SiI9034/9134 HDMI Transmitter

Programmer's Reference

Document # SiI-PR-0039-F

May 2008

Copyright Notice

Trademark Acknowledgment

Further Information

To request other materials, documentation, and information, contact your local Silicon Image, Inc. sales office or visit the Silicon Image, Inc. web site at www.siliconimage.com.

Revision History

Revision	Date	Comment
A	12/2006	First production draft
B	03/2007	Updated Data Control and HDMI Control Registers
C	04/2007	Updated DDC I^2C Status Register Information
D	08/2007	Updated Device Revision and Ri Command registers
E	12/2007	Clarified and corrected content throughout, multi-word tables broken up, editorial cleanup
F	5/2008	Added appendix for PLL set up

Table of Contents

List of Figures

List of Tables

Introduction

This document provides information about the SiI9034/9134 HDMI Transmitter so that system designers and programmers can implement the firmware and software necessary to control the device's features in a system environment.

The SiI9134 transmitter has capabilities that the SiI9034 device does not, including deep color, 14-to-8/10/12-dither, and high bit-rate audio support. Therefore, the SiI9134 register set is a superset of the SiI9034 register set. Where differences occur, the registers are explicitly called out for each of the parts.

Figure 1 shows the path along which the transmitter processes outgoing video data.

Register Maps

The registers in this document are described in groups according to function. Certain registers in each address range are reserved for future use. Detailed definitions about the reserved registers are not provided in this document.

Register addresses range from 0x00 to 0xFF on each page in the I^2C protocol. Because there are more than 255 bytes of registers in the transmitter, the device is accessible at one of two I^2C device addresses. The device address may be altered with the CI2CA pin. The level on the CI2CA pin is not latched internally and must *not* be changed during any active I^2C operations. All references to device address in this document use the default values of 0x72 and 0x7A.

Table 1 shows how the CI2CA pin state corresponds to device addresses. Table 2 provides an overview of the register address groups.

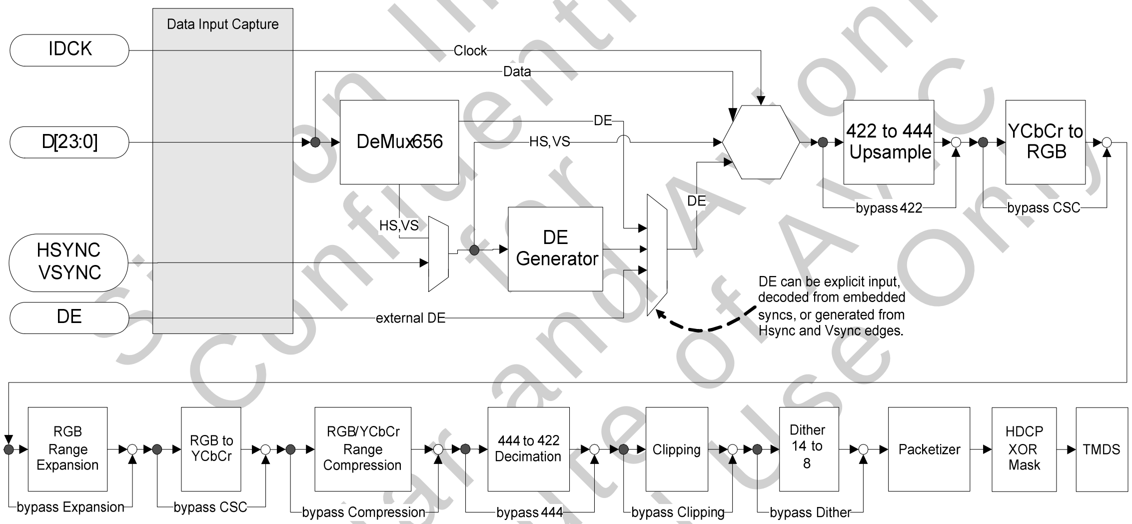

Figure 1. Transmitter Video Data Processing Path

Table 1. Control of I²C Address with CI2CA Pin

Device Address	CI2CA = HIGH	CI2CA = LOW
First Device Addr	0x76	0x72
Second Device Addr	0x7E	0x7A

Table 2. Register Address Groups

I²C Address	Address Range	Group Name	Purpose	Page
0x72	0x00-0x0E	Base	Device identification and general programming	3
	0x0F-0x2B	HDCP	HDCP authentication and other processes	6
	0x32-0x4D	Video	DE, sync decoder and encoder	11
	0x70-0x7F	Interrupt	Interrupt processing	21
	0x80-0xB2	TMDS	TMDS control	26
	0xEC-0xFF	DDC	Mastering DDC bus	29
	0xF8-0xFF	ROM	Status of HDCP keys in ROM	33
0x7A	0x00-0x3D	Audio	Audio features and translations	35
	0x3E-0xFE	CEA-861D	Support for InfoFrame packets	50

Important: Do not use I²C to write to register addresses that are not described in this document. Modifications to undocumented registers can cause unintended errors in the chip function.

Document Conventions

Bit N	Bits are numbered in little-endian format: the least-significant bit of a byte or word is referred to as bit 0.
0xNN	Hexadecimal representation of base-16 numbers is represented using C language notation, preceded by 0x.
0bNN	Binary (base-2) numbers are represented using C language notation, preceded by 0b.
NN	Decimal (base-10) numbers are represented using no additional prefixes or suffixes.
	Reserved register bits are shaded in the register description.
RSVD0	A bit in a register that is reserved and read-only, and returns a zero value.
RSVD1	A bit in a register that is reserved and read-only, and returns a one value.
RSVD	A bit in a register that is reserved and read-only, and returns an indeterminate value.
RSVDRW	A bit in a register that is reserved and read-write, returning the value written to it.
	RSVDRW0 implies a default of 0 and RSVDRW1 implies a default of 1.
	RSVDRW implies a default of 0 unless other specified.
X	A register bit defaulting to X has no defined state after hardware reset.

Base Register Set

Vendor ID Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x00	VND_IDL	Vendor ID Low Byte							
0x72	0x01	VND_IDH	Vendor ID High Byte							

Bit	Label	R/W	Description	Default	
				Low byte	**High byte**
15:0	VND_ID	R	Provides unique vendor identification through I^2C.	0x01	0x00

Device ID Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x02	DEV_IDL	Device ID Low Byte							
0x72	0x03	DEV_IDH	Device ID High Byte							

Bit	Label	R/W	Description	Default	
				Low byte	**High byte**
15:0	DEV_ID	R	Provides unique device type identification through I^2C.	0x34	9034: 0x90 9134: 0x91

Device Revision Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x04	DEV_REV	Device Revision Byte							

Bit	Label	R/W	Description	Default
7:0	DEV_REV	R	Identifies different revisions of same device.	0x01

Software Reset Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x05	SRST	RSVD0						FIFORST	SWRST

Bit	Label	R/W	Description	Default
1	FIFORST	RW	Audio FIFO reset: 0 = Normal operation 1 = Reset (flush) audio FIFO	0
0	SWRST	R/W	Software reset: 0 = Normal operation 1 = Reset all sections, including the audio FIFO, except registers that are user configurable.	0

System Control Register #1

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x08	SYS_CTRL1	RSVD0	VSYNC	VEN	HEN	RSVDRW0	BSEL	EDGE	PD#

Bit	Label	R/W	Description	Default
6	VSYNC	R	The current status of the VSYNC input pin. Refer to the INTR2 register (0x72:0x72), described on page 22, for an interrupt tied to VSYNC active edge.	X
5	VEN	R/W	VSYNC enable: 0 = Fixed LOW 1 = Follow VSYNC input	1
4	HEN	R/W	HSYNC enable: 0 = Fixed LOW 1 = Follow HSYNC input	1
2	BSEL	R/W	Input bus select: 0 = 12-bit 1 = 24-bit	1
1	EDGE	R/W	Edge select: 0 = Latch input on falling edge 1 = Latch input on rising edge	0
0	PD#	R/W	Power down mode: HIGH is normal operation. When LOW, the TMDS core is powered down and interrupts are in power-down mode. Most other register values are not affected by assertion of the PD# bit. For exceptions, see page 123.	0

System Status Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x09	SYS_STAT	VLOW	RSVD0				RSEN	HPD	P_STABLE

Bit	Label	R/W	Description	Default
7	VLOW	R	VREF mode. Always HIGH.	1
2	RSEN	R	Receiver Sense (works in DC-coupled systems only): 0 = no receiver connected 1 = receiver is connected and powered on	X
1	HPD	R	Hot Plug Detect. Provides the state of the Hot Plug Detect pin.	X
0	P_STABLE	R	IDCK to TMDS clock is stable and the Transmitter can send reliable data on the TMDS link. A change to the IDCK sets this bit LOW. After a subsequent LOW to HIGH transition, indicating a stable input clock, Silicon Image recommends performing a software reset.	0

RSEN is active when the TMDS link is terminated, usually into a powered-on TMDS receiver chip. An active RSEN implies an active HPD, because the link must also be physically connected.

The *HDCP Specification* defines a *Repeater* device and a protocol for such a device to notify the source of any change in the connection status of any downstream HDCP link. Part of this protocol toggles the Hot Plug signal whenever a downstream device is attached or detached.

Legacy Registers

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x0A	SYS_CTRL3	RSVD0					CTL		RSVD0
0x72	0x0B	LEGACY1	RSVD0							
0x72	0x0E	LEGACY3	RSVD0							

Bit	Label	R/W	Description	Default
2:1	CTL	R/W	The states of these control bits are transmitted across the TMDS link during blanking times for DVI 1.0 mode only.	0b00

System Control Register #4

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x0C	SYS_CTRL4	RSVDRW1	RSVD0		PLLF				PFEN

Bit	Label	R/W	Description	Default
4:1	PLLF	R/W	Specifies the PLL filter charge pump current: 0b0000 = 5 µA 0b0001 = 10 µA 0b0010 = 15 µA 0b0100 = 25 µA 0b0111 = 40 µA 0b1000 = 45 µA 0b1111 = 80 µA	0b0001
0	PFEN	R/W	0 = Disable PLL filter 1 = Enable PLL filter	1

Data Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x0D	DCTL	RSVD0			RSVD0		VID_BLANK	AUD_MUTE	RSVD0

Bit	Label	R/W	Description	Default
2	VID_BLANK	R/W	0 = Normal operation (video output is not blanked) 1 = Video output is blanked and the colors sent are those specified in registers 0x72:0x4B-0x4D, described on page 20.	0
1	AUD_MUTE	R/W	0 = Do not send zeros in audio packet 1 = Send zeros in audio packet	0

Table 3. Audio/Video Mute Settings

VID_BLANK	AUD_MUTE	SET_AVMUTE*	HDMI Transmitter Sends…
x	x	1	All zeros in audio packets and blank-level data in all video packets
0	1	0	All zeros in audio packets and real video in video packets
1	0	0	Blank-level data in all video packets and real audio in audio packets
0	0	0	Real video in video packet and real audio in audio packets

***Note:** Described in register 0x7A:0xDF, bit [0], on page 58.

HDCP Register Set

All multi-byte registers (such as AKSV) should be written to hardware in order from the least-significant byte to the most-significant byte. For AKSV and BKSV, the action of writing a value to the most-significant byte triggers an HDCP operation

Important: An active link clock is required to read back valid data from the HDCP registers on the E-DDC bus.

HDCP Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x0F	HDCP_CTRL	RSVDRW0	ENC_ ON	BKSV_ ERR	RX_ RPTR	TX_ ANSTOP	CP_ RESTN	RI_ RDY	ENC_ EN

Bit	Label	R/W	Description	Default
6	ENC_ON	R	Encryption status: 0 = Encryption disabled or not in progress 1 = Encryption enabled and in progress	0
5	BKSV_ERR	R	BKSV error: 0 = No error in BKSV format 1 = Error in BKSV format To clear this bit, the firmware must first set the TX_ANSTOP bit, and then perform an authentication twice with a valid BKSV value.	0
4	RX_RPTR	R/W	Repeater: 0 = Single HDMI receiver 1 = HDMI receiver is a repeater If the HDMI receiver is in a repeater, write this bit to 1 before beginning the authentication process. **Note**: This bit is written when the source device detects an attached HDCP repeater. This step is necessary for the computation of shared values in the HDCP protocol. Refer to the *HDCP Specification*.	0
3	TX_ANSTOP	R/W	AN Control. When cleared, the cipher engine runs free and the AN registers cycle through pseudo-random values. When set, the cipher engine stops and the HDCP-capable receiver can read and initialize the AN register. **Important:** To set this bit to 1, a 1 must be written to this bit two times consecutively. To clear this bit, toggle the RX_RPTR bit. This bit is also cleared when the hardware is reset or BKSV_ERR is set.	0
2	CP_RESTN	R/W	Content protection reset. 0 = Reset 1 = Normal operation	0
1	RI_RDY	R	R_i Ready. 1 = R_i first value is ready in the HDMI transmitter A hardware reset clears this bit. This bit is also cleared when the first byte of BKSV is written into the HDMI transmitter (performed at the beginning of the next authentication process).	0
0	ENC_EN	R/W	0 = Encryption disabled 1 = Encryption enabled This bit can be written to 0 or 1. A 1-to-0 transition of this bit triggers the HDCP encryption logic and sets an interrupt bit. See register 0x72:0x72[5] on page 23.	0

HDCP BKSV Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x10	BKSV1	BKSV1							
0x72	0x11	BKSV2	BKSV2							
0x72	0x12	BKSV3	BKSV3							
0x72	0x13	BKSV4	BKSV4							
0x72	0x14	BKSV5	BKSV5							

Bit	Label	R/W	Description	Default
39:0	BKSV	W	Written with the HDCP Receiver Key Selection Vector register value. Writing byte 5 triggers the authentication logic in the HDMI transmitter. Write this byte last.	0
		R	Value of the BKSV register.	

HDCP AN Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x15	AN1	AN1							
0x72	0x16	AN2	AN2							
0x72	0x17	AN3	AN3							
0x72	0x18	AN4	AN4							
0x72	0x19	AN5	AN5							
0x72	0x1A	AN6	AN6							
0x72	0x1B	AN7	AN7							
0x72	0x1C	AN8	AN8							

Bit	Label	R/W	Description	Default
63:0	AN	R/W	AN is an HDCP 64-bit pseudo-random value.	0

HDCP AKSV Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x1D	AKSV1	AKSV1							
0x72	0x1E	AKSV2	AKSV2							
0x72	0x1F	AKSV3	AKSV3							
0x72	0x20	AKSV4	AKSV4							
0x72	0x21	AKSV5	AKSV5							

Bit	Label	R/W	Description	Default
39:0	AKSV	R	HDCP-capable Transmitter Key Selection Vector. Byte 5 triggers the authentication logic in the receiver. Write this byte last.	0

HDCP Ri Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x22	RI1	RI1							
0x72	0x23	RI2	RI2							

Bit	Label	R/W	Description	Default
15:0	RI	R	R_i Register. The value of this register must be read and compared with the R_i' value from the HDMI receiver.	0

HDCP Ri 128 Compare Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x24	Ri_128_COMP	RSVDRW0	RI_128_COMP						

Bit	Label	R/W	Description	Default
6:0	RI_128_COMP	R/W	Limit counter for R_i comparison. When the frame counter (I_CNT) reaches the index set in this register, the transmitter generates an RI_128 interrupt.	0

The HDCP R_i value is updated during the vertical blanking interval for frame[j] when *j mod 128 = 0*. Refer to the *HDCP Specification*. The interrupt RI_128 in INTR1 is generated in the frame with the index that matches RI_128_COMP. RI_128_COMP defaults to generate the interrupt when j = 0. For example, setting RI_128_COMP to 0x7F creates the interrupt one frame before R_i is updated in the HDMI transmitter and receiver.

HDCP I Counter Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x25	I_CNT	RSVDR0	I_CNT						

Bit	Label	R/W	Description	Default
6:0	I_CNT	R	Current value of I counter. This register counts frames from 0 to 127, and then rolls over to zero.	0

Automatic Ri Check

The auto-synchronous R_i check compares the R_i value of the transmitter to the R_i' value of the receiver on the 0^{th} and 127^{th} frame. When reading the R_i' value of the receiver, the automatic R_i check uses the DDC output port. Use bit 0 in the RI_STAT register (0x72:0x26), described in the following table, for proper handshaking between the automatic R_i check and the master DDC functionality. The automatic R_i check is disabled by default. To enable automatic R_i check, set the Ri_EN bit to 1 in the RI_CMD register (0x72:0x27). Interrupts can be configured to trigger on an HDMI transmitter R_i and receiver R_i' mismatch condition.

Using the R_i 128 method is not required if the automatic R_i check is enabled.

Important: Enable the automatic R_i check only after the first stage of HDCP authentication is complete.

Ri Status Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x26	RI_STAT	RSVD0							Ri_ STARTED

Bit	Label	R/W	Description	Default
0	Ri_STARTED	R	R_i Check Started Status. This signal is used for handshaking between the firmware and the hardware. After the R_i check is enabled, the hardware waits for the DDC master to finish the current transaction before taking control. After this bit is set, the firmware loses the ability to use the DDC master unless it disables R_i check and this bit resets to 0.	0

Ri Command Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x27	RI_CMD	RSVD0						BCAP_EN	Ri_EN

Bit	Label	R/W	Description	Default
1	BCAP_EN	R/W	Enable polling of the BCAP_DONE bit (0x72:0x72[7]). 0 = Disable 1 = Enable **Note**: To poll the BCAP_DONE bit, the ENC_EN (0x72:0x0F[0]) bit and the Ri_EN bit must be enabled on the HDMI transmitter.	0
0	Ri_EN	R/W	Enable automatic R_i Check. Check bit 0 of the Ri_STAT register (0x72:0x26) for firmware and hardware DDC control handshaking. 0 = Disable 1 = Enable **Note**: Automatic R_i check is not affected by SET_AVMUTE (described on page 58). The HDMI transmitter R_i counter does not advance during the time that the HDCP device is in the AVMUTE state, and does not resume advancing after the AVMUTE state until the first encrypted frame.	0

Ri Line Start Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x28	RI_START	Ri_LINE_START							

Bit	Label	R/W	Description	Default
7:0	Ri_LINE_START	R/W	Indicates at what line within frame 127 or 0 to start the R_i check. **Note**: The value for this register bit represents the power of 2; 2 LSB is 0.	0x04

Ri From RX Registers

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x29	RI_RX_L	Ri_RX[7:0]							
0x72	0x2A	RI_RX_H	Ri_RX[15:8]							

Bit	Label	R/W	Description	Default
15:0	Ri_RX	R	This value represents the HDMI receiver R_i' value if any of the R_i check errors occurred.	0

Ri Debug Registers

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x2B	RI_DEBUG	RI_DBG_ TRASH	RI_DBG_ HOLD	RSVDX					

Bit	Label	R/W	Description	Default
7	RI_DBG_TRASH	R/W	0 = Continue with regular updates to R_i 1 = Force a corruption of the R_i values	0
6	RI_DBG_HOLD	R/W	0 = Continue with regular updates to R_i 1 = Hold the R_i value steady, stop updating	0

DE Generator Register Set

The HDMI transmitter provides an internal Data Enable (DE) generator for use when the attached video source does not provide a DE signal with the other video signals. The DE signal is needed for encoding the TMDS output of the HDMI transmitter. A DE signal is generated based on the values in the DE generator registers and the arrival times of the HSYNC and VSYNC pulses from the video source. This DE signal is included in the control data sent to the HDMI receiver across the link. Refer to the *SiI9034/9134 HDMI Transmitter Data Sheet* (SiI-DS-0189) for more details.

The DE generator registers are used only for DE generation. Figure 2 shows the registers diagrammatically. The vertical sync pulse (VSYNC) occurs in the top area of the frame, before the active video area. The active (leading) edge is shown with an arrow. The horizontal sync pulse (HSYNC) occurs in every line before the active video area during the DE_DLY time. The active (leading) edge of HSYNC is shown with an arrow.

Note: The VSYNC and HSYNC widths are not shown to scale.

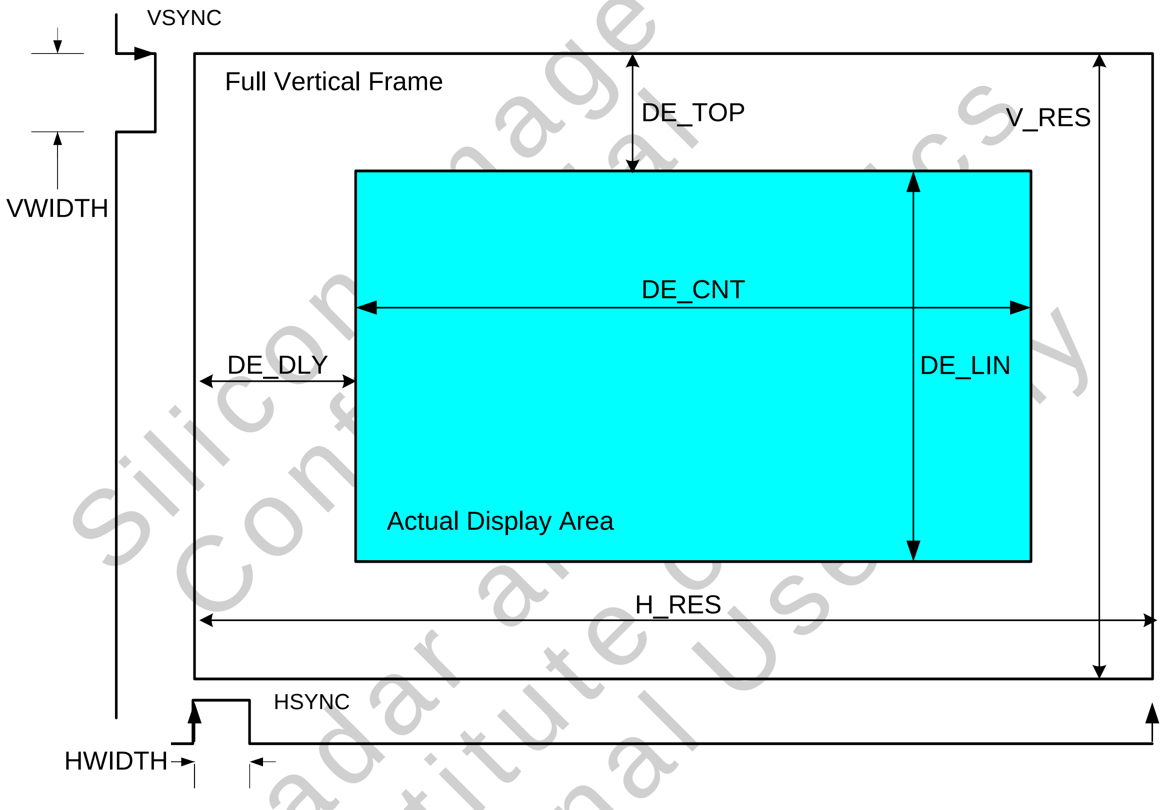

Figure 2. DE Generator Measurements

In the following definitions, *pixels* mean unique pixels. The counts in the DE generator registers, if expressed in pixels, are counted according to the original input clock even if that original input clock is multiplied within the chip. For example, a 480i field contains 720 unique pixels per line in the active video area, even when the clock is multiplied to 1440 clock cycles per active video time.

Register 0x72:0x3F records the detected polarity of VSYNC and HSYNC. The output polarities are set by register 0x72:0x33.

Note: When using the Sync Decoding module (see page 14), the DE input signal should be tied LOW.

Video DE Delay Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x32	DE_DLY	DE_DLY[7:0]							

Bit	Label	R/W	Description	Default
7:0	DE_DLY[7:0]	R/W	Width of the area to the left of the active display. The unit of measure is pixels. This register should be set to the sum of (HSYNC width) + (horizontal back porch) + (horizontal left border), and is used only for DE generation. **Note**: This 12-bit value includes four bits from register 0x72:0x33. The valid range is 1–4095. 0 is an invalid value.	0x00

Video DE Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x33	DE_CTRL	RSVD0	DE_GEN	VS_POL#	HS_POL#	DE_DLY[11:8]			

Bit	Label	R/W	Description	Default
6	DE_GEN	R/W	Generate DE signal. 0 = Disable 1 = Enable	0
5	VS_POL#	R/W	VSYNC polarity. 0 = Positive polarity (leading edge rises) 1 = Negative polarity (leading edge falls) Set this bit to the input VSYNC polarity for the source that provides VSYNC. For embedded syncs, set this bit to the desired VSYNC polarity that is generated from the embedded sync codes.	0
4	HS_POL#	R/W	HSYNC polarity. 0 = Positive polarity (leading edge rises) 1 = Negative polarity (leading edge falls) Set this bit to the input HSYNC polarity for the source that provides HSYNC. For embedded syncs, set this bit to the desired HSYNC polarity that is generated from the embedded sync codes.	0
3:0	DE_DLY[11:8]	R/W	Bits 11:8 of the DE_DLY value (refer to the Video DE Delay Register section).	0b0000

Note: If both DE signal generation and sync decoding are enabled, VS_POL# and HS_POL# define the polarities of VSYNC and HSYNC from the sync decoder. The DE generator can be used in combination with sync decoding because the encoded syncs do not carry polarity information. To set the transmitted HSYNC and VSYNC to HDMI-compliant states, the DE generator may be needed. Refer to register 0x72:0x4A (in the Video Mode Register (SiI9034) or the Video Mode Register (SiI9134) sections).

Video DE Top Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x34	DE_TOP	RSVD0	DE_TOP						

Bit	Label	R/W	Description	Default
6:0	DE_TOP	R/W	Defines the height of the area above the active display. The unit of measure is lines (HSYNC pulses). This register should be set to the sum of (VSYNC width) + (vertical back porch) + (vertical top border). The valid range is 1–127. 0 is an invalid value.	0b0000000

Video DE Count Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x36	DE_CNTL	DE_CNT[7:0]							
0x72	0x37	DE_CNTH	RSVD0				DE_CNT[11:8]			

Bit	Label	R/W	Description	Default
11:0	DE_CNT	R/W	Defines the width of the active display. The unit of measure is pixels. Set this register to the desired horizontal resolution. The valid range is 1–4095. 0 is an invalid value.	0

Note: Values measured in pixels (DE_CNT, and so on) count the total number of unique pixels on a line. If the input clock is a multiple of the pixel rate (see the DEMUX bit, described on page 18 and the ICLK bit, described on page 16), the registers indicate pixel count, which is not the same as clock count.

Video DE Line Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x38	DE_LINL	DE_LIN[7:0]							
0x72	0x39	DE_LINH	RSVD0					DE_LIN[10:8]		

Bit	Label	R/W	Description	Default
10:0	DE_LIN	R/W	Defines the height of the active display. The unit of measure is lines (HSYNC pulses). Set this register to the desired vertical resolution. For interlaced modes, set this register to the number of lines per field, which is half the overall vertical resolution. The valid range is 1–2047. 0 is an invalid value.	0

Video H Resolution Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x3A	HRES_L	H_RES[7:0]							
0x72	0x3B	HRES_H	RSVD0			H_RES[12:8]				

Bit	Label	R/W	Description	Default
12:0	H_RES	R	Measures the time between two HSYNC active edges. The unit of measure is pixels.	0

Video V Refresh Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x3C	VRES_L	V_RES[7:0]							
0x72	0x3D	VRES_H	RSVD0					V_RES[10:8]		

Bit	Label	R/W	Description	Default
10:0	V_RES	R	Measures the time between two VSYNC active edges. The unit of measure is lines.	0

The values in the DE generator read-only registers are maintained until input HSYNC and VSYNC pulses are stopped. If an input IDCK continues, the DE generator counters overflow and the registers store zero until HSYNC and VSYNC are active again. The values in these registers are accurate only when there are active HSYNC and VSYNC inputs, or in the case of embedded sync input, active SAV/EAV sequences.

Video Embedded Sync Decoding Registers

When decoding syncs from the embedded sync stream (refer to register 0x72:0x4A in the Video Mode Register (SiI9034) or Video Mode Register (SiI9134) sections), disable the DE generator block (refer to register 0x72:0x33 in the Video DE Control Register section). Also, the DE input signal should be tied LOW (refer to the Handling Interlaced Video section).

Video Interlace Adjustment Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x3E	IADJUST	RSVD0					DE_ADJ#	F2VADJ	F2VOFST

Bit	Label	R/W	Description	Default
2	DE_ADJ#	R/W	0 = Enable VSYNC adjustment 1 = Disable VSYNC adjustment Setting this bit HIGH disables VSYNC adjustments and sets the DE generator to be more compatible with existing transmitters. Clearing this bit enables detection circuits to locate the position of VSYNC relative to HSYNC and only include HSYNC edges that are greater than ¾ lines from VSYNC in the line count for DE_TOP. **Note:** Silicon Image recommends that this bit always be set to 0.	1
1	F2VADJ	R/W	If this bit is set, the VBIT_TO_VSYNC value (register 0x72:0x46) is adjusted during field 2 of an interlace frame according to the setting of the F2VOFST bit. This bit defaults to 0.	0
0	F2VOFST	R/W	If F2VADJ is set and this bit is cleared, VBIT_TO_VSYNC (register 0x72:0x46) is decremented by one during field 2 of an interlace frame. If F2VADJ and this bit are both set, VBIT_TO_VSYNC (register 0x72:0x46) is incremented by one during field 2 of an interlace frame.	0

Video SYNC Polarity Detection Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x3F	POL_DETECT	RSVD0					I_DET	VPOL_DET#	HPOL_DET#

Bit	Label	R/W	Description	Default
2	I_DET	R	Interlace detect. 0 = Non-interlaced video 1 = Interlaced video This bit is set by checking for a varying VSYNC timing characteristic of interlaced modes.	0
1	VPOL_DET#	R	Detected input VSYNC polarity, using internal circuit. 0 = Active HIGH (leading edge rises) 1 = Active LOW (leading edge falls)	0
0	HPOL_DET#	R	Detected input HSYNC polarity, using internal circuit. 0 = Active HIGH (leading edge rises) 1 = Active LOW (leading edge falls)	0

Video Hbit to HSYNC Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x40	HBIT_2HSYNC1	HBIT_TO_HSYNC[7:0]							
0x72	0x41	HBIT_2HSYNC2	RSVD0						HBIT_TO_HSYNC[9:8]	

Bit	Label	R/W	Description	Default
9:0	HBIT_TO_HSYNC	R/W	Creates HSYNC pulses. Set this register to the delay from the detection of an EAV sequence (H bit change from 1 to 0) to the active edge of HSYNC. The unit of measure is pixels. The valid range is 1–1023. 0 is an invalid value.	0

Note: Registers 0x72:0x40 and 0x72:0x41 are useful only when the input video uses 656 encoded syncs.

Video Field2 HSYNC Offset Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x42	FLD2_HS_OFSTL	FIELD2_OFST[7:0]							
0x72	0x43	FLD2_HS_OFSTH	RSVD0				FIELD2_OFST[11:8]			

Bit	Label	R/W	Description	Default
11:0	FIELD2_OFST	R/W	Determines VSYNC pixel offset for the odd field of an interlaced source. Set this register to half the number of pixels/line. The valid range is 1–4095. 0 is an invalid value.	0

Video HSYNC Length Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x44	HWIDTH1	HWIDTH[7:0]							
0x72	0x45	HWIDTH2	RSVD0						HWIDTH[9:8]	

Bit	Label	R/W	Description	Default
9:0	HWIDTH	R/W	Sets the width of the HSYNC pulses. Set this register to the desired HSYNC pulse width. The unit of measure is pixels. The valid range is 1–1023. 0 is an invalid value.	0

Video Vbit to VSYNC Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x46	VBIT_TO_VSYNC	RSVD0		VBIT_TO_VSYNC					

Bit	Label	R/W	Description	Default
5:0	VBIT_TO_VSYNC	R/W	Sets the delay from the detection of V bit changing from 1 to 0 in an EAV sequence, to the asserting edge of VSYNC. The unit of measure is lines. The valid range is 1–63. 0 is an invalid value.	0b000000

Note: Registers 0x72:0x42 through 0x46 are useful only when the input video uses 656 encoded syncs.

Video VSYNC Length Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x47	VWIDTH	RSVD0		VWIDTH					

Bit	Label	R/W	Description	Default
5:0	VWIDTH	R/W	Sets the width of VSYNC pulse. The unit of measure is lines. The valid range is 1–63. 0 is an invalid value.	0b000000

Figure 2 on page 11 shows the HWIDTH and VWIDTH dimensions relative to the complete frame time.

Video Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x48	VID_CTRL	IFPOL	RSVDRW	EXTN	CSCSEL	RSVD0		ICLK	

Bit	Label	R/W	Description	Default
7	IFPOL	R/W	Invert field polarity. 0 = Do not invert field bit 1 = Invert field bit This bit is used when the 656 Flag bit is opposite the standard polarity for Field1 and Field2. Inverting the field polarity causes the sync extraction to format HSYNC and VSYNC properly based on the F bit. In embedded sync mode, the HDMI transmitter does not detect *even* from *odd* field, except based on the setting of the F bit. With explicit syncs, the HDMI transmitter encodes HSYNC and VSYNC across the HDMI/TMDS link regardless of field sequence.	0
6	RSVDRW	R/W	Do not write this bit to 1.	0
5	EXTN	R/W	Extended Bit mode. 0 = All 8-bit input modes 1 = All 12-bit 4:2:2 input modes For 4:2:2 inputs wider than 8 bits but less than 12 bits, the unused bits should be set to 0.	0
4	CSCSEL	R/W	Color Space Conversion Standard select. 0 = BT.601 conversion 1 = BT.709 conversion	0
1:0	ICLK	R/W	Clock mode. 0b00 = Pixel data is not replicated 0b01 = Each pixel is sent twice 0b10 = RSVD 0b11 = Each pixel is sent four times **Note**: If the DEMUX bit in the VID_MODE register (0x72:0x4A[1]) is set to 0, set ICLK and the pixel replication field of the AVI v2 data byte 5 to the same value. If the DEMUX bit is set to 1, set the pixel replication field of the AVI v2 data byte 5 to the next higher pixel replication rate. For example, if DEMUX = 1 and ICLK = 0b01, set the pixel replication field of AVI v2 data byte 5 to 0b11. Refer to page 26 for examples on programming ICLK, TCLKSEL, and the pixel replication field of the AVI v2 data byte 5 for various video and audio modes.	0b00

Video Action Enable Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x49	VID_ACEN	WIDE_BUS		RSVD0	CLIP_ CS_ID	RANGE_ CLIP	RGB_2_ YCBCR	RANGE_ CMPS	DOWN_ SMPL

Bit	Label	R/W	Description	Default
7:6	WIDE_BUS	R/W	Identifies the number of bits per input video channel: 0b00 = 8 bits per channel or 24-bit bus mode 0b01 = 10 bits per channel or 30-bit bus mode 0b10 = 12 bits per channel or 36-bit bus mode 0b11 = Reserved	0b00
4	CLIP_CS_ID	R/W	Identifies the output color space on the link - used by the clipper block to determine which way to clip: 0 = Output color space is RGB 1 = Output color space is YCbCr	0
3	RANGE_CLIP	R/W	Enable range clip. 0 = Disable 1 = Enable When range clip is enabled, the range of possible values for RGB and Y is 16 to 235, and for CbCr the range of values is 16 to 240. Actual values outside of these ranges are clipped to the associated lower (16) or upper (235 or 240) limits.	0
2	RGB_2_YCBCR	R/W	Enable RGB to YCbCr color-space converter. 0 = Disable 1 = Enable	0
1	RANGE_CMPS	R/W	Enable range compression. 0 = Disable 1 = Enable When range compression is enabled, the range of possible values for RGB and Y is 16 to 235, and for CbCr the range of values is 16 to 240. All possible values from 0 to 255 are compressed (remapped) so that they are all represented within the compressed range. There may be some duplication (more than one actual value is represented by the same compressed value), but the duplication is distributed across the defined range.	0
0	DOWN_SMPL	R/W	Enable downsampler 4:4:4 to 4:2:2. 0 = Disable 1 = Enable	0

Video Mode Register (SiI9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x4A	VID_MODE	DITHER_MODE		DITHER	RANGE	CSC	UPSMP	DEMUX	SYNCEXT

Bit	Label	R/W	Description	Default
7:6	DITHER_MODE	R/W	Identifies the number of bits per output video channel: 0b00 = Dither to 8 bits 0b01 = Reserved 0b10 = Reserved 0b11 = Reserved	0b00
5	DITHER	R/W	0 = Dither disabled; the video output is truncated to the output width specified in DITHER_MODE [7:6] 1 = Dither enabled; the video output is dithered to the output width specified in DITHER_MODE [7:6]	0
4	RANGE	R/W	Data Range 16–235 to 0–255 expansion. 0 = Disable 1 = Enable When this bit is set, the HDMI transmitter expands the range of pixel data values from 16–235 into the full 8-bit range of 0–255. This feature is suitable for translating input YCbCr data into output RGB data in PC modes that use the complete range. The *HDMI Specification* allows one non-CEA-861D mode in the first and only 18-byte descriptor of the EDID 1.3 of the sink. This is the native resolution of the sink, which may be RGB. It may be a standard PC resolution (XGA, SXGA, WXGA, and so on), or a specific native resolution. In these cases (or for a sink with the Type B HDMI connector, which allows multiple PC modes), when the HDMI transmitter receives YCbCr data, the data must be expanded to full range for delivering RGB full-range modes. See the *HDMI Specification*.	0
3	CSC	R/W	YCbCr to RGB color space conversion. 0 = Disable 1 = Enable	0
2	UPSMP	R/W	Up sampling 4:2:2 to 4:4:4. 0 = Disable 1 = Enable	0
1	DEMUX	R/W	One- to two-data-channel demultiplexing. 0 = Disable 1 = Enable	0
0	SYNCEXT	R/W	Embedded sync extraction. 0 = Disable 1 = Enable	0

Video Mode Register (SiI9134)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x4A	VID_MODE	DITHER_MODE		DITHER	RANGE	CSC	UPSMP	DEMUX	SYNCEXT

Bit	Label	R/W	Description	Default
7:6	DITHER_MODE	R/W	Identifies the number of bits per output video channel: 0b00 = Dither to 8 bits 0b01 = Dither to 10 bits 0b10 = Dither to 12 bits 0b11 = Reserved	0b00
5	DITHER	R/W	0 = Dither disabled; the video output is truncated to the output width specified in DITHER_MODE [7:6] 1 = Dither enabled; the video output is dithered to the output width specified in DITHER_MODE [7:6]	0
4	RANGE	R/W	Data Range 16-to-235 to 0-to-255 expansion: 0 = Disable 1 = Enable When this bit is set, the HDMI transmitter expands the range of pixel data values from 16–235 into the full 8-bit range of 0–255. This feature is suitable for translating input YCbCr data into output RGB data in PC modes that use the complete range. The *HDMI Specification* allows one non-CEA-861D mode in the first and only 18-byte descriptor of the sink EDID 1.3. This is the native resolution of the sink, which may be RGB. It may be a standard PC resolution (XGA, SXGA, WXGA, and so on), or a specific native resolution. In these cases (or for a sink with the Type B HDMI connector, which allows multiple PC modes), when the HDMI transmitter receives YCbCr data, the data must be expanded to full range for delivering RGB full-range modes. See the *HDMI Specification*.	0
3	CSC	R/W	YCbCr to RGB color space conversion. 0 = Disable 1 = Enable	0
2	UPSMP	R/W	Up sampling 4:2:2 to 4:4:4. 0 = Disable 1 = Enable	0
1	DEMUX	R/W	One- to two-data-channel demultiplexing. 0 = Disable 1 = Enable	0
0	SYNCEXT	R/W	Embedded sync extraction. 0 = Disable 1 = Enable	0

Video Blanking Registers

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x4B	VID_BLANK1	VID_BLANK1							
0x72	0x4C	VID_BLANK2	VID_BLANK2							
0x72	0x4D	VID_BLANK3	VID_BLANK3							

Bit	Label	R/W	Description	Default
7:0	VID_BLANK1	R/W	Defines the video blanking value for Channel 1 (Blue).	0x00
7:0	VID_BLANK2	R/W	Defines the video blanking value for Channel 2 (Green).	0x00
7:0	VID_BLANK3	R/W	Defines the video blanking value for Channel 3 (Red).	0x00

Note: The VID_BLANK bit in the DCTL register (0x72:0x0D[2]) enables video blanking. Refer to page 5.

These registers are not affected by the SET_AVMUTE flag.

Video VSYNC Length Register (SiI9134)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x4E	DC_HEADER	DC_HEADER							

Bit	Label	R/W	Description	Default
7:0	DC_HEADER	R/W	This is the least significant byte of the deep color header that sends the TMDS dynamic phase once per frame.	0x03

Interrupt Registers

The interrupt registers coordinate enabling and recognizing the interrupts generated by the HDMI transmitter.

Figure 3 shows the control of the INT output pin. Each interrupt source has a bit (shown as INT*a*_bit in the figure) and a mask (shown as MASK*a*_bit), where *a* is the label of the interrupt bit and its corresponding mask bit. Each of these pairs is logically ANDed. The AND result of each pair is then ORed and latched with the active output clock and appears in the INTR bit (0x72:0x70[0]). This bit, along with the POLARITY# and OUTPUT_TYPE bits (0x72:0x79[2:1]; refer to page 25) affect INT in all modes. When RESET# goes LOW, POLARITY# defaults to 1 and OUTPUT_TYPE defaults to 0 (push-pull).

When the device is powered down by setting PD# (0x72:0x08[0] shown on page 4) to 0, RSEN (0x72:0x71[5]) is the only interrupt that affects the INT output pin. No other condition generates an interrupt when the transmitter is powered-down with PD#. If necessary, the host device must use other means to monitor the hot plug state, such as polling the System Status Register. Note that when RESET# is LOW, PD# is reset to zero.

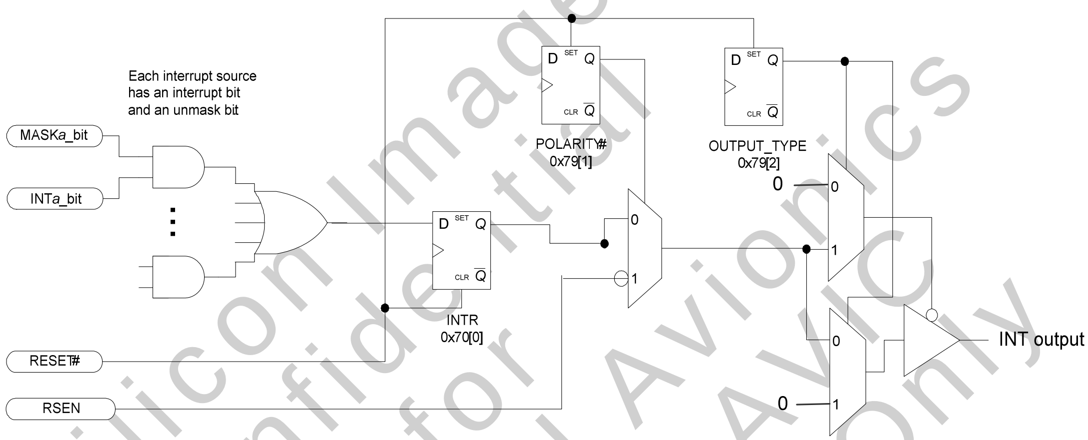

Figure 3. Interrupt Pin Control

Interrupt State Register

Dev	Addr	Name		7	6	5	4	3	2	1	0
0x72	0x70	INTR_STATE		RSVD0							INTR
Bit	**Label**		**R/W**	**Description**							**Default**
0	INTR		R	Interrupt state. When an interrupt is asserted, this bit is set to 1. The polarity of the INT output signal is set using this bit and the POLARITY# bit in the INT_CTRL register (0x72:0x79). Only INTR1, INTR2, and INTR3 bits with matching bits set in INT_UNMASK contribute to setting the INTR bit.							0

Interrupt Source Registers

When reading any interrupt bit in the following three tables, a 1 indicates that the interrupt is asserted, and a 0 indicates no interrupt occurred.

Register INTR1

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x71	INTR1	SOFT	HPD	RSEN	DROP_ SAMPLE	BI_PHASE _ERR	RI_128	OVER_ RUN	UNDER _RUN

Bit	Label	R/W	Description	Default
7	SOFT	R	Software Induced Interrupt: allows the firmware to generate an interrupt directly. Write a 1 to clear.	0
6	HPD	R	Monitor Detect Interrupt: asserted if Hot Plug Detect has changed state. The HDMI transmitter signals a change in the connectivity to a sink, either unplug or plug. HDMI specifies that Hot Plug must be active only when the sink EDID is ready to be read and that Hot Plug is toggled any time there is a change in connectivity downstream of an attached repeater. Write a 1 to clear.	0
5	RSEN	R	Receiver Sense Interrupt, asserted if RSEN has changed. Write a 1 to clear.	X
4	DROP_SAMPLE	R	New preamble forced to drop sample (S/PDIF input only). If the HDMI transmitter detects an 8-bit preamble in the S/PDIF input stream before the subframe has been captured, this interrupt is set. An S/PDIF input that stops signaling or a flat-line condition can create such a premature preamble. Write a 1 to clear.	0
3	BI_PHASE_ERR	R	Input S/PDIF stream has a bi-phase error. This can occur when there is noise or a change of the Fs rate on the S/PDIF input. Write a 1 to clear.	0
2	RI_128	R	Input counted past frame count threshold set in RI_128_COMP register. This interrupt occurs when the count written to register 0x72:0x24 is matched by the VSYNC (frame) counter in the HDMI transmitter. It should trigger the firmware to perform a link integrity check. Such a match occurs every 128 frames. Write a 1 to clear.	0
1	OVER_RUN	R	Audio FIFO Overflow. This interrupt occurs if the audio FIFO overflows when more samples are written into it than are drawn out across the HDMI link. Such a condition can occur from a transient change in the Fs or pixel clock rate. Write a 1 to clear.	0
0	UNDER_RUN	R	Audio FIFO Underflow. Similar to OVER_RUN. This interrupt occurs when the audio FIFO empties. Write a 1 to clear.	0

Register INTR2

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x72	INTR2	BCAP_ DONE	SPDIF_ PAR	ENC_ DIS	PREAM _ERR	CTS_CHG	ACR_OVR	TCLK_ STBL	VSYNC_ REC

Bit	Label	R/W	Description	Default
7	BCAP_DONE	R	If set, this interrupt detected that the FIFORDY bit (0x74:0x40[5]) is set to 1 in the HDMI receiver. To enable this interrupt, ENC_EN (0x72:0x0F[0]), BCAP_EN (0x72:0x27[1]), and Ri_EN (0x72:0x27[0]) must all be set to 1. Write a 1 to clear.	0
6	SPDIF_PAR	R	S/PDIF parity Error. The S/PDIF stream includes a parity (P) bit at the end of each sub-frame. An interrupt occurs if the calculated parity does not match the state of this bit. Write a 1 to clear.	0
5	ENC_DIS	R	The ENC_EN bit (0x72:x0F[0]) changed from 1 to 0. This interrupt occurs if encryption is turned off (0x72:0x0F[0] is set to 0). Write a 1 to clear.	0
4	PREAM_ERR	R	This condition is the opposite of the condition that causes DROP_SAMPLE (0x72:0x71[4]). This interrupt occurs if a preamble is expected but not found when the S/PDIF stream is being decoded. Write a 1 to clear.	0
3	CTS_CHG	R	Change in ACR CTS value. This interrupt occurs when the change is of an unexpected magnitude. Such an interrupt should be expected when changing Fs or pixel clock frequency. Write a 1 to clear.	0
2	ACR_OVR	R	ACR Packet Overwrite. This interrupt occurs if the HDMI transmitter puts an NCTS packet into the queue before the previous NCTS packet has been sent. This can occur if very long active data times do not allow for sufficient NCTS packet bandwidth. For all CEA-861D modes, no ACR_OVR interrupt should occur. Write a 1 to clear.	0
1	TCLK_STBL	R	TCLK_STABLE (register 0x72:0x09[0]) changes state. Whenever IDCK changes, there is a temporary instability in the internal clocking. This interrupt is set when the internal clocking has stabilized. **Note**: TCLK is a multiple of IDCK and is generated by the internal PLL. Write a 1 to clear.	0
0	VSYNC_REC	R	Asserted when VSYNC active edge is recognized. It is useful for triggering firmware actions that occur during vertical blanking. Write a 1 to clear.	0

Register INTR3

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x73	INTR3	REG_ INTR3_ STAT7	RSVD	REG_ INTR3_ STAT5	REG_ INTR3_ STAT4	REG_ INTR3_ STAT3	REG_ INTR3_ STAT2	REG_ INTR3_ STAT1	REG_ INTR3_ STAT0

Bit	Label	R/W	Description	Default
7	REG_INTR3_ STAT7	R/W	R_i not read within one frame; to clear, write a 1.	0
6	Ri_ERR#2	R/W	Reserved	0
5	REG_INTR3_ STAT5	R/W	R_i and R_i' do not match during 2nd frame (default during frame #0 = reg. 0x25); to clear, write a 1.	0
4	REG_INTR3_ STAT4	R/W	R_i and R_i' do not match during 1st frame (default during frame #127 = reg. 0x25 – 1); to clear, write a 1.	0
3	REG_INTR3_ STAT3	R/W	DDC command is complete. Asserted if set to 1. Write a 1 to clear.	0
2	REG_INTR3_ STAT2	R/W	DDC FIFO is half-full interrupt. Asserted if set to 1. Write a 1 to clear.	0
1	REG_INTR3_ STAT1	R/W	DDC FIFO is full interrupt. Asserted if set to 1. Write a 1 to clear.	0
0	REG_INTR3_ STAT0	R/W	DDC FIFO is empty. Asserted if set to 1. Write a 1 to clear.	0

Interrupt Unmask Register

Interrupts are set in the INTR1, INTR2, and INTR3 registers as they occur, but only interrupts with a corresponding bit set in the INT_UNMASK1 through INT_UNMASK3 registers are logically ORed into the INT output pin signal. The state of any interrupt can be checked at any time by reading the INTR1, INTR2, and INTR3 registers directly. The INTR_STATE register 0x72:0x70, described on page 21, is also only set for matching INT_UNMASK bits.

All bits marked RSVD in the interrupt registers should have the corresponding bit in the INT_UNMASK register cleared to zero.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x75	INT_UNMASK1	INT_UNMASK[7:0]							
0x72	0x76	INT_UNMASK2	INT_UNMASK[15:8]							
0x72	0x77	INT_UNMASK3	INT_UNMASK[23:16]							

Bit	Label	R/W	Description	Default
7:0 7:0 7:0	INT_UNMASK1 INT_UNMASK2 INT_UNMASK3	R/W	Enable or disable corresponding bit in INTR1. Enable or disable corresponding bit in INTR2. Enable or disable corresponding bit in INTR3. 0 = Disable corresponding interrupt to INT output 1 = Enable corresponding interrupt to INT output	0x00 0x00 0x00

Interrupt Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x79	INT_CTRL	RSVDRW				SOFT_INTR	OUTPUT_TYPE	POLARITY#	RSVDRW

Bit	Label	R/W	Description	Default
3	SOFT_INTR	R/W	Set software interrupt. 0 = Clear interrupt 1 = Set interrupt	0
2	OUTPUT_TYPE	R/W	INT pin output type. 0 = Push/pull 1 = Open drain **Note**: This bit must be set to 1 after reset to configure the INT pin as an open drain output.	0
1	POLARITY#	R/W	INT pin assertion level. 0 = Assert HIGH 1 = Assert LOW	1

TMDS Control Registers

The TMDS registers control Transition-Minimized Differential Signaling (TMDS). Please see the Setting up the PLL Control Registers section (p. 109) for details regarding the settings of the registers described in this section.

TMDS C Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x80	TMDS_CCTRL	RSVDRW0	RSVDRW0	FAPOSTCOUNT	RSVDRW1	RSVDRW0		RSVDRW1	

Bit	Label	R/W	Description	Default
5	FAPOSTCOUNT	R/W	Filter PLL post counter setting for the audio clock. 0 = Divided by 1 1 = Divided by 2	0

TMDS Control Register #1

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x82	TMDS_CTRL	RSVDRW0	TCLKSEL		RSVDRW0		LVBIAS	RSVDRW0	STERM

Bit	Label	R/W	Description	Default
6:5	TCLKSEL	R/W	Selects FPLL multiple of the IDCK: 0b00 = FPLL is 0.5 • IDCK 0b01 = FPLL is 1.0 • IDCK 0b10 = FPLL is 2.0 • IDCK 0b11 = FPLL is 4.0 • IDCK	0b01
2	LVBIAS	R/W	This bit should always be set to 1 after reset.	0
0	STERM	R/W	Internal source termination. 0 = Disable 1 = Enable **Note**: Silicon Image recommends enabling source termination. Refer to the respective datasheet for more information.	1

For certain combinations of video input clock frequency and audio sampling rate, the HDMI transmitter must use a higher multiple of the input pixel clock when sampling the S/PDIF input.

Set ICLK to reflect the pixel replication factor of the input data stream so that it is properly decoded. TCLKSEL indicates the factor by which the input clock (IDCK) must be multiplied to yield the output clock frequency. These settings ensure that the output clock can provide sufficient bandwidth for multi-channel audio data on an HDMI link (25 MHz at a minimum). The pixel replication count bits in the AVI InfoFrame Packet must be accurate. Refer to Table 4 for examples.

Table 4. TMDS Control Register Example

480p mode (pixel clock = 27 MHz)							
Input Clock (IDCK)[1]	**Audio Mode & max F_s**	**ICLK[2] 0x72:0x48[1:0] Input Pixel Replication[4]**	**DEMUX 0x72:0x4A[1]**	**TCLKSE[3] 0x72:0x82[6:5]**	**Output (Link) Clock[1]**	**Output Pixel Replication[10]**	**AVI InfoFrame Packet Byte 5 bits PR3:PR0[10]**
54 MHz	8 ch, 96 kHz	0b01 (2x)	0	0b01 (1.0)	54 MHz[6]	2x[7]	0b0001
54 MHz	2 ch, 192 kHz	0b01 (2x)	0	0b00 (0.5)	27 MHz	1x	0b0000
54 MHz	8 ch, 96 kHz	0b00 (1x)	1[5]	0b01	54 MHz[6]	2x[7]	0b0001
54 MHz	2 ch, 192 kHz	0b00 (1x)	1[5]	0b00	27 MHz	1x	0b0000
27 MHz	8 ch, 96 kHz	0b00 (1x)	0	0b10 (2.0)	54 MHz[6]	2x[7]	0b0001
27 MHz	2 ch, 192 kHz	0b00 (1x)	0	0b11 (4.0)	108 MHz	4x[8]	0b0011
27 MHz	8 ch, 48 kHz	0b00 (1x)	0	0b01 (1.0)	27 MHz	1x[9]	0b0000

Notes:
1. Input Clock (IDCK) and Output Clock must be within the min/max range for the HDMI transmitter.

2. For proper decoding, set ICLK to reflect the pixel replication factor of the input data stream.
3. Factor by which input clock must be multiplied to give output clock frequency.
4. There is only one pixel per 27 MHz clock cycle, so each must be replicated.
5. When YCbCr 4:2:2 data is multiplexed onto a single channel, the input clock must be doubled.
6. 54 MHz is necessary so that the blanking intervals have sufficient bandwidth to carry the 8-channel audio data sampled at frequencies up to 96 kHz.
7. Because the output clock has been doubled, pixels must be replicated.
8. Illustrates 4x pixel replication on output.
9. 27 MHz input clock provides sufficient bandwidth for 8-channel audio data sampled at frequencies 48 kHz and below. Refer to the *HDMI Specification*.
10. Bits PR0:PR3 of Byte 5 of the AVI InfoFrame packet indicate to the HDMI sink how many repetitions of each unique pixel are transmitted. Refer to Table 14 in the *CEA-861D Specification*.

These settings ensure that the output clock provides sufficient bandwidth for multi-channel audio data on an HDMI link (25 MHz minimum).

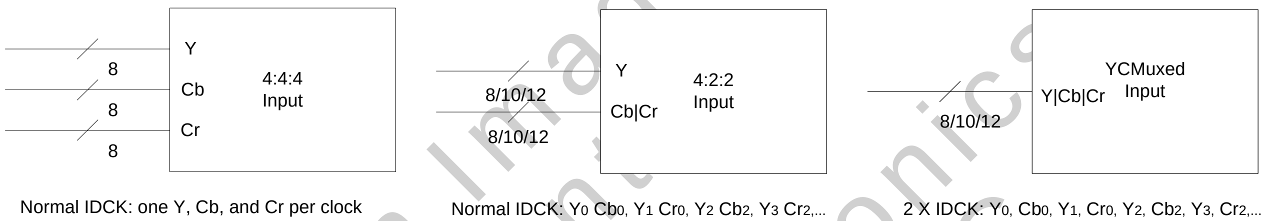

Figure 4: Input Bus Diagram for Different Formats

Note: All three input bus formats can use 656 encoded syncs.

TMDS Control Register #2

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x83	TMDS_CTRL2	POST_COUNT		FFB_COUNT			FFR_COUNT		

Bit	Label	R/W	Description	Default
7:6	POST_COUNT	R/W	Sets the divider ratio for the HDMI transmitter PLL post counter: 0b00 = Divide by 1 0b01 = Divide by 2 0b10 = Divide by 4 0b11 = Invalid	0b00
5:3	FFB_COUNT	R/W	Sets the divider ratio for the PLL filter feedback counter: 0b000 = Divide by 1 0b001 = Divide by 2 0b010 = Divide by 3 0b011 = Divide by 4 0b100 = Divide by 5 0b101 = Divide by 6 0b110 = Divide by 7	0b011
2:0	FFR_COUNT	R/W	Sets the divider ratio for the PLL filter front counter: 0b000 = Divide by 1 0b001 = Divide by 2 0b011 = Divide by 4 0b111 = Divide by 8	0b011

TMDS Control Register #3

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x84	TMDS_CTRL3	RSVDRW0	ITPLL				FPOST_COUNT		

Bit	Label	R/W	Description	Default
6:3	ITPLL	R/W	Controls the frequency response of the low pass filter for the transmitter PLL: 0b0000 = 5 µA 0b0001 = 10 µA 0b0010 = 20 µA 0b0011 = 25 µA 0b0100 = 40 µA 0b0110 = 50 µA 0b1000 = 80 µA 0b1011 = 100 0µA 0b1111 = 135 µA The filter bandwidth is approximately 4 MHz. Silicon Image recommends using the default value. However, depending on the supply voltage, operating temperature, and input frequency variations, the user may adjust that value to avoid overshoots, which amplify input clock jitter.	0b0011
2:0	FPOST_COUNT	R/W	Sets the divider ratio for the PLL filter post counter: 0b000 = Divide by 1 0b001 = Divide by 2 0b011 = Divide by 4 0b111 = Divide by 8	0b000

TMDS Control Register #4

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0x85	TMDS_CTRL4	RSVDRW0				RSVDRW0		TFR_COUNT	

Bit	Label	R/W	Description	Default
1:0	TFR_COUNT	R/W	HDMI transmitter PLL front counter setting: 0b00 = Divide by 1 0b01 = Divide by 2 0b10 = Divide by 4 0b11 = Invalid	0b01

Repeater Authentication Enable Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xCC	TMDS_CTRL4	Reserved				Repeater Authentication Enable	Reserved		

Bit	Label	R/W	Description	Default
3	Repeater Authentication Enable	R/W	**IMPORTANT:** This bit ***must*** be set to 1 if the sink connected to the SiI9034/9134 transmitter is a repeater.	0

DDC Master Registers

The following registers control the DDC output port, described on page 106. The speed of the Master DDC clock is determined by an internal oscillator in the HDMI transmitter, with a maximum of 100 kbps. There is no requirement for an active input pixel clock to the HDMI transmitter, and the DDC SCL speed is not affected by the pixel clock frequency.

The auto-synchronous R_i check also uses the DDC output port. For proper handshaking, refer to the Ri_STARTED bit in the RI_STAT register (0x72:0x26[0]), described on page 8.

Note: The DDC_CMD, DDC_DATA and DDC_STATUS registers are not accessible if PDOSC = 0 or PDTOT# = 0.

DDC I^{2}C Manual Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xEC	DDC_MAN	MAN_OVR	RSVDRW0	MAN_SDA	MAN_SCL	RSVDRW0		IO_SCL	IO_SDA

Bit	Label	R/W	Description	Default
7	MAN_OVR	R/W	Manual Override of SCL and SDA output. 0 = Normal operation 1 = Override port with MAN_SCL and MAN_SDA states	0
5	MAN_SDA	R/W	Manual SDA output.	0
4	MAN_SCL	R/W	Manual SCL output.	0
1	IO_SCL	R	DDC SCL input state.	0
0	IO_SDA	R	DDC SDA input state.	0

DDC I^{2}C Target Slave Address Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xED	DDC_ADDR	DDC_ADDR							RSVD0

Bit	Label	R/W	Description	Default
7:1	DDC_ADDR	R/W	DDC device address.	0b0000000

DDC I^{2}C Target Segment Address Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xEE	DDC_SEGM	DDC_SEGM							

Bit	Label	R/W	Description	Default
7:0	DDC_ SEGM	R/W	DDC segment address.	0x00

DDC I^{2}C Target Offset Address Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xEF	DDC_OFFSET	DDC_OFFSET							

Bit	Label	R/W	Description	Default
7:0	DDC_OFFSET	R/W	DDC offset address.	0x00

DDC I²C Data Count Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xF0	DDC_COUNT1	DDC_COUNT[7:0]							
0x72	0xF1	DDC_COUNT2	RSVDRW0						DDC_COUNT[9:8]	

Bit	Label	R/W	Description	Default
7:0 1:0	DDC_COUNT1 DDC_COUNT2	R/W	The total number of bytes to be read from the slave or written to the slave before a *Stop* bit is sent on the DDC bus. For example, if the HDCP KSV FIFO length is 635 bytes (127 devices x 5 bytes/KSV), the DDC_COUNT must be 0x27B.	0x00 0b00

DDC I²C Status Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xF2	DDC_STATUS	RSVDR0	BUS_ LOW	NO_ ACK	IN_ PROG	FIFO_ FULL	FIFO_ EMP	FRD_ USE	FWT_ USE

Bit	Label	R/W	Description	Default
6	BUS_LOW	R/W	1 = I²C transaction did not start because I²C bus is pulled LOW by an external device. This bit must be cleared to 0 by the firmware.	0
5	NO_ACK	R/W	1 = HDMI transmitter did not receive an ACK from slave device during address or data write. This bit must be cleared to 0 by the firmware.	0
4	IN_PROG	R	1 = DDC operation in progress	0
3	FIFO_FULL	R	1 = DDC FIFO full	0
2	FIFO_EMP	R	1 = DDC FIFO empty	0
1	FRD_USE	R	1 = DDC FIFO read in use	0
0	FWT_USE	R	1 = DDC FIFO write in use	0

The DDC master feature recognizes and supports clock stretching by an I²C slave device. Clock stretching is described in the *I²C Specification*.

DDC I^{2}C Command Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xF3	DDC_CMD	RSVD0		DDC_FLT_EN	SDA_DEL_EN	DDC_CMD			

Bit	Label	R/W	Description	Default
5	DDC_FLT_EN	R/W	Enable the DDC delay. 0 = Enable 1 = Disable A DDC delay is inserted into the SDA line to create a 300-ns delay for the falling edge of the DDC SDA signal to prevent an erroneous I^{2}C START condition. The real start condition must have a setup time of 600 ns so that this delay of 300 ns does not remove the real START condition. Filtering is done using a ring oscillator.	0
4	SDA_DEL_EN	R/W	Enable 3 ns glitch filtering on the DDC clock and data line: 0 = Enable 1 = Disable Filtering is done using a ring oscillator.	0
3:0	DDC_CMD	R/W	DDC command. 0b1111 = Abort transaction 0b1001 = Clear FIFO 0b1010 = Clock SCL 0b0000 = Current address read with no ACK on last byte 0b0010 = Sequential read with no ACK on last byte 0b0100 = Enhanced DDC read with no ACK on last byte 0b0110 = Sequential write ignoring ACK on last byte 0b0111 = Sequential write requiring ACK on last byte Writing to this register immediately initiates the I^{2}C transaction on the DDC bus. **Note**: The Clear FIFO command resets the FIFO read and write pointers to zero. Data formerly loaded into the FIFO cannot be read after a Clear FIFO, because the FIFO is now empty. Other command codes are reserved and may cause the DDC bus to hang if used. The Clock SCL command resets any I^{2}C devices on the DDC lines. This reset should be initiated once before initiating the DDC commands.	0b0000

DDC I²C Data Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xF4	DDC_DATA	DDC_DATA							

Bit	Label	R/W	Description	Default
7:0	DDC_DATA	R/W	DDC data input.	0x00

The FIFO supports multi-byte sequential read commands from the controller. Such a command is diagrammed in Figure 13 on page 106. Up to 16 bytes can be read in one local I²C command. Data bytes continue to be loaded into the FIFO until it is full.

DDC I²C FIFO Count Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xF5	DDC_FIFOCNT	RSVDRW0			DDC_FIFOCNT				

Bit	Label	R/W	Description	Default
4:0	DDC_FIFOCNT	R/W	FIFO data byte count (the number of bytes in the FIFO). The DDC FIFO size is 16. The maximum value for DDC_FIFOCNT is 0x10.	0b00000

ROM Registers

The following registers are used to determine the status of the HDCP keys stored in the HDMI transmitter ROM.

ROM Status Register

Write a bit in this register to 0 to clear the corresponding condition bit.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xF9	KEY_ STATUS	RSVD	BIST2 _ERR	BIST1 _ERR	RSVDRW1	RSVD	RSVD	CRC_ERR	CMD_DONE

Bit	Label	R/W	Description	Default
6	BIST2_ERR	R/W	1 = BIST self-authentication test 2 error	0
5	BIST1_ERR	R/W	1 = BIST self-authentication test 1 error	0
1	CRC_ERR	R/W	1 = CRC error	0
0	CMD_DONE	R/W	1 = Command done (last operation completed successfully)	0

ROM Command Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x72	0xFA	KEY_ COMMAND	RSVD0		LD_KSV	EPCM				

Bit	Label	R/W	Description	Default
5	LD_KSV	R/W	Enable KSV load from embedded keys: 0 = Disable 1 = Enable (write 0 before enabling again)	N/A
4:0	EPCM	R/W	I^2C master command to the embedded keys: 0b00000 = Run no BIST tests 0b00011 = Run all BIST tests 0b00100 = Run only CRC test 0b01000 = Run only BIST self-authentication test 1 (a 16-bit CRC to verify embedded key contents) 0b10000 = Run only BIST self-authentication test 2 (a 2-pass authentication that uses an inverted key selection vector to verify the HDCP cipher engine) Do not use any other values. Before writing a new value into this register, verify that the previous command is complete by checking the CMD_DONE bit (0x72:0xF9[0]) and then clearing it.	0b00000

The BIST command can be performed according to this procedure:

1. Assert hardware reset and release it. (RESET# pin is HIGH).
2. IDCK is active and is 74 MHz.
3. Set SWRST = 1 (0x72:0x05=0x01).
4. Set up PLL:
   0x72:0x80 = 0x23, 0x72:0x83 = 0x98, 0x72:0x84 = 0x63, 0x72:0x85 = 0x00.
5. Power up (0x72:0x08 = 0x01).
6. Release SWRST (0x72:0x05 = 0x00).
7. Wait until P_STABLE = 1 (read 0x72:0x09[0]).
8. Write ANSTOP 2 times:
   0x72:0x0F = 0x0C
   0x72:0x0F = 0x0C
9. Wait until 0x72:0xF9[0] = 1.
10. Write 0x72:0xF9 = 0x00.
11. BIST start 0x72:0xFa = 0x03.
12. Wait until 0x72:0xF9[0] = 1.
13. Pass if 0x72.0xF9 = 0x03, fail for other values.

Audio Registers

The HDMI link does not transport an explicit audio master clock; instead, it encodes the frequency of that clock in N/CTS Packets sent during command times. The ratio of the current pixel clock frequency to the desired MCLK frequency is defined by a numerator, *N*, and a denominator, *CTS*. Whenever the pixel clock frequency changes (the video mode changes) or the audio clock changes (the audio sampling rate changes), the value of *N* must also be updated.

ACR Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x01	ACR_CTRL	RSVD0						NCTSPKT_EN	CTS_SEL

Bit	Label	R/W	Description	Default
1	NCTSPKT_EN	R/W	CTS request enable: 0 = N/CTS packet disabled 1 = N/CTS packet enabled	1
0	CTS_SEL	R/W	CTS source select: 0 = Send HW-updated CTS value in N/CTS packet (recommended) 1 = Send SW-updated CTS value in N/CTS packet (for diagnostic use) Silicon Image recommends that this bit *not* be set to 1.	0

ACR Audio Frequency Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x02	FREQ_SVAL	RSVD0					MCLK_CONF		

Bit	Label	R/W	Description	Default
2:0	MCLK_CONF	R/W	MCLK input mode: 0b000 = MCLK is 128 • Fs 0b001 = MCLK is 256 • Fs 0b010 = MCLK is 384 • Fs 0b011 = MCLK is 512 • Fs 0b100 = MCLK is 768 • Fs 0b101 = MCLK is 1024 • Fs 0b110 = MCLK is 1152 • Fs 0b111 = MCLK is 192 • Fs The HDMI transmitter uses these bits to divide the MCLK input to produce CTS values according to the 128 • Fs formula. The ratio MCLK to Fs is for the input Fs, not the downsampled output Fs (refer to the ASRC register (0x7A:0x23), on page 45).	0b001

ACR N Software Value Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x03	N_SVAL1	N_SVAL[7:0]							
0x7A	0x04	N_SVAL2	N_SVAL[15:8]							
0x7A	0x05	N_SVAL3	RSVD0				N_SVAL[19:16]			

Bit	Label	R/W	Description	Default
7:0 7:0 3:0	N_SVAL1 N_SVAL2 N_SVAL3	R/W	N value for audio clock regeneration method. This must be written to the registers to create the correct divisor for audio clock regeneration. Only values greater than 0 are valid. This register must be written after a hardware reset.	0x00 0x00 0b0000

ACR CTS Software Value Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x06	CTS_SVAL1	CTS_SVAL[7:0]							
0x7A	0x07	CTS_SVAL2	CTS_SVAL[15:8]							
0x7A	0x08	CTS_SVAL3	RSVD0				CTS_SVAL[19:16]			

Bit	Label	R/W	Description	Default
7:0 7:0 3:0	CTS_SVAL1 CTS_SVAL2 CTS_SVAL3	R/W	CTS value for the audio clock regeneration method. For diagnostic use and applied only when the CTS_SEL bit (0x7A:0x01[0]) is set to 1.	0x00 0x00 0b0000

ACR CTS Hardware Value Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x09	CTS_HVAL1	CTS_HVAL[7:0]							
0x7A	0x0A	CTS_HVAL2	CTS_HVAL[15:8]							
0x7A	0x0B	CTS_HVAL3	RSVD0				CTS_HVAL[19:16]			

Bit	Label	R/W	Description	Default
7:0 7:0 3:0	CTS_HVAL1 CTS_HVAL2 CTS_HVAL3	R	CTS value for the audio clock regeneration method. This value is measured and stored here by the hardware when MCLK is active and N is valid, after 128 Fs/N cycles of MCLK.	X X X

Audio In Mode Register

Dev	Add	Name	7	6	5	4	3	2	1	0
0x7A	0x14	AUD_MODE	SD3_EN	SD2_EN	SD1_EN	SD0_EN	DSD_EN	RSVDRW0	SPDIF_EN	AUD_EN

Bit	Label	R/W	Description	Default
7	SD3_EN	R/W	I^2S input channel #3. 0 = Disable 1 = Enable	0
6	SD2_EN	R/W	I^2S input channel #2. 0 = Disable 1 = Enable	0
5	SD1_EN	R/W	I^2S input channel #1. 0 = Disable 1 = Enable	0
4	SD0_EN	R/W	I^2S input channel #0. 0 = Disable 1 = Enable	0
3	DSD_EN	R/W	Direct Stream Digital Audio enable. 0 = Disable 1 = Enable	0
1	SPDIF_EN	R/W	S/PDIF input stream. 0 = Disable 1 = Enable	0
0	AUD_EN	R/W	Audio input stream. 0 = Disable 1 = Enable	0

Audio input data is selected from either the S/PDIF input or the I^2S inputs. Audio input data can be disabled by clearing the AUD_EN bit (refer to Figure 5 on page 61). Bits 7:4 also apply to DSD and HBR (High Bit Rate) Audio when any of them are selected.

The Direct Stream Digital Audio (DSD) Enable bit has a lower priority than S/PDIF enable, but higher than the I^2S stream. When it is set, most of the I^2S configuration register bits become control for the DSD logic including SD3/2/1/0 enable, the I^2S FIFO map, and the Channel Status registers.

See the note on page 39 about the limitation of the HDMI transmitter in assigning the I^2S channels to audio FIFOs.

Audio In S/PDIF Control Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x15	SPDIF_CTRL	RSVDRW0				NOAUDIO	RSVDRW0	FS_OVERRIDE	RSVDRW0

Bit	Label	R/W	Description	Default
3	NOAUDIO	R	No S/PDIF audio. 0 = Detected change on the S/PDIF input 1 = No change detected on the S/PDIF input	0
1	FS_OVERRIDE	R/W	S/PDIF input stream override. 0 = Use input S/PDIF stream's detected FS 1 = Use software FS in I2S_CHST4 register (0x7A:0x21)	0

Audio In S/PDIF Extracted Fs and Length Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x18	HW_SPDIF_FS	HW_SPDIF_LEN			HW_MAXLEN	HW_SPDIF_FS			

Bit	Label	R/W	Description	Default
7:5 4	HW_SPDIF_LEN HW_MAXLEN	R R	Channel status bits 33 to 35 (bit 33 = LSB, bit 35 = MSB) Combines with HW_ SPDIF_LEN (channel status bit 32) to indicate sample size: 0 = Maximum sample length is 20 bits 1 = Maximum sample length is 24 bits Audio sample word length, indicated by bits [7:5], depends on the setting of bit 4 (HW_MAXLEN). The sample word length is shown by bits [7:4] as follows: 0b0000 = not available 0b0010 = 16 0b0100 = 18 0b1000 = 19 0b1010 = 20 0b1100 = 17 0b0001 = not available 0b0011 = 20 0b0101 = 22 0b1001 = 23 0b1011 = 24 0b1101 = 21	0b0000
3:0	HW_SPDIF_FS	R	Set to the Fs extracted from the S/PDIF input channel status bits 24–27.	0b0000

Audio In I²S Channel Swap Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x19	SWAP_I2S	SWCH3	SWCH2	SWCH1	SWCH0	RSVDRW			

Bit	Label	R/W	Description	Default
7	SWCH3	R/W	Swap left-right channels for I²S Channel 3. 0 = Do not swap left and right 1 = Swap left and right	0
6	SWCH2	R/W	Swap left-right channels for I²S Channel 2. 0 = Do not swap left and right 1 = Swap left and right	0
5	SWCH1	R/W	Swap left-right channels for I²S Channel 1. 0 = Do not swap left and right 1 = Swap left and right	0
4	SWCH0	R/W	Swap left-right channels for I²S Channel 0. 0 = Do not swap left and right 1 = Swap left and right	0
3:0	RSVDRW	R/W	Reserved; do not modify.	0x9

Note: Each SWCH[3:0] bit is active *only* when the corresponding I²S input channel is enabled with register 0x7A:0x14.

Audio Error Threshold Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1B	SPDIF_ERTH	RSVDRW0	RSVDRW1	AUD_ERR_THRESH					

Bit	Label	R/W	Description	Default
5:0	AUD_ERR_THRESH	R/W	Specifies the error threshold level. The frame is invalid if the number of bi-phase mark encoding errors in the audio stream exceeds this threshold level during frame decoding.	0b001000

Audio In I²S Data In Map Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1C	I2S_IN_MAP	FIFO3_MAP		FIFO2_MAP		FIFO1_MAP		FIFO0_MAP	

Bit	Label	R/W	Description	Default
7:6	FIFO3_MAP	R/W	Channel map to FIFO #3 (for HDMI Layout 1): 0b00 = Map SD0 to FIFO #3 0b01 = Map SD1 to FIFO #3 0b10 = Map SD2 to FIFO #3 0b11 = Map SD3 to FIFO #3	0b11
5:4	FIFO2_MAP	R/W	Channel map to FIFO #2 (for HDMI Layout 1): 0b00 = Map SD0 to FIFO #2 0b01 = Map SD1 to FIFO #2 0b10 = Map SD2 to FIFO #2 0b11 = Map SD3 to FIFO #2	0b10
3:2	FIFO1_MAP	R/W	Channel map to FIFO #1 (for HDMI Layout 1): 0b00 = Map SD0 to FIFO #1 0b01 = Map SD1 to FIFO #1 0b10 = Map SD2 to FIFO #1 0b11 = Map SD3 to FIFO #1	0b01
1:0	FIFO0_MAP	R/W	Channel map to FIFO #0 (for HDMI Layout 0 or 1): 0b00 = Map SD0 to FIFO #0 0b01 = Map SD1 to FIFO #0 0b10 = Map SD2 to FIFO #0 0b11 = Map SD3 to FIFO #0	0b00

HDMI allows for up to eight channels of audio content. Two channels pass through each of the four FIFOs listed in the description for register 0x7A:0x1C. HDMI does not restrict the source to use any specific subset of the FIFOs. For example, 4-channel content can use many combinations of two FIFOs and two fields in the Audio InfoFrame packets (indicated by each packet's B.X and SP.X bits; see the *HDMI Specification*) limited only by the channel assignment choices in EIA/CEA-861D Section 8.3.2. The HDMI transmitter logic sets the B and PR bits automatically in each Audio InfoFrame packet, using both the 0x1C register settings and the I²S channel enables in the 0x7A:0x14 register.

Some HDMI receiver chips do not make the arriving B and PR bit information accessible to the sink firmware. Therefore, the only indicator of *used audio channels* is in Data Byte 4 of the Audio InfoFrame packet.

Audio In I^{2}S Control Register (SiI9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1D	I2S_IN_CTRL	RSVD0	SCK_ EDGE	RSVD0	RSVD0	I2S_ WS	I2S_ JUST	I2S_ DIR	I2S_ SHIFT

Bit	Label	R/W	Description	Default
6	SCK_EDGE	R/W	SCK sample edge: 0 = Sample edge is falling; SD3-SD0 and WS source should change state on the rising edge of SCK 1 = Sample clock is rising; SD3-SD0 and WS source should change state on the falling edge of SCK	1
3	I2S_WS	R/W	WS polarity: 0 = Left polarity when WS is LOW 1 = Left polarity when WS is HIGH	0
2	I2S_JUST	R/W	SD justify: 0 = Data is left-justified 1 = Data is right-justified	1
1	I2S_DIR	R/W	SD direction: 0 = MSB shifted first 1 = LSB shifted first	0
0	I2S_SHIFT	R/W	WS to SD first bit shift: 0 = First bit shift (refer to the *Philips Specification*) 1 = No shift	1

Audio In I^{2}S Control Register (SiI9134)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1D	I2S_IN_CTRL	HBRA_ ON	SCK_ EDGE	CBIT_ ORDER	VBit	I2S_ WS	I2S_ JUST	I2S_ DIR	I2S_ SHIFT

Bit	Label	R/W	Description	Default
7	HBRA_ON	R/W	High Bit Rate Audio On. 0 = Input stream is not high bit rate 1 = Input stream is high bit rate. All of the I^{2}S control bits will apply to the control of the High Bit Rate Audio.	0
6	SCK_EDGE	R/W	SCK sample edge: 0 = Sample edge is falling; SD3-SD0 and WS source should change state on the rising edge of SCK 1 = Sample clock is rising; SD3-SD0 and WS source should change state on the falling edge of SCK	1
5	CBIT_ORDER	R/W	This bit should be set to 1 for High Bit Rate Audio	0
4	VBit	R/W	V bit value 0 = PCM 1 = Compressed	0
3	I2S_WS	R/W	WS polarity: 0 = Left polarity when WS is LOW 1 = Left polarity when WS is HIGH	0
2	I2S_JUST	R/W	SD justify: 0 = Data is left-justified 1 = Data is right-justified	1
1	I2S_DIR	R/W	SD direction: 0 = MSB shifted first 1 = LSB shifted first	0
0	I2S_SHIFT	R/W	WS to SD first bit shift: 0 = First bit shift (refer to the *Philips Specification*) 1 = No shift	1

Audio In I²S Channel Status Register Word 1 (9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1E	I2S_CHST1	I2S_CHST1							

Bit	Label	R/W	Description	Default
7:0	I2S_CHST1	R/W	Channel Status Byte #0	0x00

Audio In I²S Channel Status Register Word 1 (9134)*

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1E	I2S_CHST1	I2S_CHST1[7:3]					PCM_COMP	RSVD	CON_PROF

Bit	Label	R/W	Description	Default
7:3	I2S_CHST1	R/W	Channel Status Byte #0[7:3]	0b00000
2	PCM_COMP	R/W	Compression flag. 0 = PCM 1 = Compressed	0
0	CON_PRPF	R/W	Select consumer or professional. 0 = Consumer 1 = Professional	0

***Note:** When using the SiI9134 transmitter, information written to this register does not comply with the IEC 60958 audio standard. The PCM/compressed bit is not accessible at bit 1 of this register. Instead, to set high bit rate audio, write a 1 to bit 2. This correctly sets bit 1 in the CHST1 header byte sent to the receiver. The copyright-asserted bit is not accessible, but is forced to 0 (copyright protection asserted).

Audio In I²S Channel Status Register Word 2

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x1F	I2S_CHST2	I2S_CHST2							

Bit	Label	R/W	Description	Default
7:0	I2S_CHST2	R/W	Channel Status Byte #1: Category code	0x00

Audio In I²S Channel Status Register Word 3

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x20	I2S_CHST3	I2S_CHAN_NUM				I2S_SRC_NUM			

Bit	Label	R/W	Description	Default
7:4	I2S_CHAN_NUM	R/W	Channel Status Byte #2: Channel number	0b0000
3:0	I2S_SRC_NUM	R/W	Channel Status Byte #2: Source number	0b0000

Audio In I²S Channel Status Register Word 4 (9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x21	I2S_CHST4	CLK_ACCUR				SW_SPDIF_FS			

Bit	Label	R/W	Description	Default
7:4	CLK_ACCUR	R/W	Clock accuracy.	0b0000
3:0	SW_SPDIF_FS	R/W	Sampling frequency as set by software which is inserted into the S/PDIF stream if FS_OVERRIDE is enabled.	0b1111

Audio In I^{2}S Channel Status Register Word 4 (9134)*

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x21	I2S_CHST4	CLK_ACCUR				SW_SPDIF_FS			

Bit	Label	R/W	Description	Default
7:4	CLK_ACCUR	R/W	Clock accuracy.	0b0000
3:0	SW_SPDIF_FS	R/W	Sampling frequency as set by software that is inserted into the S/PDIF stream if FS_OVERRIDE is enabled. Refer to the note below.	0b1111

***Note:** When the transmitter processes high bit rate audio, set this register to 0x09.

Audio In I^{2}S Channel Status Register Word 5 (9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x22	I2S_CHST5	FS_ORIG				I2S_LEN			I2S_MAXLEN

Bit	Label	R/W	Description	Default
7:4	FS_ORIG	R/W	Original Fs.	0b0000
3:1 0	I2S_LEN I2S_MAXLEN	R/W	Bit [0] sets the maximum sample word length: 0 = 20 bits 1 = 24 bits Audio sample word length, set with bits [3:1], depends on the setting of bit 0 (I2S_MAXLEN). The word length is set using bits [3:0] as follows:[1, 2] 0b0000 = not available 0b0010 = 16 0b0100 = 18 0b1000 = 19 0b1010 = 20 0b1100 = 17 0b0001 = not available 0b0011 = 20 0b0101 = 22 0b1001 = 23 0b1011 = 24 0b1101 = 21	0b0001

Notes:

1. The word length bits [3:0] should always match the word length of the audio samples coming into the HDMI transmitter, even if the transmitter downsamples the audio. The word length of the input I^{2}S stream is set in the 0x7A:0x24 register, described on page 46. Audio stream down sampling is enabled in the 0x7A:0x23 register, described on page 45.
2. If an audio input whose sample length is less than or equal to 20 bits is downsampled, then the output sample length will always be 20 bits, and 0x7A:0x22[3:0] should be set to 0b0011. However, if the downsampled input sample length is larger than 20 bits, the output sample length is equal to the input sample length, and 0x7A:0x22[3:0] should be set accordingly.

Audio In I^{2}S Channel Status Register Word 5 (9134)[1]

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x22	I2S_CHST5	FS_ORIG				I2S_LEN			I2S_MAXLEN

Bit	Label	R/W	Description	Default
7:4	FS_ORIG	R/W	Original Fs.	0b0000
3:1 0	I2S_LEN I2S_MAXLEN	R/W	Bit [0] sets the maximum audio sample word length: 0 = 20 bits 1 = 24 bits Audio sample word length, set with bits [3:1], depends on the setting of bit 0 (I2S_MAXLEN). The word length is set using bits [3:0] as follows:[2, 3] 0b0000 = not available 0b0010 = 16 0b0100 = 18 0b1000 = 19 0b1010 = 20 0b1100 = 17 0b0001 = not available 0b0011 = 20 0b0101 = 22 0b1001 = 23 0b1011 = 24 0b1101 = 21	0b0001

Notes:

1. When the transmitter processes high bit rate audio, set this register to 0xE2.
2. With the exception of high bit rate audio, the word length bits [3:0] must always match the word length of the audio samples coming into the HDMI transmitter, even if the transmitter downsamples the audio. The word length of the input I^{2}S stream is set in the 0x7A:0x24 register, described on page 46. Audio stream down sampling is enabled in the 0x7A:0x23 register, described on page 45.
3. If an audio input whose sample length is less than or equal to 20 bits is downsampled, then the output sample length will always be 20 bits, and 0x7A:0x22[3:0] should be set to 0b0011. However, if the downsampled input sample length is larger than 20 bits, the output sample length is equal to the input sample length, and 0x7A:0x22[3:0] must be set accordingly.

HDMI 1.1 requires that accurate channel status information be transmitted in the Audio InfoFrames. When the audio is supplied to the HDMI transmitter at the S/PDIF input, the channel status is extracted from the S/PDIF stream. When the audio is supplied by I^{2}S inputs, the firmware must write accurate values into the channel status registers. Refer to IEC 60958-3 for detailed definitions of the channel status bits. The channel status information is used in the HDMI receiver to reconstruct the audio into I^{2}S format.

When down-sampling the audio stream (see the ASRC register on page 45), both CHST5 and CHST4 should be loaded with the original (input) Fs rate. The chip sends the original Fs in the audio packet channel status bits corresponding to CHST5, divides the original Fs rate (according to register ASRC), and sends that slower Fs in the bits corresponding to CHST4. Always write the input Fs into CHST4 and CHST5 when connected to an I^{2}S source. An S/PDIF source loads both bytes automatically.

If FS_OVERRIDE is set to 0 (0x7A:0x15[1]), the sampling frequency is extracted from the S/PDIF input stream. It is encoded in that stream's Channel Status bits as shown in Table 5.

The value in FS_OVERRIDE assists source systems that do not provide correct Fs information in the raw S/PDIF stream. The *HDMI Specification* requires the transmitted Fs value to be correct in the channel status bits of the audio sample packets. By setting FS_OVERRIDE to 1, the HDMI transmitter can be programmed (in SW_SPDIF_FS) with the correct Fs value and can ignore the value in the input S/PDIF stream. *Values not listed in Table 5 are reserved.*

Table 5. Encoded Audio Sampling Frequency

CH_ST4 bit				Fs Sampling Frequency
3	**2**	**1**	**0**	
0	0	0	0	44.1 kHz
1	0	0	0	88.2 kHz
1	1	0	0	176.4 kHz
0	0	1	0	48 kHz
1	0	1	0	96 kHz
1	1	1	0	192 kHz
0	0	1	1	32 kHz
0	0	0	1	not indicated
1	0	0	1	768 kHz

Audio Sample Rate Conversion Register (SiI9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x23	ASRC	RSVD0			RSDV1	RSVDRW0		RATIO	SRC_EN

Bit	Label	R/W	Description	Default
7:5	RSVD0	R	These bits are reserved and read-only, and return a zero value.	0b000
4	RSVD1	R	This bit is reserved and read-only, and returns a one value.	1
1	RATIO	R/W	Sample rate down-conversion ratio: 0 = Downsample 2-to-1 when SRC_EN is set to 1 1 = Downsample 4-to-1 when SRC_EN is set to 1	0
0	SRC_EN	R/W	Audio sample rate conversion: 0 = Disable 1 = Enable	0

Audio Sample Rate Conversion Register (SiI9134)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x23	ASRC	HBR_SPR_MASK				RSVDRW0		RATIO	SRC_EN

Bit	Label	R/W	Description	Default
7:4	HBR_SPR_MASK	R/W	Mask for the sample present and flat bit of the High Bit Rate Audio header. Each bit masks one of the subpacket sample-present bits. 0 = Mask out 1 = Unmask Bits 7:4 must be programmed to 0b0000 when HBRA mode is selected.	0b0001
1	RATIO	R/W	Sample rate down-conversion ratio: 0 = Downsample 2-to-1 when SRC_EN is set to 1 1 = Downsample 4-to-1 when SRC_EN is set to 1	0
0	SRC_EN	R/W	Audio sample rate conversion: 0 = Disable 1 = Enable	0

Sample rate conversion is applied only to 2-channel audio, either from the S/PDIF input or from the I^{2}S Channel 0 input. Setting register ASRC to 0x01 downsamples 96-kHz audio to 48 kHz and 88.2-kHz audio to 44.1 kHz. Setting ASRC to 0x02 downsamples 192 kHz-audio to 48K kHz and 176.4-kHz audio to 44.1 kHz. This conversion is performed after selecting S/PDIF or I^{2}S input paths to the HDMI transmitter. The CHST5 bits written in register 0x7A:0x22 should always indicate the sample rate before any down sampling (refer to page 41).

If an audio source is connected to both the HDMI transmitter and an audio DAC, the audio source may output 192-kHz (or 176.4-kHz) audio to drive the DAC and the transmitter. The DAC uses the higher sample rate, while the HDMI transmitter simultaneously downsamples that stream to the 48 kHz (or 44.1 kHz) sample rate. 48 kHz and 44.1 kHz are the default audio rates for HDMI.

The N value and the I^{2}S channel status registers should be set according to the input audio configuration. The N value is affected by the down sampling so that the outgoing N value in the N/CTS packets represents the video-to-audio ratio for the downsampled audio stream. For example, to input 96 kHz audio with 74.25 MHz video input and to output 48kHz audio, the N register value should be set to 12288. This N will be divided by the downsample RATIO, so that the HDMI transmitter sends packets with N = 6144.

Note: Bits 7:4 must be programmed to 0b0000 when HBRA mode is selected.

Important: Sample rate conversion works only for 2-channel PCM audio; set LAYOUT to 0 in register 0x7A:0x2F.

Audio I²S Input Length Register (SiI9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x24	I2S_IN_LEN	RSVD				IN_LENGTH			

Bit	Label	R/W	Description	Default
7:4	RSVD	R	These bits are reserved and read-only, and return an indeterminate value.	X
3:0	IN_LENGTH	R/W	Number of valid bits in the input I²S stream. Used for the extraction of the I²S data from the input stream. 0b1111 – 0b1110 = N/A 0b1101 = 21 bit 0b1100 = 17 bit 0b1011 = 24 bit 0b1010 = 20 bit 0b1001 = 23 bit 0b1000 = 19 bit 0b0111 – 0b0110 = N/A 0b0101 = 22 bit 0b0100 = 18 bit 0b0011 = N/A 0b0010 = 16 bit 0b0001 – 0b0000 = N/A	0b1011

Audio I²S Input Length Register (SiI9134)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x24	I2S_IN_LEN	HDR_PKT_ID				IN_LENGTH			

Bit	Label	R/W	Description	Default
7:4	HDR_PKT_ID	R/W	Four LS bits of the ID of the High Bit Rate Audio packet header.*	0b1001
3:0	IN_LENGTH	R/W	Number of valid bits in the input I²S stream. Used for the extraction of the I²S data from the input stream. 0b1111 – 0b1110 = N/A 0b1101 = 21 bit 0b1100 = 17 bit 0b1011 = 24 bit 0b1010 = 20 bit 0b1001 = 23 bit 0b1000 = 19 bit 0b0111 – 0b0110 = N/A 0b0101 = 22 bit 0b0100 = 18 bit 0b0011 = N/A 0b0010 = 16 bit 0b0001 – 0b0000 = N/A	0b1011

***Note:** Specified by HDMI 1.3, Table 5-28.

Audio Mode Switching Sequence

When switching audio modes, the following procedure must be followed to ensure that the frame counter is properly reset.

1. Mute the audio sent to the receiver, as shown in Table 3. (Set 0x72:0x0D[2] to 0, 0x72:0x0D[1] to 1, and 0x7A:0xDF[0] to 0).
2. Disable the audio input stream by setting 0x7A:0x14[0] to 0.
3. Set all audio mode registers to the new audio mode as needed.
4. Wait 6 ms.
5. Enable the audio input stream by setting 0x7A:0x14[0] to 1.
6. Unmute the audio sent to the receiver. (Set 0x72:0x0D[2] to 0, 0x72:0x0D[1] to 0, and 0x7A:0xDF[0] to 0).

HDMI Control Register (SiI9034)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x2F	HDMI_CTRL	RSVD0						LAYOUT	HDMI_MODE

Bit	Label	R/W	Description	Default
1	LAYOUT	R/W	Audio packet header layout indicator: 0 = Layout 0 (2-channel) 1 = Layout 1 (up to 8 channels)	0
0	HDMI_MODE	R/W	HDMI mode: 0 = Disable 1 = Enable	0

HDMI Control Register (SiI9134)

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x2F	HDMI_CTRL	RSVD0	DC_EN	PACKET_MODE			RSVD0	LAYOUT	HDMI_MODE

Bit	Label	R/W	Description	Default
6	DC_EN	R/W	Deep-color packet enable: 0 = Do not send deep-color related information in the packet to the HDMI receiver 1 = Send deep-color related information in the packet to the HDMI receiver. The following data is sent in data byte #1 of the packet: 7 = PP3 6 = PP2 5 = PP1 4 = PP0 3 = CD3 2 = CD2 1 = CD1 0 = CD0 The CD bits indicate deep-color mode (defined in the same register bits 5:3, CD3 is always 0). The PP bits indicate the phase-related information that comes from the hardware state machine.	0
5:3	PACKET_MODE	R/W	Specifies the number of bits per pixel sent to the packetizer: 0b0xx = Reserved 0b100 = 24 bits per pixel (8 bits per channel; no packing) 0b101 = 30 bits per pixel (10 bits per channel; pack to 8 bits) 0b110 = 36 bits per pixel (12 bits per channel; pack to 8 bits) 0b111 = Reserved	0b000
1	LAYOUT	R/W	Audio packet header layout indicator: 0 = Layout 0 (2-channel) 1 = Layout 1 (up to 8 channels)	0
0	HDMI_MODE	R/W	HDMI mode: 0 = Disable 1 = Enable	0

Audio Path Status Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x30	AUDO_TXSTAT	RSVD0					MUTE	NULL_ PACKET_EN VS_HIGH	NULL_ PACKET_EN

Bit	Label	R/W	Description	Default
2	MUTE	R	General Control Packet mute status: 0 = No packet with SET_AVMUTE=1 has been sent 1 = A packet with SET_AVMUTE =1 has been sent The MUTE bit is equal to the SET_AVMUTE bit in register 0x7A:0xDF, described on page 58. MUTE is not set immediately when the SET_AVMUTE bit in register 0xDF is written. In HDMI mode, MUTE is set after the General Control Packet is transmitted and after writing SET_AVMUTE to 1. In DVI mode, MUTE is set at the start of VSYNC. In HDMI mode, MUTE is cleared when a General Control Packet with CLR_AVMUTE = 1 is sent. In DVI mode, MUTE is cleared at the start of VSYNC after CLR_AVMUTE has been written to 1. **Note:** The combinations {SET_AVMUTE, CLR_AVMUTE} = {0, 0} and {1, 1} are not supported by HDMI. Such a packet is not compliant. For more details refer to page 103 and the *HDMI Specification*.	0
1	NULL_PACKET_EN VS_HIGH	R/W	Enables null packet flooding only when VSync is HIGH.	0
0	NULL_PACKET_EN	R/W	Enables null packet flooding all the time. **Note:** The HDMI compliance test will fail if this bit is set to 1. This bit should always remain 0.	0

Diagnostic Power Down Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x3D	DPD	RSVD0				RSVD	PDIDCK#	PDOSC	PDTOT#

Bit	Label	R/W	Description	Default
3	RSVD	R	This bit is reserved	1
2	PDIDCK#	R/W	Power down IDCK input: 0 = Power down; block IDCK signal to disable all IDCK-based logic 1 = Normal operation	1
1	PDOSC	R/W	Power down the internal ring oscillator. The ring oscillator clock is required for the operation of the DDC master, the ROM, and interrupts. 0 = Power down 1 = Normal operation	1
0	PDTOT#	R/W	Power down total: 0 = Power down everything 1 = Normal operation	1

Refer to page 123 for details on the restrictions for using these power-down bits.

InfoFrame Registers

For each of the seven packets described in the following registers, two bits control whether the packet is sent once or repeatedly. To send a packet one time, set the EN (enable) bit corresponding to that packet after the packet data has been written into the appropriate registers. Read the EN bit to determine if the packet has actually been transmitted across the link. When the EN bit is cleared to zero, the packet has been transmitted.

To send a packet repeatedly, write the corresponding RPT (repeat) bit to 1 at the same time as EN is set to 1. This sends the packet during every VBLANK period.

To disable repeated transmission, clear the corresponding RPT and EN bits simultaneously.

To guarantee that the HDMI receiver remains in HDMI mode, at least one HDMI packet must be transmitted every two VSYNC periods (refer to the *HDMI Specification*). This occurs automatically whenever audio is being transmitted. If audio is not yet enabled or if the HDMI transmitter is not sending audio for some other reason, transmit a null packet (all zeroes) during every VBLANK period. Set the contents of the Generic Control Packet buffer to all zeroes and then set both EN and RPT bits for that packet.

The ID, TYPE, checksum, and length fields in each HDMI InfoFrame must be loaded by the microcontroller. The SiI9034/9134 transmitter has no internal logic for calculating the checksum or any preset length value.

Note: The EN bits in registers 0x7A:0x3E and 0x7A:0x3F can be set or cleared only when IDCK is active and when the HDMI transmitter is *not* in a powered-down state. Refer to page 123 for more details on power-down bits. Data bytes can be written in each InfoFrame. The SET_AVMUTE and CLR_AVMUTE bits in the General Control Packet, along with the RPT bit for each packet type, can be set or cleared when the transmitter is powered down. Therefore, the firmware can write all necessary registers except the EN bits, de-assert any power-down bits, and then write any necessary EN bits.

Note: The EN and RPT bits for the various packet types are not affected by the state of the HDMI_MODE bit (register 0x7A:0x2F[0], described on page 46). Although packets cannot be transmitted in DVI mode (when HDMI_MODE is set to 0), and the HDMI transmitter ignores the states of registers 0x3E through 0x3F when HDMI_MODE is set to 0, the firmware should clear the packet EN and RPT bits whenever switching to DVI mode so that the status of packet sending, when read back from registers 0x3E and 0x3F, is consistent with the link mode. All packet enable and repeat bits are set to their default values after reset.

CEA-861D InfoFrame Control #1 Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x3E	PB_CTRL1	MPEG_EN	MPEG_RPT	AUD_EN	AUD_RPT	SPD_EN	SPD_RPT	AVI_EN	AVI_RPT

Bit	Label	R/W	Description	Default
7	MPEG_EN	R/W	Enable MPEG InfoFrame transmission: 0 = Disable 1 = Enable	0
6	MPEG_RPT	R/W	Repeat MPEG InfoFrame transmission: 0 = Disable (send once after EN bit is set) 1 = Enable (send in every VBLANK period)	0
5	AUD_EN	R/W	Enable Audio InfoFrame transmission: 0 = Disable 1 = Enable	0
4	AUD_RPT	R/W	Repeat Audio InfoFrame transmission: 0 = Disable (send once after EN bit is set) 1 = Enable (send in every VBLANK period)	0
3	SPD_EN	R/W	Enable SPD InfoFrame transmission: 0 = Disable 1 = Enable	0
2	SPD_RPT	R/W	Repeat SPD InfoFrame transmission: 0 = Disable (send once after EN bit is set) 1 = Enable (send in every VBLANK period)	0
1	AVI_EN	R/W	Enable AVI InfoFrame transmission: 0 = Disable 1 = Enable	0
0	AVI_RPT	R/W	Repeat AVI InfoFrame transmission: 0 = Disable (send once after EN bit is set) 1 = Enable (send in every VBLANK period)	0

Refer to the following sections of the *CEA-861D Specification*:

i. For AVI InfoFrames: Section 6.4.
ii. For Source Product Description (SPD) InfoFrames: Section 6.5.
iii. For Audio InfoFrames: Section 6.6.
iv. For MPEG InfoFrames: Section 6.7.

CEA-861D InfoFrame Control #2 Register

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x3F	PB_CTRL2	RSVD0		GEN2_EN	GEN2_RPT	GCP_EN	GCP_RPT	GEN_EN	GEN_RPT

Bit	Label	R/W	Description	Default
5	GEN2_EN	R/W	Enable generic #2 packet transmission: 0 = Disable 1 = Enable	0
4	GEN2_RPT	R/W	Repeat generic #2 packet transmission: 0 = Disable (send once after EN bit is set) 1 = Enable (send in every VBLANK period)	0
3	GCP_EN	R/W	Enable General Control Packet transmission: 0 = Disable 1 = Enable	0
2	GCP_RPT	R/W	Repeat General Control Packet transmission: 0 = Disable (send once after EN bit is set) 1 = Enable (send in every VBLANK period)	0
1	GEN_EN	R/W	Enable generic packet transmission: 0 = Disable 1 = Enable	0
0	GEN_RPT	R/W	Repeat generic packet transmission: 0 = Disable (send once after EN bit is set) 1= Enable (send in every VBLANK period)	0

Refer to the *HDMI 1.3 Standard*, section 5.3.6, for a detailed description of General Control Packets (GCP).
Refer to pages 50 and 63 for more information regarding the usage of Generic Control Packets.

CEA-861D InfoFrame Registers

Refer to the *HDMI Specification* and the *CEA-861D Specification* for a detailed explanation of these register fields.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x40	AVI_TYPE	AVI_HDR[7:0]							
0x7A	0x41	AVI_VERS	AVI_HDR[15:8]							
0x7A	0x42	AVI_LEN	AVI_HDR[23:16]							
0x7A	0x43	AVI_CHSUM	AVI_HDR[31:24]							
0x7A	0x44	AVI_DBYTE1	AVI_DATA							
0x7A	0x45	AVI_DBYTE2								
0x7A	0x46	AVI_DBYTE3								
0x7A	0x47	AVI_DBYTE4								
0x7A	0x48	AVI_DBYTE5								
0x7A	0x49	AVI_DBYTE6								
0x7A	0x4A	AVI_DBYTE7								
0x7A	0x4B	AVI_DBYTE8								
0x7A	0x4C	AVI_DBYTE9								
0x7A	0x4D	AVI_DBYTE10								
0x7A	0x4E	AVI_DBYTE11								
0x7A	0x4F	AVI_DBYTE12								
0x7A	0x50	AVI_DBYTE13								
0x7A	0x51	AVI_DBYTE14								
0x7A	0x52	AVI_DBYTE15								

Bit	Label	R/W	Description	Default
7:0	AVI_TYPE	R/W	AVI InfoFrame type code.	0x00
7:0	AVI_VERS	R/W	AVI InfoFrame version code.	0x00
7:0	AVI_LEN	R/W	AVI InfoFrame length.	0x00
7:0	AVI_CHSUM	R/W	AVI InfoFrame checksum.	0x00
	AVI_DATA	R/W	AVI InfoFrame data bytes.	

Refer to page 101 for more details on the fields and valid settings in the AVI InfoFrame.

SPD InfoFrame Registers

Refer to the *CEA-861D Specification* for more information on these register fields.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x60	SPD_TYPE	SPD_HDR[7:0]							
0x7A	0x61	SPD_VERS	SPD_HDR[15:8]							
0x7A	0x62	SPD_LEN	SPD_HDR[23:16]							
0x7A	0x63	SPD_CHSUM	SPD_HDR[31:24]							
0x7A	0x64	SPD_DBYTE1								
0x7A	0x65	SPD_DBYTE2								
0x7A	0x66	SPD_DBYTE3								
0x7A	0x67	SPD_DBYTE4								
0x7A	0x68	SPD_DBYTE5								
0x7A	0x69	SPD_DBYTE6								
0x7A	0x6A	SPD_DBYTE7								
0x7A	0x6B	SPD_DBYTE8								
0x7A	0x6C	SPD_DBYTE9								
0x7A	0x6D	SPD_DBYTE10								
0x7A	0x6E	SPD_DBYTE11								
0x7A	0x6F	SPD_DBYTE12								
0x7A	0x70	SPD_DBYTE13								
0x7A	0x71	SPD_DBYTE14	SPD_DATA							
0x7A	0x72	SPD_DBYTE15								
0x7A	0x73	SPD_DBYTE16								
0x7A	0x74	SPD_DBYTE17								
0x7A	0x75	SPD_DBYTE18								
0x7A	0x76	SPD_DBYTE19								
0x7A	0x77	SPD_DBYTE20								
0x7A	0x78	SPD_DBYTE21								
0x7A	0x79	SPD_DBYTE22								
0x7A	0x7A	SPD_DBYTE23								
0x7A	0x7B	SPD_DBYTE24								
0x7A	0x7C	SPD_DBYTE25								
0x7A	0x7D	SPD_DBYTE26								
0x7A	0x7E	SPD_DBYTE27								

Bit	Label	R/W	Description	Default
7:0	SPD_TYPE	R/W	SPD InfoFrame type code.	0x00
7:0	SPD_VERS	R/W	SPD InfoFrame version code.	0x00
7:0	SPD_LEN	R/W	SPD InfoFrame length.	0x00
7:0	SPD_CHSUM	R/W	SPD InfoFrame checksum.	0x00
	SPD_DATA	R/W	SPD InfoFrame data bytes.	

Audio InfoFrame Registers

Refer to the *HDMI Specification* and the *CEA-861D Specification* for more information on these register fields.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0x80	AUDIO_TYPE	AUDIO_HDR[7:0]							
0x7A	0x81	AUDIO_VERS	AUDIO_HDR[15:8]							
0x7A	0x82	AUDIO_LEN	AUDIO_HDR[23:16]							
0x7A	0x83	AUDIO_CHSUM	AUDIO_HDR[31:24]							
0x7A	0x84	AUDIO_DBYTE1	AUDIO_DATA							
0x7A	0x85	AUDIO_DBYTE2								
0x7A	0x86	AUDIO_DBYTE3								
0x7A	0x87	AUDIO_DBYTE4								
0x7A	0x88	AUDIO_DBYTE5								
0x7A	0x89	AUDIO_DBYTE6								
0x7A	0x8A	AUDIO_DBYTE7								
0x7A	0x8B	AUDIO_DBYTE8								
0x7A	0x8C	AUDIO_DBYTE9								
0x7A	0x8D	AUDIO_DBYTE10								

Bit	Label	R/W	Description	Default
7:0	AUDIO_TYPE	R/W	AUDIO InfoFrame type code.	0x00
7:0	AUDIO_VERS	R/W	AUDIO InfoFrame version code.	0x00
7:0	AUDIO_LEN	R/W	AUDIO InfoFrame length.	0x00
7:0	AUDIO_CHSUM	R/W	AUDIO InfoFrame checksum.	0x00
	AUDIO_DATA	R/W	AUDIO InfoFrame data bytes.	

MPEG InfoFrame Registers

Refer to the *CEA-861D Specification* for more information on these register fields.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0xA0	MPEG_TYPE	MPEG_HDR[7:0]							
0x7A	0xA1	MPEG_VERS	MPEG_HDR[15:8]							
0x7A	0xA2	MPEG_LEN	MPEG_HDR[23:16]							
0x7A	0xA3	MPEG_CHSUM	MPEG_HDR[31:24]							
0x7A	0xA4	MPEG_DBYTE1	MPEG_DATA							
0x7A	0xA5	MPEG_DBYTE2								
0x7A	0xA6	MPEG_DBYTE3								
0x7A	0xA7	MPEG_DBYTE4								
0x7A	0xA8	MPEG_DBYTE5								
0x7A	0xA9	MPEG_DBYTE6								
0x7A	0xAA	MPEG_DBYTE7								
0x7A	0xAB	MPEG_DBYTE8								
0x7A	0xAC	MPEG_DBYTE9								
0x7A	0xAD	MPEG_DBYTE10								
0x7A	0xAE	MPEG_DBYTE11								
0x7A	0xAF	MPEG_DBYTE12								
0x7A	0xB0	MPEG_DBYTE13								
0x7A	0xB1	MPEG_DBYTE14								
0x7A	0xB2	MPEG_DBYTE15								
0x7A	0xB3	MPEG_DBYTE16								
0x7A	0xB4	MPEG_DBYTE17								
0x7A	0xB5	MPEG_DBYTE18								
0x7A	0xB6	MPEG_DBYTE19								
0x7A	0xB7	MPEG_DBYTE20								
0x7A	0xB8	MPEG_DBYTE21								
0x7A	0xB9	MPEG_DBYTE22								
0x7A	0xBA	MPEG_DBYTE23								
0x7A	0xBB	MPEG_DBYTE24								
0x7A	0xBC	MPEG_DBYTE25								
0x7A	0xBD	MPEG_DBYTE26								
0x7A	0xBE	MPEG_DBYTE27								

Bit	Label	R/W	Description	Default
7:0	MPEG_TYPE	R/W	MPEG InfoFrame type code.	0x00
7:0	MPEG_VERS	R/W	MPEG InfoFrame version code.	0x00
7:0	MPEG_LEN	R/W	MPEG InfoFrame length.	0x00
7:0	MPEG_CHSUM	R/W	MPEG InfoFrame checksum.	0x00
	MPEG_DATA	R/W	MPEG InfoFrame data bytes.	

Generic Packet Registers

These registers can transmit any type of packet.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0xC0	GEN_DBYTE1								
0x7A	0xC1	GEN_DBYTE2								
0x7A	0xC2	GEN_DBYTE3								
0x7A	0xC3	GEN_DBYTE4								
0x7A	0xC4	GEN_DBYTE5								
0x7A	0xC5	GEN_DBYTE6								
0x7A	0xC6	GEN_DBYTE7								
0x7A	0xC7	GEN_DBYTE8								
0x7A	0xC8	GEN_DBYTE9								
0x7A	0xC9	GEN_DBYTE10								
0x7A	0xCA	GEN_DBYTE11								
0x7A	0xCB	GEN_DBYTE12								
0x7A	0xCC	GEN_DBYTE13								
0x7A	0xCD	GEN_DBYTE14								
0x7A	0xCE	GEN_DBYTE15								
0x7A	0xCF	GEN_DBYTE16	GEN_DATA							
0x7A	0xD0	GEN_DBYTE17								
0x7A	0xD1	GEN_DBYTE18								
0x7A	0xD2	GEN_DBYTE19								
0x7A	0xD3	GEN_DBYTE20								
0x7A	0xD4	GEN_DBYTE21								
0x7A	0xD5	GEN_DBYTE22								
0x7A	0xD6	GEN_DBYTE23								
0x7A	0xD7	GEN_DBYTE24								
0x7A	0xD8	GEN_DBYTE25								
0x7A	0xD9	GEN_DBYTE26								
0x7A	0xDA	GEN_DBYTE27								
0x7A	0xDB	GEN_DBYTE28								
0x7A	0xDC	GEN_DBYTE29								
0x7A	0xDD	GEN_DBYTE30								
0x7A	0xDE	GEN_DBYTE31								

Bit	Label	R/W	Description	Default
	GEN_DATA	R/W	Generic packet data bytes.	

General Control Packet Register

Refer to the *HDMI Specification* and the *HDCP Specification* for more information on the 0xDF register bits.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0xDF	GCP_BYTE1	RSVD0			CLR_AVMUTE	RSVD0			SET_AVMUTE

Bit	Label	R/W	Description	Default
4	CLR_AVMUTE	R/W	Clear the AVMUTE flag.	0
0	SET_AVMUTE	R/W	Set the AVMUTE flag. When the AVMUTE flag is set, the HDMI transmitter sends a General Control Packet on the TMDS link to inform the sink that the data may be incorrect. The HDMI transmitter sends blank-level data for all video packets and 0x00 for all audio packet data. When the AVMUTE flag is set, the sink assumes that no valid data is being received. Optionally, the sink can apply a mute function to the audio data or a blank function to the video data.	0

Note: Before enabling General Control Packet transmission (GCP_EN, GCP_RPT), the firmware must write 0x10 or 0x01 to this register. The default value of 0x00 is not valid in a transmitted General Control Packet.

Generic Packet #2 Registers

These registers can be used to transmit any type of packet.

Dev	Addr	Name	7	6	5	4	3	2	1	0
0x7A	0xE0	GEN2_DBYTE1								
0x7A	0xE1	GEN2_DBYTE2								
0x7A	0xE2	GEN2_DBYTE3								
0x7A	0xE3	GEN2_DBYTE4								
0x7A	0xE4	GEN2_DBYTE5								
0x7A	0xE5	GEN2_DBYTE6								
0x7A	0xE6	GEN2_DBYTE7								
0x7A	0xE7	GEN2_DBYTE8								
0x7A	0xE8	GEN2_DBYTE9								
0x7A	0xE9	GEN2_DBYTE10								
0x7A	0xEA	GEN2_DBYTE11								
0x7A	0xEB	GEN2_DBYTE12								
0x7A	0xEC	GEN2_DBYTE13								
0x7A	0xED	GEN2_DBYTE14								
0x7A	0xEE	GEN2_DBYTE15								
0x7A	0xEF	GEN2_DBYTE16	GEN2_DATA							
0x7A	0xF0	GEN2_DBYTE17								
0x7A	0xF1	GEN2_DBYTE18								
0x7A	0xF2	GEN2_DBYTE19								
0x7A	0xF3	GEN2_DBYTE20								
0x7A	0xF4	GEN2_DBYTE21								
0x7A	0xF5	GEN2_DBYTE22								
0x7A	0xF6	GEN2_DBYTE23								
0x7A	0xF7	GEN2_DBYTE24								
0x7A	0xF8	GEN2_DBYTE25								
0x7A	0xF9	GEN2_DBYTE26								
0x7A	0xFA	GEN2_DBYTE27								
0x7A	0xFB	GEN2_DBYTE28								
0x7A	0xFC	GEN2_DBYTE29								
0x7A	0xFD	GEN2_DBYTE30								
0x7A	0xFE	GEN2_DBYTE31								

Bit	Label	R/W	Description	Default
	GEN2_DATA	R/W	Generic packet #2 data bytes.	

Appendices

The following sections describe how to use the functional block of the HDMI transmitter.

Handling Audio

Enabling Audio Inputs

Audio input data is received from either the S/PDIF input or one or more I²S channels. The AUD_EN bit (register 0x7A:0x14) is logically ANDed with each channel enable bit (S/PDIF and I²S in register 0x7A:0x14). Stop audio processing by writing a 0 to the AUD_EN bit. This stops the decoding of input samples so no samples are written into the audio FIFOs. When the FIFOs are emptied by HDMI formatting, the HDMI transmitter stops sending audio packets on the HDMI link. NCTS packets continue to be sent as long as the transmitter is in HDMI mode.

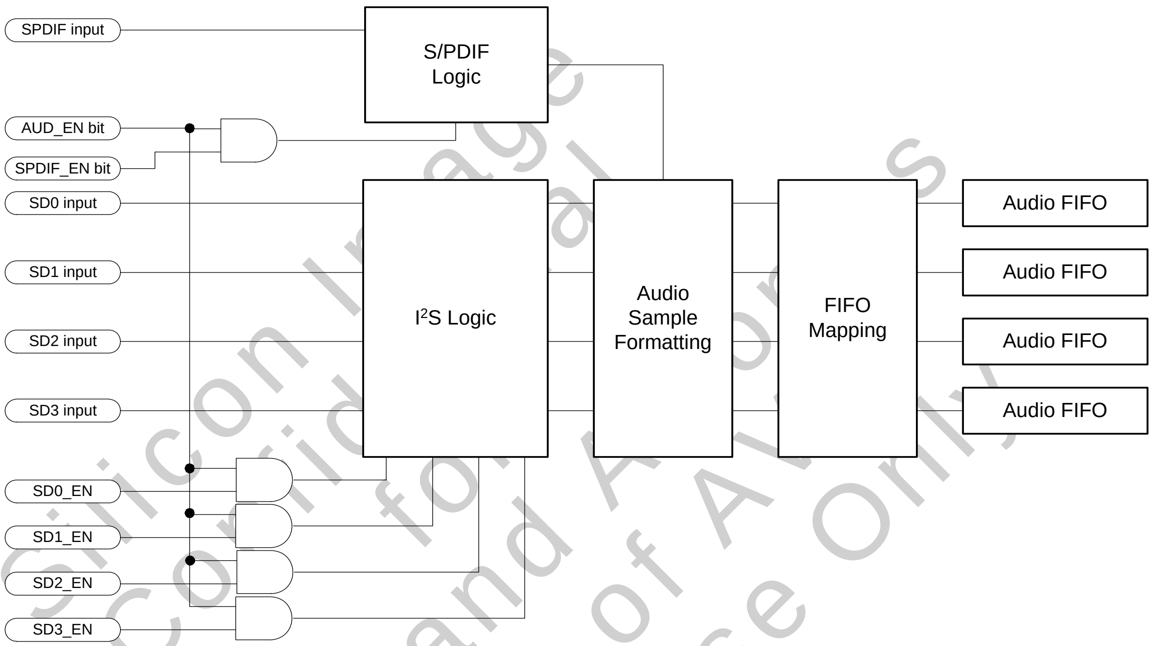

Figure 5. Audio Input Control

Encoding Audio on HDMI

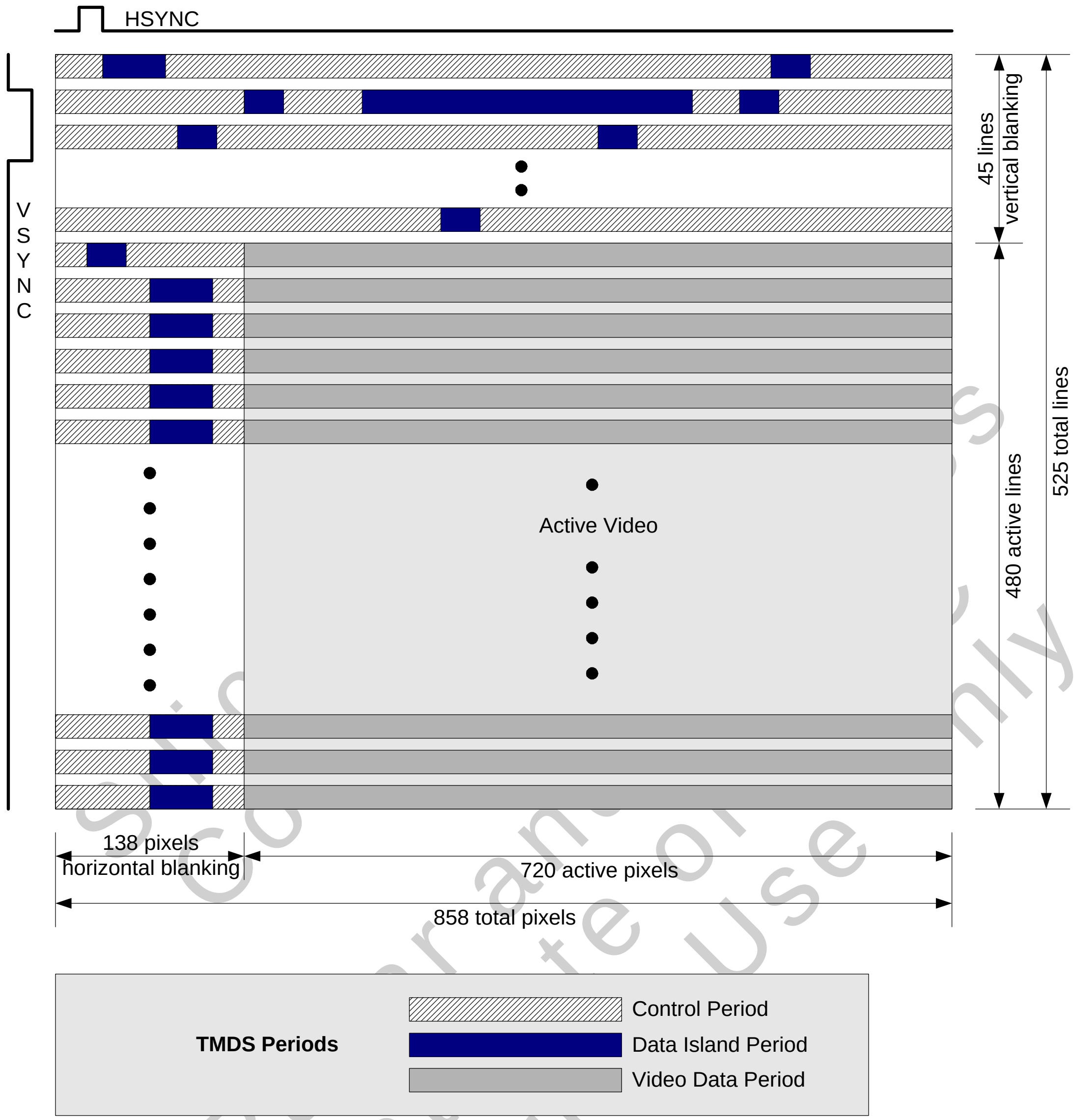

Figure 6. Overview of HDMI Operating Modes in 480p Stream

Handling DVD Audio

This section describes source support for audio content protection and ACP packet transmission.

Source Support for Audio Content Protection

HDMI 1.1 describes three new packet types to support content protection for various audio formats: ACP, ISRC1, and ISRC2. The source transmits these packets, like other packets, as data islands during the vertical blanking time. The source transmits these packets only if it recognizes that the attached sink or repeater is able to receive them, as indicated with the presence of an AI bit in the EDID VSDB. The formats for the ACP, ISRC1, and ISRC2 packets are defined in the *HDMI Specification*.

For more details on handling E-EDID and determining features of the sink based on its E-EDID, see the *HDMI Source Device Software Application Note* (SiI-AN-0117).

Transmitting ACP Packets

The HDMI transmitter allocates register space for four packets with a full-size payload of 31 data bytes. Each packet includes a 4-byte header with TYPE, VERSION, LENGTH, and CHECKSUM fields. The SPD and MPEG InfoFrames are defined in CEA-861D, although their lengths are less than 31 bytes. The HDMI transmitter expands the register space for those InfoFrame packets to accommodate any type of packet with 31 total bytes. In addition, the HDMI transmitter provides for a *Generic Packet* and *Generic Packet #2*. With these four packets, the transmitter can set up and transmit any four of the five defined packet types: SPD, MPEG, ACP, ISRC1, and ISRC2.

ISRC packets are used together to transmit International Standard Recording Code (ISRC) data from the source to the sink. When this information extends beyond 16 bytes, the ISRC_CONT field in the ISRC1 packet is set in the header and the remaining bytes are sent in the ISRC2 packet. The requirements for handling ISRC data are defined in the *DVD Specifications for Read-Only Disc, Part 4 (Version 1.0, March 1999, Annex B).*

The source device is required by HDMI 1.1 to transmit ACP packets within 300 milliseconds of changing to audio content, which requires transmission of content protection information and to repeat transmission of ACP packets at least every 300 milliseconds. If a sink does not detect ACP packets for 600 milliseconds, it assumes that no content protection information is needed. Details on transmission timing requirements for ACP and related packets are described in the *HDMI Specification*.

To control transmission of ACP and related packets with the transmitter, the packet registers must be loaded and enabled. If, in addition to ACP, ISRC1, and ISRC2, the source also needs to send MPEG and SPD InfoFrames, then one set of packet registers must alternate between one of those packet payloads and the ACP or related payload. One possible flowchart for this process is shown in Figure 7. If all five types are sent, the last register set alternates between sending SPD and MPEG InfoFrames. Refer to page 50 for register details on enabling and disabling packet transmission.

For a detailed description of the ACP, ISRC1, and ISRC2 packets, refer to the HDMI 1.3 Standard, sections 5.3.7, 5.3.8, and 5.3.9, respectively.

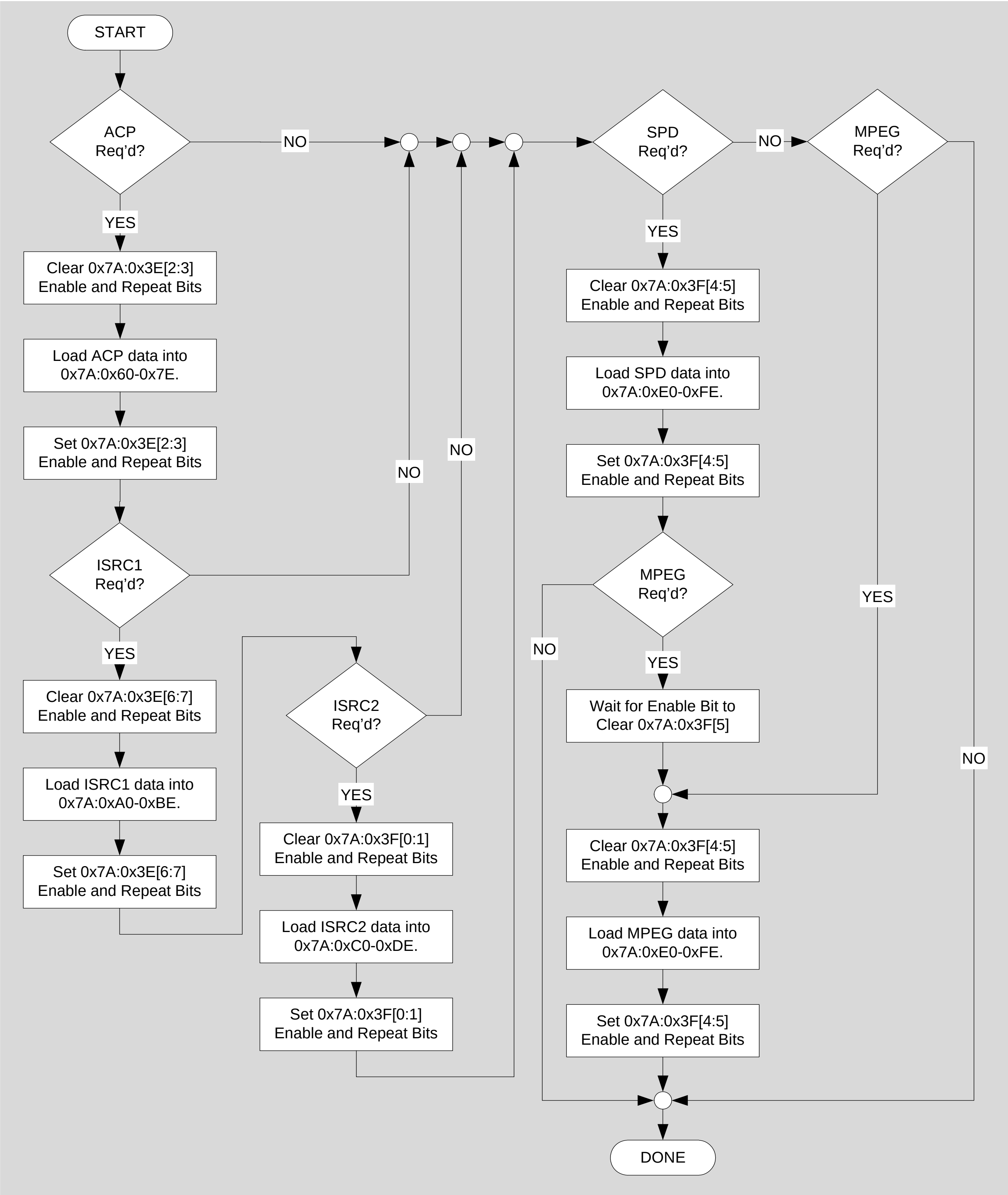

Figure 7. ACP Packet Control Flowchart

Handling Video

Programming Video Input Mode and Video Output Mode

Specific registers must be programmed according to the selection of input video bus mode and output video format. Figure 8 shows video input data processing that leads to TMDS encoding.

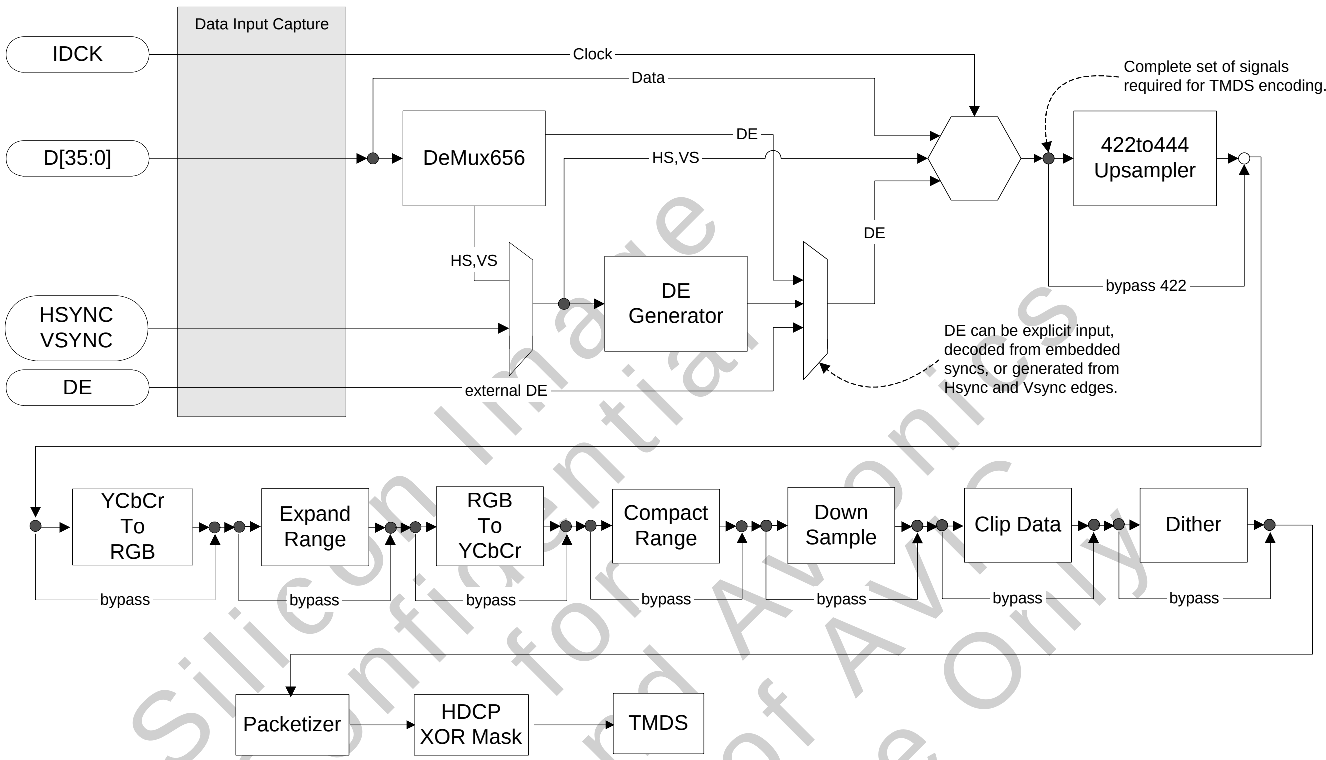

Figure 8. Transmitter Video Data Processing Path

Deep-Color Video Data (SiI9134 Transmitter Only)

The SiI9134 transmitter provides support for deep-color video data. It supports both 30-bit (10 bits per pixel component) and 36-bit (12 bits per pixel component) video input formats, and assembles the data into 8-bit data packets for encryption and TMDS encoding for transfer across the link.

When the width of the input data is more than the data size to be sent, the transmitter can be programmed to dither or truncate the video data to the desired size. For example, if the input data width is 12-bits per pixel component but the sink can only support 10-bits per pixel component, the 12-bit input data can be dithered or truncated to the desired 10-bit output data. See the VID_MODE register (0x72:0x4A[5]) on page 18.

By default, the transmitter does not send deep-color related information to the HDMI receiver. To send deep-color information, set register 0x7A:0x2F[6] to 1 (refer to page 46).

For more information about deep-color, refer to the *SiI9034/9134 HDMI Transmitter Data Sheet.*

Handling Interlaced Video

In interlaced video mode, the timing from VSYNC to pixel data changes from even to odd fields. Also, many MPEG sources do not provide standard timing. The HDMI transmitter must correct this so that the video timing across the HDMI link is compliant with CEA-861D.

Table 6 lists the registers involved in decoding embedded syncs and generating DE for interlaced modes.

Table 6. Registers Used to Handle Interlaced Video

Field	Register	Address	Use
DE_ADJ#	IADJUST	0x72:0x3E[2]	Video interface adjustment.
F2VADJ		0x72:0x3E[1]	VBIT_TO_VSYNC adjustment.
F2VOFST		0x72:0x3E[0]	Set VBIT_TO_VSYNC to increment or decrement.
I_DET	POL_DETECT	0x72:0x3F[2]	Video SYNC interlace detection.
VPOL_DET#		0x72:0x3F[1]	VSYNC polarity detection.
HPOL_DET#		0x72:0x3F[0]	HSYNC polarity detection.
HBIT_2H_SYNC	HBIT_2HSYNC	0x72:0x40–0x41	Video H bit to HSYNC.
FIELD2_OFST	FLD2_HS_OFST	0x72:0x42–0x43	Video Field to HSYNC offset.
HWIDTH	HWIDTH	0x72:0x44–0x45	Video HSYNC length.
VBIT_TO_VSYNC	VBIT_TO_VSYNC	0x72:0x46	Video V bit to VSYNC.
VWIDTH	VWIDTH	0x72:0x47	Video VSYNC length.

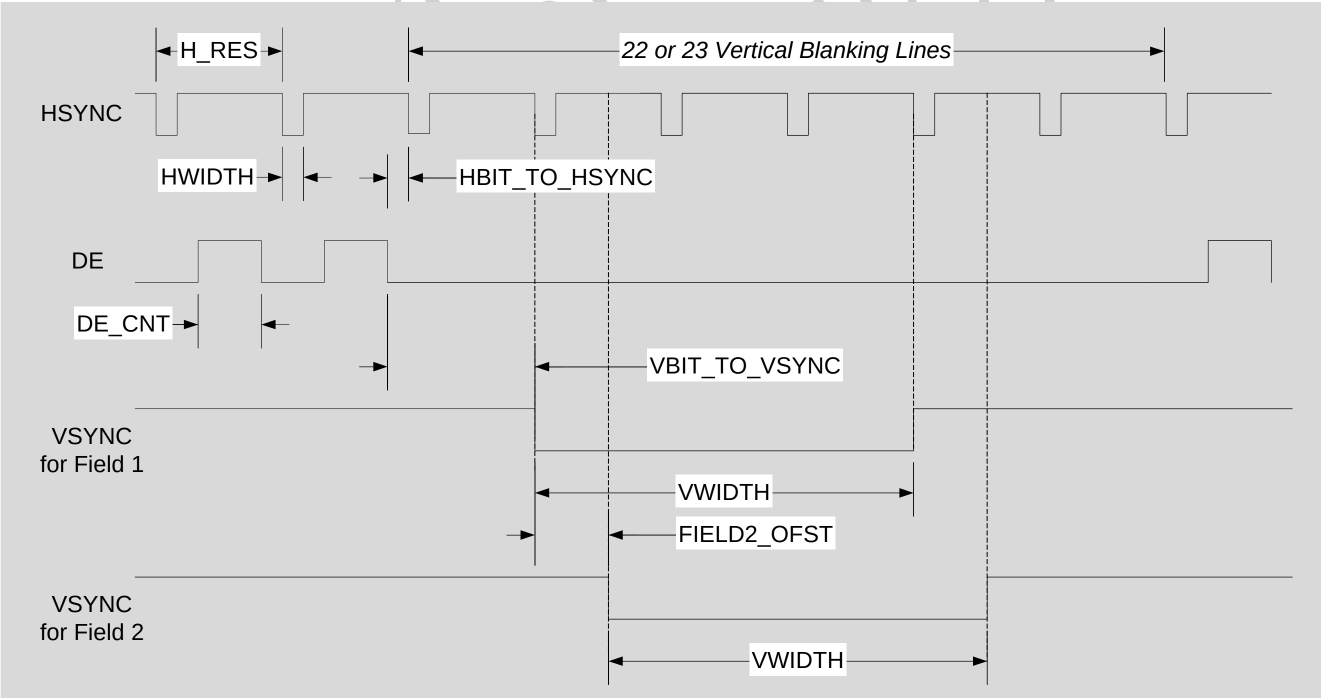

Figure 9. 480i Example for Handling Syncs

Note: Registers 0x72:0x40 through 0x46 are useful only when the input video uses 656 encoded syncs.

Summary of Video Processing Path Options

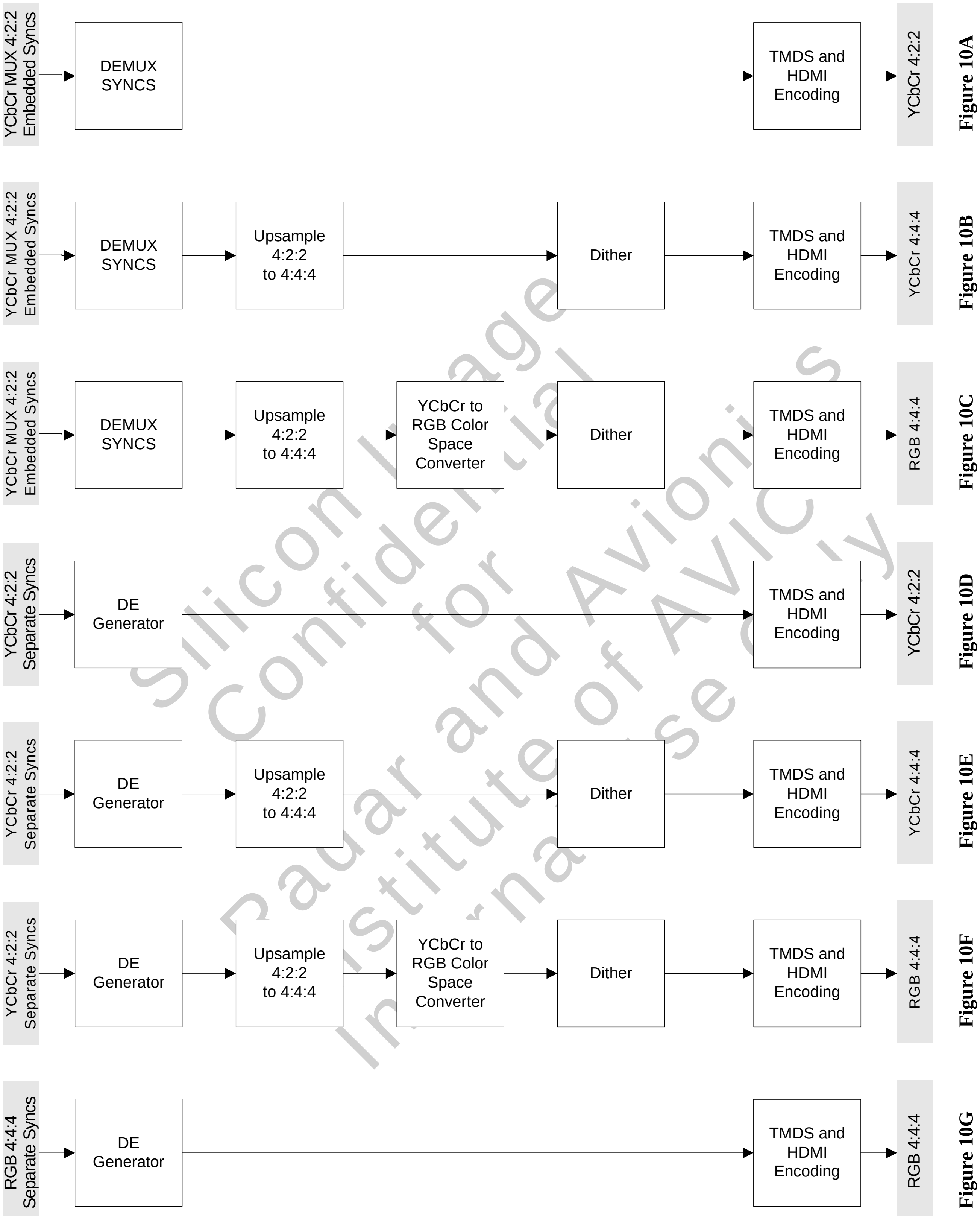

Figure 10. Video Input to Video Output Data Flow

Video Input Tables

These notes apply to the tables that follow:

1. Program the DE parameters only when the DE generator is enabled (register 0x72:0x33[6]).
2. The timings are based on the *CEA-861D Specification*.
3. The parameters are application-dependent. Consult the timing requirements of the source.
4. Set this bit to 1 only when Y and C channels are each 12 bits wide. If these bus widths are less than 12 bits, set this bit to 0 and tie all unused pins to GND.
5. HBIT_TO_HSYNC, FIELD2_OFST, and VBIT_TO_VSYNC may differ depending on the porch timings in the input stream.
6. Set RANGE to 1 whenever converting YCbCr data and sending full-range (0–255) RGB (PC mode) data across HDMI. When sending limited-range (16–235) RGB (CE mode) data, clear RANGE to 0.
7. Set WIDE_BUS to the number of bits per *input* video channel.
8. Set DITHER_MODE to the number of bits per *output* video channel supported by the sink. By default, the HDMI transmitter dithers the input (if DITHER 0x72:0x4A[5] is enabled) or truncates the input (if DITHER is disabled) to 8 bits.

480i Input

480i Multiplexed YCbCr 4:2:2 Embedded Sync Input

Input Mode			**Multiplexed YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0	00 0x13	00 0x13	00 0x13	H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0	001 0xAD	001 0xAD	001 0xAD	Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	000 0x3E	000 0x3E	000 0x3E	HSYNC Pulse Width	2	15
VBIT_TO_VSYNC	0x46	5:0	0x04	0x04	0x04	V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x03	0x03	0x03	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	1	1	1	1-to-2 Chan Demultiplex	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C		—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. Refer to page 71 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

480i Multiplexed YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			**Multiplexed YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x77	0x0 0x77	0x0 0x77	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x12	0x12	0x12	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x0 0xF0	0x0 0xF0	0x0 0xF0	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x13	00 0x13	00 0x13	H Bit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0	001 0xAD	001 0xAD	001 0xAD	Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	000 0x3E	000 0x3E	000 0x3E	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x04	0x04	0x04	V Bit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x03	0x03	0x03	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	1	1	1	1-to-2 Chan Demultiplex	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C		—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 70 for register settings that enable only the sync decoder when the source device provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

480i YCbCr 4:2:2 Multiplexed YC Separate Sync Input

Input Mode			**YCbCr 4:2:2 Mux YC Separate Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x77	0x0 0x77	0x0 0x77	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x12	0x12	0x12	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x0 0xF0	0x0 0xF0	0x0 0xF0	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	1	1	1	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F		—	68

Refer to page 69 for notes.

576i Input

576i YCbCr 4:2:2 Embedded Sync Input

Input Mode			YCbCr 4:2:2 Embedded Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x0C	00 0x0C	00 0x0C	H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0	001 0xB0	001 0xB0	001 0xB0	Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	000 0x3F	000 0x3F	000 0x3F	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x02	0x02	0x02	V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x03	0x03	0x03	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	1	1	1	1-to-2 Chan Demultiplex	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C		—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source. Refer to page 74 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

576i YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			**YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x84	0x0 0x84	0x0 0x84	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x16	0x16	0x16	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x1 0x20	0x1 0x20	0x1 0x20	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x0C	00 0x0C	00 0x0C	H Bit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0	001 0xB0	001 0xB0	001 0xB0	Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	000 0x3F	000 0x3F	000 0x3F	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x02	0x02	0x02	V Bit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x03	0x03	0x03	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	1	1	1	1-to-2 Chan Demultiplex	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 73 for register settings that enable only the sync decoder when the source device provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

576i YCbCr 4:2:2 Multiplexed YC Separate Sync Input

Input Mode			**YCbCr 4:2:2 Mux YC Separate Syncs**					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x84	0x0 0x84	0x0 0x84	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x16	0x16	0x16	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x1 0x20	0x1 0x20	0x1 0x20	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	1	1	1	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	C	—	—	68

Refer to page 69 for notes.

480p Input

480p RGB Input

Input Mode			RGB					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0			0x0 0x7A	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0			0x24	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0			0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0			0x1 0xE0	DE Lines	1, 2	13
HS_POL#	0x33	4			1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5			1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6			1	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	14
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_TO_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0			00	Pixel Replication	—	16
CSCSEL	0x48	4				Color Space Select	—	16
EXTN	0x48	5			0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0			0	Sync Extraction	—	18
DEMUX	0x4A	1			0	—	—	18
UPSMP	0x4A	2			0	Up sampling	—	18
CSC	0x4A	3			0	Color Space Convert	—	18
RANGE	0x4A	4				Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5			0	Dither Enable	—	18
Data Flow Diagram (Figure 10)					G	—	—	68

Refer to page 69 for notes.

480p YCbCr 4:4:4 Separate Sync Input

Input Mode			YCbCr 4:4:4 Separate Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 6:0		0x0 0x7A	0x0 0x7A	DE Delay	1, 2, 3	12
DE_TOP	0x34	7:0		0x24	0x24	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0		0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0		0x1 0xE0	0x1 0xE0	DE Lines	1, 2	13
HS_POL#	0x33	4		1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5		1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6		1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0		00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5		0	0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0		0	0	Sync Extraction	—	18
DEMUX	0x4A	1		0	0	—	—	18
UPSMP	0x4A	2		0	0	Up sampling	—	18
CSC	0x4A	3		0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5		0	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

480p YCbCr 4:2:2 Separate Sync Input

Input Mode			YCbCr 4:2:2 Separate Syncs					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 6:0	0x0 0x7A	0x0 0x7A	0x0 0x7A	DE Delay	1, 2, 3	12
DE_TOP	0x34	7:0	0x24	0x24	0x24	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x1 0xE0	0x1 0xE0	0x1 0xE0	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

480p YCbCr 4:2:2 Embedded Sync Input

Input Mode			**YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x10	00 0x10	00 0x10	H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x3E	0x0 0x3E	0x0 0x3E	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x09	0x09	0x09	V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x06	0x06	0x06	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. Refer to page 80 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

480p YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			**YCbCr 4:2:2 Embedded Syncs**						
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**	**BTA-T1004**			
Register								**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x7A	0x0 0x7A	0x0 0x7A	0x0 0x7A	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x24	0x24	0x24	0x24	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	0x02 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x1 0xE0	0x1 0xE0	0x1 0xE0	0x01 0xE0	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x10	00 0x10	00 0x10	00 0x10	H Bit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0				011 0x5A	Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x3E	0x0 0x3E	0x0 0x3E	0x0 0x3E	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x09	0x09	0x09	0x09	V Bit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x06	0x06	0x06	0x06	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	1	1-to-2 Chan Demultiplex	—	18
UPSMP	0x4A	2	0	1	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	1	Color Space Convert	—	18
RANGE	0x4A	4			1	1	Range Select	6	18
WIDE_BUS	0x49	7:6					Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6					Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 79 for register settings that enable only the sync decoder when the source device provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

576p Input

576p RGB Input

Input Mode			RGB					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0			0x0 0x84	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0			0x2C	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0			0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0			0x2 0x40	DE Lines	1, 2	13
HS_POL#	0x33	4			1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5			1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6			1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0			00	Pixel Replication	—	16
CSCSEL	0x48	4				Color Space Select	—	16
EXTN	0x48	5			0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0			0	Sync Extraction	—	18
DEMUX	0x4A	1			0	—	—	18
UPSMP	0x4A	2			0	Up sampling	—	18
CSC	0x4A	3			0	Color Space Convert	—	18
RANGE	0x4A	4				Range Select	—	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5			0	Dither Enable	—	18
Data Flow Diagram (Figure 10)					G	—	—	68

Refer to page 69 for notes.

576p YCbCr 4:4:4 Separate Sync Input

Input Mode			YCbCr 4:4:4 Separate Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0		0x0 0x84	0x0 0x84	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0		0x2C	0x2C	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0		0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0		0x2 0x40	0x2 0x40	DE Lines	1, 2	13
HS_POL#	0x33	4		1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5		1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6		1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0		00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5		0	0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0		0	0	Sync Extraction	—	18
DEMUX	0x4A	1		0	0	—	—	18
UPSMP	0x4A	2		0	0	Up sampling	—	18
CSC	0x4A	3		0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5		0	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

576p YCbCr 4:2:2 Separate Sync Input

Input Mode			**YCbCr 4:2:2 Separate Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x84	0x0 0x84	0x0 0x84	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x2C	0x2C	0x2C	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x2 0x40	0x2 0x40	0x2 0x40	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

576p YCbCr 4:2:2 Embedded Sync Input

Input Mode			**YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x0C	00 0x0C	00 0x0C	H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x40	0x0 0x40	0x0 0x40	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x05	0x05	0x05	V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. Refer to page 85 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

576p YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			YCbCr 4:2:2 Embedded Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0x84	0x0 0x84	0x0 0x84	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x2C	0x2C	0x2C	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x2 0x40	0x2 0x40	0x2 0x40	DE Lines	1, 2	13
HS_POL#	0x33	4	1	1	1	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	1	1	1	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x0C	00 0x0C	00 0x0C	H Bit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x40	0x0 0x40	0x0 0x40	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x05	0x05	0x05	V Bit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			0	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 84 for register settings that enable only the sync decoder when the source device provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

720p Input

720p RGB Input

Input Mode			RGB					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0			0x1 0x04	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0			0x19	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0			0x5 0x00	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0			0x2 0xD0	DE Lines	1, 2	13
HS_POL#	0x33	4			0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5			0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6			1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0			00	Pixel Replication	—	16
CSCSEL	0x48	4				Color Space Select	—	16
EXTN	0x48	5			0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0			0	Sync Extraction	—	18
DEMUX	0x4A	1			0	—	—	18
UPSMP	0x4A	2			0	Up sampling	—	18
CSC	0x4A	3			0	Color Space Convert	—	18
RANGE	0x4A	4				Range Select	—	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5			0	Dither Enable	—	18
Data Flow Diagram (Figure 10)					G	—	—	68

Refer to page 69 for notes.

720p YCbCr 4:4:4 Separate Sync Input

Input Mode			YCbCr 4:4:4 Separate Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0		0x1 0x04	0x1 0x04	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0		0x19	0x19	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0		0x5 0x00	0x5 0x00	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0		0x2 0xD0	0x2 0xD0	DE Lines	1, 2	13
HS_POL#	0x33	4		0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5		0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6		1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0		00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5		0	0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0		0	0	Sync Extraction	—	18
DEMUX	0x4A	1		0	0	—	—	18
UPSMP	0x4A	2		0	0	Up sampling	—	18
CSC	0x4A	3		0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5		0	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

720p YCbCr 4:2:2 Separate Sync Input

Input Mode			**YCbCr 4:2:2 Separate Syncs**					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x1 0x04	0x1 0x04	0x1 0x04	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x19	0x19	0x19	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x5 0x00	0x5 0x00	0x5 0x00	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

720p YCbCr 4:2:2 Embedded Sync Input

Input Mode			**YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x6E	00 0x6E	00 0x6E	H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x28	0x0 0x28	0x0 0x28	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x05	0x05	0x05	V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. Refer to page 90 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

720p YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			**YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x1 0x04	0x1 0x04	0x1 0x04	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x19	0x19	0x19	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x5 0x00	0x5 0x00	0x5 0x00	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x2 0xD0	0x2 0xD0	0x2 0xD0	DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x6E	00 0x6E	00 0x6E	H Bit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x28	0x0 0x28	0x0 0x28	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x05	0x05	0x05	V Bit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 89 for register settings that enable only the sync decoder when the source device provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

1080i Input

1080i RGB Input

Input Mode			**RGB**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0			0x0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0			0x14	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0			0x7 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0			0x2 0x1C	DE Lines	1, 2	13
HS_POL#	0x33	4			0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5			0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6			1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0			00	Pixel Replication	—	16
CSCSEL	0x48	4				Color Space Select	—	16
EXTN	0x48	5			0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0			0	Sync Extraction	—	18
DEMUX	0x4A	1			0	—	—	18
UPSMP	0x4A	2			0	Up sampling	—	18
CSC	0x4A	3			0	Color Space Convert	—	18
RANGE	0x4A	4				Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5			0	Dither Enable	—	18
Data Flow Diagram (Figure 10)					G	—	—	68

Refer to page 69 for notes.

1080i YCbCr 4:4:4 Separate Sync Input

Input Mode			**YCbCr 4:4:4 Separate Syncs**					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0		0x0 0xC0	0x0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0		0x14	0x14	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0		0x7 0x80	0x7 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0		0x2 0x1C	0x2 0x1C	DE Lines	1, 2	13
HS_POL#	0x33	4		0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5		0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6		1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0		00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5		0	0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0		0	0	Sync Extraction	—	18
DEMUX	0x4A	1		0	0	—	—	18
UPSMP	0x4A	2		0	0	Up sampling	—	18
CSC	0x4A	3		0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5		0	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

1080i YCbCr 4:2:2 Separate Sync Input

Input Mode			YCbCr 4:2:2 Separate Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0xC0	0x0 0xC0	0x0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x14	0x14	0x14	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x7 0x80	0x7 0x80	0x7 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x2 0x1C	0x2 0x1C	0x2 0x1C	DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0				H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0				V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

1080i YCbCr 4:2:2 Embedded Sync Input

Input Mode			YCbCr 4:2:2 Embedded Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x58	00 0x58	00 0x58	H Bit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0	0x4 0x4C	0x4 0x4C	0x4 0x4C	Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x2C	0x0 0x2C	0x0 0x2C	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x02	0x02	0x02	V Bit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. Refer to page 95 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

1080i YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			**YCbCr 4:2:2 Embedded Syncs**					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	3:0 7:0	0x0 0xC0	0x0 0xC0	0x0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x14	0x14	0x14	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x7 0x80	0x7 0x80	0x7 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x2 0x1C	0x2 0x1C	0x2 0x1C	DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_2_HSYNC	0x41 0x40	1:0 7:0	00 0x58	00 0x58	00 0x58	H Bit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0	0x4 0x4C	0x4 0x4C	0x4 0x4C	Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x2C	0x0 0x2C	0x0 0x2C	HSYNC Pulse Width	2	15
VBIT_2_VSYNC	0x46	5:0	0x02	0x02	0x02	V Bit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	Dither Enable	—	18
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 94 for register settings that enable only the sync decoder when the source device provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

1080p Input

1080p RGB Input

Input Mode			RGB					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	1:0 7:0			0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0			0x29	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0			0x07 0x80	DE Count	1, 2	13
DE_LIN	0x39	2:0			0x04	DE Lines	1, 2	13
	0x38	7:0			0x38			
HS_POL#	0x33	4			0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5			0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6			1	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0				HBit to HSYNC Delay	2, 5	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2,	15
VBIT_TO_VSYNC	0x46	5:0				VBit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0			00	Pixel Replication	—	16
CSCSEL	0x48	4				Color Space Select	—	16
EXTN	0x48	5			0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0			0	Sync Extraction	—	18
DEMUX	0x4A	1			0	—	—	18
UPSMP	0x4A	2			0	Up sampling	—	18
CSC	0x4A	3			0	Color Space Convert	—	18
RANGE	0x4A	4				Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5			0	10-to-8 Bit Dithering	—	18
Data Flow Diagram (Figure 10)					G	—	—	68

Refer to page 69 for notes.

1080p YCbCr 4:4:4 Separate Sync Input

Input Mode			YCbCr 4:4:4 Separate Syncs					
Output Mode			**YCbCr 4:2:2**	**YCbCr 4:4:4**	**RGB**			
Register							**Notes**	**Pg**
DE_DLY	0x33 0x32	1:0 7:0		0 0xC0	0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0		0x29	0x29	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0		0x07 0x80	0x07 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0		0x04 0x38	0x04 0x38	DE Lines	1, 2	13
HS_POL#	0x33	4		0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5		0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6		1	1	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0				HBit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_TO_VSYNC	0x46	5:0				VBit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0		00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5		0	0	Extended Bit Mode	4	16
SYNCEXT	0x4A	0		0	0	Sync Extraction	—	18
DEMUX	0x4A	1		0	0	—	—	18
UPSMP	0x4A	2		0	0	Up sampling	—	18
CSC	0x4A	3		0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5		0	1	10-to-8 Bit Dithering	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

1080p YCbCr 4:2:2 Separate Sync Input

Input Mode			YCbCr 4:2:2 Separate Syncs					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	1:0 7:0	0 0xC0	0 0xC0	0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x29	0x29	0x29	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x07 0x80	0x07 0x80	0x07 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x04 0x38	0x04 0x38	0x04 0x38	DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0				HBit to HSYNC Delay	2	15
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0				HSYNC Pulse Width	2	15
VBIT_TO_VSYNC	0x46	5:0				VBit to VSYNC Delay	2	15
VWIDTH	0x47	5:0				VSYNC Pulse Width	2	16
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	0	0	0	Sync Extraction	—	18
DEMUX	0x4A	1	0	0	0	—	—	18
UPSMP	0x4A	2	0	1	1	Up sampling	—	18
CSC	0x4A	3	0	0	1	Color Space Convert	—	18
RANGE	0x4A	4			1	Range Select	6	18
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	10-to-8 Bit Dithering	—	18
Data Flow Diagram (Figure 10)			D	E	F	—	—	68

Refer to page 69 for notes.

1080p YCbCr 4:2:2 Embedded Sync Input

Input Mode			YCbCr 4:2:2 Embedded Syncs					
Output Mode			YCbCr 4:2:2	YCbCr 4:4:4	RGB			
Register							Notes	Pg
DE_DLY	0x33 0x32	1:0 7:0				DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0				DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0				DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0				DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	0	0	0	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0	00 0x6E	00 0x6E	00 0x6E	HBit to HSYNC Delay	2	14
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x28	0x0 0x28	0x0 0x28	HSYNC Pulse Width	2	15
VBIT_TO_VSYNC	0x46	5:0	0x05	0x05	0x05	VBit to VSYNC Delay	2	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	15
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	17
DEMUX	0x4A	1	0	0	0	—	—	17
UPSMP	0x4A	2	0	1	1	Up sampling	—	17
CSC	0x4A	3	0	0	1	Color Space Convert	—	17
RANGE	0x4A	4			1	Range Select	6	17
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	10-to-8 Bit Dithering	—	17
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source. Refer to page 100 for register settings that enable both the sync decoder and the DE generator.

Refer to page 69 for additional notes.

1080p YCbCr 4:2:2 Embedded Sync Input – Adjustment for CEA-861D

Input Mode			YCbCr 4:2:2 Embedded Syncs					
Output Mode			**YCbCr**	**YCbCr**	**RGB**			
Register			**4:2:2**	**4:4:4**			**Notes**	**Pg**
DE_DLY	0x33 0x32	1:0 7:0	0 0xC0	0 0xC0	0 0xC0	DE Delay	1, 2, 3	12
DE_TOP	0x34	6:0	0x29	0x29	0x29	DE Top	1, 2, 3	12
DE_CNT	0x37 0x36	3:0 7:0	0x07 0x80	0x07 0x80	0x07 0x80	DE Count	1, 2	13
DE_LIN	0x39 0x38	2:0 7:0	0x04 0x38	0x04 0x38	0x04 0x38	DE Lines	1, 2	13
HS_POL#	0x33	4	0	0	0	HSYNC Polarity	2, 3	12
VS_POL#	0x33	5	0	0	0	VSYNC Polarity	2, 3	12
DE_GEN	0x33	6	1	1	1	DE_GEN Enable	—	12
HBIT_TO_HSYNC	0x41 0x40	1:0 7:0	00 0x6E	00 0x6E	00 0x6E	HBit to HSYNC Delay	2, 5	14
FIELD2_OFST	0x43 0x42	3:0 7:0				Odd Field Offset	2, 5	15
HWIDTH	0x45 0x44	1:0 7:0	0x0 0x28	0x0 0x28	0x0 0x28	HSYNC Pulse Width	2	15
VBIT_TO_VSYNC	0x46	5:0	0x05	0x05	0x05	VBit to VSYNC Delay	2, 5	15
VWIDTH	0x47	5:0	0x05	0x05	0x05	VSYNC Pulse Width	2	15
ICLK	0x48	1:0	00	00	00	Pixel Replication	—	16
CSCSEL	0x48	4			1	Color Space Select	—	16
EXTN	0x48	5	1	1	1	Extended Bit Mode	4	16
SYNCEXT	0x4A	0	1	1	1	Sync Extraction	—	17
DEMUX	0x4A	1	0	0	0	—	—	17
UPSMP	0x4A	2	0	1	1	Up sampling	—	17
CSC	0x4A	3	0	0	1	Color Space Convert	—	17
RANGE	0x4A	4			1	Range Select	6	17
WIDE_BUS	0x49	7:6				Bits per Input Video Channel	7	17
DITHER_MODE	0x4A	7:6				Bits per Output Video Channel	8	18
DITHER	0x4A	5	0	1	1	10-to-8 Bit Dithering	—	17
Data Flow Diagram (Figure 10)			A	B	C	—	—	68

The video reconstructed by the transmitter may not be compliant with CEA-861D timings, depending on the sync timings received from the source. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861D compliant. Refer to page 99 for register settings that enable only the sync decoder when the source provides compliant SAV/EAV timings.

Refer to page 69 for additional notes.

Handling InfoFrames and General Control Packets

AVI InfoFrames

The following descriptions are adapted from the *CEA-861D Specification* and from the *HDMI Specification*. AVI InfoFrame packets are enabled by AVI_EN and AVI_RPT in the Packet Buffer Control #1 Register (0x7A:0x3E), described on page 51.

Table 7. AVI InfoFrame Layout

			Bit Number								
Reg.			**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Notes**
0x40	**0**	**Type Code**	0x82								1
0x41	**1**	**Version**	0x02								2
0x42	**2**	**Length**	0x0D								3
0x43	**3**	**Checksum**									4
0x44	**4**	**Data Byte 1**	0	Y1	Y0	A0	B1	B0	S1	S0	5, 6
0x45	**5**	**Data Byte 2**	C1	C0	M1	M0	R3	R2	R1	R0	
0x46	**6**	**Data Byte 3**	0	0	0	0	0	0	SC1	SC0	
0x47	**7**	**Data Byte 4**	0	VIC6	VIC5	VIC4	VIC3	VIC2	VIC1	VIC0	
0x48	**8**	**Data Byte 5**	0	0	0	0	PR3	PR2	PR1	PR0	
0x49	**9**	**Data Byte 6**	Bar info for Letterbox and Pillarbox transmissions								
0x4A	**10**	**Data Byte 7**									
0x4B	**11**	**Data Byte 8**									
0x4C	**12**	**Data Byte 9**									
0x4D	**13**	**Data Byte 10**									
0x4E	**14**	**Data Byte 11**									
0x4F	**15**	**Data Byte 12**									
0x50	**16**	**Data Byte 13**									
0x51	**17**	**Data Byte 14**									7
0x52	**18**	**Data Byte 15**									

Notes:

1. Although the *CEA-861D Specification* defines TYPE as 0x02, the *HDMI Specification* defines TYPE as 0x82 with bit 7 set.
2. VERSION 0x02 defines this as a CEA-861D AVI InfoFrame.
3. LENGTH should be set to 13 (0x0D). Additional data bytes are ignored. See the *HDMI Specification*.
4. CHECKSUM is calculated so that the modulo-256 sum of {TYPE + VERSION + LENGTH + CHECKSUM + Data Bytes 1 to LENGTH} is zero. See the *HDMI Specification*.
5. Shaded bits are RESERVED and are not to be set to other values.
6. Refer to CEA-861D, Sections 6.1.2–6.1.3 for more details on the labeled fields in Data Bytes 1 through 13. The fields are summarized in Table 8, below.
7. CEA-861D defines only 13 data bytes. HDMI defines 27 data bytes. The HDMI transmitter has 15 data byte registers. Load data bytes 14 and 15 with 0x00.

Table 8. AVI InfoFrame Field Settings

Field				Definition
Y1		**Y0**		**Color Space**
0		0		RGB
0		1		YCbCr 4:2:2
1		0		YCbCr 4:4:4
1		1		Future
		A0		**Active Format Information Present**
		0		No format data present.
		1		Active format identification data is present in the AVI InfoFrame.
B1		**B0**		**Bar Info**
0		0		Bar data not valid.
0		1		Vertical bar info valid.
1		0		Horizontal bar info valid.
1		1		Both vertical and horizontal bar info valid.
S1		**S0**		**Scan Information**
0		0		No data.
0		1		Over scanned (television).
1		0		Under scanned (computer).
1		1		Future
SC1		**SC0**		**Non-Uniform Picture Scaling**
0		0		No known non-uniform scaling.
0		1		Picture has been scaled horizontally.
1		0		Picture has been scaled vertically.
1		1		Picture has been scaled both horizontally and vertically.
C1		**C0**		**Colorimetry**
0		0		No data.
0		1		SMTPE 170M and ITU 601 (for standard definition TV)
1		0		ITU 709 (for advanced and high definition TV)
1		1		Future
M1		**M0**		**Picture Aspect Ratio**
0		0		No data.
0		1		4:3
1		0		16:9
1		1		Future
R3	**R2**	**R1**	**R0**	**Active Format Aspect Ratio**
1	0	0	0	Same as Picture Aspect Ratio (M1:M0 field).
1	0	0	1	4:3
1	0	1	0	16:9
1	0	1	1	14:9
other values				As specified by the DVB AFD active_format field.

General Control Packets

General Control Packets control the flow of video and audio data across the HDMI link. The General Control Packet is defined in the *HDMI Specification*. This data byte is controlled in the transmitter with the GCP_BYTE1 register (0x7A:0xDF), described on page 58. The General Control Packet is transmitted only during the vertical blanking period, after the active edge of VSYNC.

To change the content of the GCP_BYTE1 register, GCP_EN must be zero. The firmware should keep GCP_EN cleared until the desired value is written into SET_AVMUTE and CLR_AVMUTE. Then GCP_EN and GCP_RPT should be set to 1 (refer to page 50 more information about EN and RPT bits). The SET_AVMUTE bit and CLR_AVMUTE bit cannot both be set at the same time (refer to page 58). Because the General Control Packet is synchronized to VSYNC, the action from SET_AVMUTE and CLR_AVMUTE takes effect immediately after the next VSYNC pulse.

Table 9. Mute Actions for Setting or Clearing SET_AVMUTE and CLR_AVMUTE Bits

SET_AVMUTE	CLR_AVMUTE	Action
0	0	Default setting after RESET#. Same action as SET_AVMUTE = 0 with CLR_AVMUTE = 1.
0	1	Allow pixel data and audio sample data to pass across the link.
1	0	The transmitter sends a General Control Packet on the TMDS link to inform the sink that the data may be incorrect, and sends blank level data for all video packets and 0x00 for all audio packet data.
1	1	*NOT ALLOWED BY HDMI SPECIFICATION.*

When the HDCP link fails, the transmitter must carefully manage the video and audio content to minimize the disruption of video output from the receiver. The firmware should follow the process described in Figure 12 on page 105.

When transmitting in DVI mode, setting SET_AVMUTE to 1 changes the video content to 0x00. CTL3 pulses stop at the next frame, so the receiver stops decrypting and sends the 0x00 data as a blank screen. Although HDMI receivers reset their HDCP engines with a new authentication (at the writing of the last byte of AKSV), earlier DVI receivers may not do so. To guarantee interoperability with all DVI-HDCP sinks, the source firmware should stop the video signaling before beginning a new authentication. This causes the receiver to create an SCDT event, which resets the receiver HDCP engine. The new authentication can then complete normally, and video transmission and reception resumes with a decrypted picture.

When a Link Integrity check fails in DVI mode, the transmitter should stop the video signaling briefly before beginning a new authentication. This break in DE causes an SCDT event in the DVI-HDCP receiver, which resets its HDCP engine.

Initiating HDMI Video after Hardware Reset and Successful HDCP Authentication

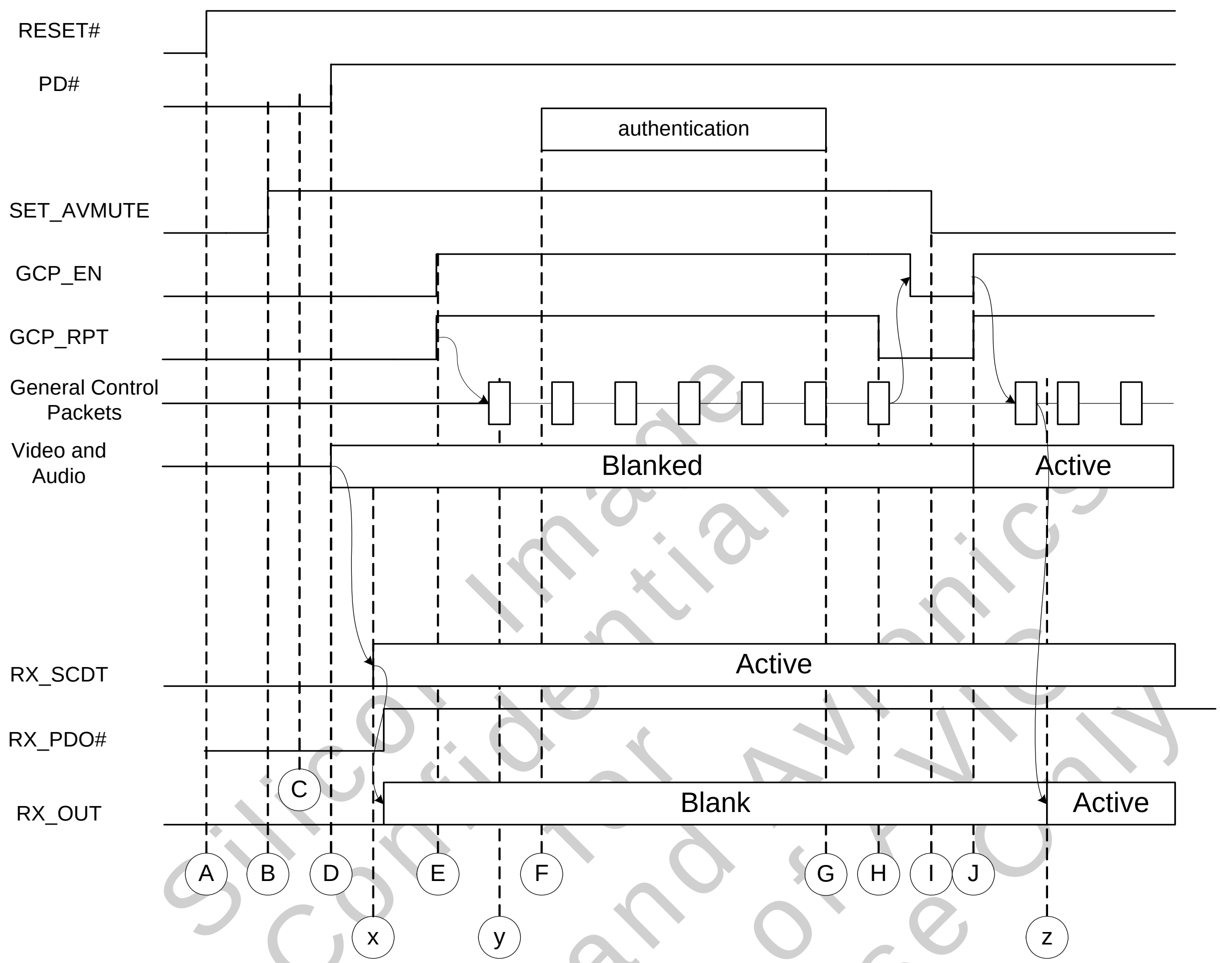

Figure 11. Muting Video and Audio from Reset to Authentication

Transmitter Actions

A Transmitter comes out of hardware reset.

B Assert SET_AVMUTE to blank video and audio when they become active.

C Set up video path and audio path before powering on.

D Set PD# = 1 to enable video and audio output, which is blanked.

E Set GCP_EN and GCP_RPT bits for General Control Packets, which flow to receiver at next VSYNC.

F Source begins authentication.

G Authentication completes successfully.

H Clear GCP_RPT, wait for GCP_EN = 0.

I Set CLR_AVMUTE = 1.

J Set GCP_EN and GCP_RPT together. The transmitter enables video output and begins sending new General Control Packets with CLR_AVMUTE.

Receiver Actions

x The receiver chip senses the start of HDMI signaling and asserts its SCDT. The sink's firmware recognizes the SCDT = 1 state, and enables the receiver chip's video and audio output pins.

y The receiver chip detects a General Control Packet with SET_AVMUTE = 1 and automatically blanks the video output.

z The receiver chip detects a General Control Packet with CLR_AVMUTE = 1 and automatically begins decrypting and providing active content.

If the receiver chip senses that HDMI signaling has stopped (when the HDMI transmitter is reset), the receiver firmware blanks the video output. The receiver enables content at the video outputs only after it receives active video signaling, completes HDCP authentication, and receives a CLR_AVMUTE General Control Packet.

Controlling HDMI Video through an HDCP Link Failure and Re-Authentication

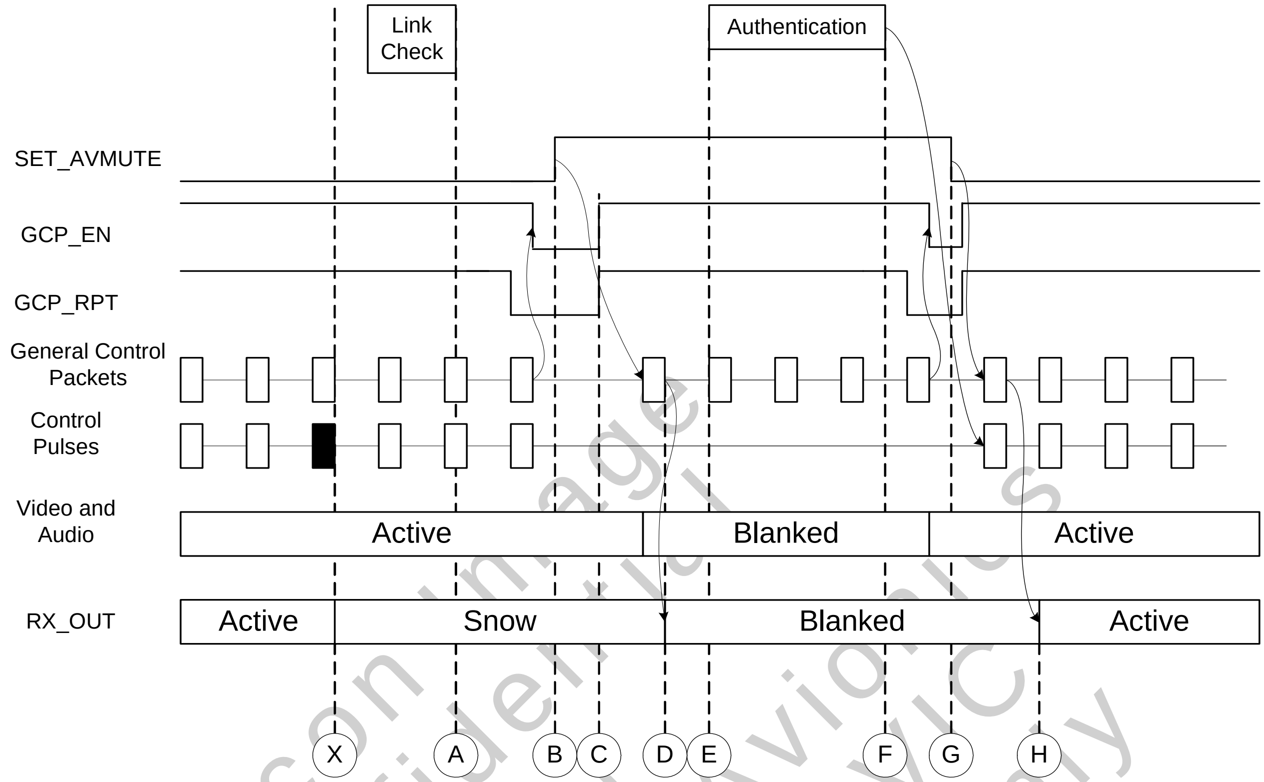

Figure 12. Muting Video and Audio from Link Failure to Authentication

X Extra or missing control pulse gets decryption out of step with encryption, and the receiver produces snow.

A Transmitter firmware fails Link Integrity check of polling of R_i.

B Clear GCP_RPT and wait for the transmitter to reset GCP_EN to 0.

C Set SET_AVMUTE = 1, then set GCP_EN and GCP_RPT. New General Control Packets are sent.

D Receiver senses SET_AVMUTE and sends BLANKLEVEL video.

E Source firmware begins authentication. BKSV write resets the transmitter HDCP engine. AKSV write resets the receiver HDCP engine.

F Successful authentication begins encryption and decryption on the next frame. Firmware clears GCP_RPT and waits for GCP_EN = 0.

G Write SET_AVMUTE = 0, then set GCP_EN = 1 and GCP_RPT = 1 again.

H Video content resumes from the transmitter on the next frame and is decrypted in the receiver.

During this process, the video from the transmitter is not interrupted. An SCDT event does not occur on the receiver side because sync information continues to arrive from the transmitter. Although snow may appear when the link fails (before the Link Integrity check, but for no longer than 2 seconds), a blank screen in the correct color space quickly replaces it. Live video content resumes smoothly after authentication.

Handling Audio Content Protection (ACP) Packets

HDMI 1.1 defines new packets for handling content protection for audio. Refer to the explanations beginning on page 63.

Operating DDC Master

The transmitter includes a logic block to drive the E-DDC bus, which supports a variety of I^2C commands. The individual registers are described on page 29. The speed of the I^2C clock is determined by an internal oscillator in the transmitter and is not dependent on or a function of any input pixel clock. The I^2C frequency does not exceed 100 kHz.

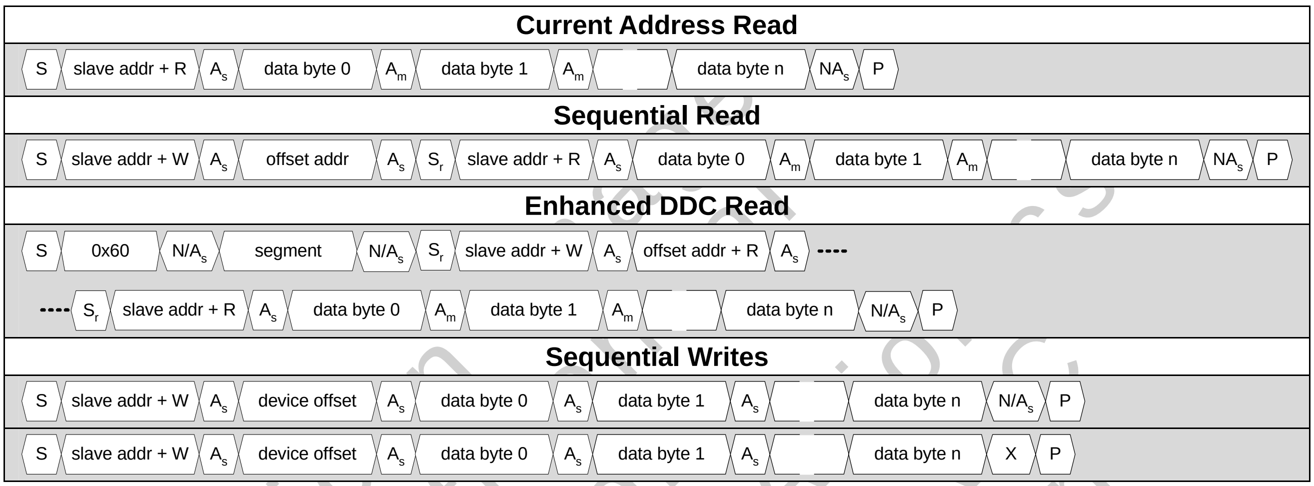

Figure 13. Supported Master I^2C Transactions

Current Address Read: Reads from the last offset address so that no offset needs to be written to the slave. Multiple bytes can be read from consecutive addresses, which increment in the slave with each ACK received from the master.

Sequential Read: Reads from a specific start address, which is sent as a write command to the slave. Multiple bytes may be read from consecutive addresses. Although the FIFO in the transmitter can hold only 16 bytes, the sequential read command can be of any length up to the 10-bit value in the DDC_COUNT register (0x72:0xF0 and 0xF1). A *Stop* bit is sent by the DDC master only when the entire DDC_COUNT is complete.

Enhanced DDC Read: A special command, defined by the *VESA E-DDC Specification*, that writes a segment address to a separate I^2C device address, then sends an offset address to the slave device, and finally reads one or more data bytes beginning from address *256 • segment + offset*. Multiple bytes within the same segment can be read, as the slave increments the offset with each ACK received from the master. The segment register in the slave is reset at the end of each command. A NACK or ACK is required from the slave device if the segment is not zero, but is ignored if the segment is zero. Refer to the *E-DDC Specification*.

Sequential Write: Similar to the sequential read, this command sends one or more bytes to the slave, beginning at the explicit offset address. Multiple bytes may be written, as the slave increments the address until the master sends a stop bit. Two command op codes are available that either wait for ACK/NACK or ignore ACK/NACK on the last byte.

Device Addresses

Table 10 lists the standardized device addresses for the E-DDC bus, as specified in the E-DDC and HDCP standards.

Table 10. HDCP DDC Standard Device Addresses

Device	Address
EDID PROM	0xA0
E-DDC Segment Address	0x60
HDCP Receiver	0x74

DDC Read Operation

The firmware sets up the read command when it loads values for the device address, the offset address, and the byte count. If the read command is an extended-DDC read, the segment address must also be loaded. After loading these values, the command register is loaded with the read command code.

The transmitter uses a FIFO to hold data read across the DDC bus. Up to 16 bytes can be held in the FIFO, but as many as 1023 bytes can be read in one master DDC operation by setting the overall count in DDC_COUNT. The IN_PROG bit (0x72:0xF2[4]) is cleared when the last byte has been read. Figure 14 illustrates how to perform a *short read* of 16 bytes or less. All *N* bytes can be read from the finished FIFO with one local I^2C read command.

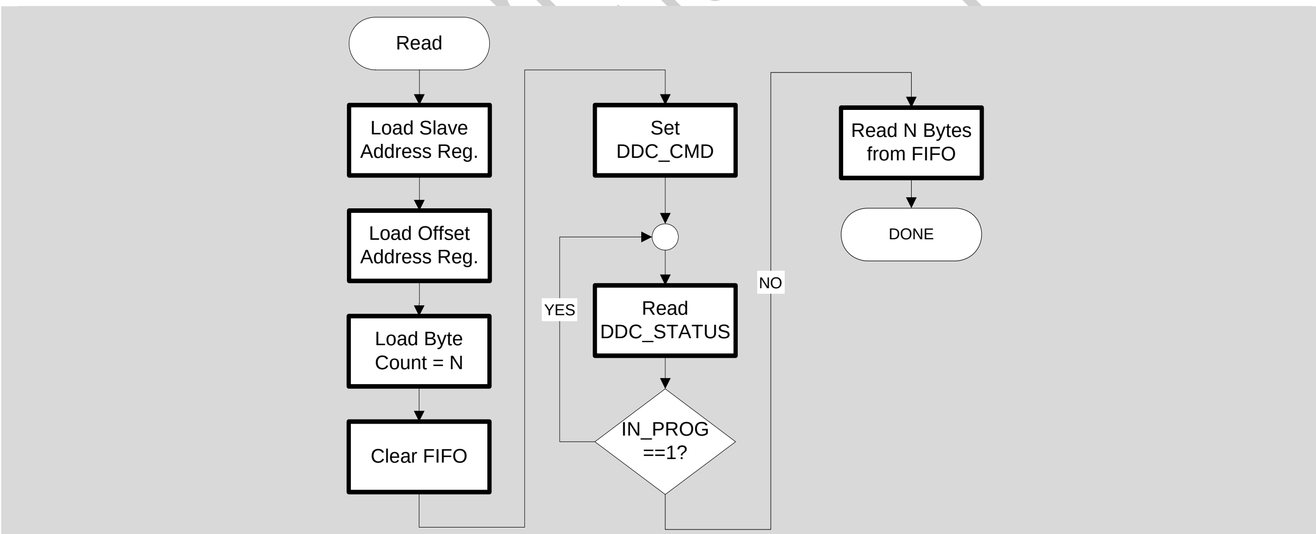

Figure 14. Master I^2C Read Command Flowchart

To perform lengthy, multi-byte read operations that exceed the depth of the physical FIFO, the firmware should poll the status of the master DDC block FIFO by reading the value of the DDC_FIFOCNT register (0x72:0xF5). Each time the FIFO fills up to contain a certain number of bytes (up to 16), the firmware should read that number of bytes, wait for the FIFO to refill, and then repeat the process until the required total number of bytes is read. At that point, the master DDC issues a stop command. The source firmware can pull bytes from the FIFO continuously while the transmitter performs the actual read cycles on the DDC bus.

Silicon Image recommends that the designer add timeout protection in case the FIFO fails to fill up to the required number of bytes.

Note: During such a read, the DDC bus is busy. No other operation can take place on the DDC bus until the read is complete.

Write Command

Write commands are similar to read commands. The FIFO must be written by the firmware before initiating the command, after emptying the FIFO as described above. If fewer than 17 bytes will be written to the DDC channel, the firmware must only load the bytes to the FIFO and then write the DDC_CMD register with the write command code. If more than 16 bytes will be written, the firmware must fill the FIFO, wait for it to begin sending, and then write the remaining bytes into the FIFO without causing it to overflow. *Because neither HDMI nor HDCP require a write of more than 16 bytes, such an operation is not described in this Programmer's Reference.*

Each master I^2C operation begins with writes to several registers in the transmitter. For a write command, the destination device address is followed by the offset address and the byte count. The firmware must also clear the FIFO before writing data to it. When the byte count is complete, the transmitter automatically sends the stop bit to the DDC bus.

Figure 15 shows how to write up to 16 bytes from the firmware to the DDC bus. All the data fits into the FIFO so that a multi-byte I^2C write can be used from the firmware to the transmitter. The WRITE command is then issued to the DDC_CMD register and all data bytes are transferred across the DDC bus. Some type of timeout should be used when checking that the Master DDC module is available, either before starting a new command or before returning from the subroutine after beginning a command.

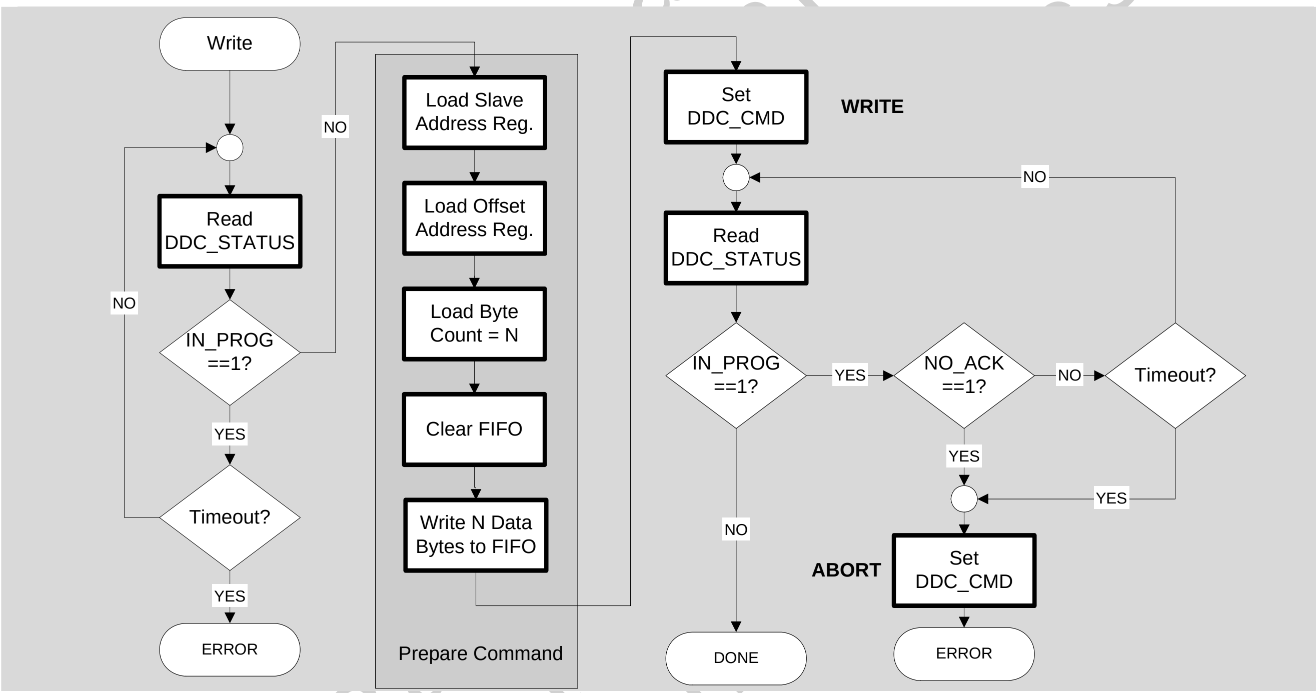

Figure 15. Master I^2C Write Command Flowchart

Figure 15 includes a check before beginning the write command to be certain that the master DDC block is not already in use, and a check after triggering the write command to check for bus hangs. If the write command times out or fails to receive an ACK from the slave device, an abort command is issued.

Abort Command

A read or write command can be interrupted by the firmware at any time by writing an abort command to the DDC_CMD register (0x72:0xF3). The abort command issues a STOP, followed by nine SCK clock cycles, but only if a previous command is incomplete. If the abort command is issued when no other command is in progress, no action occurs on the DDC link. Firmware should allow for DDC bus hangs and include a timeout in the loop that waits for the completion of a write or read command. This can be done by setting up a watchdog timer with an interrupt service routine and aborting the DDC command if the timer expires. It can also be done by polling the DDC_STATUS byte and deciding that the command is stalled, then issuing an abort command.

Setting up the PLL Control Registers

General

The following seven registers need to be set for a proper operation of the HDMI Transmitter's PLL (the register names in this section are the names used in this Programmer's Reference):

- **0x72:0x0C – System Control Register #4** enables/disables the PLL filter and sets its Charge pump current.
- **0x72:0x49 – Video Action Enable Register** defines the input video bus width, and specifically the color depth of the HDMI Transmitter output.
- **0x72:0x80** – **TMDS C Control Register** sets the Filter PLL post counter for the audio clock **FAPOSTCOUNT**.
- **0x72:0x82 - TMDS Control Register #1** sets the multiplication ratio **TCLKSEL** between the FPLL frequency and the input pixel clock (**IDCK**) frequency.
- **0x72:0x83 - TMDS Control Register #2** sets the divider ratio for the HDMI transmitter PLL post counter **POST_COUNT**, the divider ratio for the PLL filter feedback counter **FFB_COUNT**, and the divider ratio for the PLL filter front counter **FFR_COUNT**.
- **0x72:0x84 - TMDS Control Register #3** sets **ITPLL**, which controls the current of the low pass filter (LPF) of the PLL, and the divider ratio for the PLL post counter, **FPOST_COUNT.**
- **0x72:0x85 - TMDS Control Register #4** sets the PLL front counter divide ratio **TFR_COUNT**.

The settings depend on the input pixel clock frequency, which varies with the input resolution and on the required color depth (24/30/36 bits).

PLL Setup Tables

To set the PLL control registers, Table 11 through Table 16 are used.

The **mpll settings** tables define the link (HDMI output) clock frequency as a function of **TPOSTCOUNT** (0x72:0x83[7:6]), **ITPLL** (0x72:0x84[6:3]) and **TFRCOUNT** (0x72:0x85[1:0]).

The **ifpll settings** tables define the following outputs:

- The outputs of the PLL's phase frequency detector (PFD)
- The PLL's VCO
- The output clock

Each of these outputs is a function of some or all of the following:

- The color depth (set in 0x72:0x49[7:6])
- The Link-frequency-to-input-IDCK-multiplication ratio (set in 0x72:0x82[6:5])
- IPLLF (set in 0x72:0x0C[4:1])
- FFRCOUNT (set in 0x72:0x83[2:0])
- FFBCOUNT (set in 0x72:0x83[5:3])
- FPOSTCOUNT (set in 0x72:0x84[1:0]
- FAPOSTCOUNT (set in 0x72:0x80[5]).

Note: The Programmer's Reference, Table 11 through Table 16, and the reference firmware use slightly different names for some of the variables, enumerations and macro definitions that represent similar register fields. This appendix explicitly re-defines these terms, using the register full address and bit-field.

Basic Requirements and Assumptions

The VCO frequency of the PLL (refer to the previous section) must always be between 100 MHz and 250 MHz for proper operation.

In the SiI9134 reference code, **TCLKSEL** (0x72:0x82[6:5]) is always set to 0b10, meaning that **FPLL** is the same as **IDCK** (the input pixel clock). Therefore, the firmware only uses the sections marked as **'1x'** in Table 11 through Table 14 for the PLL setup.

The reference firmware does not use Table 16, **Recommended Setting #2.** Refer to the next section.

Note: The user is free to use any another **TCLKSEL** and **Recommended Setting #2** as needed.

PLL Setup Procedure – Overview

The following steps are implemented by Silicon Image reference firmware to set up the PLL control registers.

Table 15 and Table 16 contain the recommended settings for **ITPLL** (0x72:0x84[6:3]) which sets the PLL LPF frequency response; **TPOSTCOUNT** (0x72:0x83[7:6]) which sets the post count divider; and **TFRCOUNT** (0x72:0x85[1:0]) that sets the PLL front counter. These settings define, for each of the two groups, three frequency ranges for the link (output) clock frequency.

The SiI9034/9134 reference firmware uses Table 15, Recommended Setting #1, and does not use Table 16. Therefore, 0x72:0x84[6:3] (the **ITPLL** field) is set to 0b0110, which sets the charge pump current to 50 µA.

Table 15 and Table 16 define three ranges for the link clock frequency, which is the clock that will be sent from the SiI9134 HDMI output. For 24-bit color depth the link frequency is the standard pixel clock frequency of the format sent from the HDMI transmitter. The link frequency must be multiplied by a factor of 30/24 for 30 bit depth and by 36/24 for 36 bit depth.

The three regions are named **Blue**, **Yellow** and **Orange**. In Table 11 through Table 16, the left column is nicknamed **Orange**, the middle column is nicknamed **Yellow** and the right column is nicknamed **Blue**. The three regions are defined by their lowest and highest frequencies (corner points), which are the link frequencies of 25, 64, 126 and 270 MHz, respectively.

After a region (such as Yellow) is selected, based on the input pixel clock (**IDCK**) frequency and the input color depth, the function **SiI_Mpll_setup()** is called by the PLL setup function **SiI_TMDS_setup**. **SiI_Mpll_setup()** sets 0x72:0x83[7:6] (**tpostcount**) and 0x72:0x85[1:0] (**tfrcount**). It takes the values of these parameters from the header of the corresponding regions of Table 15.

PLL Setup Firmware Implementation

The function **SiI_TMDS_setup()** is the main function called to set the PLL. It takes as a parameter an index into the firmware table **VModeTables[]**, from which it extracts the proper input pixel clock frequency (**IDCK**). It first reads 0x72:0x82[6:5] to find the ratio between the link clock frequency and the input pixel clock. In the current SiI9134 reference firmware that ratio is set to 1 by setting 0x72:0x82[6:5] to 0b10. The local variable tclk is then set to **x1**, which is a macro definition for 0x20.

SiI_TMDS_setup() then reads 0x72:0x49[7:6] to determine the input bus width. This value must already have been set by the host. If using the Silicon Image starter kit, be sure that it is running only with **HDMIGear** 3.11.

SiI_TMDS_setup() sets local variable iLowRange to 25, which is lowest value of the link clock acceptable for HDMI. Based on the value of 0x72:0x49[7:6], the switch statement **switch (bRegVal)** sets the values of local variable **nFFBCOUNT** (that will eventually be written to 0x72:0x83[5:3]) and the values of corner points **iMidRange1**, **iMidRange2**, **iHghRange**.

For a color depth of 24 bits, **iMidRange1**, **iMidRange2**, and **iHghRange** are the corner points as defined in the previous section. However, for color depths of 30 and 36 bits the clock values must be multiplied by 30/24 and 36/24, respectively. The corner points for larger depths must be made lower to select higher PLL multipliers. Therefore, **iMidRange1** becomes 64.84•24/30 = 48.63, and 53 is selected from Table 15. For 36 bits, **iMidRange1** becomes 64•24/36 = 42.66, so 44 is selected. Similarly, **iMidRange2** is 126•24/30 = 93.75 (but 104, which is the first available value higher than 93.75 is selected) and 126•24/36 = 84 (86 which is the closest value above it is selected). Similarly **iHghRange** is set to 203 and 168, respectively.

The value for **nFFBCOUNT**, which needs to be written to 0x72:0x83[5:3], is taken from the header of the corresponding section of Table 11 through Table 14. In the reference firmware, 0x72:0x82[6:5] = 0b10. As a result, only the tables marked as **x1** are used (there are **three** such tables for each color depth, one for the **Orange**, one for the **Yellow** and one for the **Blue** region). If, for example, the frequency range matches the **Orange** region, then, for 24 bit color depth, the **8bit to 8bit; 1x Orange** table should be used. That sets **nFFBCOUNT** to 0b011. For 30 bit color depth the **10bit to 8bit; 1x Orange** table should be used, which sets **nFFBCOUNT** to 0b100, and for 36 bit color depth the **12bit to 8bit; 1x Orange** table should be used, which sets **nFFBCOUNT** to 0b101.

nFPOSTCOUNT, which goes into 0x72:0x84[2:0], is first set to 0x03 which is the correct value for the Blue region, but is adjusted to 1 and 0 for the Yellow and Orange ranges, respectively, if needed, as specified in the headers of the **x1** tables (8-bit to 8-bit, 10-bit to 8-bit, 12-bit to 8-bit) of the three frequency regions.

After the adjusted frequency range (defined by its corner points) is determined, **nFFRCOUNT** (to be written to 0x72:0x83[2:0]) is selected from the header of the proper frequency range table. For each frequency range, **nFFRCOUNT** has the same value for all three possible color depths (0b011 for **Orange**, 0b001 for **Yellow** and 0b000 for **Blue**).

After **nFFRCOUNT** is set, **SiI_TMDS_setup()** calls function **SiI_Mpll_setup()** with the proper frequency range as an argument. That sets 0x72:0x83[7:6] (**tpostcount**) and 0x72:0x85[1:0] (**tfrcount**). Refer to the previous section.

SiI_TMDS_setup() then calls function **SiI_FApost_setup()**, which takes the frequency range, the input pixel clock frequency, and the color depth as parameters. Based on the input frequency range (Blue, for instance) and on the color depth, **SiI_FApost_setup()** takes the value of **nFAPOSTCOUNT** (0x72:0x80[5]) from the corresponding **x1** section of Table 11 through Table 14.

For example, if the input pixel clock is 74.25MHz, and the color depth is set to 36 bits, **SiI_TMDS_setup()** selects the **Yellow** range, because its **iMidRange1** is 44 and **iMidRange2** is 86. The **Yellow** frequency range will be passed as a parameter to **SiI_FApost_setup()**, together with **IDCK** of 74 and color depth of 36 bit (enumerated as 2). Because **IDCK** is more than 58, from the **Yellow** 12bit-to-8bit-1x columns of Table 13, **nFAPOSTCOUNT** is set to 1.

SiI_FApost_setup() then writes the selected value of **nFAPOSTCOUNT** to 0x72:0x80[5].

SiI_TMDS_setup() writes the recommended value of nIPLLF to 0x72:0x0C[4:1] to set the PLL filter charge pump current to 10uA; **nFFRCOUNT** to 0x72:0x83[2:0] and **nFPOSTCOUNT** to 0x72:0x84[2:0].

A Numerical Example

The following example follows the procedure described in the previous section for an input format of 1280 x 720p, and a color depth of 36 bits. Please refer to the listing of function **SiI_TMDS_setup()** below.

For 720p, the input parameter bVMode for the function **SiI_TMDS_setup()** is 2. The function uses it to access input mode table VModeTables[2], to extract the value of the pixel clock for 720p and divide it by 100, setting variable idclk_freq to 74.

Reading 0x82[6:5], the function finds that **FPLL** is the same as **IDCK**, setting tclk to **x1**. This value is later used to calculate **nFPOSTCOUNT**.

The input video bus width (color depth) is found by reading 0x49[7:6].

Based on the color depth found in 0x49[7:6], **SiI_TMDS_setup()** sets:

nFFBCOUNT = 0x05;

iMidRange1 = 44;

iMidRange2 = 86;

iHghRange = 168;

The value of **nFFBCOUNT** is then written to 0x72:0x83[5:3].

IPLLF (0x72:0x0C[4:1]) is set to 1, which is the value listed in the tables.

Because tclk was set to 1, the function initially sets **nFPOSTCOUNT** to 0x03. This value is adjusted later.

SiI_TMDS_setup() then checks between what pair of values (defined in the TMDS setup) the input pixel clock falls. Because idclk_freq is 74 MHz, it is between 44 and 86 MHz, which is the **Yellow** color range for 36-bit color depth. This range dictates (from the **12 bits to 8 bits 1x Yellow** columns of Table 13) that **nFFRCOUNT** (0x72:0x83[2:0]) be 0x03 and **nFPOSTCOUNT** be 0.

SiI_TMDS_setup() calls **SiI_Mpll_setup(yellow)**. **SiI_Mpll_setup** handles the settings defined in Table 15 and Table 16. For Recommended Settings #1 (Table 15), the value of (0x72:0x84[6:3]) is always 6. It also sets both **tpostcount** (0x72:0x83[7:6] and **tfrcount** (0x72:0x85[1:0]) to 0b01, since these are the values defined at the top of the **Yellow** column of the Table 15. **SiI_Mpll_setup()** writes these values to 0x72:0x83[7:6], 0x72:0x84[6:3] and 0x72:0x85[1:0].

SiI_TMDS_setup() then calls **SiI_FApost_setup(yellow, idclk_freq, bRegVal)**, where **yellow** is the second table column, **idclk_freq** is 74 and **bRegVal** is the color depth. In this example, the table region is **Yellow** and the color depth is **36**.

SiI_FApost_setup() then checks if the input pixel clock frequency is higher than 58 MHz, because this is the value in the Yellow column of the **12bit to 8bit;1x** in Table 13 where **FAPOSTCOUNT** (0x72:0x80[5]) needs to be set to 1. For a 74-MHz pixel clock, that is the case, so **SiI_FApost_setup()** sets (0x72:0x80[5] to 1 and returns.

SiI_TMDS_setup() writes the values of **nIPLLF**, **nFFRCOUNT**, **nFPOSTCOUNT** to 0x72:0x0C, 0x72:0x83 and 0x72:0x84 respectively. This step completes the settings of the PLL control registers.

SiI9134 Register Setting Implementation

```
//-------------------------------------------------------------------------------
// SiI_TMDS_setup

//-------------------------------------------------------------------------------
byte SiI_TMDS_setup(byte bVMode)
{
    int idclk_freq, iLowRange, iMidRange1, iMidRange2, iHghRange;
    TCLK_SEL tclk;
    byte bRegVal;
    byte bRegVal2;
    byte nIPLLF, nFFRCOUNT, nFFBCOUNT, nFPOSTCOUNT;

    printf ("[TXVIDP.C](SiI_TMDS_setup): Start...\n");
    idclk_freq = (int) VModeTables[bVMode].PixClk / 100;

bRegVal = ReadByteHDMITXP0 ( TX_TMDS_CTRL_ADDR ) & 0x60;          // get TCLSEL
value from 0x72:0x82[6:5]. In this prgram it is always x1 (==0b01)
switch (bRegVal)
{
        case 0x00: tclk = x0_5; printf ("[TXVIDP.C](SiI_TMDS_setup): 0.5x
tclk\n"); break;
        default:
        case 0x20: tclk = x1;   printf ("[TXVIDP.C](SiI_TMDS_setup): 1.0x
tclk\n"); break;
        case 0x40: tclk = x2;   printf ("[TXVIDP.C](SiI_TMDS_setup): 2.0x
tclk\n"); break;
        case 0x60: tclk = x4;   printf ("[TXVIDP.C](SiI_TMDS_setup): 4.0x
tclk\n"); break;
    }

    bRegVal = ReadByteHDMITXP0 ( VID_ACEN_ADDR );
    bRegVal = bRegVal & (~VID_ACEN_DEEP_COLOR_CLR);         // 0x72:0x49[7:6]
      (bus width => color depth - 24/30/36 bit)
    bRegVal = bRegVal >> 6;

    iLowRange = 25;                 // "Blue" range lower point
switch (bRegVal)
{                                   //
        case SiI_DeepColor_24bit:
            nFFBCOUNT = 0x03;             // 0x72:0x83[5:3]
            iMidRange1 =  64;   // "Blue" upper freq and "Yellow" range lower
freq for 24 bit color depth
            iMidRange2 = 126;   // "Yellow" range upper freq and "Orange" range
lower freq for 24 bit           // color depth
            iHghRange  = 270;    // "Orange" range highest freq for 24 bit color
depth
            break;

case SiI_DeepColor_30bit:
            nFFBCOUNT = 0x04; // 0x72:0x83[5:3]
```

```
            iMidRange1 =  53; // "Blue" range upper freq and "Yellow" range
lower freq for 30 bit
// color depth
iMidRange2 = 104; // "Yellow" range upper freq and and "Orange" range lower freq
for 30 // bit color depth
            iHghRange  = 203; // "Orange" range highest freq for 30 bit color
depth
            break;

        case SiI_DeepColor_36bit:
            nFFBCOUNT = 0x05;       // 0x72:0x83[5:3]
            iMidRange1 =  44; // "Blue" range upper freq and "Yellow" range
lower freq for      // 36 bit color depth
iMidRange2 =  86; // "Yellow" range upper freq and "Orange" range lower freq
for 36 bit color depth
            iHghRange  = 168;       // "Orange" range highest freq for 36 bit
color depth
            break;
    }

    // Set FFBCount field in 0x72:0x83[5:3]:
    bRegVal2 = ReadByteHDMITXP0 ( TX_TMDS_CTRL2_ADDR );
    bRegVal2 &= CLR_BITS_5_4_3;
    bRegVal2 |= (nFFBCOUNT << 3);
    WriteByteHDMITXP0 ( TX_TMDS_CTRL2_ADDR, bRegVal2 );           // 72:83

    nIPLLF = 0x01;
    switch (tclk) {
        case x0_5:  nFPOSTCOUNT = 0x07; break;
        case x1:    nFPOSTCOUNT = 0x03; break;      // This is the value set in
0x72:0x84[2:0]
        case x2:    nFPOSTCOUNT = 0x01; break;
        case x4:    nFPOSTCOUNT = 0x00; break;
    }

    // Out of Range
if ((idclk_freq < iLowRange) || (idclk_freq > iHghRange))
{           // example: iLowRange == 25 (always) for DC 36bit - IHghRange == 168
        return TMDS_SETUP_FAILED;
    }

    // Blue range
if ((idclk_freq >= iLowRange) && (idclk_freq <= iMidRange1))
{
        nFFRCOUNT = 0x00;
        SiI_Mpll_setup(blue);
        SiI_FApost_setup(blue, idclk_freq, bRegVal);
    }
    else
        // Yellow range
        if ((idclk_freq > iMidRange1) && (idclk_freq <= iMidRange2))
{
            if (tclk == x4)
    {
```

```
                return TMDS_SETUP_FAILED;
            }
            nFFRCOUNT = 0x01;
            nFPOSTCOUNT >>= 1;
            SiI_Mpll_setup(yellow);
            SiI_FApost_setup(yellow, idclk_freq, bRegVal);
        }
        else
            // Orange range
            if ((idclk_freq > iMidRange2) && (idclk_freq <= iHghRange))
     {
                if ((tclk == x4) || (tclk == x2))
        {
                    return TMDS_SETUP_FAILED;
               }
                nFFRCOUNT = 0x03;
                nFPOSTCOUNT >>= 2;
                SiI_Mpll_setup(orange);
                SiI_FApost_setup(orange, idclk_freq, bRegVal);
            }

    // TX_SYS_CTRL4_ADDR            72:0x0C
    //  [7:5]   reserved
//  [4:1]   IPLLF = 0x01* - Set 72:0x0C[4:1] to "1" => set the PLL filter charge
pump current to       // 10uA
    // [0]     reserved
 WriteByteHDMITXP0 (TX_SYS_CTRL4_ADDR,   ((ReadByteHDMITXP0(TX_SYS_CTRL4_ADDR) &
0xE1 ) | (nIPLLF << 1))); // 72:0x0C

    // TX_TMDS_CTRL2_ADDR           72:83
    //  [7:6]   TPOSTCOUNT
    //  [5:3]   FFBCOUNT = 0x03*

    //  [2:0]   FFRCOUNT*

       WriteByteHDMITXP0 (TX_TMDS_CTRL2_ADDR,
((ReadByteHDMITXP0(TX_TMDS_CTRL2_ADDR) & 0xF8) | (nFFRCOUNT)));
// Value set after the "switch (bRegVal)" statement in this function.

    // TX_TMDS_CTRL3_ADDR
    //  [7]     reserved
    //  [6:3]   ITPLL
    //  [2:0]   FPOSTCOUNT*
    WriteByteHDMITXP0 (TX_TMDS_CTRL3_ADDR,
((ReadByteHDMITXP0(TX_TMDS_CTRL3_ADDR) & 0xF8)  | (nFPOSTCOUNT)));

    return TMDS_SETUP_PASSED;
}

//--------------------------------------------------------------------------------
-
// SiI_Mpll_setup
// Use "Recommended Setting #1" (table 5)
```

```
//------------------------------------------------------------------------------
-
void SiI_Mpll_setup(byte MpllSet)
{
    byte itpll, tpostcount, tfrcount;

    itpll = 0x06;                               // always
    switch (MpllSet) {
        default:
        case blue:
            tpostcount = 0x02;
            tfrcount   = 0x00;
            break;
        case yellow:
            tpostcount = 0x01;
            tfrcount   = 0x01;
            break;
        case orange:
            tpostcount = 0x00;
            tfrcount   = 0x02;
            break;
    }
    // TX_TMDS_CTRL2_ADDR
    //  [7:6]   TPOSTCOUNT*
    //  [5:3]   FFBCOUNT
    //  [2:0]   FFRCOUNT
    WriteByteHDMITXP0 (TX_TMDS_CTRL2_ADDR,
((ReadByteHDMITXP0(TX_TMDS_CTRL2_ADDR) & 0x3F) | (tpostcount << 6)));

    // TX_TMDS_CTRL3_ADDR     72:84
    //  [7]     reserved
    //  [6:3]   ITPLL*
    //  [2:0]   FPOSTCOUNT
    WriteByteHDMITXP0 (TX_TMDS_CTRL3_ADDR,
((ReadByteHDMITXP0(TX_TMDS_CTRL3_ADDR) & 0x87)  | (itpll << 3))); // 72:84

    // TX_TMDS_CTRL4_ADDR           72:85
    //  [7:2]   reserved
    //  [1:0]   TFRPOSTCOUNT*
    WriteByteHDMITXP0 (TX_TMDS_CTRL4_ADDR,
((ReadByteHDMITXP0(TX_TMDS_CTRL4_ADDR) & 0xFC) | (tfrcount)));          // 72:85
}
//------------------------------------------------------------------------------
// SiI_FApost_setup
//------------------------------------------------------------------------------
void SiI_FApost_setup(byte RangeSet, int idclk_freq, byte bpp)
{
    byte nFAPOSTCOUNT = 0;
    switch (RangeSet) {
        default:
        case blue:
            switch (bpp)
              {
                default:
```

```
                case SiI_DeepColor_Off:
                case SiI_DeepColor_24bit: if (idclk_freq >= 44) nFAPOSTCOUNT =
1; break;
                case SiI_DeepColor_30bit: if (idclk_freq >= 33) nFAPOSTCOUNT =
1; break;
                case SiI_DeepColor_36bit: if (idclk_freq >= 30) nFAPOSTCOUNT =
1; break;
            }
            break;

case yellow:
            switch (bpp)
    {
                default:
                case SiI_DeepColor_Off:
                case SiI_DeepColor_24bit: if (idclk_freq >= 86) nFAPOSTCOUNT =
1; break;
                case SiI_DeepColor_30bit: if (idclk_freq >= 71) nFAPOSTCOUNT =
1; break;
                case SiI_DeepColor_36bit: if (idclk_freq >= 58) nFAPOSTCOUNT =
1; break;
            }
            break;

case orange:
            switch (bpp) {
                default:
                case SiI_DeepColor_Off:
                case SiI_DeepColor_24bit: if (idclk_freq >= 168) nFAPOSTCOUNT =
1; break;
                case SiI_DeepColor_30bit: if (idclk_freq >= 139) nFAPOSTCOUNT =
1; break;
                case SiI_DeepColor_36bit: if (idclk_freq >= 114) nFAPOSTCOUNT =
1; break;
            }
            break;
    }


           // TX_TMDS_CCTRL_ADDR
    //  [7:6]   reserved
    //  [5]     FAPOSTCOUNT*
    //  [4:0]   reserved
    WriteByteHDMITXP0 (TX_TMDS_CCTRL_ADDR,
((ReadByteHDMITXP0(TX_TMDS_CCTRL_ADDR) & 0xDF) | (nFAPOSTCOUNT << 5))); // 72:80
}
```

Table 11. IFPLL Setting: 8-bit to 8-bit, 1x

IPLLF[3:0]	0001					0001					0001				
FFRCOUNT [2:0]	011					001					000				
FFBCOUNT [2:0]	011					011					011				
FPOST COUNT[2:0]	000					001					011				
	D=	4				D=	2				D=	1			
	N=	4				N=	4				N=	4			
	P=	1				P=	2				P=	4			
Pixel Clock Freq.	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**
25.00	6.25	25.00	25.00	1.00		12.50	50.00	25.00	1.00		25.00	100.00	25.00	1.00	0
27.50	6.88	27.50	27.50	1.00		13.75	55.00	27.50	1.00		27.50	110.00	27.50	1.00	0
30.25	7.56	30.25	30.25	1.00		15.13	60.50	30.25	1.00		30.25	121.00	30.25	1.00	0
33.28	8.32	33.28	33.28	1.00		16.64	66.55	33.28	1.00		33.28	133.10	33.28	1.00	0
36.60	9.15	36.60	36.60	1.00		18.30	73.21	36.60	1.00		36.60	146.41	36.60	1.00	0
40.26	10.07	40.26	40.26	1.00		20.13	80.53	40.26	1.00		40.26	161.05	40.26	1.00	0
44.29	11.07	44.29	44.29	1.00		22.14	88.58	44.29	1.00		44.29	177.16	44.29	1.00	1
48.72	12.18	48.72	48.72	1.00		24.36	97.44	48.72	1.00		48.72	194.87	48.72	1.00	1
53.59	13.40	53.59	53.59	1.00		26.79	107.18	53.59	1.00	0	53.59	214.36	53.59	1.00	1
58.95	14.74	58.95	58.95	1.00		29.47	117.90	58.95	1.00	0	58.95	235.79	58.95	1.00	1
64.84	16.21	64.84	64.84	1.00		32.42	129.69	64.84	1.00	0	64.84	259.37	64.84	1.00	1
71.33	17.83	71.33	71.33	1.00		35.66	142.66	71.33	1.00	0	71.33	285.31	71.33	1.00	
78.46	19.62	78.46	78.46	1.00		39.23	156.92	78.46	1.00	0	78.46	313.84	78.46	1.00	
86.31	21.58	86.31	86.31	1.00		43.15	172.61	86.31	1.00	1	86.31	345.23	86.31	1.00	
94.94	23.73	94.94	94.94	1.00		47.47	189.87	94.94	1.00	1	94.94	379.75	94.94	1.00	
104.43	26.11	104.43	104.43	1.00	0	52.22	208.86	104.43	1.00	1	104.43	417.72	104.43	1.00	
114.87	28.72	114.87	114.87	1.00	0	57.44	229.75	114.87	1.00	1	114.87	459.50	114.87	1.00	
126.36	31.59	126.36	126.36	1.00	0	63.18	252.72	126.36	1.00	1	126.36	505.45	126.36	1.00	
139.00	34.75	139.00	139.00	1.00	0	69.50	278.00	139.00	1.00		139.00	555.99	139.00	1.00	
152.90	38.22	152.90	152.90	1.00	0	76.45	305.80	152.90	1.00		152.90	611.59	152.90	1.00	
168.19	42.05	168.19	168.19	1.00	0	84.09	336.37	168.19	1.00		168.19	672.75	168.19	1.00	
185.01	46.25	185.01	185.01	1.00	1	92.50	370.01	185.01	1.00		185.01	740.02	185.01	1.00	
203.51	50.88	203.51	203.51	1.00	1	101.75	407.01	203.51	1.00		203.51	814.03	203.51	1.00	
223.86	55.96	223.86	223.86	1.00	1	111.93	447.72	223.86	1.00		223.86	895.43	223.86	1.00	
246.24	61.56	246.24	246.24	1.00	1	123.12	492.49	246.24	1.00		246.24	984.97	246.24	1.00	
270.87	67.72	270.87	270.87	1.00	1	135.43	541.74	270.87	1.00		270.87	1083.47	270.87	1.00	
297.95															

Table 12. IFPLL Setting: 10-bit to 8-bit, 1x

IPLLF[3:0]	**0001**					**0001**					**0001**				
FFRCOUNT [2:0]	**011**					**001**					**000**				
FFBCOUNT [2:0]	**011**					**011**					**011**				
FPOST COUNT[2:0]	**000**					**001**					**011**				
	D=	4				D=	2				D=	1			
	N=	5				N=	5				N=	5			
	P=	1				P=	2				P=	4			
Pixel Clock Freq.	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**
25.00	6.25	31.25	31.25	1.25		12.50	62.50	31.25	1.25		25.00	125.00	31.25	1.25	0
27.50	6.88	34.38	34.38	1.25		13.75	68.75	34.38	1.25		27.50	137.50	34.38	1.25	0
30.25	7.56	37.81	37.81	1.25		15.13	75.63	37.81	1.25		30.25	151.25	37.81	1.25	0
33.28	8.32	41.59	41.59	1.25		16.64	83.19	41.59	1.25		33.28	166.38	41.59	1.25	1
36.60	9.15	45.75	45.75	1.25		18.30	91.51	45.75	1.25		36.60	183.01	45.75	1.25	1
40.26	10.07	50.33	50.33	1.25		20.13	100.66	50.33	1.25	0	40.26	201.31	50.33	1.25	1
44.29	11.07	55.36	55.36	1.25		22.14	110.72	55.36	1.25	0	44.29	221.45	55.36	1.25	1
48.72	12.18	60.90	60.90	1.25		24.36	121.79	60.90	1.25	0	48.72	243.59	60.90	1.25	1
53.59	13.40	66.99	66.99	1.25		26.79	133.97	66.99	1.25	0	53.59	267.95	66.99	1.25	1
58.95	14.74	73.69	73.69	1.25		29.47	147.37	73.69	1.25	0	58.95	294.74	73.69	1.25	
64.84	16.21	81.05	81.05	1.25		32.42	162.11	81.05	1.25	0	64.84	324.22	81.05	1.25	
71.33	17.83	89.16	89.16	1.25		35.66	178.32	89.16	1.25	1	71.33	356.64	89.16	1.25	
78.46	19.62	98.08	98.08	1.25		39.23	196.15	98.08	1.25	1	78.46	392.30	98.08	1.25	
86.31	21.58	107.88	107.88	1.25	0	43.15	215.77	107.88	1.25	1	86.31	431.53	107.88	1.25	
94.94	23.73	118.67	118.67	1.25	0	47.47	237.34	118.67	1.25	1	94.94	474.69	118.67	1.25	
104.43	26.11	130.54	130.54	1.25	0	52.22	261.08	130.54	1.25	1	104.43	522.16	130.54	1.25	
114.87	28.72	143.59	143.59	1.25	0	57.44	287.19	143.59	1.25		114.87	574.37	143.59	1.25	
126.36	31.59	157.95	157.95	1.25	0	63.18	315.90	157.95	1.25		126.36	631.81	157.95	1.25	
139.00	34.75	173.75	173.75	1.25	1	69.50	347.49	173.75	1.25		139.00	694.99	173.75	1.25	
152.90	38.22	191.12	191.12	1.25	1	76.45	382.24	191.12	1.25		152.90	764.49	191.12	1.25	
168.19	42.05	210.23	210.23	1.25	1	84.09	420.47	210.23	1.25		168.19	840.94	210.23	1.25	
185.01	46.25	231.26	231.26	1.25	1	92.50	462.52	231.26	1.25		185.01	925.03	231.26	1.25	
203.51	50.88	254.38	254.38	1.25	1	101.75	508.77	254.38	1.25		203.51	1017.53	254.38	1.25	
223.86	55.96	279.82	279.82	1.25		111.93	559.64	279.82	1.25		223.86	1119.29	279.82	1.25	
246.24	61.56	307.80	307.80	1.25		123.12	615.61	307.80	1.25		246.24	1231.22	307.80	1.25	
270.87	67.72	338.58	338.58	1.25		135.43	677.17	338.58	1.25		270.87	1354.34	338.58	1.25	
297.95															

Table 13. IFPLL Setting: 12-bit to 8-bit, 1x

IPLLF[3:0]	0001					0001					0001				
FFRCOUNT [2:0]	011					001					000				
FFBCOUNT [2:0]	011					011					011				
FPOST COUNT[2:0]	000					001					011				
	D=	4				D=	2				D=	1			
	N=	6				N=	6				N=	6			
	P=	1				P=	2				P=	4			
Pixel Clock Freq.	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**
25.00	6.25	37.50	37.50	1.50		12.50	75.00	37.50	1.50		25.00	150.00	37.50	1.50	0
27.50	6.88	41.25	41.25	1.50		13.75	82.50	41.25	1.50		27.50	165.00	41.25	1.50	0
30.25	7.56	45.38	45.38	1.50		15.13	90.75	45.38	1.50		30.25	181.50	45.38	1.50	1
33.28	8.32	49.91	49.91	1.50		16.64	99.83	49.91	1.50		33.28	199.65	49.91	1.50	1
36.60	9.15	54.90	54.90	1.50		18.30	109.81	54.90	1.50	0	36.60	219.62	54.90	1.50	1
40.26	10.07	60.39	60.39	1.50		20.13	120.79	60.39	1.50	0	40.26	241.58	60.39	1.50	1
44.29	11.07	66.43	66.43	1.50		22.14	132.87	66.43	1.50	0	44.29	265.73	66.43	1.50	1
48.72	12.18	73.08	73.08	1.50		24.36	146.15	73.08	1.50	0	48.72	292.31	73.08	1.50	
53.59	13.40	80.38	80.38	1.50		26.79	160.77	80.38	1.50	0	53.59	321.54	80.38	1.50	
58.95	14.74	88.42	88.42	1.50		29.47	176.85	88.42	1.50	1	58.95	353.69	88.42	1.50	
64.84	16.21	97.27	97.27	1.50		32.42	194.53	97.27	1.50	1	64.84	389.06	97.27	1.50	
71.33	17.83	106.99	106.99	1.50	0	35.66	213.98	106.99	1.50	1	71.33	427.97	106.99	1.50	
78.46	19.62	117.69	117.69	1.50	0	39.23	235.38	117.69	1.50	1	78.46	470.76	117.69	1.50	
86.31	21.58	129.46	129.46	1.50	0	43.15	258.92	129.46	1.50	1	86.31	517.84	129.46	1.50	
94.94	23.73	142.41	142.41	1.50	0	47.47	284.81	142.41	1.50		94.94	569.62	142.41	1.50	
104.43	26.11	156.65	156.65	1.50	0	52.22	313.29	156.65	1.50		104.43	626.59	156.65	1.50	
114.87	28.72	172.31	172.31	1.50	1	57.44	344.62	172.31	1.50		114.87	689.25	172.31	1.50	
126.36	31.59	189.54	189.54	1.50	1	63.18	379.09	189.54	1.50		126.36	758.17	189.54	1.50	
139.00	34.75	208.50	208.50	1.50	1	69.50	416.99	208.50	1.50		139.00	833.99	208.50	1.50	
152.90	38.22	229.35	229.35	1.50	1	76.45	458.69	229.35	1.50		152.90	917.39	229.35	1.50	
168.19	42.05	252.28	252.28	1.50	1	84.09	504.56	252.28	1.50		168.19	1009.12	252.28	1.50	
185.01	46.25	277.51	277.51	1.50		92.50	555.02	277.51	1.50		185.01	1110.04	277.51	1.50	
203.51	50.88	305.26	305.26	1.50		101.75	610.52	305.26	1.50		203.51	1221.04	305.26	1.50	
223.86	55.96	335.79	335.79	1.50		111.93	671.57	335.79	1.50		223.86	1343.15	335.79	1.50	
246.24	61.56	369.36	369.36	1.50		123.12	738.73	369.36	1.50		246.24	1477.46	369.36	1.50	
270.87	67.72	406.30	406.30	1.50		135.43	812.60	406.30	1.50		270.87	1625.21	406.30	1.50	
297.95															

Table 14. IFPLL Setting: 16-bit to 8-bit, 1x

IPLLF[3:0]	**0001**					**0001**				
FFRCOUNT [2:0]	**011**					**001**				
FFBCOUNT [2:0]	**011**					**011**				
FPOST COUNT[2:0]	**000**					**001**				
	D=	2				D=	1			
	N=	4				N=	4			
	P=	1				P=	2			
Pixel Clock Freq.	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**	**PFD**	**VCO**	**Out Clk**	**Ratio**	**FAPOSTCOUNT[0]**
25.00	12.50	50.00	50.00	2.00		25.00	100.00	50.00	2.00	0
27.50	13.75	55.00	55.00	2.00		27.50	110.00	55.00	2.00	0
30.25	15.13	60.50	60.50	2.00		30.25	121.00	60.50	2.00	0
33.28	16.64	66.55	66.55	2.00		33.28	133.10	66.55	2.00	0
36.60	18.30	73.21	73.21	2.00		36.60	146.41	73.21	2.00	0
40.26	20.13	80.53	80.53	2.00		40.26	161.05	80.53	2.00	0
44.29	22.14	88.58	88.58	2.00		44.29	177.16	88.58	2.00	1
48.72	24.36	97.44	97.44	2.00		48.72	194.87	97.44	2.00	1
53.59	26.79	107.18	107.18	2.00	0	53.59	214.36	107.18	2.00	1
58.95	29.47	117.90	117.90	2.00	0	58.95	235.79	117.90	2.00	1
64.84	32.42	129.69	129.69	2.00	0	64.84	259.37	129.69	2.00	1
71.33	35.66	142.66	142.66	2.00	0	71.33	285.31	142.66	2.00	
78.46	39.23	156.92	156.92	2.00	0	78.46	313.84	156.92	2.00	
86.31	43.15	172.61	172.61	2.00	1	86.31	345.23	172.61	2.00	
94.94	47.47	189.87	189.87	2.00	1	94.94	379.75	189.87	2.00	
104.43	52.22	208.86	208.86	2.00	1	104.43	417.72	208.86	2.00	
114.87	57.44	229.75	229.75	2.00	1	114.87	459.50	229.75	2.00	
126.36	63.18	252.72	252.72	2.00	1	126.36	505.45	252.72	2.00	
139.00	69.50	278.00	278.00	2.00		139.00	555.99	278.00	2.00	
152.90	76.45	305.80	305.80	2.00		152.90	611.59	305.80	2.00	
168.19	84.09	336.37	336.37	2.00		168.19	672.75	336.37	2.00	
185.01	92.50	370.01	370.01	2.00		185.01	740.02	370.01	2.00	
203.51	101.75	407.01	407.01	2.00		203.51	814.03	407.01	2.00	
223.86	111.93	447.72	447.72	2.00		223.86	895.43	447.72	2.00	
246.24	123.12	492.49	492.49	2.00		246.24	984.97	492.49	2.00	
270.87	135.43	541.74	541.74	2.00		270.87	1083.47	541.74	2.00	
297.95										

Table 15. MPLL Setting #1

ITPLL[3:0]	**0110**	**0110**	**0110**
TPOSTCOUNT[1:0]	**00**	**01**	**10**
TFRCOUNT[1:0]	**10**	**01**	**00**
Link Clock Frequency			
25.00			
27.50			
30.25			
33.28			
36.60			
40.26			
44.29			
48.72			
53.59			
58.95			
64.84			
71.33			
78.46			
86.31			
94.94			
104.43			
114.87			
126.36			
139.00			
152.90			
168.19			
185.01			
203.51			
223.86			
246.24			
270.87			
297.95			

Notes:
0x083[7:6]: TPOSTCOUNT[1:0]: TXPLL post counter
0x084[6:3]: ITPLL[3:0]: TXPLL charge pump current
0x085[1:0]: TPRCOUNT[1:0]: TXPLL pre-counter

Table 16. MPLL Setting #2

ITPLL[3:0]	**0110**	**0110**	**0110**
TPOSTCOUNT[1:0]	**00**	**01**	**10**
TFRCOUNT[1:0]	**10**	**01**	**00**
Link Clock Frequency			
25.00			
27.50			
30.25			
33.28			
36.60			
40.26			
44.29			
48.72			
53.59			
58.95			
64.84			
71.33			
78.46			
86.31			
94.94			
104.43			
114.87			
126.36			
139.00			
152.90			
168.19			
185.01			
203.51			
223.86			
246.24			
270.87			
297.95			

Notes:
0x083[7:6]: TPOSTCOUNT[1:0]: TXPLL post counter
0x084[6:3]: ITPLL[3:0]: TXPLL charge pump current
0x085[1:0]: TPRCOUNT[1:0]: TXPLL pre-counter

Muting Video and Audio in HDCP Applications

Some HDCP applications may require that video and audio be muted until authentication is complete. Whenever the transmitter starts from a powered-down state (either after RESET or after assertion of one or more power-down register bits), the video and audio has to be suppressed until authentication is complete, because authentication cannot be performed when the transmitter is powered-down.

Muting video and audio with General Control Packets in HDMI mode is described on page 103. In DVI mode, there are no General Control Packets, but the SET_AVMUTE bit in register GCP_BYTE1 (refer to page 58) sets the output pixel values to 0x00 when set to 1. Because there is no audio in DVI mode, all content is muted as soon as SET_AVMUTE is set to 1.

Power Down Control

Logic Blocks Affected by Power Down

The transmitter provides four register bits to control power down of various sections of the chip. PD# is defined on page 4. PDIDCK#, PDOSC, and PDTOT# are defined on page 49.

Table 17. Power Down Control Bit Effects

PD#	PDIDCK#	PDOSC	PDTOT#	Function	Note
X	X	X	0	Powers down everything.	1
0	1	1	1	Powers down TMDS core and PLL. Digital logic is switching with an active IDCK. Registers are accessible via I^2C with the exceptions listed Table 18.	2
1	0	1	1	Powers down internal digital clock tree.	3
1	1	0	1	Powers down internal ring oscillator. Disables internal read of HDCP keys and KSV. Disables master DDC block.	4, 5

Notes:

1. This combination delivers the lowest power consumption if input signals are switching.
2. An attached HDMI receiver sees no switching clock or data and should react by disabling that HDMI input port until switching is detected again.
3. A quiet internal clock tree significantly reduces power consumption.
4. HDCP keys and KSV are read only at the rising edge of RESET#. PDOSC = 1 by default; therefore, after RESET#, the keys and KSV will be read completely before the firmware can write a 0 to PDIDCK#.
5. Set PDOSC = 1 and PDTOT# = 1 whenever Master DDC block is used to write or read across the DDC bus for HDCP. Master DDC can be used when PD# = 0 or PDIDCK# = 0, such as for reading EDID when the attached HDMI receiver is not powered on.

Registers Affected by Power Down

The registers shown in Table 18 require PD# = 1 AND PDIDCK# = 1 AND PDOSC = 1 AND PDTOT# = 1.

Table 18. Registers Affected by PD Bits

Device	Offset	Register	Device	Offset	Register
0x7A	0x3E	PB_CTRL1	0x72	0x15	AN1
0x7A	0x3F	PB_CTRL2	0x72	0x16	AN2
0x72	0x0F	HDCP_CTRL	0x72	0x17	AN3
0x72	0x10	BKSV1	0x72	0x18	AN4
0x72	0x11	BKSV2	0x72	0x19	AN5
0x72	0x12	BKSV3	0x72	0x1A	AN6
0x72	0x13	BKSV4	0x72	0x1B	AN7
0x72	0x14	BKSV5	0x72	0x1C	AN8

References

Standards Documents

Table 19 lists the abbreviations of standards used in this document. Contact the responsible standards groups for more information on these Specifications.

Table 19. Referenced Documents

Abbreviation	Specification
HDMI	*High-bandwidth Digital Multimedia Interface*, Revision 1.3, HDMI Consortium; June 2006
HDCP	*High-bandwidth Digital Content Protection*, Revision 1.2, Digital Content Protection, LLC; June 2006
DVI	*Digital Visual Interface*, Revision 1.0, Digital Display Working Group; April 1999
E-EDID	*Enhanced Extended Display Identification Data Standard*, Release A Revision 1, Video Electronics Standards Association (VESA); February 2000
CEA861D	*A DTV Profile For Uncompressed High Speed Digital Interfaces*, EIA/CEA; August 2005
EDDC	*Enhanced Display Data Channel Standard*, Version 1, VESA; September 1999
DVD	*DVD Specifications for Read-Only Disc*, Part 4 (Version 1.0, March 1999, Annex B), DVD Forum

Silicon Image Documents

Table 20 lists the documents relevant to this programmer's reference available from your Silicon Image sales representative.

Table 20. Silicon Image Documents

Document Number	Document Name
SiI-AN-0073	*Handling EDID and CEA-861D Application Note*
SiI-DS-0189	*SiI9034 HDMI Transmitter Data Sheet*

Disclaimers

These materials are provided on an "AS IS" basis. Silicon Image, Inc. and its affiliates disclaim all representations and warranties (express, implied, statutory or otherwise), including but not limited to: (i) all implied warranties of merchantability, fitness for a particular purpose, and/or non-infringement of third party rights; (ii) all warranties arising out of course-of-dealing, usage, and/or trade; and (iii) all warranties that the information or results provided in, or that may be obtained from use of, the materials are accurate, reliable, complete, up-to-date, or produce specific outcomes. Silicon Image, Inc. and its affiliates assume no liability or responsibility for any errors or omissions in these materials, makes no commitment or warranty to correct any such errors or omissions or update or keep current the information contained in these materials, and expressly disclaims all direct, indirect, special, incidental, consequential, reliance and punitive damages, including WITHOUT LIMITATION any loss of profits arising out of your access to, use or interpretation of, or actions taken or not taken based on the content of these materials.

Silicon Image, Inc. and its affiliates reserve the right, without notice, to periodically modify the information in these materials, and to add to, delete, and/or change any of this information.

Notwithstanding the foregoing, these materials shall not, in the absence of authorization under U.S. and local law and regulations, as required, be used by or exported or re-exported to (i) any U.S. sanctioned or embargoed country, or to nationals or residents of such countries; or (ii) any person, entity, organization or other party identified on the U.S. Department of Commerce's Denied Persons or Entity List, the U.S. Department of Treasury's Specially Designated Nationals or Blocked Persons List, or the Department of State's Debarred Parties List, as published and revised from time to time; (iii) any party engaged in nuclear, chemical/biological weapons or missile proliferation activities; or (iv) any party for use in the design, development, or production of rocket systems or unmanned air vehicles.

Products and Services

The products and services described in these materials, and any other information, services, designs, know-how and/or products provided by Silicon Image, Inc. and/or its affiliates are provided on as "AS IS" basis, except to the extent that Silicon Image, Inc. and/or its affiliates provides an applicable written limited warranty in its standard form license agreements, standard Terms and Conditions of Sale and Service or its other applicable standard form agreements, in which case such limited warranty shall apply and shall govern in lieu of all other warranties (express, statutory, or implied). EXCEPT FOR SUCH LIMITED WARRANTY, SILICON IMAGE, INC. AND ITS AFFILIATES DISCLAIM ALL REPRESENTATIONS AND WARRANTIES (EXPRESS, IMPLIED, STATUTORY OR OTHERWISE), REGARDING THE INFORMATION, SERVICES, DESIGNS, KNOW-HOW AND PRODUCTS PROVIDED BY SILICON IMAGE, INC. AND/OR ITS AFFILIATES, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND/OR NON-INFRINGEMENT OF THIRD PARTY RIGHTS. YOU ACKNOWLEDGE AND AGREE THAT SUCH INFORMATION, SERVICES, DESIGNS, KNOW-HOW AND PRODUCTS HAVE NOT BEEN DESIGNED, TESTED, OR MANUFACTURED FOR USE OR RESALE IN SYSTEMS WHERE THE FAILURE, MALFUNCTION, OR ANY INACCURACY OF THESE ITEMS CARRIES A RISK OF DEATH OR SERIOUS BODILY INJURY, INCLUDING, BUT NOT LIMITED TO, USE IN NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION, EMERGENCY SYSTEMS, OR OTHER SYSTEMS WITH A SIMILAR DEGREE OF POTENTIAL HAZARD. NO PERSON IS AUTHORIZED TO MAKE ANY OTHER WARRANTY OR REPRESENTATION CONCERNING THE PERFORMANCE OF THE INFORMATION, PRODUCTS, KNOW-HOW, DESIGNS OR SERVICES OTHER THAN AS PROVIDED IN THESE TERMS AND CONDITIONS.

1060 E. Arques Avenue
Sunnyvale, CA 94085
T 408.616.4000   F 408.830.9530
www.siliconimage.com

1060 E. Arques Avenue
Sunnyvale, CA 94085
T 408.616.4000   F 408.830.9530
www.siliconimage.com