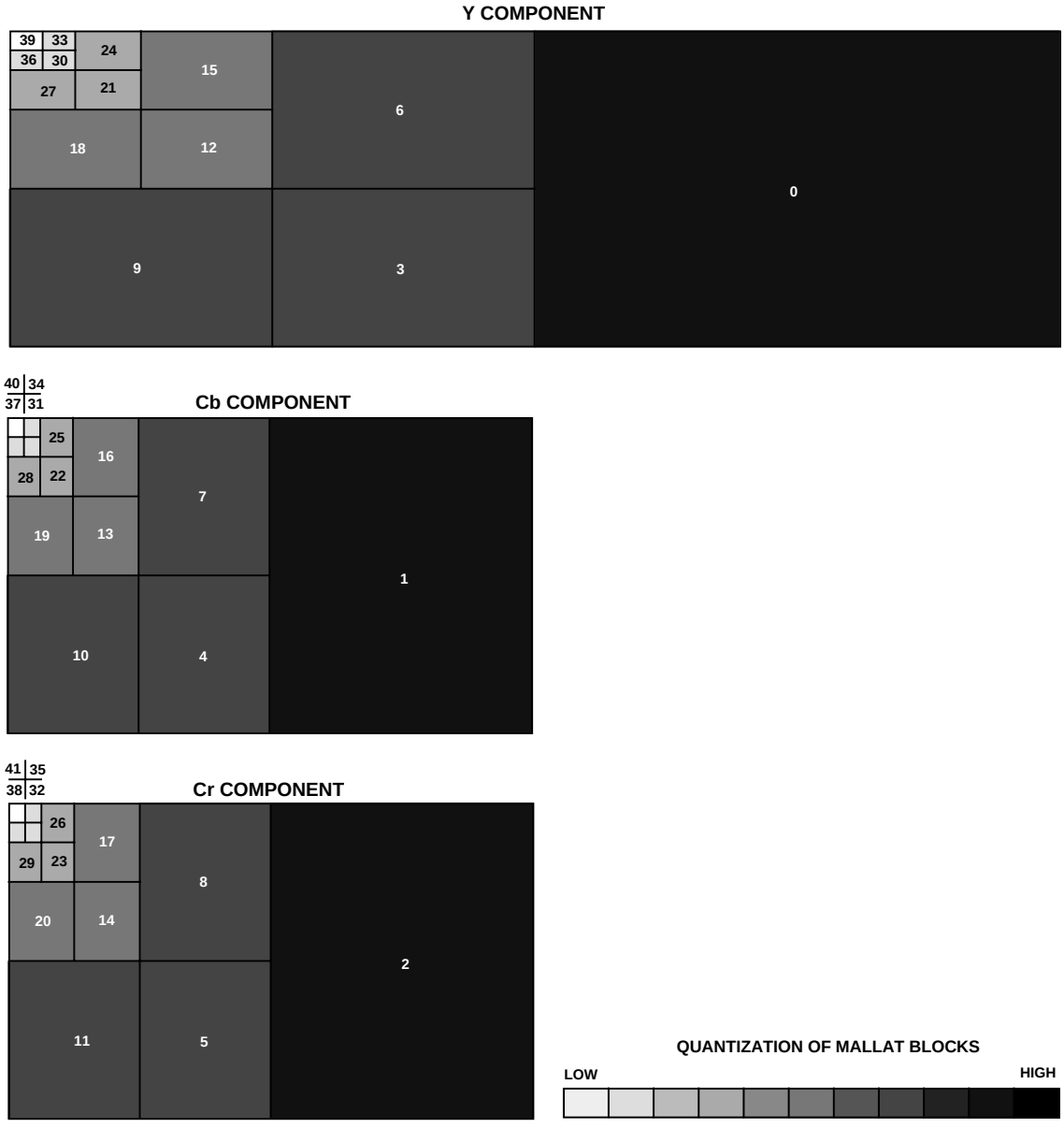


Figure 8. Typical Quantization of Mallat Data Blocks (Graphed)

Table II. ADV601LC Typical Quantization of Mallat Data Block Data[1]

Mallat Blocks	Bin Width Factors	Reciprocal Bin Width Factors
39	0x007F	0x0810
40	0x009A	0x06a6
41	0x009A	0x06a6
36	0x00BE	0x0564
33	0x00BE	0x0564
30	0x00E4	0x047e
34	0x00E6	0x0474
35	0x00E6	0x0474
37	0x00E6	0x0474
38	0x00E6	0x0474
31	0x0114	0x03b6
32	0x0114	0x03b6
27	0x0281	0x0199
24	0x0281	0x0199
21	0x0301	0x0155
25	0x0306	0x0153
26	0x0306	0x0153
28	0x0306	0x0153
29	0x0306	0x0153
22	0x03A1	0x011a
23	0x03A1	0x011a
5	0x0A16	0x0066
18	0x0A16	0x0066
12	0x0C1A	0x0055
20	0x0C2E	0x0054
19	0x0C2E	0x0054
17	0x0C2E	0x0054
16	0x0C2E	0x0054
14	0x0E9D	0x0046
13	0x0E9D	0x0046
6	0x1DDC	0x0022
9	0x1DDC	0x0022
3	0x23D5	0x001d
11	0x2410	0x001c
10	0x2410	0x001c
8	0x2410	0x001c
7	0x2410	0x001c
5	0x2B46	0x0018
4	0x2B46	0x0018
0	0xA417	0x0006
2	0xC62B	0x0005
1	0xC62B	0x0005

NOTE
[1]The Mallat block numbers, Bin Width factors, and Reciprocal Bin Width factors in Table II correspond to the shading percent fill) of Mallat blocks in Figure 8.

THE RUN LENGTH CODER AND HUFFMAN CODER

This block contains two types of entropy coders that achieve mathematically loss-less compression: run length and Huffman. The run-length coder looks for long strings of zeros and replaces it with short hand symbols. Table III illustrates an example of how compression is possible.

The Huffman coder is a digital compressor/decompressor that can be used for compressing any type of digital data. Essentially, an ideal Huffman coder creates a table of the most commonly occurring code sequences (typically zero and small values near zero) and then replaces those codes with some shorthand. The ADV601LC employs three fixed Huffman tables; it does not create tables.

The filters and the quantizer increase the number of zeros and strings of zeros, which improves the performance of the entropy coders. The higher the selected compression ratio, the more zeros and small value sequences the quantizer needs to generate. The transformed image in Figure 5 shows that the filter bank concentrates zeros and small values in the higher frequency blocks.

Encoding vs. Decoding

The decoding of compressed video follows the exact path as encoding but in reverse order. There is no need to calculate Bin Widths during decode because the Bin Width is stored in the compressed image during encode.

PROGRAMMER'S MODEL

A host device configures the ADV601LC using the Host I/O Port. The host reads from status registers and writes to control registers through the Host I/O Port.

Table IV. Register Description Conventions

Register Name	
Register Type (Indirect or Direct, Read or Write) and Address	
Register Functional Description Text	
Bit [#] or Bit Range [High:Low]	Bit or Bit Field Name and Usage Description
	0 Action or Indication When Bit Is Cleared (Equals 0)
	1 Action or Indication When Bit Is Set (Equals 1)

Table III. Uncompressed Versus Compressed Data Using Run-Length Coding

00000000000000000000000000000000000000000000000000000000000000000(uncompressed)
57 Zeros (Compressed)

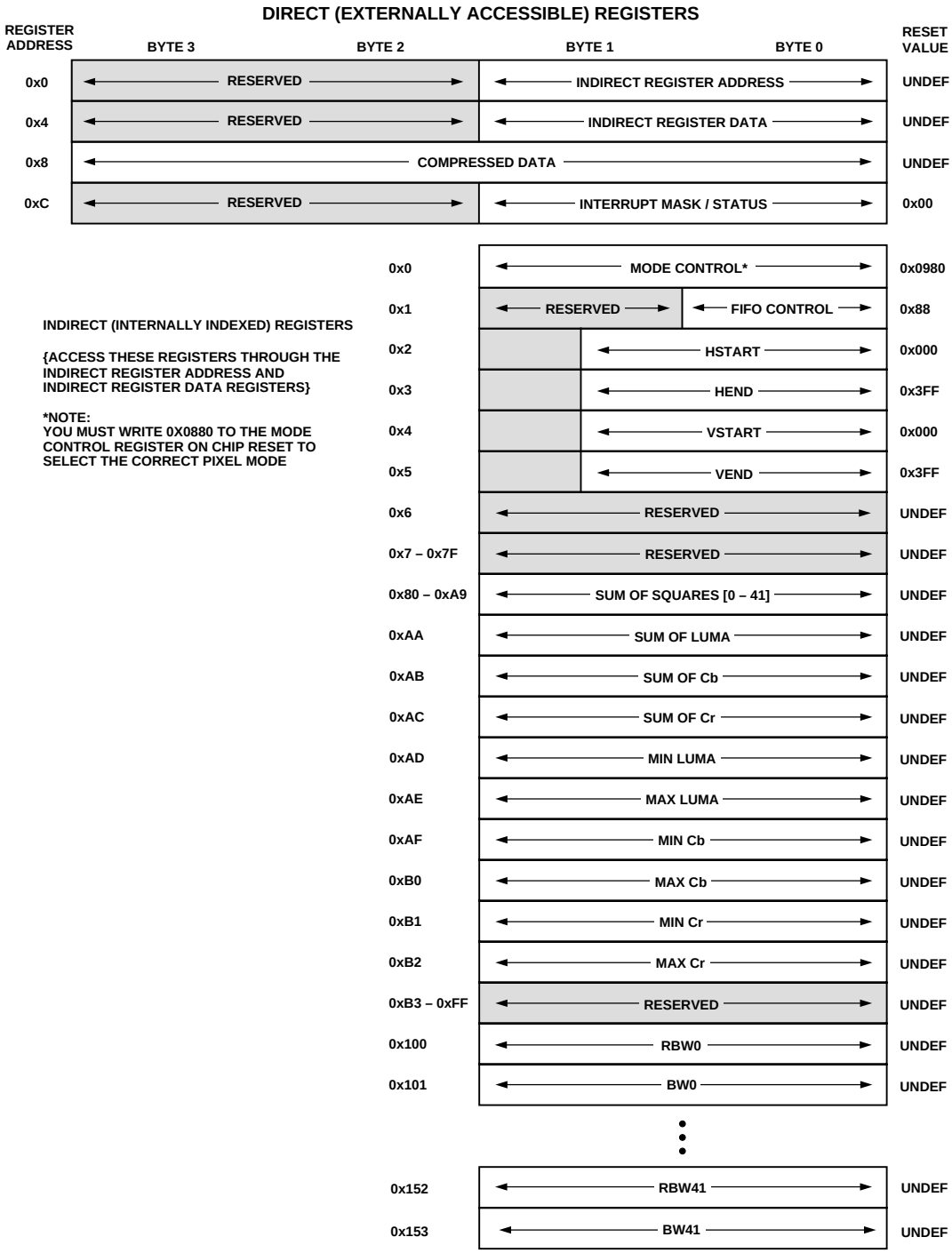


Figure 9. Map of ADV601LC Direct and Indirect Registers

ADV601LC REGISTER DESCRIPTIONS

Indirect Address Register

Direct (Write) Register Byte Offset 0x00.

This register holds a 16-bit value (index) that selects the indirect register accessible to the host through the indirect data register. All indirect write registers are 16 bits wide. The address in this register is auto-incremented on each subsequent access of the indirect data register. This capability enhances I/O performance during modes of operation where the host is calculating Bin Width controls.

[15:0] Indirect Address Register, **IAR[15:0]**. Holds a 16-bit value (index) that selects the indirect register to read or write through the indirect data register (undefined at reset)

[31:16] Reserved (undefined read/write zero)

Indirect Register Data

Direct (Read/Write) Register Byte Offset 0x04

This register holds a 16-bit value read or written from or to the indirect register indexed by the Indirect Address Register.

[15:0] Indirect Register Data, **IRD[15:0]**. A 16-bit value read or written to the indexed indirect register. Undefined at reset.

[31:16] Reserved (undefined read/write zero)

Compressed Data Register

Direct (Read/Write) Register Byte Offset 0x08

This register holds a 32-bit sequence from the compressed video bit stream. This register is buffered by a 512 position, 32-bit FIFO. For Word (16-bit) accesses, access Word0 (Byte 0 and Byte 1) then Word1 (Byte 2 and Byte 3) for correct auto-increment. For a description of the data sequence, see the Compressed Data Stream Definition section.

[31:0] Compressed Data Register, **CDR[31:0]**. 32-bit value containing compressed video stream data. At reset, contents undefined.

Interrupt Mask / Status Register

Direct (Read/Write) Register Byte Offset 0x0C

This 16-bit register contains interrupt mask and status bits that control the state of the ADV601LC's $\overline{\text{HIRQ}}$ pin. With the seven mask bits (IE_LCODE, IE_STATSR, IE_FIFOSTP, IE_FIFOSRQ, IE_FIFOERR, IE_CCIRER, IE_MERR); select the conditions that are ORed together to determine the output of the $\overline{\text{HIRQ}}$ pin.

Six of the status bits (LCODE, STATSR, FIFOSTP, MERR, FIFOERR, CCIRER) indicate active interrupt conditions and are sticky bits that stay set until read. Because sticky status bits are cleared when read, and these bits are set on the positive edge of the condition coming true, they cannot be read or tested for stable level true conditions multiple times.

The FIFOSRQ bit is not sticky. This bit can be polled to monitor for a FIFOSRQ true condition. Note: Enable this monitoring by using the FIFOSRQ bit and correctly programming DSL and ESL fields within the FIFO control registers.

[0] CCIR-656 Error in CCIR-656 data stream, **CCIRER**. This read only status bit indicates the following:

- 0 No CCIR-656 Error condition, *reset value*
- 1 Unrecoverable error in CCIR-656 data stream (missing sync codes)

[1] Statistics Ready, **STATSR**. This read only status bit indicates the following:

- 0 No Statistics Ready condition, *reset value* (STATS_R pin LO)
- 1 Statistics Ready for BW calculator (STATS_R pin HI)

[2] Last Code Read, **LCODE**. This read only status bit indicates the last compressed data word for field will be retrieved from the FIFO on the next read from the host bus.

- 0 No Last Code condition, *reset value* (LCODE pin LO)
- 1 Next read retrieves last word for field in FIFO (LCODE pin HI)

[3] FIFO Service Request, **FIFOSRQ**. This read only status bit indicates the following:

- 0 No FIFO Service Request condition, *reset value* (FIFO_SRQ pin LO)
- 1 FIFO is nearly full (encode) or nearly empty (decode) (FIFO_SRQ pin HI)

[4] FIFO Error, **FIFOERR**. This condition indicates that the host has been unable to keep up with the ADV601LC's compressed data supply or demand requirements. If this condition occurs during encode, the data stream will not be corrupted until MERR indicates that the DRAM is also overflowed. If this condition occurs during decode, the video output will be corrupted. If the system overflows the FIFO (disregarding a FIFOSTP condition) with too many writes in decode mode, FIFOERR is asserted. This read only status bit indicates the following:

0 No FIFO Error condition, *reset value* (FIFO_ERR pin LO)
1 FIFO overflow (encode) or underflow (decode) (FIFO_ERR pin HI)

[5] FIFO Stop, **FIFOSTP**. This condition indicates that the FIFO is full in decode mode and empty in encode mode. In decode mode only, FIFOSTP status actually behaves more conservatively than this. In decode mode, even when FIFOSTP is indicated, there are still 32 empty Dwords available in the FIFO and 32 more Dword writes can safely be performed. This status bit indicates the following:

0 No FIFO Stop condition, *reset value* (FIFO_STP pin LO)
1 FIFO empty (encode) or full (decode) (FIFO_STP pin HI)

[6] Memory Error, **MERR**. This condition indicates that an error has occurred at the DRAM memory interface. This condition can be caused by a defective DRAM, the inability of the Host to keep up with the ADV601LC compressed data stream, or bit errors in the data stream. Note that the ADV601LC recovers from this condition without host intervention.

0 No memory error condition, *reset value*
1 Memory error

[7] Reserved (always read/write zero)

[8] Interrupt Enable on CCIRER, **IE_CCIRER**. This mask bit selects the following:

0 Disable CCIR-656 data error interrupt, *reset value*
1 Enable interrupt on error in CCIR-656 data

[9] Interrupt Enable on STATR, **IE_STATR**. This mask bit selects the following:

0 Disable Statistics Ready interrupt, *reset value*
1 Enable interrupt on Statistics Ready

[10] Interrupt Enable on LCODE, **IE_LCODE**. This mask bit selects the following:

0 Disable Last Code Read interrupt, *reset value*
1 Enable interrupt on Last Code Read from FIFO

[11] Interrupt Enable on FIFOSRQ, **IE_FIFOSRQ**. This mask bit selects the following:

0 Disable FIFO Service Request interrupt, *reset value*
1 Enable interrupt on FIFO Service Request

[12] Interrupt Enable on FIFOERR, **IE_FIFOERR**. This mask bit selects the following:

0 Disable FIFO Stop interrupt, *reset value*
1 Enable interrupt on FIFO Stop

[13] Interrupt Enable on FIFOSTP, **IE_FIFOSTP**. This mask bit selects the following:

0 Disable FIFO Error interrupt, *reset value*
1 Enable interrupt on FIFO Error

[14] Interrupt Enable on MERR, **IE_MERR**. This mask bit selects the following:

0 Disable memory error interrupt, *reset value*
1 Enable interrupt on memory error

[15] Reserved (always read/write zero)

Mode Control Register

Indirect (Write Only) Register Index 0x00

This register holds configuration data for the ADV601LC's video interface format and controls several other video interface features. For more information on formats and modes, see the Video Interface section. Bits in this register have the following functions:

[3:0] Video Interface Format, **VIF[3:0]**. These bits select the interface format. Valid settings include the following (all other values are reserved):

0x0 CCIR-656, *reset value*
0x2 MLTPX (Philips)

[4] VCLK Output Divided by two, **VCLK2**. This bit controls the following:

0 Do not divide VCLK output (VCLKO = VCLK), *reset value*
1 Divide VCLK output by two (VCLKO = VCLK/2)

[5] Video Interface Master/Slave Mode Select, **M/S**. This bit selects the following:
- 0 Slave mode video interface (External control of video timing, HSYNC-VSYNC-FIELD are inputs), *reset value*
- 1 Master mode video interface (ADV601LC controls video timing, HSYNC-VSYNC are outputs)

[6] Video Interface 525/625 (NTSC/PAL) Mode Select, **P/N**. This bit selects the following:
- 0 525 mode video interface, *reset value*
- 1 625 mode video interface

[7] Video Interface Encode/Decode Mode Select, **E/D**. This bit selects the following:
- 0 Decode mode video interface (compressed-to-raw)
- 1 Encode mode video interface (raw-to-compressed), *reset value*

[8] Reserved (always write zero)

[9] Video Interface Bipolar/Unipolar Color Component Select, **BUC**. This bit selects the following:
- 0 Bipolar color component mode video interface, *reset value*
- 1 Unipolar color component mode video interface

[10] Reserved (always write zero)

[11] Video Interface Software Reset, **SWR**. This bit has the following effects on ADV601LC operations:
- 0 Normal operation
- 1 Software Reset. This bit is set on hardware reset and must be cleared before the ADV601LC can begin processing. (*reset value*)

  When this bit is set during encode, the ADV601LC completes processing the current field then suspends operation until the SWR bit is cleared. When this bit is set during decode, the ADV601LC suspends operation immediately and does not resume operation until the SWR bit is cleared. Note that this bit must be set whenever any other bit in the Mode register is changed.

[12] HSYNC pin Polarity, **PHSYNC**. This bit has the following effects on ADV601LC operations:
- 0 HSYNC is HI during blanking, *reset value*
- 1 HSYNC is LO during blanking (HI during active)

[13] $\overline{\text{HIRQ}}$ pin Polarity, **PHIRQ**. This bit has the following effects on ADV601LC operations:
- 0 $\overline{\text{HIRQ}}$ is active LO, *reset value*
- 1 $\overline{\text{HIRQ}}$ is active HI

[15:14] Reserved (always write zero)

FIFO Control Register

Indirect (Read/Write) Register Index 0x01

This register holds the service-request settings for the ADV601LC's host interface FIFO, causing interrupts for the "nearly full" and "nearly empty" levels. Because each register is four bits in size, and the FIFO is 512 positions, the 4-bit value must be multiplied by 32 (decimal) to determine the exact value for encode service level (nearly full) and decode service level (nearly empty). The ADV601LC uses these setting to determine when to generate a FIFO Service Request related host interrupt (FIFOSRQ bit and FIFO_SRQ pin).

[3:0] Encode Service Level, **ESL[3:0]**. The value in this field determines when the FIFO is considered nearly full on encode; a condition that generates a FIFO service request condition in encode mode. Since this register is four bits (16 states), and the FIFO is 512 positions, the step size for each bit in this register is 32 positions. The following table summarizes sample states of the register and their meaning.

ESL	Interrupt When . . .
0000	Disables service requests (FIFO_SRQ never goes HI during encode)
0001	FIFO has only 32 positions filled (FIFO_SRQ when >= 32 positions are filled)
1000	FIFO is 1/2 full, *reset value*
1111	FIFO has only 32 positions empty (480 positions filled)

[7:4] Decode Service Level, **DSL[7:4]**. The value in this field determines when the FIFO is considered nearly empty in decode; a condition that generates a FIFO service request in decode mode. Because this register is four bits (16 states), and the FIFO is 512 positions, the step size for each bit in this register is 32 positions. The following table summarizes sample states of the register and their meaning.

DSL	Interrupt When . . .
0000	Disables service requests (FIFO_SRQ never goes HI)
0001	FIFO has only 32 positions filled (480 positions empty)
1000	FIFO is 1/2 empty, *reset value*
1111	FIFO has only 32 positions empty (FIFO_SRQ when >= 32 positions are empty)

[15:8] Reserved (always write zero)

VIDEO AREA REGISTERS

The area defined by the HSTART, HEND, VSTART and VEND registers is the active area that the wavelet kernel processes. Video data outside the active video area is set to minimum luminance and zero chrominance (black) by the ADV601LC. These registers allow cropping of the input video during compression (encode only), but do not change the image size. Figure 10 shows how the video area registers work together.

Some comments on how these registers work are as follows:

- The vertical numbers include the blanking areas of the video.

  Specifically, a VSTART value of 21 will include the first line of active video, and the first pixel in a line corresponds to a value HSTART of 0 (for NTSC regular).

  Note that the vertical coordinates start with 1, whereas the horizontal coordinates start with 0.

- The default cropping mode is set for the entire frame. Specifically, Field 2 starts at a VSTART value of 283 (for NTSC regular).

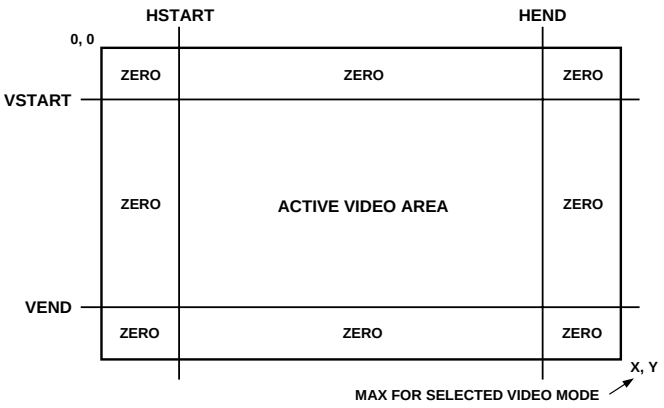


Figure 10. Video Area and Video Area Registers

HSTART Register

Indirect (Write Only) Register Index 0x02

This register holds the setting for the horizontal start of the ADV601LC's active video area. The value in this register is usually set to zero, but in cases where you wish to crop incoming video it is possible to do so by changing HST.

[9:0] Horizontal Start, **HST[9:0]**. 10-bit value defining the start of the active video region. (0 at reset)

[15:10] Reserved (always write zero)

HEND Register

Indirect (Write Only) Register Index 0x03

This register holds the setting for the horizontal end of the ADV601LC's active video area. If the value is larger than the max size of the selected video mode, the ADV601LC uses the max size of the selected mode for HEND.

[9:0] Horizontal End, **HEN[9:0]**.10-bit value defining the end of the active video region. (0x3FF at reset this value is larger than the max size of the largest video mode)

[15:10] Reserved (always write zero)

VSTART Register

Indirect (Write Only) Register Index 0x04

This register holds the setting for the vertical start of the ADV601LC's active video area. The value in this register is usually set to zero unless you want to crop the active video.

To vertically crop video while encoding, program the VSTART and VEND registers with actual video line numbers, which differ for each field. The VSTART and VEND contents must be updated on each field. Perform this updating as part of the field-by-field BW register update process. To perform this dynamic update correctly, the update software must keep track of which field is being processed next.

[9:0] Vertical Start, **VST[9:0]**. 10-bit value defining the starting line of the active video region, with line numbers from 1-to-625 in PAL and 1-to-525 in NTSC. (0 at reset)

[15:10] Reserved (always write zero)

VEND Register

Indirect (Write Only) Register Index 0x05

This register holds the setting for the vertical end of the ADV601LC's active video area. If the value is larger than the max size of the selected video mode, the ADV601LC uses the max size of the selected mode for VEND.

To vertically crop video while encoding, program the VSTART and VEND registers with actual video line numbers, which differ for each field. The VSTART and VEND contents must be updated on each field. Perform this updating as part of the field-by-field BW register update process. To perform this dynamic update correctly, the update software must keep track of which field is being processed next.

[9:0] Vertical End, **VEN[9:0]**. 10-bit value defining the ending line of the active video region, with line numbers from 1-to-625 in PAL and 1-to-525 in NTSC. (0x3FF at reset—this value is larger than the max size of the largest video mode)

[15:10] Reserved (always write zero)

Sum of Squares [0–41] Registers

Indirect (Read Only) Register Index 0x080 through 0x0A9

The Sum of Squares [0–41] registers hold values that correspond to the summation of values (squared) in corresponding Mallat blocks [0–41]. These registers let the Host or DSP read sum of squares statistics from the ADV601LC; using these values (with the Sum of Value, MIN Value, and MAX Value) the host or DSP can then calculate the BW and RBW values. The ADV601LC indicates that the sum of squares statistics have been updated by setting (1) the STATR bit and asserting the STAT_R pin. Read the statistics at any time. The Host reads these values through the Host Interface.

[15:0] Sum of Squares, **STS[15:0]**. 16-bit values [0-41] for corresponding Mallat blocks [0-41] (undefined at reset). Sum of Square values are 16-bit codes that represent the Most Significant Bits of values ranging from 40 bits for small blocks to 48 bits for large blocks. The 16-bit codes have the following precision:

Blocks	Precision	Sum of Squares Precision Description
0–2	48.–32	48.-bits wide, left shift code by 32-bits, and zero fill
3–11	46.–30	46.-bits wide, left shift code by 30-bits, and zero fill
12–20	44.–28	44.-bits wide, left shift code by 28-bits, and zero fill
21–29	42.–26	42.-bits wide, left shift code by 26-bits, and zero fill
30–41	40.–24	40.-bits wide, left shift code by 24-bits, and zero fill

If the Sum of Squares code were 0x0025 for block 10, the actual value would be 0x000940000000; if using that same code, 0x0025, for block 30, the actual value would be 0x0025000000.

[31:0] Reserved (always read zero)

Sum of Luma Value Register

Indirect (Read Only) Register Index 0x0AA

The Sum of Luma Value register lets the host or DSP read the sum of pixel values for the Luma component in block 39. The Host reads these values through the Host Interface.

[15:0] Sum of Luma, **SL[15:0]**. 16-bit component pixel values (undefined at reset)

[31:0] Reserved (always read zero)

Sum of Cb Value Register

Indirect (Read Only) Register Index 0x0AB

The Sum of Cb Value register lets the host or DSP read the sum of pixel values for the Cb component in block 40. The Host reads these values through the Host Interface.

[15:0] Sum of Cb, **SCB[15:0]**. 16-bit component pixel values (undefined at reset)

[31:0] Reserved (always read zero)

Sum of Cr Value Register

Indirect (Read Only) Register Index 0x0AC

The Sum of Cr Value register lets the host or DSP read the sum of pixel values for the Cr component in block 41. The Host reads these values through the Host Interface.

[15:0] Sum of Cr, **SCR[15:0]**. 16-bit component pixel values (undefined at reset)

[31:0] Reserved (always read zero)

MIN Luma Value Register

Indirect (Read Only) Register Index 0x0AD

The MIN Luma Value register lets the host or DSP read the minimum pixel value for the Luma component in the unprocessed data. The Host reads these values through the Host Interface.

[15:0] Minimum Luma, **MNL[15:0]**. 16-bit component pixel value (undefined at reset)

[31:0] Reserved (always read zero)

MAX Luma Value Register

Indirect (Read Only) Register Index 0x0AE

The MAX Luma Value register lets the host or DSP read the maximum pixel value for the Luma component in the unprocessed data. The Host reads these values through the Host Interface.

[15:0] Maximum Luma, **MXL[15:0]**. 16-bit component pixel value (undefined at reset)

[31:0] Reserved (always read zero)

MIN Cb Value Register
Indirect (Read Only) Register Index 0x0AF

The MIN Cb Value register lets the host or DSP read the minimum pixel value for the Cb component in the unprocessed data. The Host reads these values through the Host Interface.

[15:0] Minimum Cb, **MNCB[15:0]**, 16-bit component pixel value (undefined at reset)

[31:0] Reserved (always read zero)

MAX Cb Value Register
Indirect (Read Only) Register Index 0x0B0

The MAX Cb Value register lets the host or DSP read the maximum pixel value for the Cb component in the unprocessed data. The Host reads these values through the Host Interface.

[15:0] Maximum Cb, **MXCB[15:0]**.16-bit component pixel value (undefined at reset)

[31:0] Reserved (always read zero)

MIN Cr Value Register
Indirect (Read Only) Register Index 0x0B1

The MIN Cr Value register lets the host or DSP read the minimum pixel value for the Cr component in the unprocessed data. The Host reads these values through the Host Interface.

[15:0] Minimum Cr, **MNCR[15:0]**. 16-bit component pixel value (undefined at reset)

[31:0] Reserved (always read zero)

MAX Cr Value Register
Indirect (Read Only) Register Index 0x0B2

The MAX Cr Value register lets the host or DSP read the maximum pixel value for the Cr component in the unprocessed data. The Host reads these values through the Host Interface.

[15:0] Maximum Cr, **MXCR[15:0]**. 16-bit component pixel value (undefined at reset)

[31:0] Reserved (always read zero)

Bin Width and Reciprocal Bin Width Registers
Indirect (Read/Write) Register Index 0x0100-0x0153

The RBW and BW values are calculated by the host or DSP from data in the Sum of Squares [0-41], Sum of Value, MIN Value, and MAX Value registers; then are written to RBW and BW registers during encode mode to control the quantizer. The Host writes these values through the Host Interface.

These registers contain a 16-bit interleaved table of alternating RBW/BW (RBW-even addresses and BW-odd addresses) values as indexed on writes by address register. Bin Widths are 8.8, unsigned, 16-bit, fixed-point values. Reciprocal Bin Widths are 6.10, unsigned, 16-bit, fixed-point values. Operation of this register is controlled by the host driver or the DSP (84 total entries) (undefined at reset).

[15:0] Bin Width Values, **BW[15:0]**

[15:0] Reciprocal Bin Width Values, **RBW[15:0]**

PIN FUNCTION DESCRIPTIONS

Clock Pins

Name	Pins	I/O	Description
VCLK/XTAL	2	I	A single clock (VCLK) or crystal input (across VCLK and XTAL). An acceptable 50% duty cycle clock signal is 27 MHz (CCIR-601 NTSC/PAL). If using a clock crystal, use a parallel resonant, microprocessor grade clock crystal. If using a clock input, use a TTL level input, 50% duty cycle clock with 1 ns (or less) jitter (measured rising edge to rising edge). Slowly varying, low jitter clocks are acceptable; up to 5% frequency variation in 0.5 sec.
VCLKO	1	O	VCLK Output or VCLK Output divided by two. Select function using Mode Control register.

Video Interface Pins

Name	Pins	I/O	Description
VSYNC	1	I or O	Vertical Sync or Vertical Blank. This pin can be either an output (Master Mode) or an input (Slave Mode). The pin operates as follows: • Output (Master) HI during inactive lines of video and LO otherwise • Input (Slave) a HI on this input indicates inactive lines of video
HSYNC	1	I or O	Horizontal Sync or Horizontal Blank. This pin can be either an output (Master Mode) or an input (Slave Mode). The pin operates as follows: • Output (Master) HI during inactive portion of video line and LO otherwise • Input (Slave) a HI on this input indicates inactive portion of video line Note that the polarity of this signal is modified using the Mode Control register. For detailed timing information, see the Video Interface section.
FIELD	1	I or O	Field # or Frame Sync. This pin can be either an output (Master Mode) or an input (Slave Mode). The pin operates as follows: • Output (Master) HI during Field1 lines of video and LO otherwise • Input (Slave) a HI on this input indicates Field1 lines of video
ENC	1	O	Encode or Decode. This output pin indicates the coding mode of the ADV601LC and operates as follows: • LO Decode Mode (Video Interface is output) • HI Encode Mode (Video Interface is input) Note that this pin can be used to control bus enable pins for devices connected to the ADV601LC Video Interface.
VDATA[7:0]	8	I/O	4:2:2 Video Data (8-bit digital component video data). These pins are inputs during encode mode and outputs during decode mode. When outputs (decode) these pins are compatible with 50 pF loads (rather than 30 pF as all other busses) to meet the high performance and large number of typical loads on this bus. The performance of these pins varies with the Video Interface Mode set in the Mode Control register, see the Video Interface section of this data sheet for pin assignments in each mode. Note that the Mode Control register also sets whether the color component is treated as either signed or unsigned.

DRAM Interface Pins

Name	Pins	I/O	Description
DDAT[15:0]	16	I/O	DRAM Data Bus. The ADV601LC uses these pins for 16-bit data read/write operations to the external 256K × 16-bit DRAM. (The operation of the DRAM interface is fully automatic and controlled by internal functionality of the ADV601LC.) These pins are compatible with 30 pF loads.
DADR[8:0]	9	O	DRAM Address Bus. The ADV601LC uses these pins to form the multiplexed row/column address lines to the external DRAM. (The operation of the DRAM interface is fully automatic and controlled by internal functionality of the ADV601LC.) These pins are compatible with 30 pF loads.
$\overline{RAS}$	1	O	DRAM Row Address Strobe. This pin is compatible with 30 pF loads.
$\overline{CAS}$	1	O	DRAM Column Address Strobe. This pin is compatible with 30 pF loads.
$\overline{WE}$	1	O	DRAM Write Enable. This pin is compatible with 30 pF loads. Note that the ADV601LC does not have a DRAM $\overline{OE}$ pin. Tie the DRAM's $\overline{OE}$ pin to ground.

Host Interface Pins

Name	Pins	I/O	Description
DATA[31:0]	32	I/O	Host Data Bus. These pins make up a 32-bit wide host data bus. The host controls this asynchronous bus with the $\overline{WR}$, $\overline{RD}$, $\overline{BE}$, and $\overline{CS}$ pins to communicate with the ADV601LC. These pins are compatible with 30 pF loads.
ADR[1:0]	2	I	Host DWord Address Bus. These two address pins let you address the ADV601LC's four directly addressable host interface registers. For an illustration of how this addressing works, see the Control and Write Register Map figure and Status and Read Register Map figure. The ADR bits permit register addressing as follows: ADR1 ADR0 DWord Address Byte Address 0 0 0 0x00 0 1 1 0x04 1 0 2 0x08 1 1 3 0x0C
$\overline{BE0}$–$\overline{BE3}$	2	I	Host Word Enable pins. These two input pins select the words that the ADV601LC's direct and indirect registers access through the Host Interface; $\overline{BE0}$–$\overline{BE1}$ access the least significant word, and $\overline{BE2}$–$\overline{BE3}$ access the most significant word. For a 32-bit interface only, tie these pins to ground, making all words available. Some important notes for 16-bit interfaces are as follows: • When using these byte enable pins, the byte order is always the lowest byte to the higher bytes. • The ADV601LC advances to the next 32-bit compressed data FIFO location after the $\overline{BE2}$–$\overline{BE3}$ pin is asserted then de-asserted (when accessing the Compressed Data register); so the FIFO location only advances when and if the host reads or writes the MSW of a FIFO location. • The ADV601LC advances to the next 16-bit indirect register after the $\overline{BE0}$–$\overline{BE1}$ pin is asserted then de-asserted; so the register selection only advances when and if the host reads or writes the MSW of a 16-bit indirect register.
$\overline{CS}$	1	I	Host Chip Select. This pin operates as follows: • LO Qualifies Host Interface control signals • HI Three-states DATA[31:0] pins
$\overline{WR}$	1	I	Host Write. Host register writes occur on the rising edge of this signal.
$\overline{RD}$	1	I	Host Read. Host register reads occur on the low true level of this signal.

Host Interface Pins (*Continued*)

Name	Pins	I/O	Description
$\overline{\text{ACK}}$	1	O	Host Acknowledge. The ADV601LC acknowledges completion of a Host Interface access by asserting this pin. Most Host Interface accesses (other than the compressed data register access) result in $\overline{\text{ACK}}$ being held high for at least one wait cycle, but some exceptions to that rule are as follows: • A full FIFO during decode operations causes the ADV601LC to de-assert (drive HI) the $\overline{\text{ACK}}$ pin, holding off further writes of compressed data until the FIFO has one available location. • An empty FIFO during encode operations causes the ADV601LC to de-assert (drive HI) the $\overline{\text{ACK}}$ pin, holding off further reads until one location is filled.
FIFO_SRQ	1	O	FIFO Service Request. This pin is an active high signal indicating that the FIFO needs to be serviced by the host. (see FIFO Control register). The state of this pin also appears in the Interrupt Mask/Status register. Use the interrupt mask to assert a Host interrupt ($\overline{\text{HIRQ}}$ pin) based on the state of the FIFO_SRQ pin. This pin operates as follows: • LO No FIFO Service Request condition (FIFOSRQ bit LO) • HI FIFO needs service is nearly full (encode) or nearly empty (decode) During encode, FIFO_SRQ is LO when the SWR bit is cleared (0) and goes HI when the FIFO is nearly full (see FIFO Control register). During decode, FIFO_SRQ is HI when the SWR bit is cleared (0), because FIFO is empty, and goes LO when the FIFO is filled beyond the nearly empty condition (see FIFO Control register).
STATS_R	1	O	Statistics Ready. This pin indicates the Wavelet Statistics (contents of Sum of Squares, Sum of Value, MIN Value, MAX Value registers) have been updated and are ready for the Bin Width calculator to read them from the host interface. The frequency of this interrupt will be equal to the field rate. The state of this pin also appears in the Interrupt Mask/Status register. Use the interrupt mask to assert a Host interrupt ($\overline{\text{HIRQ}}$ pin) based on the state of the STATS_R pin. This pin operates as follows: • LO No Statistics Ready condition (STATSR bit LO) • HI Statistics Ready for BW calculator (STATSR bit HI)
LCODE	1	O	Last Compressed Data (for field). This bit indicates the last compressed data word for field will be retrieved from the FIFO on the next read from the host bus. The frequency of this interrupt is similar to the field rate, but varies depending on compression and host response. The state of this pin also appears in the Interrupt Mask/Status register. Use the interrupt mask to assert a Host interrupt ($\overline{\text{HIRQ}}$ pin) based on the state of the LCODE pin. This pin operates as follows: • LO No Last Code condition (LCODE bit LO) • HI Last data word for field has been read from FIFO (LCODE bit HI)
$\overline{\text{HIRQ}}$	1	O	Host Interrupt Request. This pin indicates an interrupt request to the Host. The Interrupt Mask/Status register can select conditions for this interrupt based on any or all of the following: FIFOSTP, FIFOSRQ, FIFOERR, LCODE, STATR or CCIR-656 unrecoverable error. Note that the polarity of the $\overline{\text{HIRQ}}$ pin can be modified using the Mode Control register.
$\overline{\text{RESET}}$	1	I	ADV601LC Chip Reset. Asserting this pin returns all registers to reset state. Note that the ADV601LC must be reset at least once after power-up with this active low signal input. For more information on reset, see the SWR bit description.

Power Supply Pins

Name	Pins	I/O	Description
GND	16	I	Ground
VDD	13	I	+5 V dc Digital Power

Video Interface

The ADV601LC video interface supports two types of component digital video (D1) interfaces in both compression (input) and decompression (output) modes. These digital video interfaces include support for the Multiplexed Philips 4:2:2 and CCIR-656/SMPTE125M—international standard.

Video interface master and slave modes allow for the generation or receiving of synchronization and blanking signals. Definitions for the different formats can be found later in this section. For recommended connections to popular video decoders and encoders, see the Connecting The ADV601LC To Popular Video Decoders and Encoders section. A complete list of supported video interfaces and sampling rates is included in Table V.

Table V. Component Digital Video Interfaces

Name	Bits/ Component	Color Space	Sampling	Nominal Date Rate (MHz)	I/F Width
CCIR-656	8	YCrCb	4:2:2	27	8
Multiplex Philips	8	YUV	4:2:2	27	8

Internally, the video interface translates all video formats to one consistent format to be passed to the wavelet kernel. This consistent internal video standard is 4:2:2 at 16 bits accuracy.

VITC and Closed Captioning Support

The video interface also supports the direct loss-less extraction of 90-bit VITC codes during encode and the insertion of VITC codes during decode. Closed Captioning data (found on active Video Line 21) is handled just as normal active video on an active scan line. As a result, no special dedicated support is necessary for Closed Captioning. The data rates for Closed Captioning data are low enough to ensure robust operation of this mechanism at compression ratios of 50:1 and higher. Note that you must include Video Line 21 in the ADV601LC's defined active video area for Closed Caption support.

27 MHz Nominal Sampling

There is one clock input (VCLK) to support all internal processing elements. This is a 50% duty cycle signal and must be synchronous to the video data. Internally this clock is doubled using a phase locked loop to provide for a 54 MHz internal processing clock. The clock interface is a two pin interface that allows a crystal oscillator to be tied across the pins or a clock oscillator to drive one pin. The nominal clock rate for the video interface is 27 MHz. Note that the ADV601LC also supports a pixel rate of 13.5 MHz.

Video Interface and Modes

In all, there are seven programmable features that configure the video interface. These are:

- *Encode-Decode Control*
  In addition to determining what functions the internal processing elements must perform, this control determines the direction of the video interface. In decode mode, the video interface outputs data. In encode mode, the interface receives data. The state of the control is reflected on the ENC pin. This pin can be used as an enable input by external line drivers. This control is maintained by the host processor.
- *Master-Slave Control*
  This control determines whether the ADV601LC generates or receives the VSYNC, HSYNC, and FIELD signals. In master mode, the ADV601LC generates these signals for external hardware synchronization. In slave mode, the ADV601LC receives these signals. Note that some video formats require the ADV601LC to operate in slave mode only. This control is maintained by the host processor.
- *525-625 (NTSC-PAL) Control*
  This control determines whether the ADV601LC is operating on 525/NTSC video or 625/PAL video. This information is used when the ADV601LC is in master and decode modes so that the ADV601LC knows where and when to generate the HSYNC, VSYNC, and FIELD Pulses as well as when to insert the SAV and EAV time codes (for CCIR-656 only) in the data stream. This control is maintained by the host processor. Table VI shows how the 525-625 Control in the Mode Control register works.

Table VI. Square Pixel Control, 525-625 Control, and Video Formats

525-625 Control	Max Horizontal Size	Max Field Size	NTSC-PAL
0	720	243	CCIR-601 NTSC
1	720	288	CCIR-601 PAL

- *Bipolar/Unipolar Color Component*
  This mode determines whether offsets are used on color components. In Philips mode, this control is usually set to Bipolar, since the color components are normal twos-compliment signed values. In CCIR-656 mode, this control is set to Unipolar, since the color components are offset by 128. Note that it is likely the ADV601LC will function if this control is in the wrong state, but compression performance will be degraded. It is important to set this bit correctly.
- *Active Area Control*
  Four registers HSTART (horizontal start), HEND (horizontal end), VSTART (vertical start) and VEND (vertical end) determine the active video area. The maximum active video area is 720 by 288 pixels for a single field.
- *Video Format*
  This control determines the video format that is supported. In general, the goal of the various video formats is to support glueless interfaces to the wide variety of video formats peripheral components expect. This control is maintained by the host processor. Table VII shows a synopsis of the supported video formats. Definitions of each format can be found later in this section. For Video Interface pins descriptions, see the Pin Function Descriptions.

Clocks and Strobes

All video data is synchronous to the video clock (VCLK). The rising edge of VCLK is used to clock all data into the ADV601LC.

Synchronization and Blanking Pins

Three signals, which can be configured as inputs or outputs, are used for video frame and field horizontal synchronization and blanking. These signals are VSYNC, HSYNC, and FIELD.

VDATA Pins Functions With Differing Video Interface Formats

The functionality of the Video Interface pins depends on the current video format. Table VIII defines how Video data pins are used for the various formats.

Table VIII. VDATA[7:0] Pin Functions Under CCIR-656 and Multiplex Philips

VDATA[7:0] Pins	CCIR-656	Multiplex Philips
7	Data9	Data9
6	Data8	Data8
5	Data7	Data7
4	Data6	Data6
3	Data5	Data5
2	Data4	Data4
1	Data3	Data3
0	Data2	Data2

Table VII. Component Digital Video Formats

Name	Bit/ Component	Color Space	Sampling	Nominal Data Rate (MHz)	Master/ Slave	I/F Width	Format Number
CCIR-656	8	YCrCb	4:2:2	27	Master	8	0x0
Multiplex Philips	8	YUV	4:2:2	<=29.5	Either	8	0x2

Video Formats—CCIR-656

The ADV601LC supports a glueless video interface to CCIR-656 devices when the Video Format is programmed to CCIR-656 mode. CCIR-656 requires that 4:2:2 data (8 bits per component) be multiplexed and transmitted over a single 8-bit physical interface. A 27 MHz clock is transmitted along with the data. This clock is synchronous with the data. The color space of CCIR-656 is YCrCb.

When in master mode, the CCIR-656 mode does not require any external synchronization or blanking signals to accompany digital video. Instead, CCIR-656 includes special time codes in the stream syntax that define horizontal blanking periods, vertical blanking periods, and field synchronization (horizontal and vertical synchronization information can be derived). These time codes are called End-of-Active-Video (EAV) and Start-of-Active-Video (SAV). Each line of video has one EAV and one SAV time code. EAV and SAV have three bits of embedded information to define HSYNC, VSYNC and Field information as well as error detection and correction bits.

VCLK is driven with a 27 MHz, 50% duty cycle clock which is synchronous with the video data. Video data is clocked on the rising edge of the VCLK signal. When decoding, the VCLK signal is typically transmitted along with video data in the CCIR-656 physical interface.

Electrically, CCIR-656 specifies differential ECL levels to be used for all interfaces. The ADV601LC, however, only supports unipolar, TTL logic thresholds. Systems designs that interface to strictly conforming CCIR-656 devices (especially when interfacing over long cable distances) must include ECL level shifters and line drivers.

The functionality of HSYNC, VSYNC and FIELD Pins is dependent on three programmable modes of the ADV601LC: Master-Slave Control, Encode-Decode Control and 525-625 Control. Table IX summarizes the functionality of these pins in various modes.

Table IX. CCIR-656 Master and Slave Modes HSYNC, VSYNC, and FIELD Functionality

HSYNC, VSYNC and FIELD Functionality for CCIR-656	Master Mode (HSYNC, VSYNC and FIELD Are Outputs)	Slave Mode (HSYNC, VSYNC and FIELD Are Inputs)
Encode Mode (video data is input to the chip)	Pins are driven to reflect the states of the received time codes: EAV and SAV. This functionality is independent of the state of the 525-625 mode control. An encoder is most likely to be in master mode.	Undefined—Use Master Mode
Decode Mode (video data is output from the chip)	Pins are output to the precise timing definitions for CCIR-656 interfaces. The state of the pins reflect the state of the EAV and SAV timing codes that are generated in the output video data. These definitions are different for 525 and 625 line systems. The ADV601LC completely manages the generation and timing of these pins.	Undefined—Use Master Mode

Video Formats — Multiplexed Philips Video

The ADV601LC supports a hybrid mode of operation that is a cross between standard dual lane Philips and single lane CCIR-656. In this mode, video data is multiplexed in the same fashion in CCIR-656, but the values 0 and 255 are not reserved as signaling values. Instead, external HSYNC and VSYNC pins are used for signaling and video synchronization. VCLK may range up to 29.5 MHz.

VCLK is driven with up to a 29.5 MHz 50% duty cycle clock synchronous with the video data. Video data is clocked on the rising edge of the VCLK signal. The functionality of HSYNC, VSYNC, and FIELD pins is dependent on three programmable modes of the ADV601LC: Master-Slave Control, Encode-Decode Control, and 525-625 Control. Table X summarizes the functionality of these pins in various modes.

Table X. Philips Multiplexed Video Master and Slave Modes HSYNC, VSYNC, and FIELD Functionality

HSYNC, VSYNC and FIELD Functionality for Multiplexed Philips	Master Mode (HSYNC, VSYNC and FIELD Are Outputs)	Slave Mode (HSYNC, VSYNC and FIELD Are Inputs)
Encode Mode (video data is input to the chip)	The ADV601LC completely manages the generation and timing of these pins. The device driving the ADV601LC video interface must use these outputs to remain in sync with the ADV601LC. It is expected that this combination of modes would not be used frequently.	These pins are used to control the blanking of video and sequencing.
Decode Mode (video data is output from the chip)	The ADV601LC completely manages the generation and timing of these pins.	These pins are used to control the blanking of video and sequencing.

Video Formats—References

For more information on video interface standards, see the following reference texts.

- For the definition of CCIR-601:
  1992 – *CCIR Recommendations RBT series Broadcasting Service (Television) Rec. 601-3 Encoding Parameters of digital television for studios*, page 35, September 15, 1992.
- For the definition of CCIR-656:
  1992 – *CCIR Recommendations RBT series Broadcasting Service (Television) Rec. 656-1 Interfaces for digital component video signals in 525 and 626 line television systems operating at the 4:2:2 level of Rec. 601*, page 46, September 15, 1992.

Host Interface

The ADV601LC host interface is a high performance interface that passes all command and real-time compressed video data between the host and codec. A 512 position by 32-bit wide, bidirectional FIFO buffer passes compressed video data to and from the host. The host interface is capable of burst transfer rates of up to 132 million bytes per second (4 × 33 MHz). For host interface pins descriptions, see the Pin Function Descriptions section. For host interface timing information, see the Host Interface Timing section.

DRAM Manager

The DRAM Manager provides a sorting and reordering function on the sub-band coded data between the Wavelet Kernel and the Programmable Quantizer. The DRAM manager provides a pipeline delay stage to the ADV601LC. This pipeline lets the ADV601LC extract current field image statistics (min/max pixel values, sum of pixel values, and sum of squares) used in the calculation of Bin Widths and re-order wavelet transform data. The use of current field statistics in the Bin Width calculation results in precise control over the compressed bit rate. The DRAM manager manages the entire operation and refresh of the DRAM.

The interface between the ADV601LC DRAM manager and DRAM is designed to be transparent to the user. The ADV601LC DRAM pins should be connected to the DRAM as called out in the Pin Function Descriptions section. The ADV601LC requires one 256K word by 16-bit, 60 ns DRAM. The following is a selected list of manufacturers and part numbers. All parts can be used with the ADV601LC at all VCLK rates except where noted. Any DRAM used with the ADV601LC must meet the minimum specifications outlined for the Hyper Mode DRAMs listed in Table XI. For DRAM Interface pins descriptions, see the Pin Function Descriptions.

Table XI. ADV601LC Compatible DRAMs

Manufacturer	Part Number	Notes
Toshiba	TC514265DJ/DZ/DFT-60	None
NEC	µPD424210ALE-60	None
NEC	µPD42S4210ALE-60	CBR Self Refresh feature of this product is not needed by the ADV601LC.
Hitachi	HM514265CJ-60	None

Compressed Data-Stream Definition

Through its Host Interface the ADV601LC outputs (during encode) and receives (during decode) compressed digital video data. This stream of data passing between the ADV601LC and the host is hierarchically structured and broken up into blocks of data as shown in Figure 11. Table IV shows pseudo code for a video data transfer that matches the transfer order shown in Figure 11 and uses the code names shown in Table XIV. The blocks of data listed in Figure 11 correspond to wavelet compressed sections of each field illustrated in Figure 12 as a modified Mallat diagram.

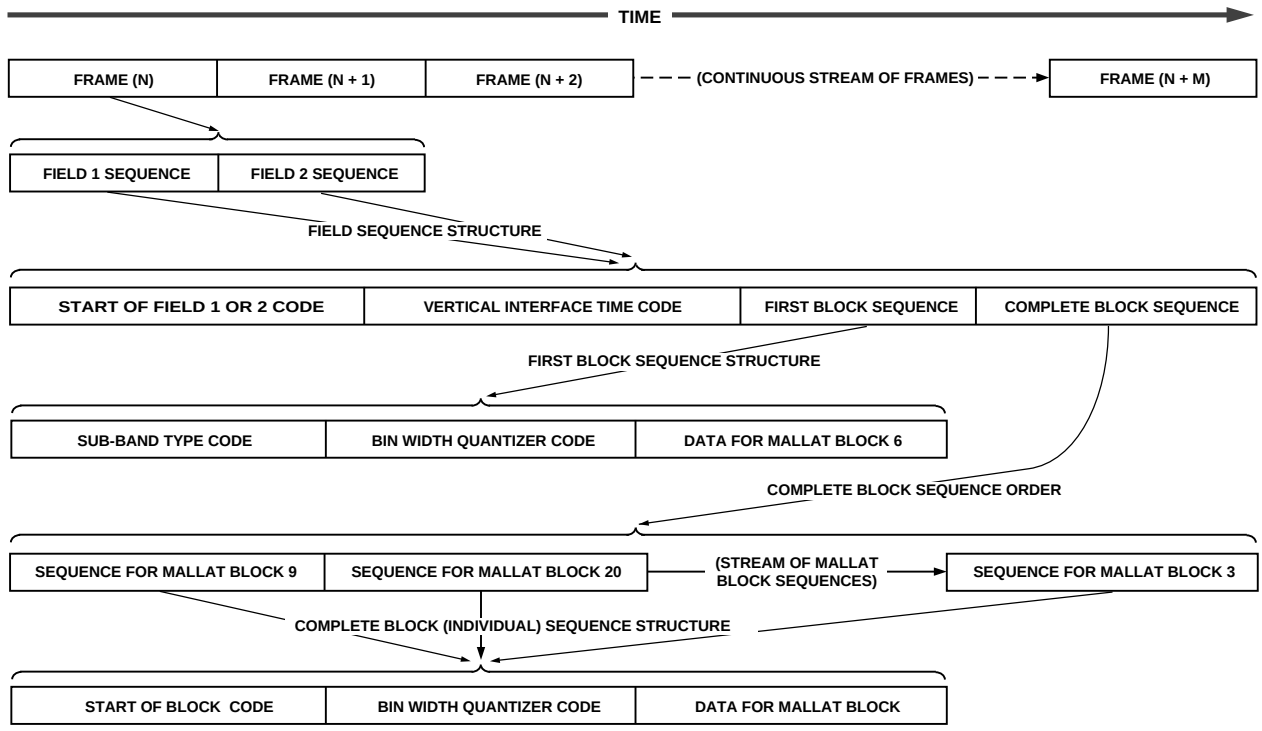


Figure 11. Hierarchical Structure of Wavelet Compressed Frame Data (Data Block Order)

Table XII. Pseudo-Code Describing a Sequence of Video Fields

Complete Sequence:	
<Field 1 Sequence>	"Frame N; Field 1"
<Field 2 Sequence>	"Frame N; Field 2"
<Field 1 Sequence>	"Frame N+1; Field 1"
<Field 2 Sequence>	"Frame N+1; Field 2"
(Field Sequences)	
<Field 1 Sequence>	"Frame N+M; Field 1"
<Field 2 Sequence>	"Frame N+M; Field 2"
#EOS	"Required in decode to let the ADV601LC know the sequence of fields is complete."
Field 1 Sequence:	
#SOF1	
<VITC>	
<First Block Sequence>	
<Complete Block Sequence>	
Field 2 Sequence:	
#SOF2	
<VITC>	
<First Block Sequence>	
<Complete Block Sequence>	
First Block Sequence:	
<TYPE4>	
<BW>	
<Huff_Data>	
Complete Block Sequence:	
<Block Sequence>	
. . .	
(Block Sequences)	
. . .	
<Block Sequence>	
Block Sequence:	
#SOB1, #SOB2, #SOB3, #SOB4 or #SOB5	
<BW>	
<Huff_Data>	

In general, a Frame of data is made up of odd and even Fields as shown in Figure 11. Each Field Sequence is made up of a First Block Sequence and a Complete Block Sequence. The First Block Sequence is separate from the Complete Block Sequence. The Complete Block Sequence contains the remaining 41 Block Sequences (see block numbering in Figure 12). Each Block Sequence contains a start of block delimiter, Bin Width for the block and actual encoder data for the block. A pseudo code bit stream example for one complete field of video is shown in Table XIII. A pseudo code bit stream example for one sequence of fields is shown in Table XIV. An example listing of a field of video in ADV601LC bitstream format appears in Table XVI.

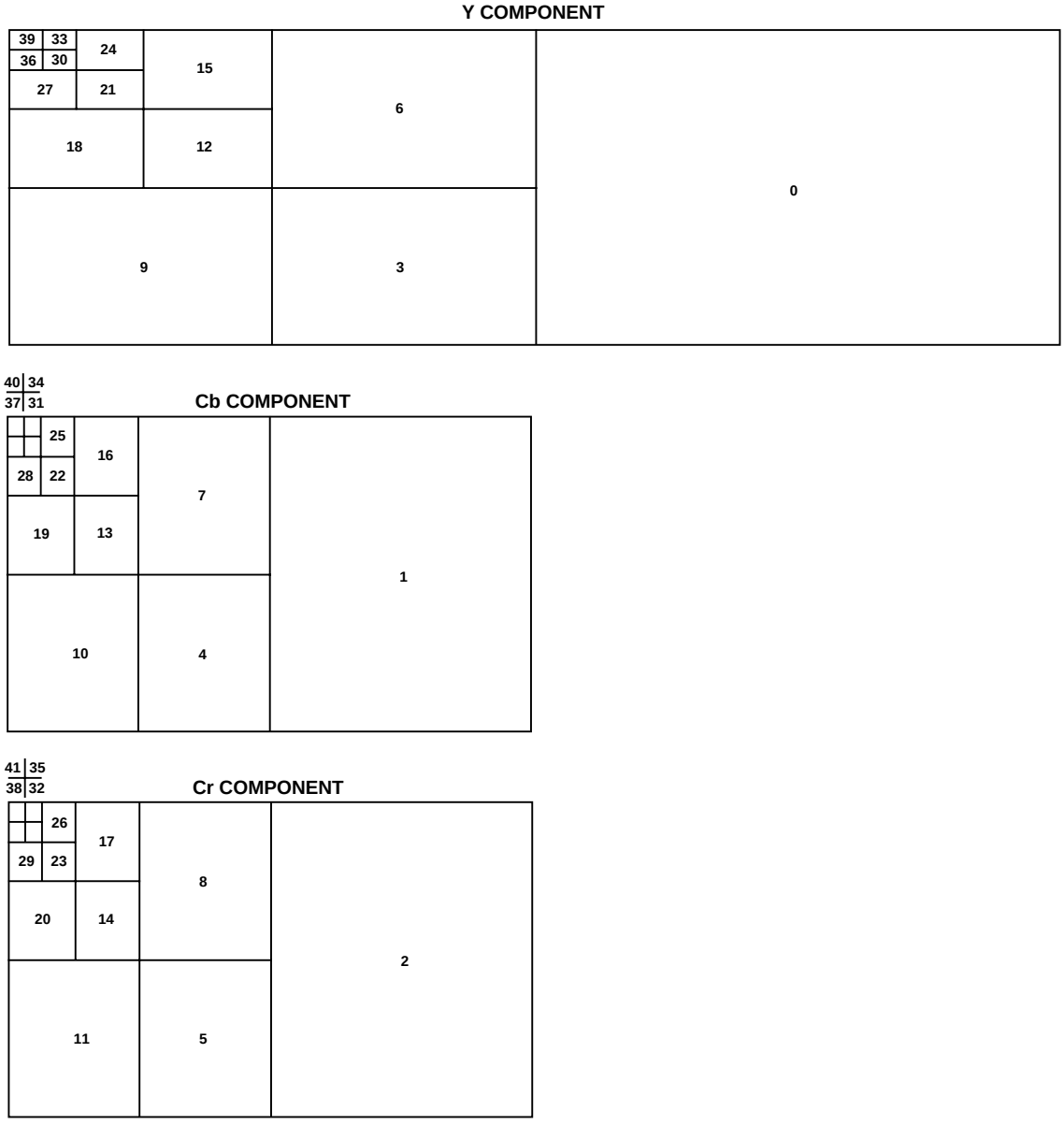


Figure 12. Block Order of Wavelet Compressed Field Data (Modified Mallat Diagram)

Table XIII. Pseudo-Code of Compressed Video Data Bitstream for One Field of Video

Block Sequence Data	For Mallat Block Number . . .
#SOFn<VITC><TYPE4><BW><Huff_Data>	n indicates field 1 or 2 Huff_Data indicates Mallat block 6 data A typical Bin Width (BW) factor for this block is 0x1DDC
#SOB4<BW><Huff_Data>	Mallat block 9 data—Typical BW = 0x1DDC
#SOB3<BW><Huff_Data>	Mallat block 20 data—Typical BW = 0x0C2E
#SOB3<BW><Huff_Data>	Mallat block 22 data—Typical BW = 0x03A1
#SOB3<BW><Huff_Data>	Mallat block 19 data—Typical BW = 0x0C2E
#SOB3<BW><Huff_Data>	Mallat block 23 data—Typical BW = 0x03A1
#SOB3<BW><Huff_Data>	Mallat block 17 data—Typical BW = 0x0C2E
#SOB3<BW><Huff_Data>	Mallat block 25 data—Typical BW = 0x0306
#SOB3<BW><Huff_Data>	Mallat block 16 data—Typical BW = 0x0C2E
#SOB3<BW><Huff_Data>	Mallat block 26 data—Typical BW = 0x0306
#SOB3<BW><Huff_Data>	Mallat block 14 data—Typical BW = 0x0E9D
#SOB3<BW><Huff_Data>	Mallat block 28 data—Typical BW = 0x0306
#SOB3<BW><Huff_Data>	Mallat block 13 data—Typical BW = 0x0E9D
#SOB3<BW><Huff_Data>	Mallat block 29 data—Typical BW = 0x0306
#SOB3<BW><Huff_Data>	Mallat block 11 data—Typical BW = 0x2410
#SOB1<BW><Huff_Data>	Mallat block 31 data—Typical BW = 0x0114
#SOB3<BW><Huff_Data>	Mallat block 10 data—Typical BW = 0x2410
#SOB1<BW><Huff_Data>	Mallat block 32 data—Typical BW = 0x0114
#SOB3<BW><Huff_Data>	Mallat block 8 data—Typical BW = 0x2410
#SOB1<BW><Huff_Data>	Mallat block 34 data—Typical BW = 0x00E5
#SOB3<BW><Huff_Data>	Mallat block 7 data—Typical BW = 0x2410
#SOB1<BW><Huff_Data>	Mallat block 35 data—Typical BW = 0x00E6
#SOB3<BW><Huff_Data>	Mallat block 5 data—Typical BW = 0x2B46
#SOB1<BW><Huff_Data>	Mallat block 37 data—Typical BW = 0x00E6
#SOB3<BW><Huff_Data>	Mallat block 4 data—Typical BW = 0x2B46
#SOB1<BW><Huff_Data>	Mallat block 38 data—Typical BW = 0x00E6
#SOB3<BW><Huff_Data>	Mallat block 2 data—Typical BW = 0xC62B
#SOB1<BW><Huff_Data>	Mallat block 40 data—Typical BW = 0x009A
#SOB3<BW><Huff_Data>	Mallat block 1 data—Typical BW = 0xC62B
#SOB1<BW><Huff_Data>	Mallat block 41 data—Typical BW = 0x009A
#SOB4<BW><Huff_Data>	Mallat block 0 data—Typical BW = 0xA417
#SOB2<BW><Huff_Data>	Mallat block 39 data—Typical BW = 0x007F
#SOB4<BW><Huff_Data>	Mallat block 12 data—Typical BW = 0x0C1A
#SOB2<BW><Huff_Data>	Mallat block 36 data—Typical BW = 0x00BE
#SOB4<BW><Huff_Data>	Mallat block 15 data—Typical BW = 0x0A16
#SOB2<BW><Huff_Data>	Mallat block 33 data—Typical BW = 0x00BE
#SOB4<BW><Huff_Data>	Mallat block 18 data—Typical BW = 0x0A16
#SOB2<BW><Huff_Data>	Mallat block 30 data—Typical BW = 0x00E4
#SOB2<BW><Huff_Data>	Mallat block 21 data—Typical BW = 0x0301
#SOB2<BW><Huff_Data>	Mallat block 27 data—Typical BW = 0x0281
#SOB2<BW><Huff_Data>	Mallat block 24 data—Typical BW = 0x0281
#SOB4<BW><Huff_Data>	Mallat block 3 data—Typical BW = 0x23D5

Table XIV specifies the Mallat block transfer order and associated Start of Block (SOB) codes. Any of these SOB codes can be replaced with an SOB#5 code for a zero data block.

Table XIV. Pseudo-Code of Compressed Video Data Bitstream for One Sequence of Video Fields

Block Sequence Data	For Mallat Block Number
#SOF1<VITC><TYPE4><BW><Huff_Data> ... (41 #SOBn blocks)	/* Mallat block 6 data */
#SOF2<VITC><TYPE4><BW><Huff_Data> ... (41 #SOBn blocks) . (any number of Fields in sequence)	/* Mallat block 6 data */
#EOS	/* Required in decode to end field sequence*/

Table XV. ADV601LC Field and Block Delimiters (Codes)

Code Name	Code	Description (Align all #Delimiter Codes to 32-Bit Boundaries)
#SOF1	0xffffffff40000000	Start of Field delimiter identifies Field1 data. #SOF1 resets the Huffman decoder and is sufficient on its own to reset the processing of the chip during decode. Please note that this code or #SOF2 are the only delimiters necessary between adjacent fields. #SOF1 operates identically to #SOF2 except that during decode it can be used to differentiate between Field1 and Field2 in the generation of the Field signal (master mode) and/or SAV/EAV codes for CCIR-656 modes.
#SOF2	0xffffffff41000000	Start of Field delimiter identifies Field2 data. #SOF resets the Huffman decoder and is sufficient on its own to reset the processing of the chip during decode. Please note that this code or #SOF1 are the only delimiters necessary between adjacent fields. #SOF2 operates identically to #SOF1 except that during decode it can be used to differentiate between Field2 and Field1 in the generation of the Field signal (master mode) and/or SAV/EAV codes for CCIR-656 modes.
<VITC>	(96 bits)	This is a 12-byte string of data extracted by the video interface during encode operations and inserted by the video interface into the video data during decode operations. The data content is 90 bits in length. For a complete description of VITC format, see pages 175-178 of Video Demystified: A Handbook For The Digital Engineer (listed in References section).
<TYPE1>	0x81	This is an 8-bit delimiter-less type code for the first sub-band block of wavelet data. (Model 1 Chroma)
<TYPE2>	0x82	This is an 8-bit delimiter-less type code for the first sub-band block of wavelet data. (Model 1 Luma)
<TYPE3>	0x83	This is an 8-bit delimiter-less type code for the first sub-band block of wavelet data. (Model 2 Chroma)
<TYPE4>	0x84	This is an 8-bit delimiter-less type code for the first sub-band block of wavelet data. (Model 2 Luma)
#SOB1 #SOB2 #SOB3 #SOB4 #SOB5	0xffffffff81 0xffffffff82 0xffffffff83 0xffffffff84 0xffffffff8f	Start of Block delimiter identifies the start of Huffman coded sub-band data. This delimiter will reset the Huffman decoder if a system ever experiences bit errors or gets out of sync. The order of blocks in the frame is fixed and therefore implied in the bit stream and no unique #SOB delimiters are needed per block. There are 41 #SOB delimiters and associated BW and Huffman data within a field. #SOB1 is differentiated from #SOB2, #SOB3 and #SOB4 in that they indicate which model and Huffman table was used in the Run Length Coder for the particular block: #SOB1 Model 1 Chroma #SOB2 Model 1 Luma #SOB3 Model 2 Chroma #SOB4 Model 2 Luma #SOB5 Zero data block. All data after this delimiter and before the next start of block delimiter is ignored (if present at all) and assumed zero including the BW value.

Table XVI. ADV601LC Field and Block Delimiters (Codes)

Code Name	Code	Description (Align all #Delimiter Codes to 32-Bit Boundaries) (*Continued*)
<BW>	(16 bits, 8.8)	This data code is not entropy coded, is always 16 bits in length and defines the Bin Width Quantizer control used on all data in the block sub-band. During decode, this value is used by the Quantizer. If this value is set to zero during decode, all Huffman data is presumed to be zero and is ignored, but must be included. During encode, this value is calculated by the external Host and is inserted into the bit stream by the ADV601LC (this value is not used by the quantizer). Another value calculated by the Host, 1/BW is actually used by the Quantizer during encode.
<HUFF_DATA>	(Modulo 32)	This data is the quantized and entropy coded block sub-band data. The data's length is dependent on block size and entropy coding so it is therefore variable in length. This field is filled with 1s making it Modulo 32 bits in length. Any Huffman decode process can be interrupted and reset by any unexpectedly received # delimiter following a bit error or synchronization problem.
#EOS	0xffffffffc0ffffff	The host sends the #EOS (End of Sequence) to the ADV601LC during decode after the last field in a sequence to indicate that the field sequence is complete. The ADV601LC does not append this code to the end of encoded field sequences; it must be added by the host.

Table XVII. Video Data Bitstream for One Field In a Video Sequence[1]

ffff	**ffff**	**40**00	0000	0000	0000	0000	0000	0000	0000	8400	00ff	df0d	8eff	**ffff**	**ffff**
8400	00ff	df0c	daff	**ffff**	**ffff**	**83**00	00ff	609f	ffff	**ffff**	**ffff**	**83**00	00fe	c5af	ffff
ffff	**ffff**	**83**00	00ff	609f	ffff	**ffff**	**ffff**	**83**00	00fe	c5af	ffff	**ffff**	**ffff**	**83**00	00ff
609f	ffff	**ffff**	**ffff**	**83**00	00fe	c70f	ffff	**ffff**	**ffff**	**83**00	00ff	609f	ffff	**ffff**	**ffff**
8300	00fe	c70f	ffff	**ffff**	**ffff**	**83**00	00ff	609f	ffff	**ffff**	**ffff**	**83**00	00fe	c78f	ffff
ffff	**ffff**	**83**00	00ff	609f	ffff	**ffff**	**ffff**	**83**00	00fe	c78f	ffff	**ffff**	**ffff**	**83**00	00ff
6894	3fff	**ffff**	**ffff**	**81**1d	40f0	90ff	ffff	**ffff**	**ffff**	**83**00	00ff	6894	3fff	**ffff**	**ffff**
811d	40f0	90ff	ffff	**ffff**	**ffff**	**83**00	00ff	68aa	bfff	**ffff**	**ffff**	**81**16	80f0	9bff	ffff
ffff	**ffff**	**83**00	00ff	68aa	bfff	**ffff**	**ffff**	**81**16	80f0	9bff	ffff	**ffff**	**ffff**	**83**00	00ff
6894	3fff	**ffff**	**ffff**	**81**16	80f0	9fff	ffff	**ffff**	**ffff**	**83**00	00ff	6894	3fff	**ffff**	**ffff**
8116	80f0	9fff	ffff	**ffff**	**ffff**	**83**00	00ff	fe62	a2ff	**ffff**	**ffff**	**81**03	e6e9	d74d	75d7
5d75	d75a	f8f9	74eb	d7af	5ebd	7af5	ebf0	f8f8	f979	7979	7979	7979	79fd	5f5f	c7e3
f1f8	fc7e	3f1f	8fc7	e5fa	ff6f	d5f6	7d9f	67d9	f67d	9f67	d9f6	7edf	abec	f87c	3e1f
0f87	c3e1	f0f8	fd9f	1f1f	2f2f	2f2f	2f2f	2f2f	2f1f	2ebd	7af5	ebd7	ae9d	74e9	a56d
6b5a	d6b5	a2b0	d249	24a5	ce36	db6d	b6db	6db7	c6fd	fd3d	3d3d	3d3d	3d3d	3d3b	7a7b
fbfb	fbfb	fbfb	fbfb	fcfd	bdfe	dfb7	edfb	7eef	bbee	fbbe	dfbb	dbe7	f6fd	ff7f	dff7
fdff	7fdf	f7fd	feff	3fbb	effb	feff	bfef	fbfe	ffbf	efff	ffff	**ffff**	**ffff**	**83**00	00ff
fe62	a2ff	**ffff**	**ffff**	**81**03	e6fd	bfab	f9bf	57d5	f2eb	18f4	f9fd	ffb7	f5ff	3feb	fafc
7431	e9f4	fbff	77eb	fd3f	b3ec	f2d5	efeb	f6fe	1fbb	f67e	afdb	f0f3	aaed	edf7	fe3f
57ed	fd7f	bbe3	d2d3	dfe7	f87e	5f57	eefd	9fbb	e5d6	2fdf	e7f8	7eff	abf7	7ecf	ddf2
eb17	eff3	fc3f	7fd5	fbbf	67ee	f975	8bf7	f9fe	1fbf	eafd	dfb3	f77c	bac5	fbfc	ff0f
dff5	7eef	d9fb	be5d	62fd	fe7f	87ef	fabf	77ec	fddf	2eb1	7eff	3fc3	f7fd	5fbb	f67e
ef97	58bf	7f9f	e1fb	feaf	ddfb	3f77	cbac	5fbf	cff0	fdff	57ee	fd9f	bbe5	d62f	dfe7
f87e	ffaf	f77e	cfab	e5d6	2fe9	f3fc	7f7f	d9f5	7edf	abc7	431e	9f4f	c7f8	7fff	ffff
ffff	**ffff**	**84**00	00ff	dfb7	c5ff	df0d	7fff	**ffff**	**ffff**	**82**02	9afc	3eff	b7e9	ede9	e9e9
e9e9	e9e9	e9e9	e9e9	e9e9	e9e9	e9e9	dbef	fbbe	9efe	9dbb	76ed	dbb7	6edd	bb76	eddb
b76e	ddb7	fbbe	df9f	af6d	b6db	6db6	db6d	b6db	6db6	db6d	aff6	fd3d	bbed	7bde	f7bd
ef7b	def7	bdef	75f4	f7f4	dee9	2492	4924	924c	fa7b	77da	6991	f4f7	efb4	d323	e9ed
df69	a647	d3db	bed3	4c8f	a7b7	7da6	991f	4f7e	fb4d	323e	9edd	f69a	647d	3dbb	ed34
c8fa	7b77	da69	647c	fd7b	6100	0000	0045	bdfd	37bb	8888	8888	8888	8888	8aff	ffff
ffff	**ffff**	**84**00	00ff	c9a7	1fff	**ffff**	**ffff**	**82**0f	00ff	7704	4fff	**ffff**	**ffff**	**84**00	00ff
c9a7	1fff	**ffff**	**ffff**	**82**0f	00ff	7704	bfff	**ffff**	**ffff**	**84**00	00ff	c9a7	1fff	**ffff**	**ffff**
8213	80ff	7703	5fff	**ffff**	**ffff**	**82**00	00ff	7743	1fff	**ffff**	**ffff**	**82**00	00ff	7743	1fff
ffff	**ffff**	**82**00	00ff	7745	efff	**ffff**	**ffff**	**84**00	00ff	df0c	daff				

NOTE
[1]This table shows ADV601LC compressed data for one field in a color ramp video sequence. The SOF# and SOB# codes in the data are in **bold** text.

Bit Error Tolerance

Bit error tolerance is ensured because a bit error within a Huffman coded stream does not cause #delimiter symbols to be misread by the ADV601LC in decode mode. The worst error that can occur is loss of a complete block of Huffman data. With the ADV601LC, this type of error results only in some blurring of the decoded image, not complete loss of the image.

APPLYING THE ADV601LC

This section includes the following topics:

- Using the ADV601LC in computer applications
- Using the ADV601LC in standalone applications
- Configuring the host interface for 6- or 32-bit data paths
- Connecting the video interface to popular video encoders and decoders
- Getting the most out of the ADV601LC

The following Analog Devices products should be considered in ADV601LC designs:

- ADV7175/ADV7176—Digital YUV to analog composite video encoder
- AD722—Analog RGB to analog composite video encoder
- AD1843—Audio codec with embedded video synchronization
- ADSP-21xx—Family of fixed-point digital signal processors
- AD8xxx—Family of video operational amplifiers

Using the ADV601LC in Computer Applications

Many key features of the ADV601LC were driven by the demanding cost and performance requirements of computer applications. The following ADV601LC features provide key advantages in computer applications:

- *Host Interface*
  The 512 double word FIFO provides necessary buffering of compressed digital video to deal with PCI bus latency.
- *Low Cost External DRAM*
  Unlike many other real-time compression solutions, the ADV601LC does not require expensive external SRAM transform buffers or VRAM frame stores.

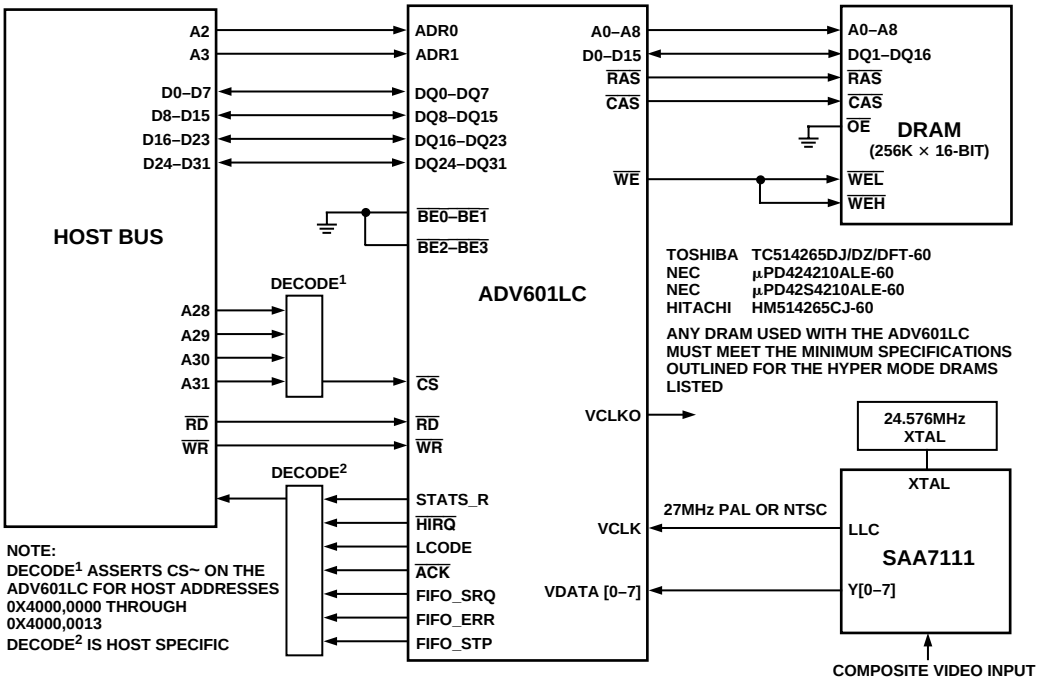


Figure 13. A Suggested PC Application Design

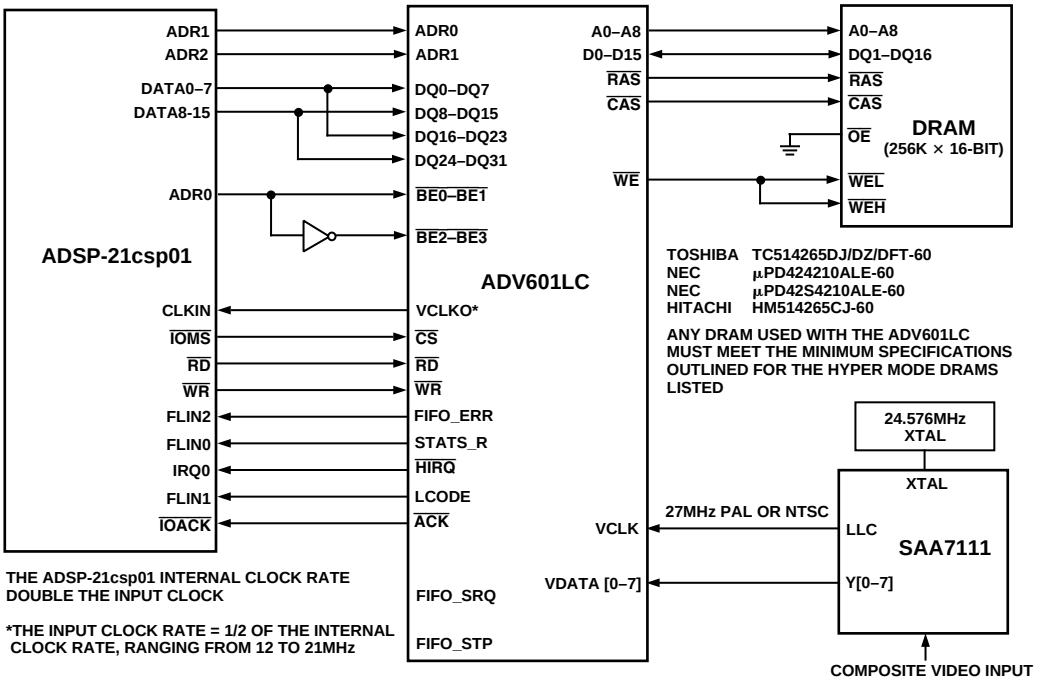


Figure 14. Alternate Standalone Application Design

Using the ADV601LC In Standalone Applications

Figure 14 shows the ADV601LC in a noncomputer based applications. Here, an ADSP-21csp01 digital signal processor provides Host control and BW calculation services. Note that all control and BW operations occur over the host interface in this design.

Connecting the ADV601LC to Popular Video Decoders and Encoders

The following circuits are recommendations only. Analog Devices has not actually built or tested these circuits.

Using the Philips SAA7111 Video Decoder

The SAA7111 example circuit, which appears in Figure 15, is used in this configuration on the ADV601LC Video Lab demonstration board.

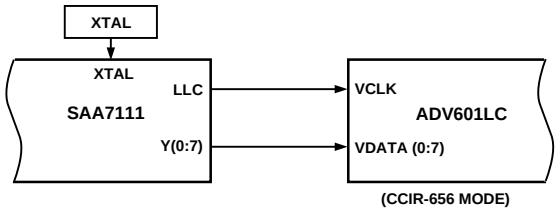


Figure 15. ADV601LC and SAA7111 Example Interfacing Block Diagram

Using the Analog Devices ADV7175 Video Encoder

Because the ADV7175 has a CCIR-656 interface, it connects directly with the ADV601LC without "glue" logic. Note that the ADV7175 can only be used at CCIR-601 sampling rates.

The ADV7175 example circuit, which appears in Figure 16, is used in this configuration on the ADV601LC Video Lab demonstration board.

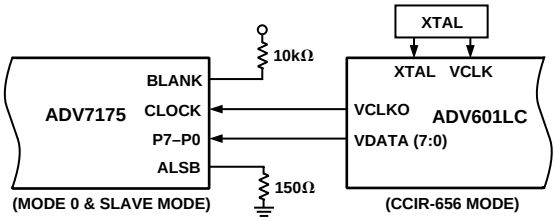


Figure 16. ADV601LC and ADV7175 Example Interfacing Block Diagram

Using the Raytheon TMC22173 Video Decoder

Raytheon has a whole family of video parts. Any member of the family can be used. The user must select the part needed based on the requirements of the application. Because the Raytheon part does not include the A/Ds, an external A/D is necessary in this design (or a pair of A/Ds for S video).

The part can be used in CCIR-656 (D1) mode for a zero control signal interface. Special attention must be paid to the video output modes in order to get the right data to the right pins (see the following diagram).

Note that the circuit in Figure 17 has not been built or tested.

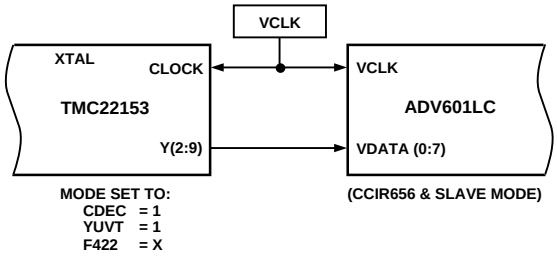


Figure 17. ADV601LC and TMC22153 Example CCIR-656 Mode Interface

GETTING THE MOST OUT OF ADV601LC

The unique sub-band block structure of luminance and color components in the ADV601LC offers many unique application benefits. Analog Devices will offer a Feature Software Library as well as separate feature application documentation to help users exploit these features. The following section provides an overview of only some of the features and how they are achieved with the ADV601LC. Please refer to Figures 2 and 3 as necessary.

Higher Compression With Interfield Techniques

The ADV601LC normally operates as a field-independent codec. However, through use of the sub-bands it is possible to use the ADV601LC with interfield techniques to achieve even higher levels of compression. In such applications, each field is not compressed separately, thus accessing the compressed bit stream can only be done at specific points in time. There are two general ways this can be accomplished:

- *Subsampling high frequency blocks*
  The human visual system is more sensitive to interframe motion of low frequency block than to motion in high frequency blocks. The host software driver of the ADV601LC allows exploitation of this option to achieve higher compression. Note that the compressed bit stream can only be accessed at points where the high frequency blocks have just been updated.
- *Updating the image with motion detection*
  In applications where the video is likely to have no motion for extended periods of time (video surveillance in a vacant building, for instance), it is only necessary to update the image either periodically or when motion occurs. By using the wavelet sub-bands to detect motion (see later in this section), it is possible to achieve very high levels of compression when motion is infrequent.

Scalable Compression Technology

The ADV601LC offers many different options for scaling the image, the compressed bit stream bandwidth and the processing horsepower for encode or decode. Because the ADV601LC employs decimators, interpolators and filters in the filter bank, the scaling function creates much higher quality images than achieved through pixel dropping. Mixing and matching the many scaling options is useful in network applications where transmission pipes may vary in available bit rate, and decode/encode capabilities may be a mix of software and hardware. These are the key options:

- *Extract scaled images by factors of 2 from the compressed bit stream*
  This is useful in video editing applications where thumbnail sketches of fields need to be displayed. In this case, editing software can quickly extract and decode the desired image. This technique eliminates the burden of decoding an entire image and then scaling to the desired size.
- *Use software to decode bit stream*
  Decoding an entire CCIR-601 resolution image in real time at 50/60 fields per second does require the ADV601LC hardware. Analog Devices provides a bit-exact ADV601LC simulator that can decode a scaled image in real time or a full-size image off-line. Image size and frame rates depend on the performance of the host processor.
- *Scale bit stream*
  The compressed video bit stream was created with simple parsing in mind. This type of parsing means that a lower resolution/lower bandwidth bit stream can be extracted with little computational burden. Generally, this effect is accomplished by selecting a subset of lower frequency blocks. This technique is useful in applications where the same video source material must be sent over a range of different communication pipes {i.e., ISDN (128 Kbps), T1 (1.5 Mbps) or T3 (45 Mbps)}.
- *Use software to encode*
  In this case, a host CPU could encode a smaller image size and fill in high frequency blocks with zeros. Again, image quality would depend on the performance of the host. The Bin Width may be set to zero, zeroing out the data in any particular Mallat block.

Parametric Image Filtering

The ADV601LC offers a unique set of image filtering capabilities not found in other compression technologies. The ADV601LC quantizer is capable of attenuating any or all of the luminance or chrominance blocks during encode or decode. Here are some of the possible applications:

- *Parametric softening of color saturation and contrast during encode or decode*
  Trade off image softness for higher compression. Attenuation of the higher frequency blocks during encode leads to softer images, but it can lead to much higher compression performance.
- *Color saturation control*
  This effect is achieved by controlling gain of low pass chrominance blocks during encode or decode.
- *Contrast control*
  This effect is achieved by controlling the gain of the low frequency luminance blocks during encode or decode.
- *Fade to black*
  This effect achieved by attenuation of luminance blocks.

Mixing of Two or More Images

Blocks from different images can be mixed into the bit stream and then sent to the ADV601LC during decode. The result is high quality mixing of different images. This also provides the capability to fade from one image to the next.

Edge or Motion Detection

In certain remote video surveillance and machine vision applications, it is desirable to detect edges or motion. Edges can be quickly found through evaluation of the high frequency blocks. Motion searches can be achieved in two ways:

- *Evaluation of the smallest luminance block*. Because the size of the smallest block is so mcuh smaller than the others, the computational burden is significantly less than doing an evaluation over the entire image.
- *Polling the Sum of Squares registers.* Because large changes in the video data create patterns, it is possible to detect motion in the video by polling the Sum of Squares registers, looking for patterns and changes.

The ADV601LC Video Codec uses a Bi-Orthogonal (7, 9) Wavelet Transform.

RECOMMENDED OPERATING CONDITIONS

Parameter	Description	Min	Max	Unit
V_{DD}	Supply Voltage	4.50	5.50	V
T_{AMB}	Ambient Operating Temperature	0	+70	°C

ELECTRICAL CHARACTERISTICS

Parameter	Description	Test Conditions	Min	Max	Unit
V_{IH}	Hi-Level Input Voltage	@ V_{DD} = max	2.0	N/A	V
V_{IL}	Lo-Level Input Voltage	@ V_{DD} = min	N/A	0.8	V
V_{OH}	Hi-level Output Voltage	@ V_{DD} = min, I_{OH} = –0.5 mA	2.4	N/A	V
V_{OL}	Lo-Level Output Voltage	@ V_{DD} = min, I_{OL} = 2 mA	N/A	0.4	V
I_{IH}	Hi-Level Input Current	@ V_{DD} = max, V_{IN} = V_{DD} max	N/A	10	µA
I_{IL}	Lo-Level Input Current	@ V_{DD} = max, V_{IN} = 0 V	N/A	10	µA
I_{OZH}	Three-State Leakage Current	@ V_{DD} = max, V_{IN} = V_{DD} max	N/A	10	µA
I_{OZL}	Three-State Leakage Current	@ V_{DD} = max, V_{IN} = 0 V	N/A	10	µA
C_I	Input Pin Capacitance	@ V_{IN} = 2.5 V, f_{IN} = 1.0 MHz, T_{AMB} = 25°C	N/A	8*	pF
C_O	Output Pin Capacitance	@ V_{IN} = 2.5 V, f_{IN} = 1.0 MHz, T_{AMB} = 25°C	N/A	8*	pF

*Guaranteed but not tested.

ABSOLUTE MAXIMUM RATINGS*

Parameter	Description	Min	Max	Unit
V_{DD}	Supply Voltage	–0.3	+7	V
V_{IN}	Input Voltage	N/A	V_{DD} ± 0.3	V
V_{OUT}	Output Voltage	N/A	V_{DD} ± 0.3	V
T_{AMB}	Ambient Operating Temperature	0	+70	°C
T_S	Storage Temperature	–65	+150	°C
T_L	Lead Temperature (5 sec) LQFP	N/A	+280	°C

*Stresses greater than those listed above under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the Pin Definitions section of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

SUPPLY CURRENT AND POWER

Parameter	Description	Test Conditions	Min	Max	Unit
I_{DD}	Supply Current (Dynamic)	@ V_{DD} = max, t_{VCLK_CYC} = 37 ns (at 27 MHz VCLK)	0.11	0.27	A
I_{DD}	Supply Current (Soft Reset)	@ V_{DD} = max, t_{VCLK_CYC} = 37 ns (at 27 MHz VCLK)	0.08	0.17	A
I_{DD}	Supply Current (Idle)	@ V_{DD} = max, t_{VCLK_CYC} = None	0.01	0.02	A

ENVIRONMENTAL CONDITIONS

Parameter	Description	Max	Unit
θ_{CA}	Case-to-Ambient Thermal Resistance	30	°C/W
θ_{JA}	Junction-to-Ambient Thermal Resistance	35	°C/W
θ_{JC}	Junction-to-Case Thermal Resistance	5	°C/W

CAUTION

The ADV601LC is an ESD (electrostatic discharge) sensitive device. Electrostatic charges readily accumulate on the human body and equipment and can discharge without detection. Permanent damage may occur to devices subjected to high energy electrostatic discharges. Proper ESD precautions are strongly recommended to avoid functional damage or performance degradation.

The ADV601LC latchup immunity has been demonstrated at ≥100 mA/–100 mA on all pins when tested to industry standard/JEDEC methods.

TEST CONDITIONS

Figure 18 shows test condition voltage reference and device loading information. These test conditions consider an output as *disabled* when the output stops driving and goes from the measured high or low voltage to a high impedance state. Tests measure output disable time ($t_{DISABLE}$) as the time between the reference input signal crossing +1.5 V and the time that the output reaches the high impedance state (also +1.5 V). Similarly, these tests conditions consider an output as *enabled* when the output leaves the high impedance state and begins driving a measured high or low voltage. Tests measure output enable time (t_{ENABLE}) as the time between the reference input signal crossing +1.5 V and the time that the output reaches the measured high or low voltage.

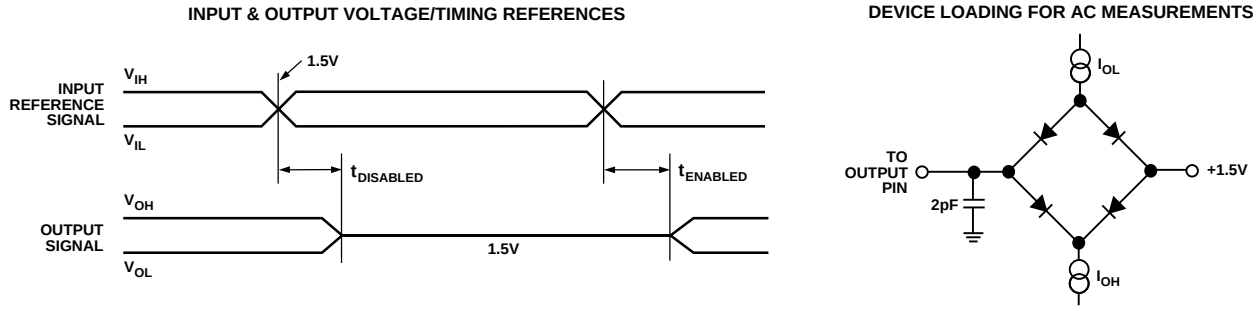


Figure 18. Test Condition Voltage Reference and Device Loading

TIMING PARAMETERS

This section contains signal timing information for the ADV601LC. Timing descriptions for the following items appear in this section:

- Clock signal timing
- Video data transfer timing (CCIR-656, and Multiplexed Philips formats)
- Host data transfer timing (direct register read/write access)

Clock Signal Timing

The diagram in this section shows timing for VCLK input and VCLKO output. All output values assume a maximum pin loading of 50 pF.

Table XVIII. Video Clock Period, Frequency, Drift and Jitter

Video Format	Min VCLK_CYC Period	Nominal VCLK_CYC Period (Frequency)	Max VCLK_CYC Period[1, 2]
CCIR-601 PAL	35.2 ns	37 ns (27 MHz)	38.9 ns
CCIR-601 NTSC	35.2 ns	37 ns (27 MHz)	38.9 ns

NOTES
[1]VCLK Period Drift = ±0.1 (VCLK_CYC/field.
[2]VCLK edge-to-edge jitter = 1 ns.

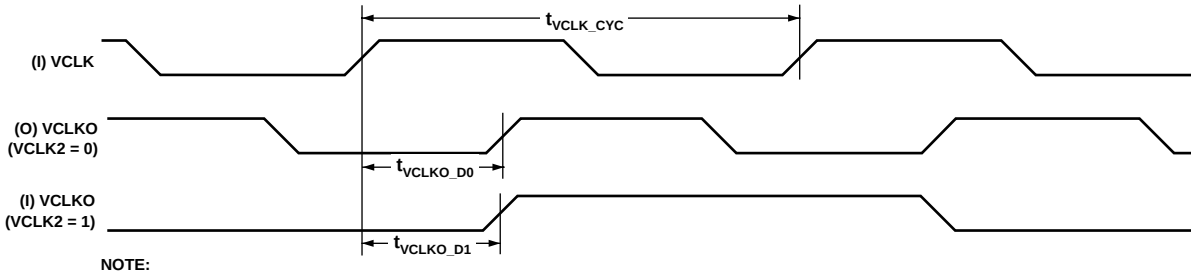
Table XIX. Video Clock Duty Cycle

	Min	Nominal	Max
VCLK Duty Cycle[1]	(40%)	(50%)	(60%)

NOTE
[1]VCLK Duty Cycle = $t_{VCLK\_HI}/(t_{VCLK\_LO}) \times 100$.

Table XX. Video Clock Timing Parameters

Parameter	Description	Min	Max	Unit
t_{VCLK_CYC}	VCLK Signal, Cycle Time (1/Frequency) at 27 MHz	(See Video Clock Period Table)		
t_{VCLKO_D0}	VCLKO Signal, Delay (when VCLK2 = 0) at 27 MHz	10	29	ns
t_{VCLKO_D1}	VCLKO Signal, Delay (when VCLK2 = 1) at 27 MHz	10	29	ns



NOTE:
USE VCLK FOR CLOCKING VIDEO-ENCODE OPERATIONS AND USE VCLKO FOR CLOCKING VIDEO-DECODE OPERATIONS.
DO NOT TRY TO USE EITHER CLOCK FOR BOTH ENCODE AND DECODE.

Figure 19. Video Clock Timing

CCIR-656 Video Format Timing

The diagrams in this section show transfer timing for pixel (YCrCb), line (horizontal), and frame (vertical) data in CCIR-656 video mode. All output values assume a maximum pin loading of 50 pF. Note that in timing diagrams for CCIR-656 video, the label *CTRL* indicates the VSYNC, HSYNC, and FIELD pins.

Table XXI. CCIR-656 Video—Decode Pixel (YCrCb) Timing Parameters

Parameter	Description	Min	Max	Units
$t_{VDATA\_DC\_D}$	VDATA Signals, Decode CCIR-656 Mode, Delay	N/A	14	ns
$t_{VDATA\_DC\_OH}$	VDATA Signals, Decode CCIR-656 Mode, Output Hold	4	N/A	ns
$t_{CTRL\_DC\_D}$	CTRL Signals, Decode CCIR-656 Mode, Delay	N/A	11	ns
$t_{CTRL\_DC\_OH}$	CTRL Signals, Decode CCIR-656 Mode, Output Hold	5	N/A	ns

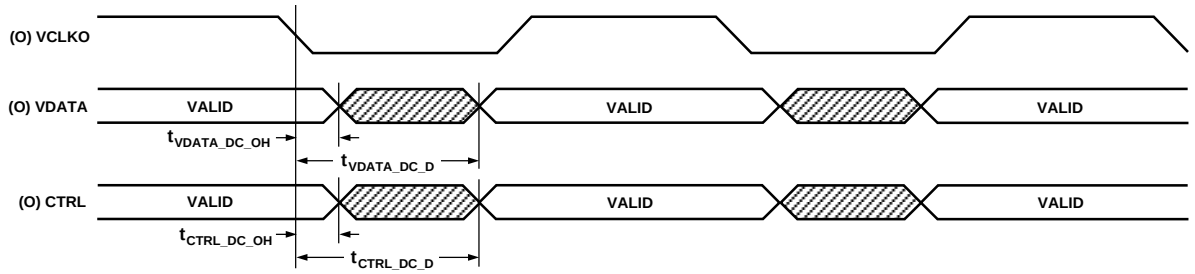


Figure 20. CCIR-656 Video—Decode Pixel (YCrCb) Transfer Timing

Table XXII. CCIR-656 Video—Encode Pixel (YCrCb) Timing Parameters

Parameter	Description	Min	Max	Units
$t_{VDATA\_EC\_S}$	VDATA Bus, Encode CCIR-656 Mode, Setup	2	N/A	ns
$t_{VDATA\_EC\_H}$	VDATA Bus, Encode CCIR-656 Mode, Hold	5	N/A	ns
$t_{CTRL\_EC\_D}$	CTRL Signals, Encode CCIR-656 Mode, Delay	N/A	33	ns
$t_{CTRL\_EC\_OH}$	CTRL Signals, Encode CCIR-656 Mode, Output Hold	20	N/A	ns

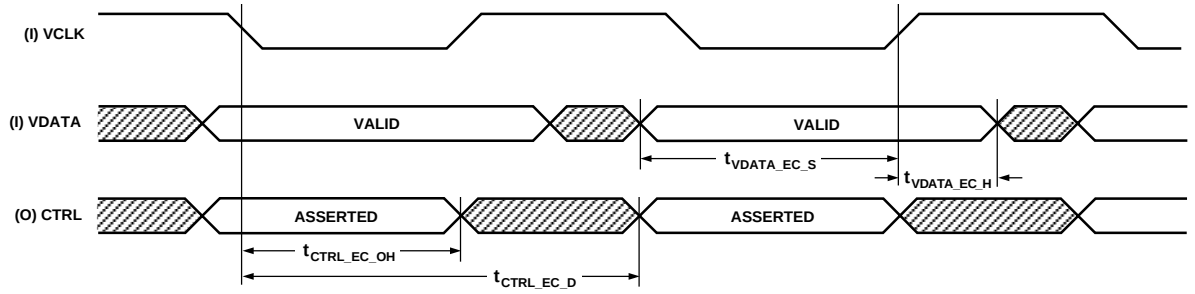


Figure 21. CCIR-656 Video—Encode Pixel (YCrCb) Transfer Timing

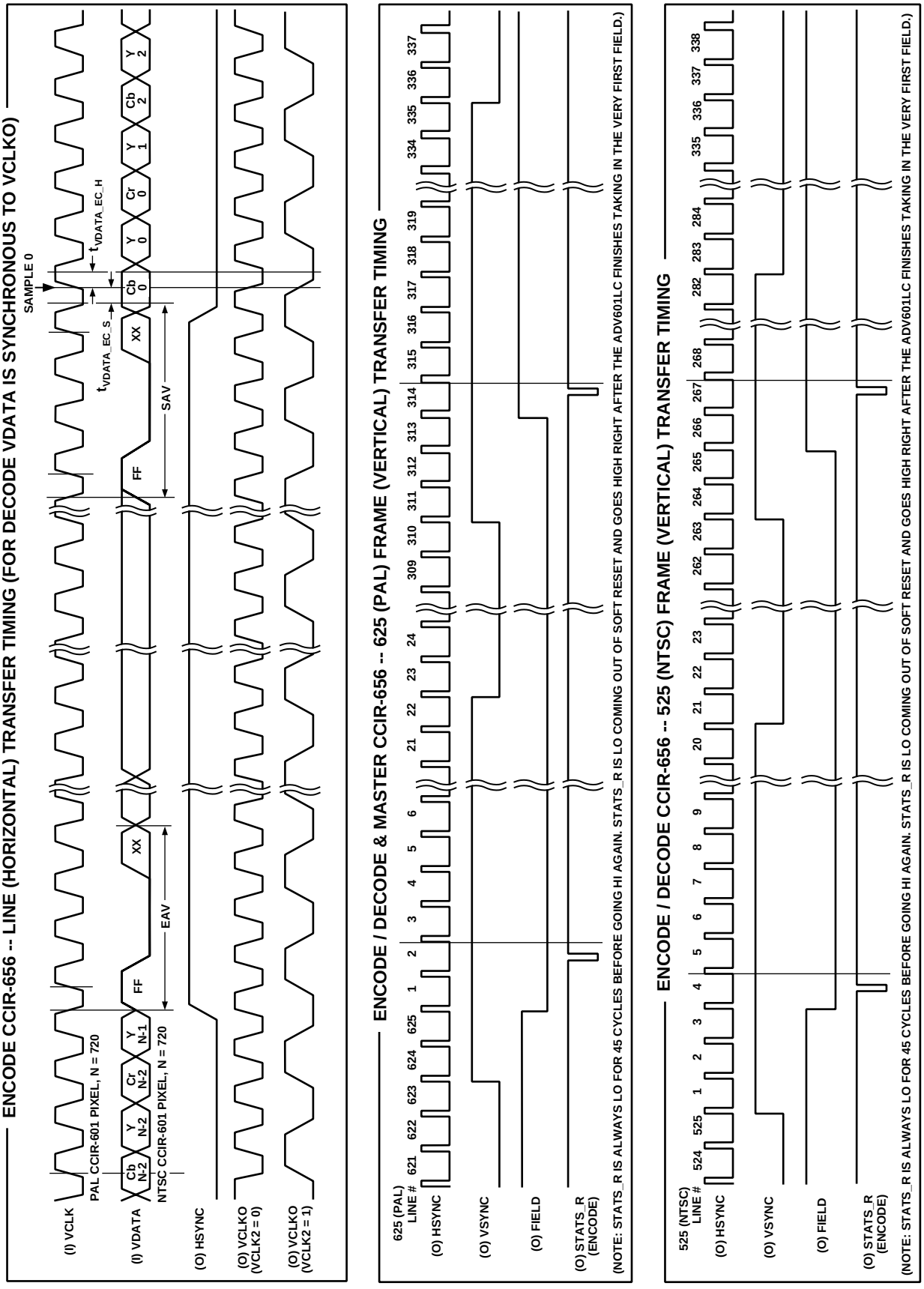


Figure 22. CCIR-656 Video—Line (Horizontal) and Frame (Vertical) Transfer Timing

Note that for CCIR-656 Video—Decode and Master Line (Horizontal) timing, VDATA is synchronous with VCLKO.

Multiplexed Philips Video Timing

The diagrams in this section show transfer timing for pixel (YCrCb) data in Multiplexed Philips video mode. For line (horizontal) and frame (vertical) data transfer timing, see Figure 25. All output values assume a maximum pin loading of 50 pF. Note that in timing diagrams for Multiplexed Philips video, the label CTRL indicates the VSYNC, HSYNC and FIELD pins.

Table XXIII. Multiplexed Philips Video—Decode and Master Pixel (YCrCb) Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{VDATA\_DMM\_D}$	VDATA Bus, Decode Master Multiplexed Philips, Delay	N/A	14	ns
$t_{VDATA\_DMM\_OH}$	VDATA Bus, Decode Master Multiplexed Philips, Output Hold	4	N/A	ns
$t_{CTRL\_DMM\_D}$	CTRL Signals, Decode Master Multiplexed Philips, Delay	N/A	11	ns
$t_{CTRL\_DMM\_OH}$	CTRL Signals, Decode Master Multiplexed Philips, Output Hold	5	N/A	ns

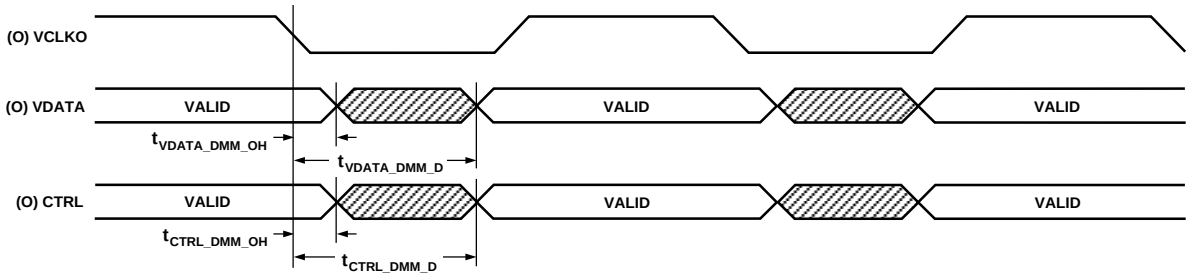


Figure 23. Multiplexed Philips Video—Decode and Master Pixel (YCrCb) Transfer Timing

Table XXIV. Multiplexed Philips Video—Decode and Slave Pixel (YCrCb) Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{VDATA\_DSM\_D}$	VDATA Bus, Decode Slave Multiplexed Philips, Delay	N/A	14	ns
$t_{VDATA\_DSM\_OH}$	VDATA Bus, Decode Slave Multiplexed Philips, Output Hold	4	N/A	ns
$t_{CTRL\_DSM\_S}$	CTRL Signals, Decode Slave Multiplexed Philips, Setup	16	N/A	ns
$t_{CTRL\_DSM\_H}$	CTRL Signals, Decode Slave Multiplexed Philips, Hold	42	N/A	ns

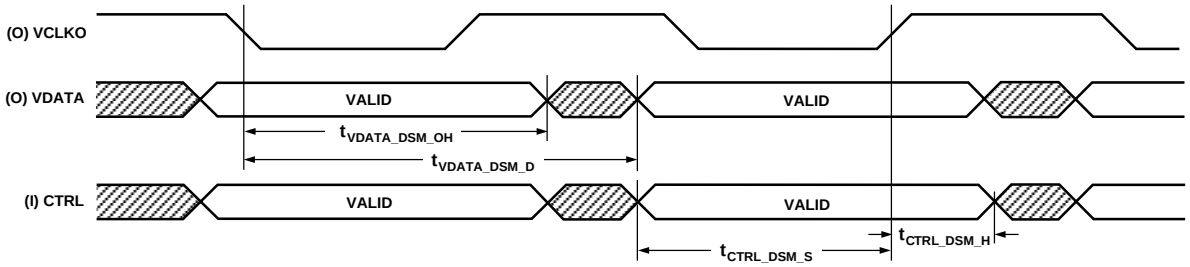


Figure 24. Multiplexed Philips Video—Decode and Slave Pixel (YCrCb) Transfer Timing

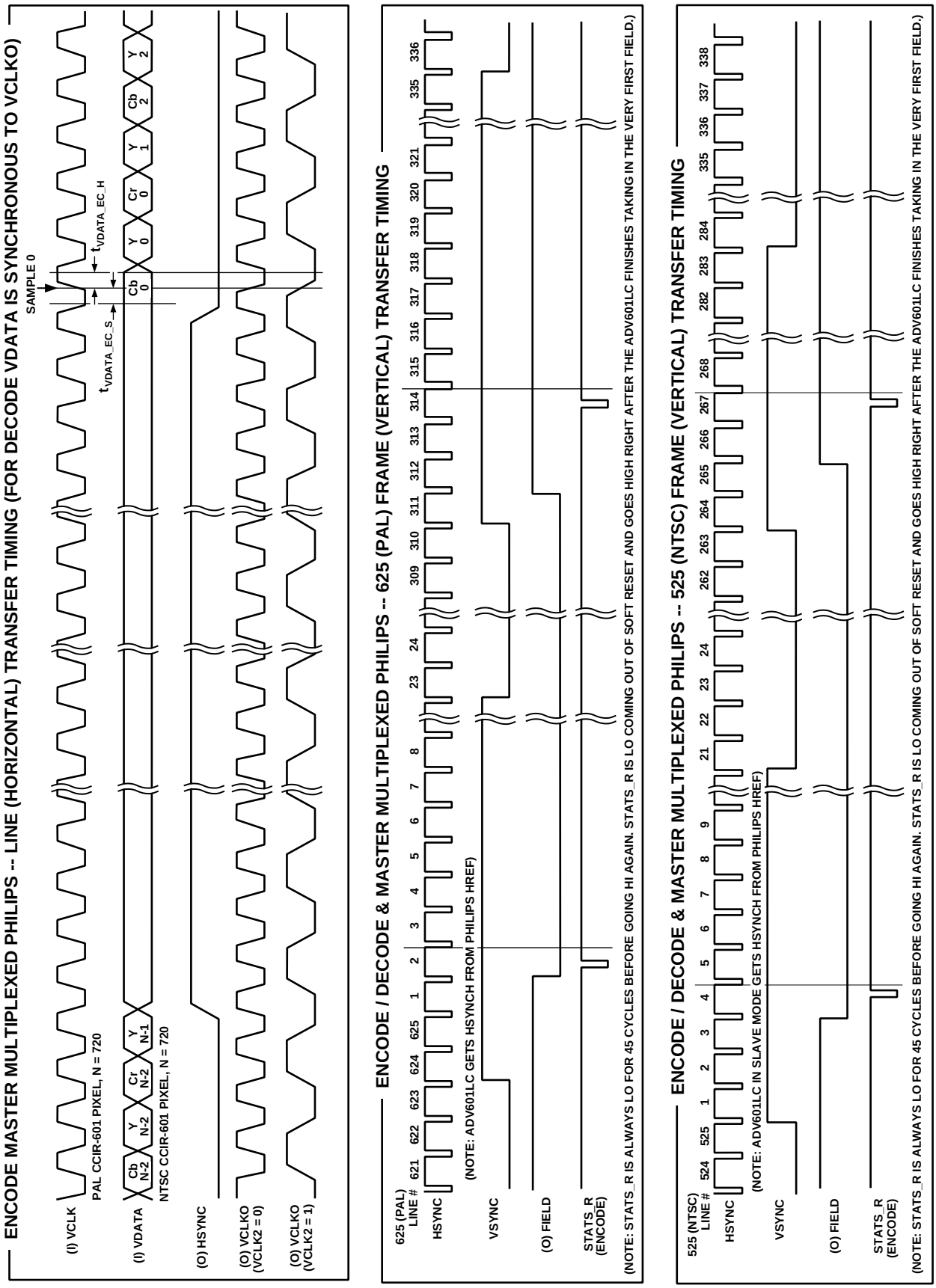


Figure 25. Multiplexed Philips Video–Line (Horizontal) and Frame (Vertical) Transfer Timing

Table XXV. Multiplexed Philips Video—Encode and Master Pixel (YCrCb) Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{VDATA\_EMM\_S}$	VDATA Bus, Encode Master Multiplexed Philips, Setup	2	N/A	ns
$t_{VDATA\_EMM\_H}$	VDATA Bus, Encode Master Multiplexed Philips, Hold	5	N/A	ns
$t_{CTRL\_EMM\_D}$	CTRL Signals, Encode Master Multiplexed Philips, Delay	N/A	33	ns
$t_{CTRL\_EMM\_OH}$	CTRL Signals, Encode Master Multiplexed Philips, Output Hold	20	N/A	ns

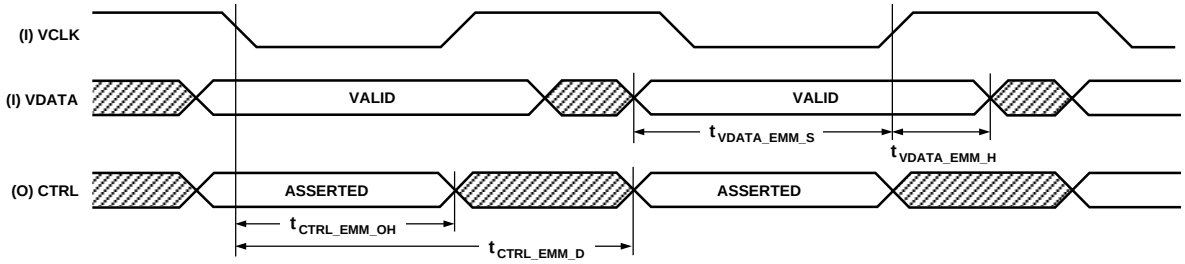


Figure 26. Multiplexed Philips Video—Encode and Master Pixel (YCrCb) Transfer Timing

Table XXVI. Multiplexed Philips Video—Encode and Slave Pixel (YCrCb) Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{VDATA\_ESM\_S}$	VDATA Bus, Encode Slave Multiplexed Philips Mode, Setup	2	N/A	ns
$t_{VDATA\_ESM\_H}$	VDATA Bus, Encode Slave Multiplexed Philips Mode, Hold	5	N/A	ns
$t_{CTRL\_ESM\_S}$	CTRL Signals, Encode Slave Multiplexed Philips Mode, Setup	5	N/A	ns
$t_{CTRL\_ESM\_H}$	CTRL Signals, Encode Slave Multiplexed Philips Mode, Hold	5	N/A	ns

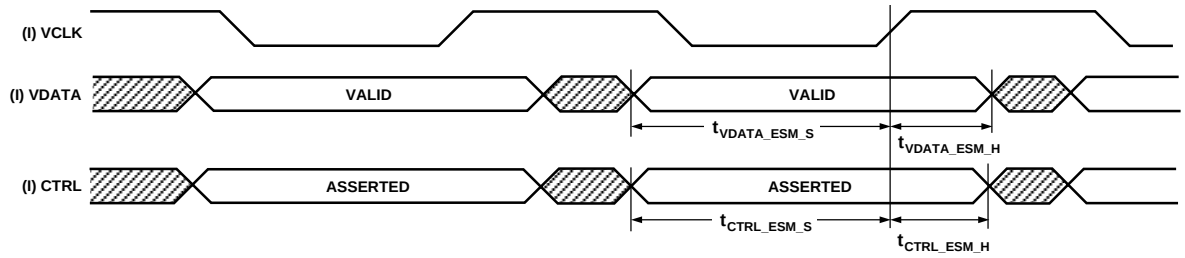


Figure 27. Multiplexed Philips Video—Encode and Slave Pixel (YCrCb) Transfer Timing

Host Interface (Indirect Address, Indirect Register Data, and Interrupt Mask/Status) Register Timing

The diagrams in this section show transfer timing for host read and write accesses to all of the ADV601LC's direct registers, except the Compressed Data register. Accesses to the Indirect Address, Indirect Register Data, and Interrupt Mask/Status registers are *slower* than access timing for the Compressed Data register. For information on access timing for the Compressed Data direct register, see the Host Interface (Compressed Data) Register Timing section. Note that for accesses to the Indirect Address, Indirect Register Data and Interrupt Mask/Status registers, your system *MUST* observe $\overline{ACK}$ and $\overline{RD}$ or $\overline{WR}$ assertion timing.

Table XXVII. Host (Indirect Address, Indirect Data, and Interrupt Mask/Status) Read Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{RD\_D\_RDC}$	$\overline{RD}$ Signal, Direct Register, Read Cycle Time (at 27 MHz VCLK)	N/A[1]	N/A	ns
$t_{RD\_D\_PWA}$	$\overline{RD}$ Signal, Direct Register, Pulsewidth Asserted (at 27 MHz VCLK)	N/A[1]	N/A	ns
$t_{RD\_D\_PWD}$	$\overline{RD}$ Signal, Direct Register, Pulsewidth Deasserted (at 27 MHz VCLK)	5	N/A	ns
$t_{ADR\_D\_RDS}$	ADR Bus, Direct Register, Read Setup	2	N/A	ns
$t_{ADR\_D\_RDH}$	ADR Bus, Direct Register, Read Hold	2	N/A	ns
$t_{DATA\_D\_RDD}$	DATA Bus, Direct Register, Read Delay	N/A	171.6[2, 3]	ns
$t_{DATA\_D\_RDOH}$	DATA Bus, Direct Register, Read Output Hold (at 27 MHz VCLK)	26	N/A	ns
$t_{RD\_D\_WRT}$	$\overline{WR}$ Signal, Direct Register, Read-to-Write Turnaround (at 27 MHz VCLK)	48.7[4]	N/A	ns
$t_{ACK\_D\_RDD}$	$\overline{ACK}$ Signal, Direct Register, Read Delayed (at 27 MHz VCLK)	8.6	287.1[5, 6]	ns
$t_{ACK\_D\_RDOH}$	$\overline{ACK}$ Signal, Direct Register, Read Output Hold (at 27 MHz VCLK)	11	N/A	ns

NOTES
[1]$\overline{RD}$ input must be asserted (low) until ACK is asserted (low).
[2]Maximum $t_{DATA\_D\_RDD}$ varies with VCLK according to the formula: $t_{DATA\_D\_RDD\ (MAX)}$ = 4 (VCLK Period) +16.
[3]During STATS_R deasserted (low) conditions, $t_{DATA\_D\_RDD}$ may be as long as 52 VCLK periods.
[4]Minimum $t_{RD\_D\_WRT}$ varies with VCLK according to the formula: $t_{RD\_D\_WRT\ (MIN)}$ = 1.5 (VCLK Period) –4.1.
[5]Maximum $t_{ACK\_D\_RDD}$ varies with VCLK according to formula: $t_{ACK\_D\_RDD\ (MAX)}$ = 7 (VCLK Period) +14.8.
[6]During STATS_R deasserted (low) conditions, $t_{ACK\_D\_RDD}$ may be as long as 52 VCLK periods.

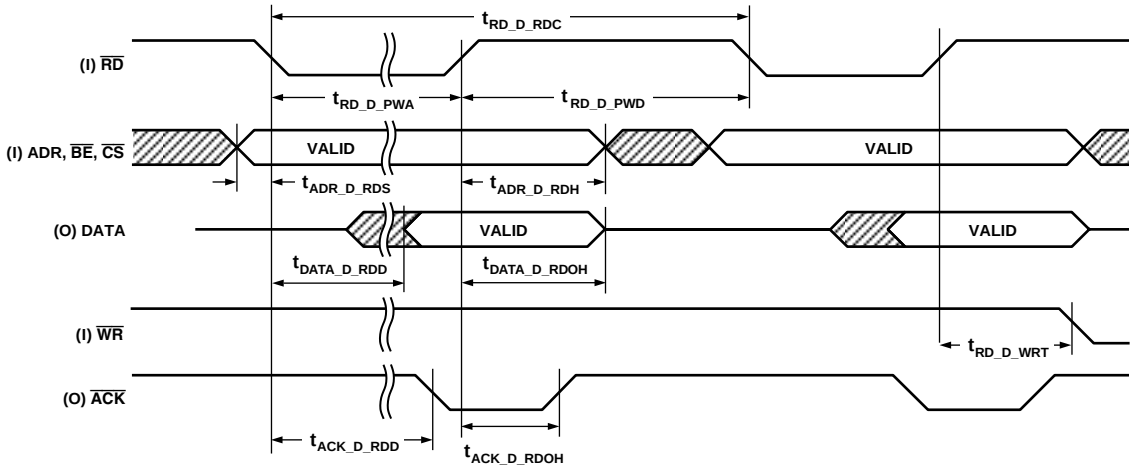


Figure 28. Host (Indirect Address, Indirect Register Data, and Interrupt Mask/Status) Read Transfer Timing

Table XXVIII. Host (Indirect Address, Indirect Data, and Interrupt Mask/Status) Write Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{WR\_D\_WRC}$	$\overline{WR}$ Signal, Direct Register, Write Cycle Time (at 27 MHz VCLK)	N/A[1]	N/A	ns
$t_{WR\_D\_PWA}$	$\overline{WR}$ Signal, Direct Register, Pulsewidth Asserted (at 27 MHz VCLK)	N/A[1]	N/A	ns
$t_{WR\_D\_PWD}$	$\overline{WR}$ Signal, Direct Register, Pulsewidth Deasserted (at 27 MHz VCLK)	5	N/A	ns
$t_{ADR\_D\_WRS}$	ADR Bus, Direct Register, Write Setup	2	N/A	ns
$t_{ADR\_D\_WRH}$	ADR Bus, Direct Register, Write Hold	2	N/A	ns
$t_{DATA\_D\_WRS}$	DATA Bus, Direct Register, Write Setup	–10	N/A	ns
$t_{DATA\_D\_WRH}$	DATA Bus, Direct Register, Write Hold	0	N/A	ns
$t_{WR\_D\_RDT}$	$\overline{WR}$ Signal, Direct Register, Read Turnaround (After a Write) (at 27 MHz VCLK)	35.6[2]	N/A	ns
$t_{ACK\_D\_WRD}$	$\overline{ACK}$ Signal, Direct Register, Write Delay (at 27 MHz VCLK)	8.6	182.1[3, 4]	ns
$t_{ACK\_D\_WROH}$	$\overline{ACK}$ Signal, Direct Register, Write Output Hold	11	N/A	ns

NOTES
[1]$\overline{WR}$ input must be asserted (low) until ACK is asserted (low).
[2]Minimum $t_{WR\_D\_RDT}$ varies with VCLK according to the formula: $t_{WR\_D\_RDT\ (MIN)}$ = 0.8 (VCLK Period) +7.4.
[3]Maximum $t_{WR\_D\_WRD}$ varies with VCLK according to the formula: $t_{ACK\_D\_WRD\ (MAX)}$ = 4.3 (VCLK Period) +14.8.
[4]During STATS_R deasserted (low) conditions, $t_{ACK\_D\_WRD}$ may be as long as 52 VCLK periods.

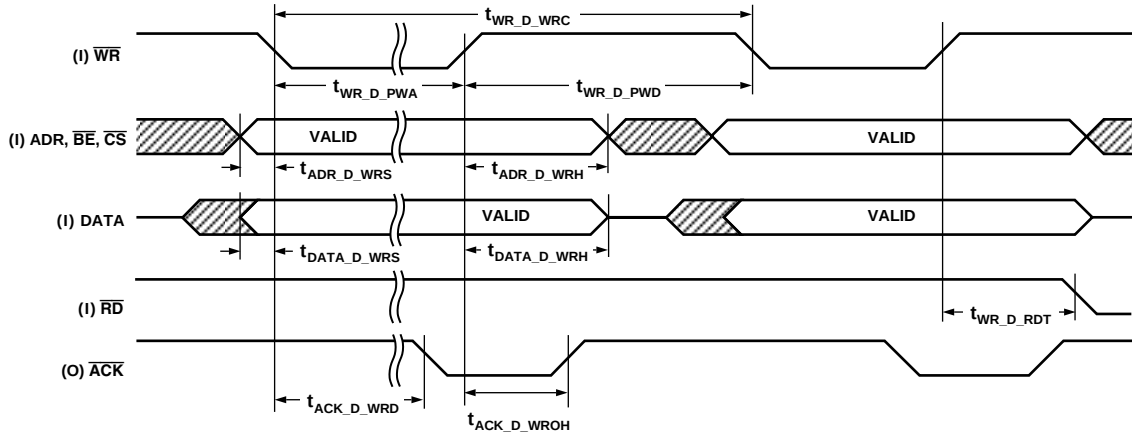


Figure 29. Host (Indirect Address, Indirect Register Data, and Interrupt Mask/Status) Write Transfer Timing

Host Interface (Compressed Data) Register Timing

The diagrams in this section show transfer timing for host read and write transfers to the ADV601LC's Compressed Data register. Accesses to the Compressed Data register are faster than access timing for the Indirect Address, Indirect Register Data, and Interrupt Mask/Status registers. For information on access timing for the other registers, see the Host Interface (Indirect Address, Indirect Register Data, and Interrupt Mask/Status) Register Timing section. Also note that as long as your system observes the $\overline{RD}$ or $\overline{WR}$ signal assertion timing, your system does *NOT* have to wait for the $\overline{ACK}$ signal between new compressed data addresses.

Table XXIX. Host (Compressed Data) Read Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{RD\_CD\_RDC}$	$\overline{RD}$ Signal, Compressed Data Direct Register, Read Cycle Time	28	N/A	ns
$t_{RD\_CD\_PWA}$	$\overline{RD}$ Signal, Compressed Data Direct Register, Pulsewidth Asserted	10	N/A	ns
$t_{RD\_CD\_PWD}$	$\overline{RD}$ Signal, Compressed Data Direct Register, Pulsewidth Deasserted	10	N/A	ns
$t_{ADR\_CD\_RDS}$	ADR Bus, Compressed Data Direct Register, Read Setup	2	N/A	ns
$t_{ADR\_CD\_RDH}$	ADR Bus, Compressed Data Direct Register, Read Hold (at 27 MHz VCLK)	2	N/A	ns
$t_{DATA\_CD\_RDD}$	DATA Bus, Compressed Data Direct Register, Read Delay	N/A	10	ns
$t_{DATA\_CD\_RDOH}$	DATA Bus, Compressed Data Direct Register, Read Output Hold	18	N/A	ns
$t_{ACK\_CD\_RDD}$	$\overline{ACK}$ Signal, Compressed Data Direct Register, Read Delay	N/A	18	ns
$t_{ACK\_CD\_RDOH}$	$\overline{ACK}$ Signal, Compressed Data Direct Register, Read Output Hold	9	N/A	ns

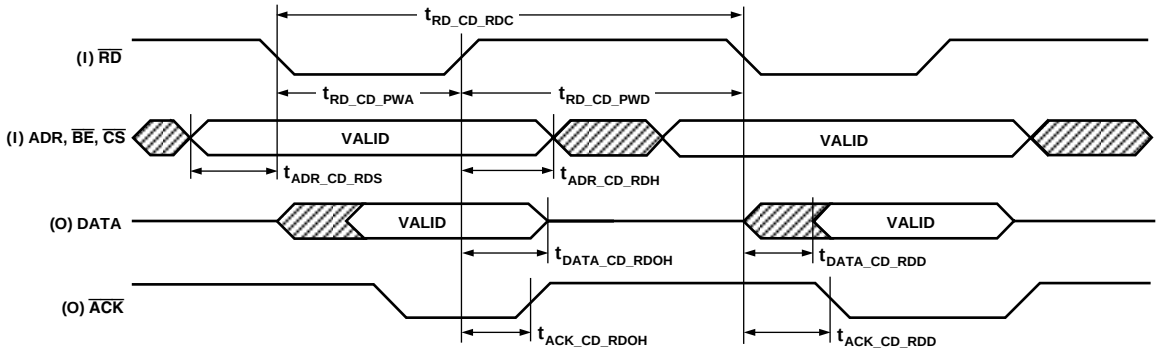


Figure 30. Host (Compressed Data) Read Transfer Timing

Table XXX. Host (Compressed Data) Write Timing Parameters

Parameter	Description	Min	Max	Unit
$t_{WR\_CD\_WRC}$	$\overline{WR}$ Signal, Compressed Data Direct Register, Write Cycle Time	28	N/A	ns
$t_{WR\_CD\_PWA}$	$\overline{WR}$ Signal, Compressed Data Direct Register, Pulsewidth Asserted	10	N/A	ns
$t_{WR\_CD\_PWD}$	$\overline{WR}$ Signal, Compressed Data Direct Register, Pulsewidth Deasserted	10	N/A	ns
$t_{ADR\_CD\_WRS}$	ADR Bus, Compressed Data Direct Register, Write Setup	2	N/A	ns
$t_{ADR\_CD\_WRH}$	ADR Bus, Compressed Data Direct Register, Write Hold	2	N/A	ns
$t_{DATA\_CD\_WRS}$	DATA Bus, Compressed Data Direct Register, Write Setup	2	N/A	ns
$t_{DATA\_CD\_WRH}$	DATA Bus, Compressed Data Direct Register, Write Hold	2	N/A	ns
$t_{ACK\_CD\_WRD}$	$\overline{ACK}$ Signal, Compressed Data Direct Register, Write Delay	N/A	19	ns
$t_{ACK\_CD\_WROH}$	$\overline{ACK}$ Signal, Compressed Data Direct Register, Write Output Hold	9	N/A	ns

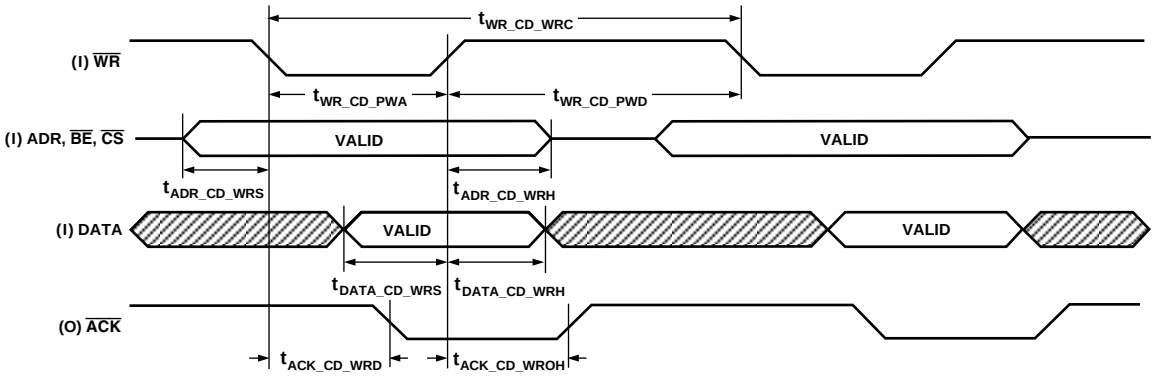


Figure 31. Host (Compressed Data) Write Transfer Timing

PINOUTS

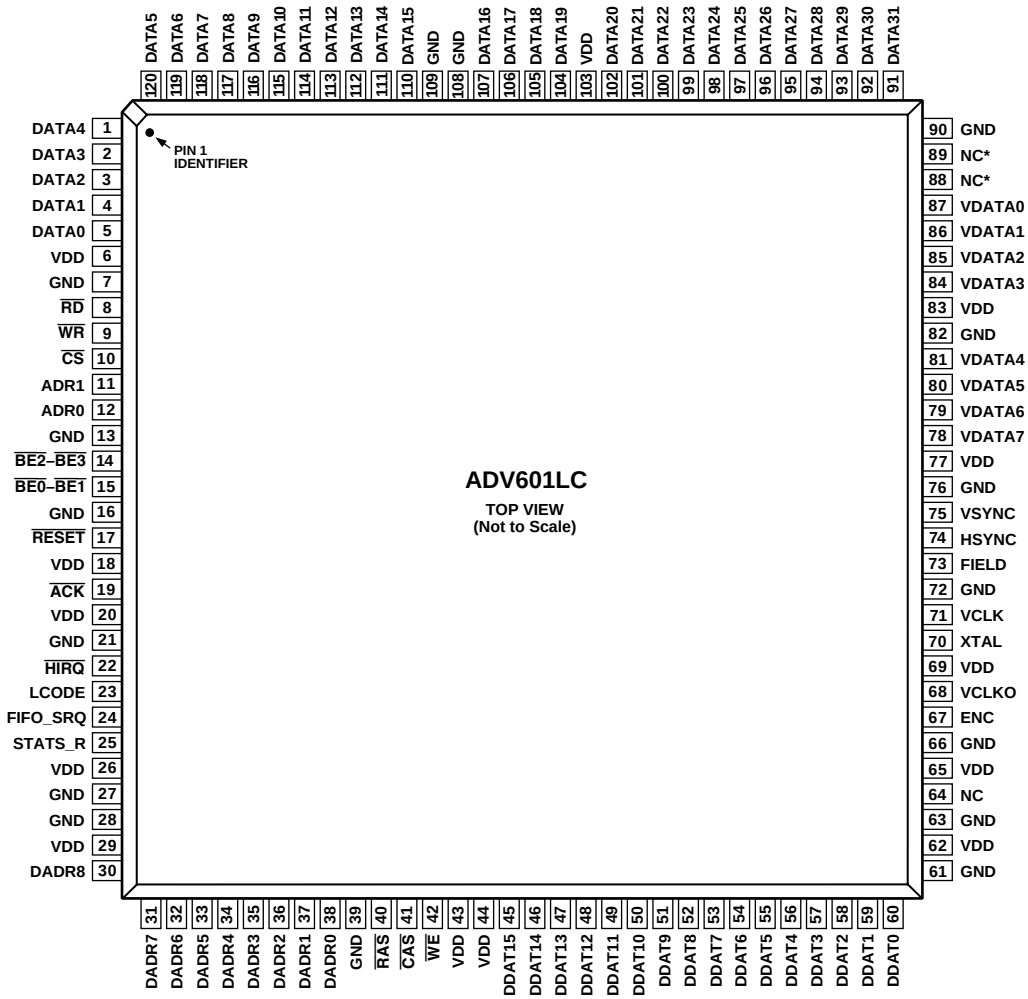
Pin	Pin Name	Pin Type
1	DATA4	I/O
2	DATA3	I/O
3	DATA2	I/O
4	DATA1	I/O
5	DATA0	I/O
6	VDD	POWER
7	GND	GROUND
8	$\overline{RD}$	I
9	$\overline{WR}$	I
10	$\overline{CS}$	I
11	ADR1	I
12	ADR0	I
13	GND	GROUND
14	$\overline{BE2}$–$\overline{BE3}$	I
15	$\overline{BE0}$–$\overline{BE1}$	I
16	GND	GROUND
17	$\overline{RESET}$	I
18	VDD	POWER
19	$\overline{ACK}$	O
20	VDD	POWER
21	GND	GROUND
22	$\overline{HIRQ}$	O
23	LCODE	O
24	FIFO_SRQ	O
25	STATS_R	O
26	VDD	POWER
27	GND	GROUND
28	GND	GROUND
29	VDD	POWER
30	DADR8	O
31	DADR7	O
32	DADR6	O
33	DADR5	O
34	DADR4	O
35	DADR3	O
36	DADR2	O
37	DADR1	O
38	DADR0	O
39	GND	GROUND
40	$\overline{RAS}$	O

Pin	Pin Name	Pin Type
41	$\overline{CAS}$	O
42	$\overline{WE}$	O
43	VDD	POWER
44	VDD	POWER
45	DDAT15	I/O
46	DDAT14	I/O
47	DDAT13	I/O
48	DDAT12	I/O
49	DDAT11	I/O
50	DDAT10	I/O
51	DDAT9	I/O
52	DDAT8	I/O
53	DDAT7	I/O
54	DDAT6	I/O
55	DDAT5	I/O
56	DDAT4	I/O
57	DDAT3	I/O
58	DDAT2	I/O
59	DDAT1	I/O
60	DDAT0	I/O
61	GND	GROUND
62	VDD	POWER
63	GND	GROUND
64	NC	NC
65	VDD	I/O
66	GND	GROUND
67	ENC	O
68	VCLKO	O
69	VDD	POWER
70	XTAL	I
71	VCLK	I
72	GND	GROUND
73	FIELD	I OR O
74	HSYNC	I OR O
75	VSYNC	I OR O
76	GND	GROUND
77	VDD	POWER
78	VDATA7	I/O
79	VDATA6	I/O
80	VDATA5	I/O

Pin	Pin Name	Pin Type
81	VDATA4	I/O
82	GND	GROUND
83	VDD	POWER
84	VDATA3	I/O
85	VDATA2	I/O
86	VDATA1	I/O
87	VDATA0	I/O
88	NC*	NC
89	NC*	NC
90	GND	GROUND
91	DATA31	I/O
92	DATA30	I/O
93	DATA29	I/O
94	DATA28	I/O
95	DATA27	I/O
96	DATA26	I/O
97	DATA25	I/O
98	DATA24	I/O
99	DATA23	I/O
100	DATA22	I/O
101	DATA21	I/O
102	DATA20	I/O
103	VDD	POWER
104	DATA19	I/O
105	DATA18	I/O
106	DATA17	I/O
107	DATA16	I/O
108	GND	GROUND
109	GND	GROUND
110	DATA15	I/O
111	DATA14	I/O
112	DATA13	I/O
113	DATA12	I/O
114	DATA11	I/O
115	DATA10	I/O
116	DATA9	I/O
117	DATA8	I/O
118	DATA7	I/O
119	DATA6	I/O
120	DATA5	I/O

*Apply a 10 kΩ pull-down resistor to this pin.

PIN CONFIGURATION

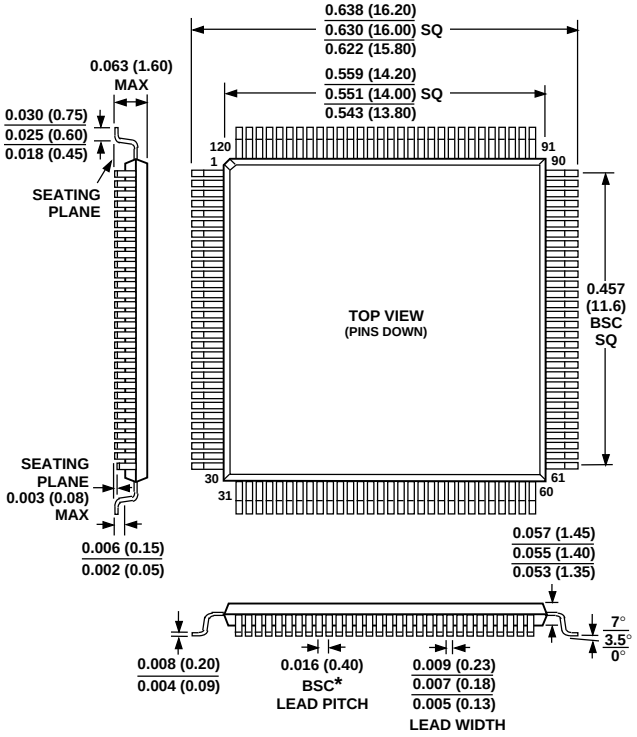


*APPLY A 10kΩ PULL DOWN RESISTOR TO THIS PIN

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

120-Lead LQFP
(ST-120)



* THE ACTUAL POSITION OF EACH LEAD IS WITHIN 0.003 (0.07) FROM ITS IDEAL POSITION WHEN MEASURED IN THE LATERAL DIRECTION.
CENTER FIGURES ARE TYPICAL UNLESS OTHERWISE NOTED

ORDERING GUIDE

Part Number	Ambient Temperature Range[1]	Package Description	Package Option[2]
ADV601LCJST	0°C to +70°C	120-Lead LQFP	ST-120

NOTES
[1]J = Commercial temperature range (0°C to +70°C).
[2]ST = Plastic Thin Quad Flatpack.

C3164–3–1/99

PRINTED IN U.S.A.