DATA SHEET

SAA7145
PCI Multimedia Bridge

Product Specifications

November 30th, 1995

Philips Semiconductors

PHILIPS

Legal stuff:

This document contains preliminary information and is intended for internal use only. Data subject to change without notice.

Philips Semiconductors and Philips Electronics North America Corporation reserve the right to make changes, without notice, in the product describe herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of this product, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representation or warranties that these products are free from patent, copyright, or mask work right infringement unless otherwise specified. Application that are described herein are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS
Philips Semiconductors and Philips Electronics North America Corporation products are not designed for use in life support appliances, devices, or systems where malfunction of a Philips Semiconductors and Philips Electronics North America Corporation Product can reasonably be expected to result in a personal injury. Semiconductors and Philis Electronics North America Corporation customers using or selling Semiconductors and Philips Electronics North America Corporation Products for use in such applications do so at their own risk and agree to fully indemnity Semiconductors and Philips Electronics North America Corporation for any damages resulting from such improper use or sale.

Semiconductors and Philips Electronics North America Corporation register eligible circuits under the Semiconductors Chip Protection Act.

Table of Contents

1. Features

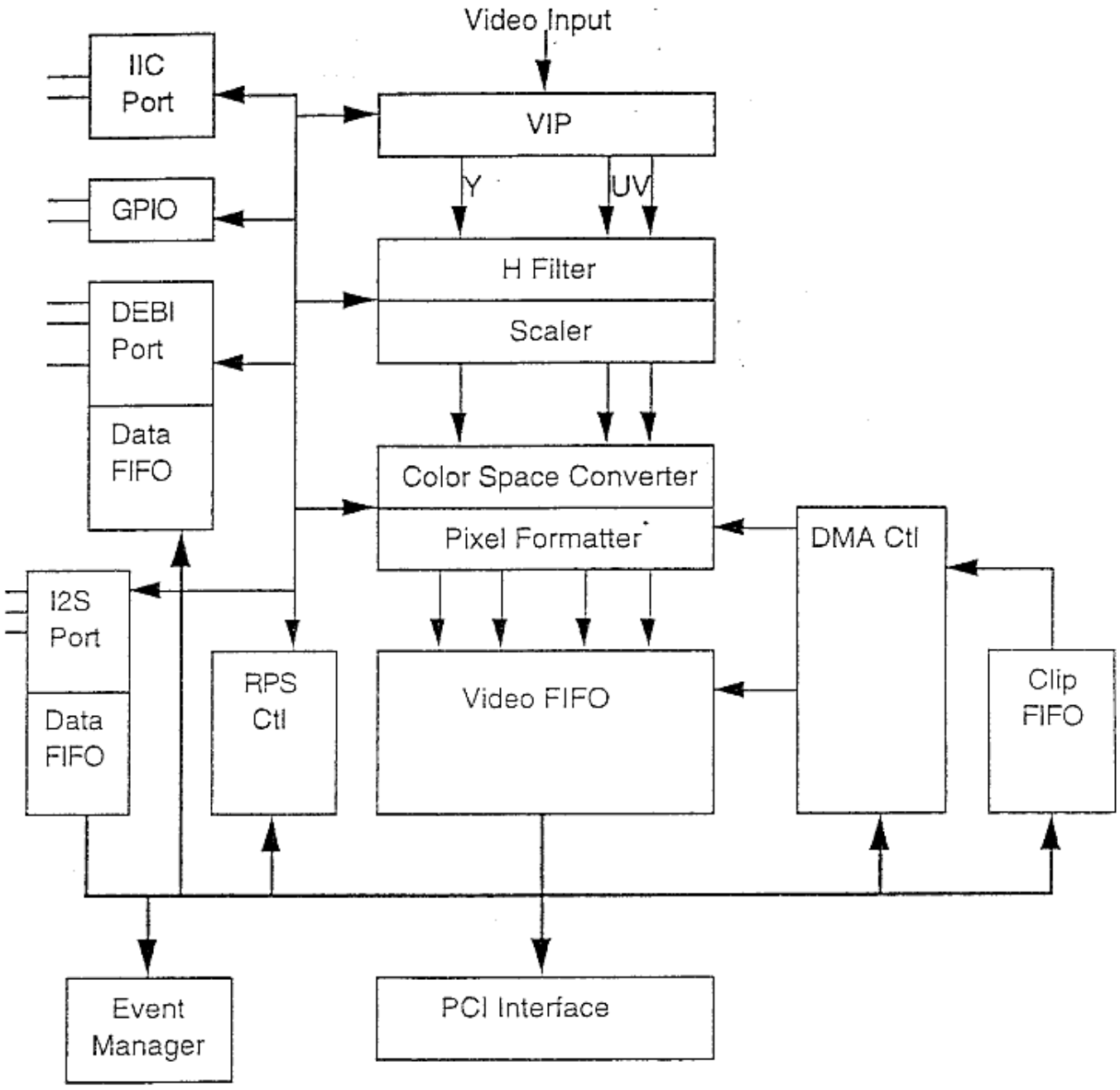
- Full size, full speed video delivery to frame buffer or system memory
- Full Bandwidth PCI bus Master (up to 132 MB/sec)
- D1 (CCIR-656) Video Input Port
- DMSD-2 compatible
- 128 D word Video FIFO
- Simple Arbitrary Size Scaling
- Horizontal Filtering
- Color Space Conversion
- Support for all standard packed pixel formats
- Support for Pixel Dithering
- Video Mask Clipping
- Video synchronized programming control (RPS)
- Audio Capture via I^2S input port
- I^2C Bus port
- DEBI (Data Expansion Bus Interface) port for use with external devices
- FIFO Overflow Detection and “graceful” recovery
- Support for event analysis
- 160 pin package

2. General Description

The SAA7145 is a Multimedia PCI bridge. The SAA7145 can be used to place a real time video anywhere in system or video memory. It supports many different output formats for extended compatibility with any frame buffer. The video image can be scaled to an arbitrary size by the internal scaling unit. The video can also be cropped to an arbitrary shape to support video window occlusions and graphics overlay.
The SAA7145 also has support for audio capture. Audio capture is accomplished though a I^2S interface. The audio port i configurable to work with a range of audio A to D converters.
In general it offers higher performance than it's predecessor the SAA7116 and adds several new features.

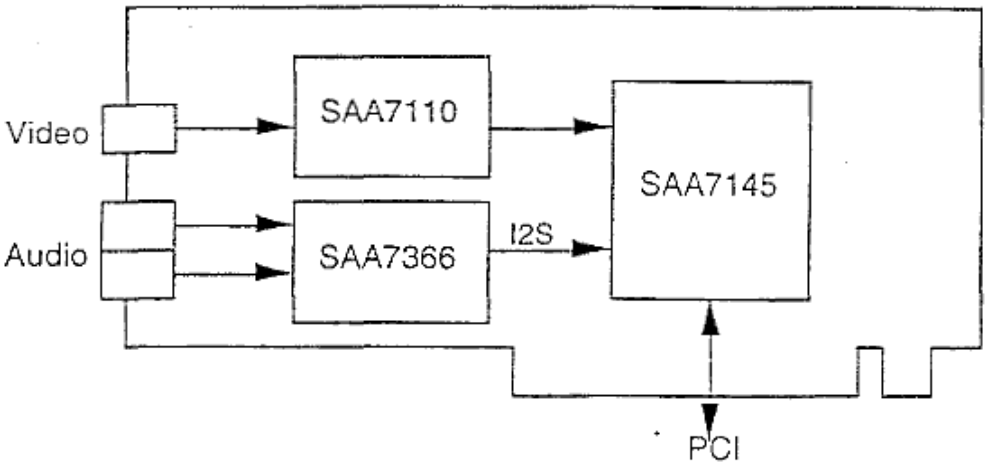
Function	7116	7145	7146
video to PCI	√	√	√
PCI to video			√
Packed Output Modes	√	√	√
RGB32	√	√	√
RGB24 wrapped		√	√
RGB16		√	√
RGB15	√	√	√
Dithering		√	√
Planar Output Modes	√		√
Video Flip	√	√	√
D1 Video Port		√	2
DMSD2 Compatible		√	√
24 bit Video Port	√		
Arbitrary Video Scaling		√	√
Vertical Filtering			√
Horizontal Filtering		√	√
Zoom up			√
B-C-S Control			√
I^2C Port	√	√	√
DEBI Port		√	√
Audio Input Port		√	2
Audio Output Port			2
GPIO Ports		4	4
Rectangular Clipping			√
Mask Clipping		√	√
Chroma Key Clipping			√
RPS Control		√	√
Gamma Correction			√
Color Space Converter		√	√
PCI Retry Capable		√	√
Package Size	160	160	206

Below is a diagram showing the major blocks of the SAA7145.

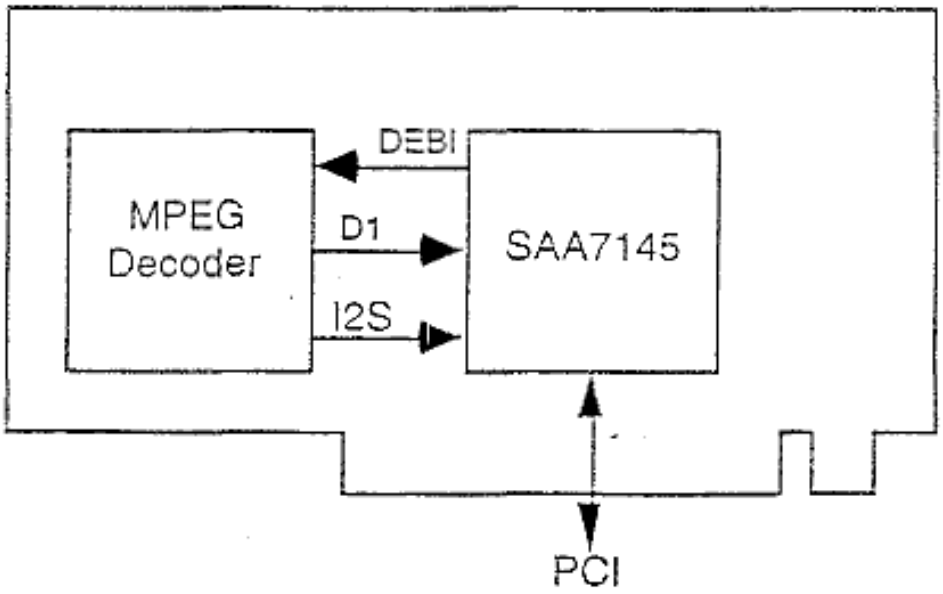


3. Implementation Examples

Below is the simplest use of the SAA7145. A very low cost video input card can be achieved in this way. Only two major components are required: SAA7145 VPCI and the SAA7110 OFC1 or SAA7111 VIP.



Below in another use of the SAA7145. A low cost MPEG playback card can be achieved with a small amount of glue logic.



4. SAA7145 Register Set

A table of the internal registers of the SAA7145 is provided below. These registers are accessed by adding the offset to the value contained in the PCI Configuration Base Address Register. Accesses to the Register space are permitted by Dwords only. Byte and Word accesses will not cause an error by will not correctly program the register.

OFFSET	NAME	Shadowed	BITS	DESCRIPTION
00	VDBA	yes	31:0	Video DMA Base Address
04	**VDPA**	yes	23:2	Video DMA Protection Address
	VDPEN		0	Enable Video DMA Protection
08	**VDMAC**	yes		Video DMA Control Register
	EVID		31	Enable Video
	VBS		27:25	Video Data Burst Transfer Size
	VTHR		23:21	Video FIFO Threshold
	VEND		20:19	Endianness Controls
	PFMT		18:16	Output Format (rgb32, etc.)
	VFLIP		15	Vertical Flip
	VPCH		12:0	Line Pitch
0C	**VPCTL**	yes	15:0	Video Port Control
10	**XWC**	yes		Horizontal Window Control Register
	XWS		25:16	Horizontal Window Start
	XWZ		9:0	Horizontal Window Size
14	**YWC**	yes		Vertical Window Control Register
	YWS		25:16	Vertical Window Start
	YWZ		9:0	Vertical Window Size
18	**FSCTL**	yes		Filter and Scaler Control Register
	ILO		31	Interlace Output
	FMODE		30:28	Horizontal Filter Mode
	XSCI		26:16	Horizontal Scaling Increment
	YSCI		10:0	Vertical Scaling Increment
1C	**YPR**	yes		Vertical Phase Register
	YPO		25:16	Vertical Phase Offset Odd
	YPE		9:0	Vertical Phase Offset Even
20	**CDBA**	yes	31:2	Clip Mask DMA Base Address
24	**CDF**	yes		Clip Data Format Register
	CPO		28:24	Pixel Offset into Dword
	CPIX		21:16	Number of Dwords per Line of Clip Mask
	CLINE		9:0	Number of Lines of Clip Mask
28	**CDMAC**	yes		Clipper DMA Control Register
	ECLIP		31	Enable Clipper
	CEND		20:19	Endianness Control
	CPOL		16	Clipper Polarity
	CPCH		9:0	Line Pitch

OFFSET	NAME	Shadowed	BITS	DESCRIPTION
2C	**LCR**	yes		Line Counter Register
	SLCT		25:16	Source line counter threshold
	TLCT		9:0	Target line counter threshold
30	**ADBA**	no	31:2	Audio DMA Base Address Register
34	**ADPA**	no	16:2	Audio DMA Protection Address
38	**ADMAC**	no		Audio DMA Control Register
	EAUD		31	Enable Audio DMA
	AEND		19	Audio Data Endianness
	AFMT		17:16	Audio Format Control
	ARNG		3:0	DMA Interrupt Range
3C	**RPSPTR**	no	31:2	RPS Pointer Address
	ERPS		0	Enable RPS bit
40	**RPSR**	no		RPS Page Register
	RPSP		31:11	RPS Page Address
	ERPSP		0	RPS Page Error Enable
44	**RPSTO**	no		RPS Time Out
	TOSEL		29:28	RPS Time Out Selector
	TOV		27:24	RPS Video Time Out Value
	TOP		23:0	RPS PCI Time Out Value
48	**ISR**	no	31:0	Interrupt Status
4C	**IER**	no	31:0	Interrupt Enable
50	**IMR**	no	31:0	Interrupt Monitor
54	**ECR**	no	19:0	Event Counter Register
58	**ECT**	no	19:0	Event Counter Threshold
5C	**ECC**	no	15:0	Event Counter Control
60	**GPIOC**	no	31:0	GPIO Control
64	**DEBICFG**	no	31:0	DEBI Configuration Register
68	**DEBIDAT**	no	31:0	DEBI Data Register
6C	**DEBICMD**	no	31:0	DEBI Command Register
70	**IICTRF**	no		I^2C Transfer Control Register
	IICB2		31:24	I^2C Data Byte 2
	IICB1		23:16	I^2C Data Byte 1
	IICB0		15:8	I^2C Data Byte 0
	IICCTL		7:0	I^2C Transfer Control
74	**IICSC**	no		I^2C Status and Clock Register
	IICCC		10:8	I^2C Clock Control
	IICSTA		7:0	I^2C Status
78	**UTIL**	no	31:0	Utility Register
7C	**rsvd**	no	31:0	Reserved for future use

More detailed register description can be found in the associated section.

5. Video Port

The video port of the SAA7145 is a YUV 4:2:2 input only port. It can be configured to accept 8-bit time-multiplexed YUV data in accordance with CCIR-656 or the port can also be configured as a 16-bit port for backward compatibility with OCF1 and DMSD2. A pixel clock input and a pixel qualifier are used to clock in the video samples. The port circuitry can detect or create a field sequence if needed of field information can be brought in on the FID pin. If the FID pin is used a high value indicates an odd field, low indicates even.

5.1. Video Port Control

The Video port of the SAA7145 is controlled by the Video Port Control (VPCTL) register. This register configures the video port to match the incoming stream and configures sync and field detection.
The Video Port Size (VPZ) bit selects between an 8 or 16 bit port.
The External Sync Source (XSS) bit selects between encoded sync signals or sync brought in the pins. 8 bit mode only.
Sync Output Enable (SOE) enables the decoded sync signals to be placed on the HS and VS pins. 8 bit mode only.
The Field Detect bits (FD[1:0]) are used to select how field changes are detected. Modes are Normal (00), Noise Limit (01), Force Toggle (10) and External (11). In Normal and Noise Limited modes the Reference Edge of Vertical sync (defined below) is used to detect the field ID. If H Sync is high when the reference edge occurs that field is detected as even, if low the field is odd.
Force Odd (FODD) forces all fields to be handled as odd fields. Swap Fields (SF) inverts the field bit. If FODD is active then all fields are even.
Reference Edge for Vertical Sync (REVS): this bit is used to select the polarity of the trailing edge of vertical sync. This edge is also used to start the source line counter. A one in this bit resets the source line counter on the rising edge of vertical sync, a zero resets the counter on the falling edge. This bit also defines the edge of vertical sync on which fields are detected.
Reference Edge for Horizontal Sync (REHS): this bit is used to select the polarity of the leading edge of horizontal sync. This edge is also used to start the source pixel counter. A one in this bit resets, the source pixel counter on the rising edge of horizontal sync, a zero resets the counter on the falling edge.
EVEND: setting this bit disables the display of even fields.
ODDD: setting this bit disables the display of odd fields.
TWOS: U and V components in Twos complement. If this bit is set then the U and V components of the input stream are in twos complement format rather than unitary.
TEST: this bit is used only in testing. This bit must be zero for normal operation.

VPCTL		Offset: 0C
Bits	**Name**	**Description**

VPCTL		Offset: 0C
15	TEST	1 = Enter Test mode 0 = Normal Operation
14:12	rsvd	Reserved
11	EVEND	Turn off even fields
10	ODDD	Turn off odd fields
9	REVS	Reference Edge for Vertical Sync
8	REHS	Reference Edge for Horizontal Sync
7	TWOS	U and V Components in Two's Complement
6	SF	Swap Fields (1 enables)
5	FODD	Force Odd (1 enables)
4:3	FD[1:0]	Field Detect Mode 00 = Normal 01 = Noise Limited 10 = Force Toggle 11 = External Input (FID)
2	SOE	Sync Output Enable 1 = Encoded Sync placed on pins
1	XSS	External Sync Source 1 = Sync comes in on pins 0 = Sync is encoded in stream
0	VPZ	Video Port Size 1 = 16 bit port 0 = 8 bit port

5.2. CCIR-656 Mode

In CCIR-656 mode (also called D1 video), sync signals can be encoded in the stream or brought in on separate sync inputs. The sync pins can be also used as outputs when external synchronization to the input source is required. The pixel rate is equal to one half the clock rate with a luminance sample arriving every other clock. Chrominance samples arrive on the 1st and 3rd clocks in the pattern U-Y-V-Y. Pixel data may be qualified by the user of PXQ.
In this mode video data comes in on VPI[15..8] pins. VPI[7..0] are ignored.

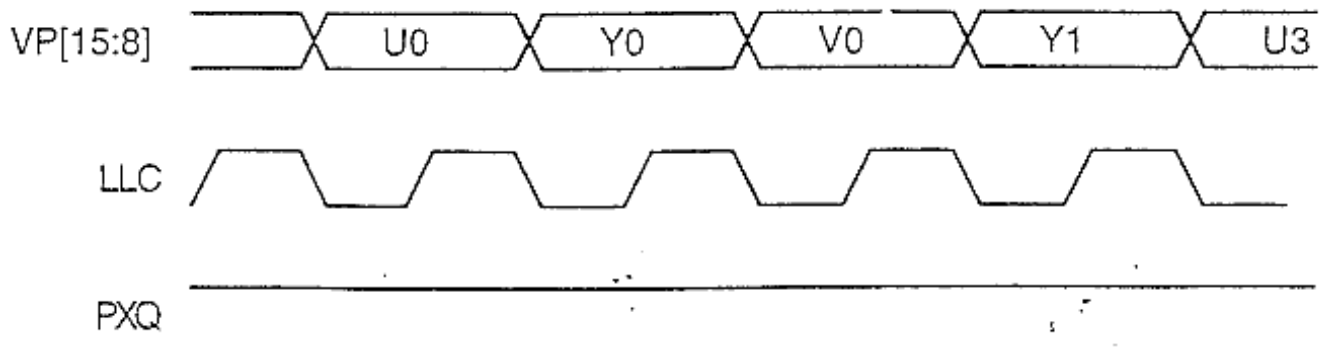
clock edge	sample	VPI[15:8]
n	1	U
n+1	1	Y
n+2	2	V
n+3	2	Y

In this mode vertical and horizontal synchronization signals can be encoded into the video stream. The SAA7145 is capable of decoding these codes. Each code is preceded by the prefix FF,00,00 the next 8 bits contain the encoded field number, vertical and horizontal information and 4 bit parity nibble. The SAA7145 does not do parity checking or error correction of encoded data.

Bit No.							
7 1	6 F	5 V	4 H	3 P_3	2 P_2	1 P_1	0 P_0

F = Field bit. 0 for odd fields
1 for even fields.
V = VBI bit. 1 during Vertical Blanking
0 elsewhere
H = HBI bit. 0 indicates Start of Active Video (SAV)
1 indicates End of Active Video (EAV)

This mode is compatible with the SAA7111 and other devices.

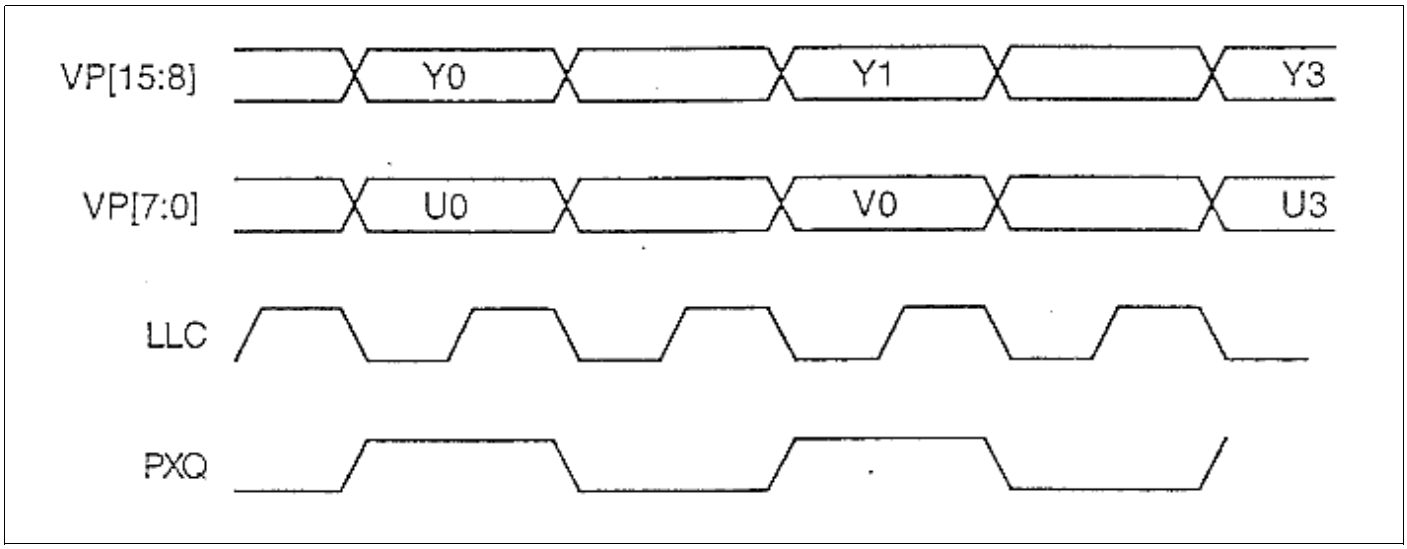


5.3. DMSD2 Mode

The DMSD2 format has luminance (Y) data on one 8 bit port and chrominance (U&V) data time multiplexed on a second 8 bit port. The clock runs at twice pixel rate in this mode with a clock qualifier on PXQ at one half the clock rate. Sync signals are always brought in on the sync input pins.
In this mode video data arrives in byte pairs of YU and YV with Y data appearing on VPI[15..8] and the UV data appearing on VPI[7..0]. U samples are odd and V samples are even.

clock edge	sample	VPI[15:8]	VPI[7..0]
n	1	Y	U
n+1	1	Y	V
n+2	2	Y	U
n+3	2	Y	V

This mode is compatible with the SAA7110, SAA7191 and other 16 bit devices.

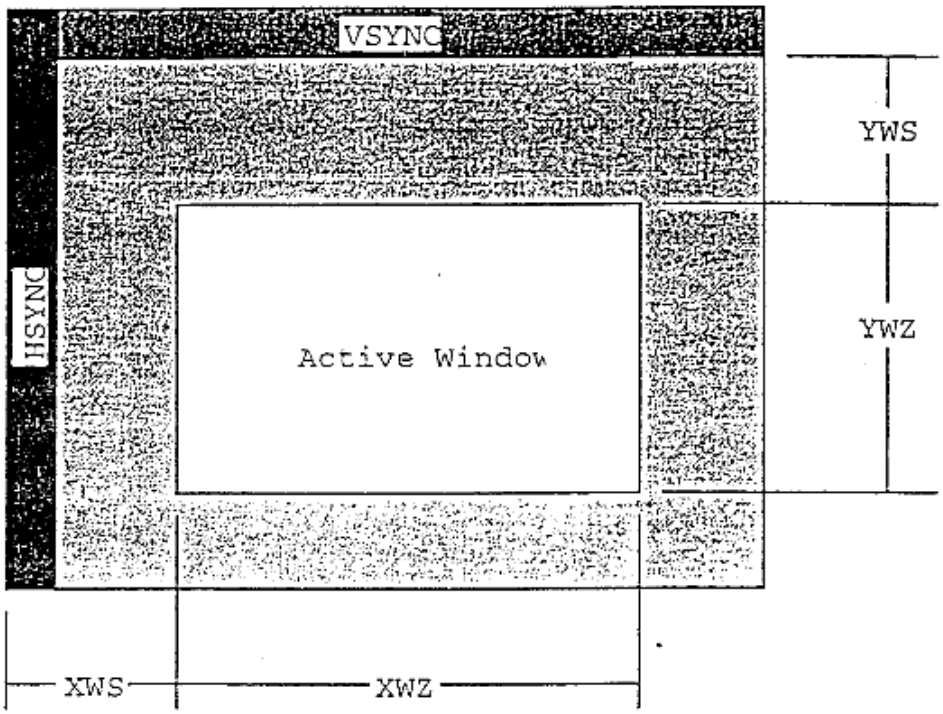


6. Scaler

The Scaler and Horizontal Filter are in the same physical block.
Horizontal scaling is accomplished by a combination of pixel dropping and averaging. Vertical scaling is done by line dropping. For interlaced video, line dropping can be done singly or by dropping line pairs. A window is selected from the incoming video stream and is called the Active Video Window. Only the part of video located inside the window is scaled, the rest is ignored.

6.1. Active Window Selection

Selection of the active window of the video stream is accomplished by a set of four registers. Only the data within the active window is passed along to the scaling unit, all other video data is discarded. The vertical window is chosen using the Y Windows Start (YWS) and Y Window Size (YWZ) registers. The YWS register is loaded with the line number of the first line of active video. The YWX register is loaded with the total number of lines to be passed on to the scaler. A horizontal window within a line is chosen from the input stream by two similar registers, X Window Start (XWS) and X Window Size (XWZ).
Y Window Start (YWS): count this number of lines from the Reference Edge of vertical sync before starting the capture. Example: if the register contains a value of 20 the the 20th line after the reference edge will be the first one sent to the scaler. This register is used as an offset to the start of active video from the sync edge programmed in the VPCTL. YWS has a range of 1 to 1024, but must be programmed with a value less than the total number of lines in one field of video.
Y Window Size (YWZ): this is the number of lines to be sent to the scaler. Usually this is programmed with the number of lines of active video. YWZ has a range of 1 to 1024. The Y window size can be programmed with a maximum value of the total number of lines in a field (including blanking) minus two.
X Window Start (XWS): count this number of pixels from the Reference Edge of horizontal sync before starting the capture. Example: if the register contains a value of 40 then the 40th pixel after the reference edge will be the first one sent to the scaler. This register is used as an offset to the start of active video from the sync edge programmed in the VPCTL. XWS has a range of 1 to 1024, but must be programmed with a value less than the total number of pixels in one line.
X Window Size (XWZ): this is the number of pixels to send to the scaler. Usually this is programmed with the number of pixels is a line of video. XWZ has a range of 1 to 1024. The X window size can be programmed with a maximum value of the total number of pixels in a line (including blanking) minus 8.



The above diagram shows the typical use of Active Window Registers. It is possible however to capture smaller or larger windows or even more than one. In the case where more than one window is captured the second window's YWS is still measured from the Vertical Sync edge, not the previous window.

6.2. Scaling

Scaling id done by line and pixel dropping. Both the horizontal and vertical scaling and filter modes are set by programmable registers.
Video scaling is controlled by four registers: Y Scaling Increment (YSCI), Y Scaling Offset (YPO), Y Scaling Offset (YPE) and X Scaling Increment (XSCI).
The values for each of there registers is determined by formulas shown below.
S_Y = #output lines / #input lines
The number of input lines is programmed in YWZ.
YSCI = INT (1024 x S_Y)
In interlace mode, for single line scaling:
YPO > YPE
Recommend:
YPO = 768, YPE = 256 for S_Y = 1 to 2/3 and 1/3 to 0
YPO = 683, YPE = 341 for S_Y = 2/3 to 1/3
In interlace mode, for line pair scaling
YPO = YPE = 512
In non interlace mode,
YPO = YPE = 512

S_X = #output pixels / #input pixels (per line)
The number of input pixels is programmed in XWZ.
XSCI = INT (1024 x S_X)

FSCTL		Offset: 18
Bits	**Name**	**Description**
31	ILO	Interlace Output
30:28	FMODE	Horizontal Filter Mode
26:16	XSCI	Horizontal Scaling Increment
10:0	YSCI	Vertical Scaling Increment

6.3. Filtering

The Horizontal filter must be programmed to match the scaling factor chosen. Lower filter modes can be used but for best result use the formula below. Using too high of a filter factor will yield unpredictable results.
Filter Mode (FMODE): this register control the filtering function of the SAA7145. The FMODE value is programmed in the FSCTL register. If filtering is to be used the following formula should be used to calculate the proper value for FMODE. If filtering is bypassed FMODE should be loaded with zero (or one).

$$FMODE = TRUNC(LOG_2(1/S_X)) + 1$$

7. Pixel Formatter

The Pixel formatter is responsible for putting the pixel information in the correct byte lanes and sizing it to the proper output format. It is manager by the Video DMA controller to ensure that the data is properly aligned with the masking information received from the Clipper. The pixel formatter organizes the information from the scaler based on bit depth, color space and whether the target wants big or little endian format. The PFMT value is programmed in the VDMAC register.
The Color Space Converter is selected automatically by the modes which require it.
The pixel formatter supports the following modes:

PFMT	Mode
0	RGB32 (aRGB)
1	RGB24 (RGBR...)
2	RGB16 (5:6:5)
3	RGB15 (a:5:5:5)
4	YUV16 (CCIR)
5	Grayscale (YYYY)
6	RGB16 dither
7	RGB15 dither

8. Video DMA Control

The Video DMA controller of the SAA7145 is similar to the SAA7116 with several enhancements. The SAA7145 is capable of multiple transfers per bus request. The burst size is programmable from 1 to 65 in powers of two plus a setting for unlimited length. Since the SAA7145 also has multiple masters an internal arbiter helps to schedule efficient bus use. The SAA71445 is also capable of accepting and properly handling PCI cycles terminated with a Retry.
The SAA7145 also contains logic for “gracefully” handling FIFO overflow and address protection violations. A FIFO over flow is usually caused by a momentary increase in bus usage. The SAA7145 recovers from an overflow as soon as space is available in the FIFO. A protection violation is usually caused by a loss of signal integrity of improper programming. The SAA7145 allows video to continue running from the next vertical sync after the violation. Each of these conditions can generate an interrupt, if enabled.
DMA Base Address (VDBA): the byte address of the first pixel to be displayed. This is the upper left corner of the video picture. The upper eight bits of the address are fixed, only the bottom 24 bits are incremented during DMA. This has the effect of locking video into one 16 megabyte space.
Protection Address (VPA): This value is used as a safety valve for address overruns. In the event that the DMA master tries to overwrite the end of the target buffer. This protection ensures that the program data areas and system control registers are not inadvertently written to. This feature is enabled by writing a one to bit zero (VDPEN). In normal operation the Protection address checks to see if the target address is above threshold. If the VFLIP bit is set (see below) the Protection address checks to see if the target address is below threshold.
The Video DMA Control Register (VDMAC) contains many controls for video DMA. They are outlined below.

VDMAC		Offset: 08
Bits	**Name**	**Description**
31	EVID	Enable Video Capture
28:25	VBS	Video Data Burst Transfer Size
23:21	VTHR	Video FIFO Threshold
20:19	VEND	Endianness Controls
18:16	PFMT	Output Format
15	VFLIP	Vertical Flip
12:0	VPCH	Video Line Pitch

Enable Video (EVID): enables video data to be placed in the video FIFO and enabled DMA transfers.
Burst Size (VBS): a number programming the size of the transfer in the power of 2. After this the DMA yields ownership of the bus to the next internal device. Programming a 7 in this location enables unlimited length transfers, at least until the latency Timer runs out. This is not recommended when other functions are in use.

Value	Burst Size
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	unlimited

FIFO Threshold (VTHR): This value determines how many D words to store in the FIFO before trying to acquire the bus. This register with multiples of 16 Dwords as shown below. Regardless of the amount of data in the FIFO the data is always flushed at the end of the horizontal window.

Value	FIFO Threshold
0	1
1	16
2	32
3	48
4	64
5	80
6	96
7	112

Endianness (VEND): These two bits select whether big or little endian addressing is used. Also provides 2 byte swap.

Value	Swapping
00	none
01	2 byte (1234 => 3412)
10	4 byte (1234 => 4321)
11	reserved

Vertical Flip (VFLIP): if this bit is set the number programmed in VPCH is in two’s complement format to indicate a vertical (top-down) flip. Clipping in not available in this mode.
Video Pitch (VPCH): Pitch is defined as the difference in address between two vertically adjacent pixels. This number can be programmed in two’s complement format to indicate a vertical (top-down) flip if the VPCH bit is set.

8.1. Non-Interlaced Video Output

To display non-interlaced video the SAA7145 starts at each vertical sync by loading the Video Destination Pointer (VDP) with the contents of Video DMA Base Address Register (VDBA). When the start of the active window is reached the SAA7145 begins bus master writes to the target. At the end of each line the Pitch register is added to the VDP and the SAA7145 is then ready to precess the next line. This continues until the end of the Active Video Window.
The VDBA is loaded with the address of the top left pixel to be displayed. The Protection Address is loaded with the address of the bottom left pixel to be displayed plus 1 line pitch plus 1 Dword.

8.2. Interlaced Video Output

There are some special considerations to be made when displaying interlaced video. To enable interlacing the ILO bit must be set to 1. This has several effects. On even fields the Pitch is automatically added one to the Video DMA Base Address after it has been loaded into the Video Destination Pointer. Normal full size interlace requires the pitch to be added twice between each line, this occurs automatically when ILO is set.
As the image is scaled down and some lines are removed, the scaler and the DMA controller work together to make sure each line of video is properly interlaced. Scaling can be by line pairs (YPO=YPE) or by single lines (YPO>YPE).
The arrangement of clip data in memory remains the same as if a larger non-interlaced image were displayed. The SAA7145 takes care to use the proper line of clip data.
The VDBA is loaded with the address of the top left pixel to be displayed. The Protection Address is loaded with the address of the bottom left pixel to be displayed plus 2 line pitched plus 1 Dword.

9. Audio Input Port

The audio input port of the SAA7145 is a stereo I^2S port.
The Audio DMA controller operates much like the video DMA. The Audio Base Address Register (ADBA) points to the first location in memory where data is to be stored. The Protection Address (ADPA) is used as a data wrapping point, when the DMA pointer reaches this address it reinitializes itself with the Base Address and immediately continues. Since audio is contiguous data, the DMA fills this memory repeatedly in a circular fashion. The ADPA in only writable in the bottom 16 bits, the upper bits are taken from ADBA. This limits the size of the audio memory buffer to a maximum of 64 Kbytes.
Larger buffers can be emulated by splitting the buffer into smaller parts and collecting the data in one part while the DMA is writing to the other. To ensure that the audio data is collected properly by the Host, an interrupt can be generated as the DMA pointer crosses thresholds determined by the ARNG register. These thresholds are related to the target address boundaries. A value of 000 sets the interrupt whenever the bottom 8 bits of the target address are zeroes or every 256 bytes. 001 sets the interrupt when the bottom 9 bits are zeroes on up until the value of 111 for 15 zeros or every 32 K.
The I^2S interface is a timing slave. It receives the audio data and parallelizes it into 16 or 8 bit words. These words are placed in the audio FIFO. The audio FIFO is only 8 Dwords deep, but is still big enough to store 2 video lines of stereo 16 bit audio data at 44.1 Khz. The FIFO unloader always attempts bursts of 4 Dwords of data across the PCI bus to the target location. With such a low data rate the probability of a FIFO overflow is remote, but if it does occur the Audio FIFO Overflow (AFO) interrupt bit is set.
Acquisition always begins with the left channel, indicated by WS being low.

ADBA		Offset: 30
Bits	**Name**	**Description**
31:2	ADBA	Audio DMA Base address

ADPA		Offset: 34
Bits	**Name**	**Description**
16:2	ADPA	Audio DMA Protection address

ADMAC		Offset: 38
Bits	**Name**	**Description**
31	EAUD	Enable Audio DMA
19	AEND	Endian swapping control 0 = no swapping 1 = four byte swap (4321 => 1234)
17:16	AFMT	Audio Format Control 00 = 16 bit, rising edge 01 = 8 bit, rising edge 10 = 16 bit, falling edge 11 = 8 bit, falling edge
2:0	ARNG	DMA Interrupt Range

10. DEBI Port

DEBI (Data Expansion Bus Interface) is a sixteen bit auxiliary data port which can be used to control an external parallel device. Large amounts of data can be transferred using the data FIFO. It is this port that enables the SAA7145 to support MPEG decoders and other parallel access ICs. The DEBI port can be accessed directly or via RPS.
The DEBI port is a 16 bit multiplexed address and data bus with control signals. The five control signals can be configured to operate as Intel or Motorola style of signaling. A sixth pin is used to transmit an interrupt request to the PCI bus. This interrupt is maskable.
The DEBI port is controlled by the use of three registers. The configuration register (DEBICFG) contains information regarding the transfer mode of operation (Intel/Motorola, Big/Little Endian, etc.). The Command Register (DEBICMD) contains the target address and transfer direction and the length of the data phase. The transfer is initiated by the write to the DEBICMD register so it must be the last register programmed. If the data phase id 4 bytes or less then the data is placed in the Data Register (DEBIDAT). If the transfer is more than 4 bytes then the address in system memory of the first long word of data is placed in the DEBIDAT register. If the data phase is less than four bytes the relevant data should always be placed in the LS byte D[7:0].
16 bit accesses to unaligned slave addresses are automatically stepped down to 8 bit transfers.
Long bursts of data are transported to or from the DEBI port by using the DEBI data FIFO. This FIFO is used as a flow control for DEBI data since the process will most likely be asynchronous to RPS and PCI. The DEBI data FIFO is bi-directional and is 32 D words deep.
The Smart Slave / Dump Slave option allows transfers to devices which do not respond with RDY/DTACK. A dump slave transfer is terminated by the time out value only. On dump Intel style reads the data is sampled at the rising edge of RDN. On dump Motorola style reads the data is sampled at the rising edge of xDS.
XIRQ can be used to regulate the flow of data to/from the DEBI port if the XIRW_WAIT flag is set. If set, assertion of XIRQ will cause the DEBI port to stall after the current transaction. Execution will continue if XIRQ is de-asserted, or the host can use this time to reprogram the DEBI slave device. The active level of XIRQ is programmable via the XIRQPOL pin.
If INCR is enabled the address presented to the slave is incremented between each data phase. If it is disabled the same address is used repeatedly. The latter mode is useful for accessing external FIFOs.
Is SLAVE16 is enabled the 7145 presents 16 bits of data per data phase. If disabled the 7145 will present the data 8 bits per data phase. Data is presented only on XAD[7:0] in this mode.
Data Bandwidth is largely dependent on slave speed. However if the target acknowledges immediately the cycle times are 7 PCI clocks per slave write (PCI read) and 10 PCI clocks for slave reads (PCI write). From this it can be determined that the maximum slave write bandwidth is about 9.4 MB/s and for reads is about 6.6 MB/s for a 16 bit slave. 8 bit performance is exactly half of the 16 bit values.

DEBICFG		Offset: 64
Bits	**Name**	**Description**
29	XIRQPOL	This bit represents the active level of XIRQ
28	SS/DS	Smart Slave / Dump Slave 1 = Smart 0 = Dump
27	XIRQ_WAIT	If enable, XIRQ causes the DEBI port to hold off pending accesses until cleared
26	rsvd	Reserved
25:22	TIMEOUT	Terminate the transfer after this many PCI clocks
21:20	SWAP	Endian swapping control 00 = no swapping 01 = two byte swap (4321 => 2143) 10 = four byte swap (4321 => 1234) 11 = reserved
19	SLAVE16	Indicated the slave is able to server 16 bit transfer
18	INCR	Enables address increment for block transfer
17	IN/MO	Intel style bus if high, Motorola style is low
16	TIEN	Timer Enable, 1 enables

DEBIDAT		Offset: 68
Bits	**Name**	**Description**
31:0	DEBIDAT	Data or Address of Data

DEBICMD		Offset: 6C
Bits	**Name**	**Description**
31:17	BLOCKLEN	Data Block Length in Bytes
16	WRITE_N	Transfer Direction 1 = Read 0 = Write
15:0	A16_IN	Slave Target Address

10.1. DEBI Timing

On the following pages are timing diagrams outlining the basic functionality of the DEBI port in both the Intel and Motorola style signaling protocols. Below are the timing tables.

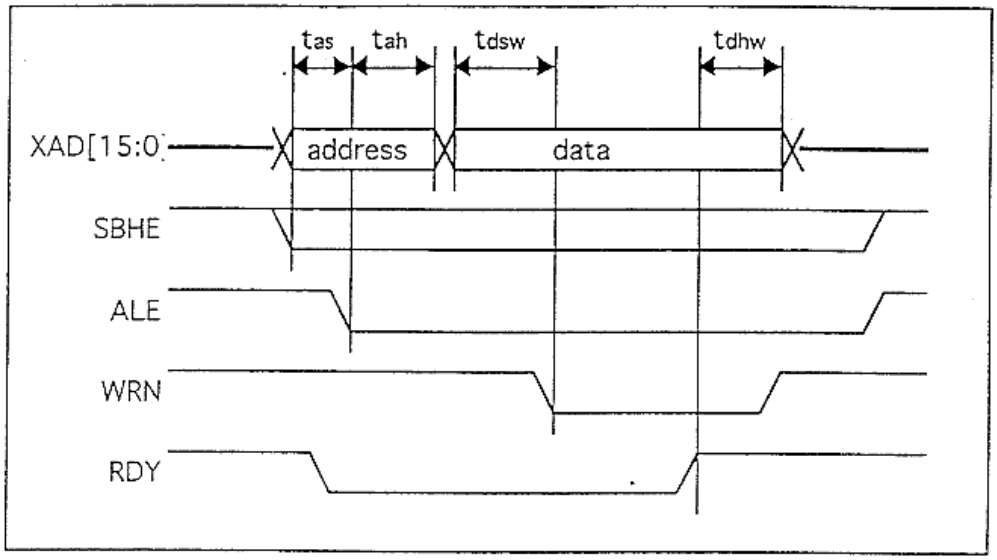
Parameter	min.	max.	units
t_{as}	20	T_c	ns
t_{ah}	20	T_c	ns
t_{dsr}	20	-	ns
t_{dhr}	5	-	ns
t_{dsw}	20	T_c	ns
t_{dhw}	0	T_c	ns
t_{dz}	0	4 x T_c	ns
t_{acc}	0	*	ns
t_{dtack}	T_c	**	ns

T_C = Period of the PCI Clock

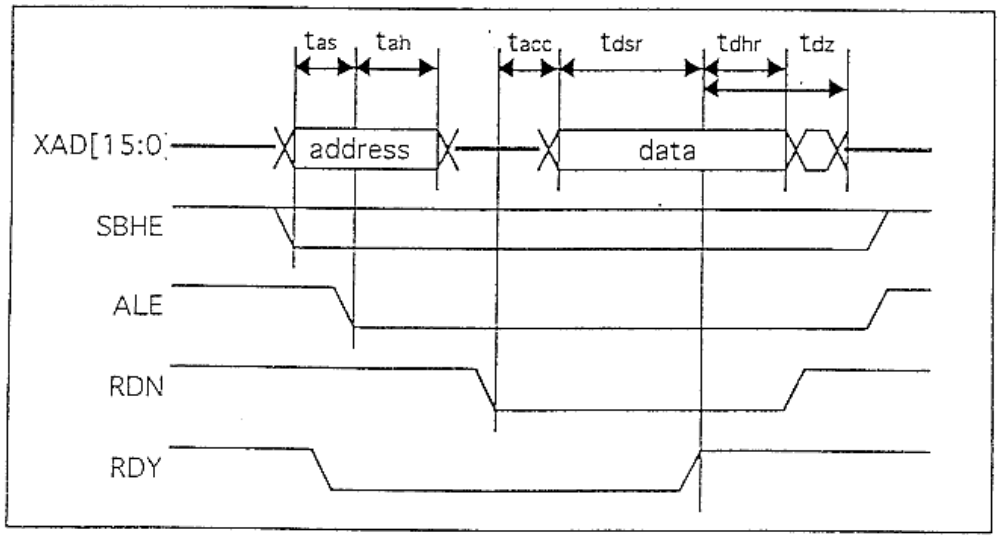
* Define by TIMEOUT value int the DEBICFG register. The time out counter begins counting when AS or ALE fails. If the time out value is reached before an acknowledge (RDY going high, DTACK going low) is observed then the transfer is terminated and the DEBI_TO flag is set. The time out value is measured int PCI clocks.

In Intel mode a cycle can be terminated in three ways. If RDY is high when RDN or WRN falls, the device is assumed to be ready and the cycle will complete. If RDY is low when RDN or WRN falls, the cycle terminates when RDY returns high or when the timer expires. If RDY does not return high after the time out, the DEBI controller will not start another transaction until RDY is high.
In Motorola mode a cycle can be terminated in two ways. The falling edge of DTACK or when the timer expires.
** A new transaction will not be started until after DTACK returns high.

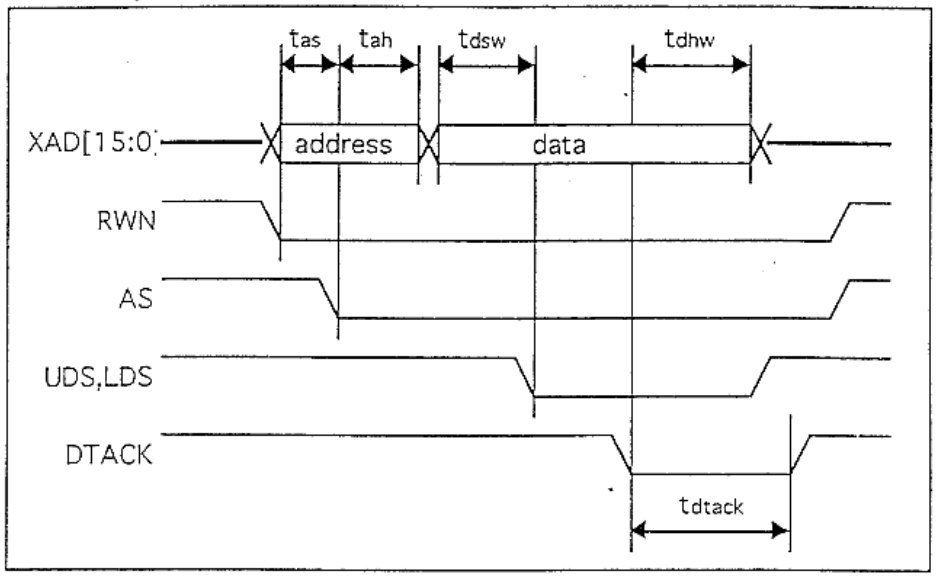
Intel Style Write



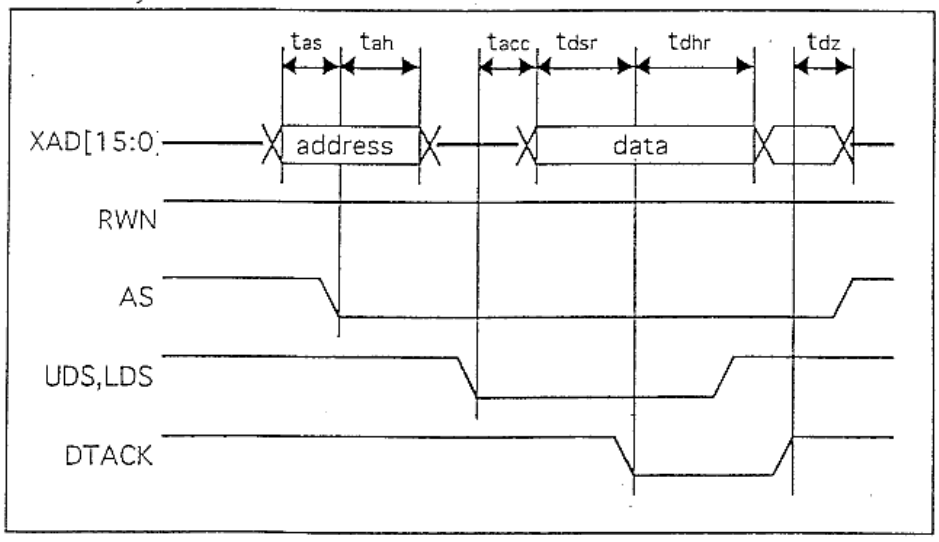
Intel Style Read



Motorola Style Write



Motorola Style Read



11. I²C Port

The SAA7145 is an I²C bus master. The I²C bus is used to configure other Philis components in the system. It can be accessed directly or via RPS.
I²C performs serial byte transfers. The clock chosen is divided down from the PCI system clock to arrive at the I²C clock of around 100 KHz.
Two registers are used to generate I²C transfers. The Transfer Control Register (IICTRF) contains the actual data to be transferred and the transfer type of each byte. The act of placing data in the IICTRF initiates the transfer. The Status and Clock Register (IICSC) contains status and control information used to determine the source of an interrupt and set the speed of the I²C clock.
Each of the three bytes in the IICTRF has two attribute bits associated with it. These two bits are used to describe control function for the corresponding byte. Each byte can be configured as a start byte (11), continue byte (10), end byte (01) or nop byte (00). The start byte determines the direction of the data transfer with bit 0. Continue bytes can be sub-addresses or data. A stop byte is the last byte in a transfer. A nop byte is ignored.

Example:
To transfer the byte A5 to address 40 sub-address 21 the IICTRF should be programmed with 4121A5E4.

IICTRF		Offset: 70
Bits	**Name**	**Description**
31:24	BYTE2	Data/Address Byte 2
23:16	BYTE1	Data/Address Byte 1
15:8	BYTE0	Data/Address Byte 0
7:5	ATTR2	Attribute Information for BYTE2
5:4	ATTR1	Attribute Information for BYTE1
3:2	ATTR0	Attribute Information for BYTE0
1	ERR	General Error Flag
0	BUSY	Transfer in Progress

IICSC		Offset: 74
Bits	**Name**	**Description**
10:8	IICCC	I^2C Clock Control
7	IICIRQ	Interrupt Request
6	rsvd	Reserved
5	APERR	Address Phase Error
4	DTERR	Data Transmission Error
3	rsvd	Reserved
2	rsvd	Reserved
1	ERR	General Error Flag
0	BUSY	Transfer in Progress

IICCC	
Value	**Clock Rate**
7*	PCI Clk / 12
6	PCI Clk / 320
5	PCI Clk / 6400
4	PCI Clk / 480
3*	PCI Clk / 8
2	PCI Clk / 80
1	PCI Clk / 3200
0	PCI Clk / 120

* Do not use these rates, data for reference purposes only.

12. Register Programming Sequencer (RPS)

The SAA7145 uses RPS to provide automatic programming of the internal registers. RPS commands are executed from an external memory location accessed by DMA. These commands take the form of a program that the SAA7145's RPS unit sequentially execute.
To prevent access to registers in active areas of the chip, a register buffer or Shadow RAM is used to store values to be loaded into the internal registers at a "safe" time. This function, called "upload", is performed at the resolution of the timing commands WAIT and CHECK. The shadowed registers are located in addresses 00h to 2Ch. All 12 of the shadowed registers are updated during the upload regardless of whether the values have changed or not. During the upload, internal functions are temporarily halted so care must be taken in using these commands. The PAUSE command can be used for a wait which does not need an upload.
The Event Manager, DEBI, GPIO and I^2C controls are not shadowed so these commands are executed immediately, however the RPS command stream does contain commands to aid in properly timing their execution.
The Utility Register is provided for temporary storage or anything else. The register has no effect on operation.

12.1. RPS Control

RPS activity is controlled by the RPSP register. The address pointer is programmed with the address of the first command in the RPS list. The Enable RPS (ERPS) bit must be set while writing the start address in the RPSP register. Immediately after the ERPS but is set the RPS DMA begins to fetch commands from memory beginning with the command at the address pointer's location. Four Dwords are fetched at a time and loaded into an instruction queue. The RPS unit executes each command sequentially to the end of the queue at which time the RPS DMA loaded the next four Dwords in the RPS list. After each command is executed the address pointer is incremented. If a WAIT, CHECK or PAUSE instruction is encountered in a RPS list, the remainder of the queue is considered invalid and will be reloaded after the conditional has been met.
When an RPS command is accessing the registers, host access to the registers is momentarily held off. The 7145 will execute PCI retry cycles until the register set is available and the host cycle will be completed normally. No special action is required by software. The possibility of a lock up condition is negated by 7145 issuing the Target Disconnect-Retry to the master attempting to access the 7145's internal registers. If such an event does occur, the RPS unit of the 7145 goes into a special state after the current command is completed, until the host retries the disconnected transaction.
The RPS pointer addresses must be Dword aligned. If the RPSP register is read it will indicate the address of command about to be executed. Bit 0 of the RPSP returns the status of the ERPS bit. If RPS is active, any attempt to write to the address pointer will shut down RPS after the current instruction finishes execution.
The RPS Page Address register can be used control RPS writes to system memory. If enabled, accesses outside the RPS page are not allowed and an interrupt is generated. The Page Error Enable (ERPSP) bit is located at bit 0 of the RPS page register, a one enables page errors. This is

bit is a 0 after a reset.
RPS has a safety valve for commands that cause a wait in case the event does not occur. A waiting time out value can be programmed in units of vertical synchs or PCI clocks or both. The feature may be disabled by selecting none.

TOSEL	Time Out Used
00	None
01	PCI Counter Time Out
10	V Sync Counter Time Out
11	Both

12.2. Command Set

The RPS Command set is simple data flow model. Commands are embedded in the data stream itself. There are ten commands available for use in the SAA7145, they are:

- LDREG
- STREG
- STOP
- JUMP
- CHECK
- WAIT
- PAUSE
- IRQ
- CLR
- SET

Commands are always one D Word and the instruction code is always the top byte of the D Word (1st nibble, bits 31:28). The second byte is always reserved and must be written with zeroes. The bottom two bytes contain command control and the meaning varies from command to command. Operand D Words may or may not follow the command.

12.2.1.LDREG

Load Register commands place data into the SAA7145's internal registers. These commands have the data immediately following the command word. The instruction code for LDREG is 5h. The command word has two control bytes, bits 15:8 indicate the number of data D Words the follow. The SAA7145 does not allow for multiple D Word transfers however, so this byte should always be written with a 1. Bits 7:0 indicate the destination starting address of the data word(s). The starting address is the DWORD address of the first register.

Ex.1 Put the value 21400280 into register 18h:

```
50000118
21400280
```

12.2.2.STREG

Store Register commands places the contents of internal registers into system memory. The operand of the Store Register command is an address. The address must be inside the RPS page or an interrupt is generated and RPS stops. Page Errors can be disabled by setting bit 0 in the RPS page register to 1. The instruction code for STREG is 6h. The command word has two control bytes, bits 15:8 indicate the number of data D Words the follow. The SAA7145 does not allow for multiple D Word transfers however, so this byte should always be written with a 1. Bits 7:0 indicate the source starting address of the data word(s). The starting address is the DWORD address of the first register.

Ex.2 Put the value of register 28h into memory address location 034BC004:

```
60000128
034BC004
```

12.2.3.STOP

The Stop instruction halts execution of RPS. It has no operands. The instruction code for the STOP is 3h. When this command is executed the ERPS bit is set to zero and execution stops.

Ex.3 The stop instruction has only one variation:

```
30000000
```

12.2.4.JUMP

The Jump instruction is used to change the RPS Address Pointer. The instruction code for the JUMP is 4h. The D Word following the JUMP indicates the address at which to continue execution. The JUMP command can be made to depend on a status flag. If the specified flag is set when the instruction is executed the JUMP is take, if not then it is ignored. A JUMP with no status flag set is taken immediately. The lower two bytes of the command specify which flags are used to control the JUMP. See the section on Flags for their description.

Ex.4 Start execution from address 00FC0800 (unconditional)

```
40000000
00FC0800
```

12.2.5.IRQ

A IRQ instruction generates an interrupt on the PCI bus. The interrupt is maskable. It's use is up to the programmer. The instruction code for IRQ is 2h. The IRQ command can be made to depend on a status flag. If the specified flag is set when the instruction is executed the RPS_I bit is set, if not the IRQ is ignored. A IRQ with no status flag set is done immediately. The lower two bytes of the command specify which flags are used to control the IRQ. See the section on Flags for their description.

Ex.5 An unconditional IRQ:

```
20000000
```

12.2.6.NOP

A NOP can be executed in many ways. A set of clear flag with unspecified control is good example.

Ex.6 The NOP instruction (by CLR):

```
00000000
```

12.2.7.WAIT

The Wait command is used to temporarily stop execution of the RPS list based on the state of selected status flags. The WAIT command will stop the RPS until the selected event occurs. When the event occurs, execution continues immediately after the registers have been uploaded. If the event has already occurred then it is late, the RPS late bit is set and RPS stops. The instruction code for a WAIT is 9h. The lower two bytes of the command specify which flags are used to control the WAIT. See the section on Flags for their description.

Ex.7 Wait for beginning of an even field:

```
90000004
```

12.2.8.CHECK

The Check Wait command is used to temporarily stop execution of the RPS list based on the state of selected status flags. The CHECK command will stop the RPS until the selected event occurs. If the event has already occurred then execution continues immediately after the registers have been uploaded. The instruction code for a CHECK is 8h. The lower two bytes of the command specify which flags are used to control the CHECK. See the section on Flags for their description. A CHECK command with no flags set performs an unconditional upload.

12.2.9.PAUSE

The Pause command is used to temporarily stop execution of the RPS list based on the state of selected status flags. The CHECK command will stop the RPS until the selected event occurs. If the event has already occurred then execution continues immediately. Registers are not uploaded after a pause command. The instruction code for PAUSE is Ah. The lower two byte of the command specify which flags are used to control the Pause. See the section on Flags for their description.

12.2.10.SET

The Set Flags command is used to set a particular flag (or set of flags). It is most likely that is will only be used for debugging and optimizing purposes. The instruction code for SET is 1h. The lower two bytes of the command specify which flags are set by the SET. See the section on Flags for their description.

12.2.11.CLR

The Clear Flag command is used to clear a particular flag (or set of flags). Clear Flag is used to set up the flags for a wait. The instruction code for CLR is 0h. The lower two bytes of the command specify which flags are cleared by the CLR. See the section on Flags for their description.

12.3. RPS Flags

There are a set of flags used by RPS commands. These are:

DEBID: DEBI Done. Set when DEBI finishes an access. Cleared by the start of a DEBI access.

IICD: I^2C Done. Set when the I^2C bus transaction is finished. Cleared when an I^2C access is initiated.

EVEN: Even field. High during an Even Field. Low during and Odd Field.

ODD: Odd field. High during an Odd Field. Low during an Even Field.

HS: H Count Source. Set when the source line counter reaches the value of the SLCT register. Cleared by Vertical Sync, or when the source line counter is reprogrammed to a line higher than the current line. The Source Line Counter stops after it reaches it's maximum value, it does not roll over to zero and continue counting.

HT: H Count Target. Set when the target line counter reaches the value of the TLCT register. Cleared by the Start of Active Video, or when the target line counter is reprogrammed to a line higher than the current line.

EOL: End of Line. Set when the last pixel of a line is placed in the FIFO, cleared by the start of the next line.

EAW: End of Active Window. Set when the last pixel of the current active window (as defined by the X and Y windowers) is put into the FIFO. Cleared by Start of the next Active Video Window.

GPIO3: GPIO3. Set when the GPIO3 pin goes high. Cleared by the resolution of the wait.

GPIO2: GPIO2. Set when the GPIO3 pin goes high. Cleared by the resolution of the wait.

GPIO1: GPIO1. Set when the GPIO3 pin goes high. Cleared by the resolution of the wait.

GPIO0: GPIO0. Set when the GPIO3 pin goes high. Cleared by the resolution of the wait.

VFE: Video FIFO Empty. Set when the FIFO goes empty, cleared if any data is present int the FIFO.

RPS_S: RPS Semaphore. Set by software in the IMR. Cleared on use.

ANY: ANY/ALL. This flag indicated whether all of the selected flags must be set (logical AND) for a true condition or if any of the selected flags (logical OR) indicate a true condition. This flag has no meaning in SET or CLR commands.

INV: Invert Flag. If this flag is set then the condition is met when leaving the state rather than entering it. This flag has no meaning in SET or CLR commands.

The Bit position of the flags are the same for all commands.

Bit	15	14	13	12	11	10	9	8
Flag	ANY	INV	RPS_S	VFE	GPIO3	GPIO2	GPIO1	GPIO0
Bit	7	6	5	4	3	2	1	0
Flag	EAW	EOL	HS	HT	ODD	EVEN	DEBID	IICD

13. Clipper

The Clipper in the SAA7145 uses a software generated clip mask to provide video window occlusion capability. This method provides an arbitrary number window clips of any size or share. A bit mask is stored in memory by the Host and is Master read via the clipper's own DMA.
The clip mask data is used bit for pixel to determine whether or not a pixel is written to the target. The user selects whether a zero or a one determines active pixels. The Clipper does not operate in the YUV16 output mode because of the UV subsampling.
In the case that the line clip mask is unable to be read in the time required, all the remaining pixels of that line will be clipped and operation continues with the next line.
Software should take care to only change the state of the Clipper outside of the Active Window to assure proper operation.
The Clipper has its own DMA controller and is controlled by the following registers.

Clip Mask DMA Base Address (CDBA): Contains the base address for DMA transfers. This is the address containing the first 32 bits of clipping information. Like the VDBA only the bottom 24 bits are incremented, so the clip mask is limited to a 16 megabyte page.

CDF		Offset: 24
Bits	**Name**	**Description**
28:24	CPO	Pixel Offset into Dword
21:26	CPIX	Number of Dwords per Line of Clip Mask
9:0	CLINE	Number of Lines of Clip Mask

First Pixel Offset (CPO): contains the offset into the first Dword of Clip Data. The number programmed represents the bit position to start from after the optional byte swap. This offset is only used on the first Dword of clip data in a line.
CPIX: contains the number of clip data long words to fetch per line.
CLINE: contains the number of clip data lines of clip data to read.

CDMAC		Offset: 28
Bits	**Name**	**Description**
31	ECLIP	Enable Clipper
20:19	CEND	Endianness Controls
16	CPOL	Clipper Polarity
9:0	CPCH	Line Pitch

Enable Clipping (ECLIP): enables the Clipper DMA to fetch data and present clip data to the Pixel Formatter.
Clip Data Polarity (CPOL): the polarity of this bit is the polarity used for clipping pixels.
Clip Data Pitch (CPCH): this is the difference in address (in system memory) from the first clip data D word of one line to the first clip data D word of the next line.
Endianness (CEND): controls an option byte swap and which end of the Dword to start from. This first and second bits show below are as they appear on the PCI bus (CEND[0] = 0).

CEND[0]	**Swapping**	**1st, 2nd bit**
0	No swap	31,30
1	4 byte swap	7,6

CEND[1]	**Start from**	**1st, 2nd bit**
0	Big End	31,30
1	Little End	0,1

14. PCI Interface

The PCI interface of the SAA7145 is compliant with the PCI Local Bus Specification Revision 2.1 and is both a master and a slave. The PCI configuration space use is given below in the table. Internal registers of the SAA7145 are addressed by using the base address register and adding the register offset. This register offset is also used in RPS programming.

OFFSET	NAME	BIT	TYPE/VALUE	DESCRIPTION
00h	Device ID	31:16	RO/7145h	SAA7145
	Vendor ID	15:0	RO/1131h	Philips Semiconductors
04h	Status Register	31	RO	Detected Parity Error
		30	RO/0b	Signaled System Error
		29	RO	Received Master Abort
		28	RO	Received Target Abort
		27	RO/0b	Signaled Target Abort
		26:25	RO/01b	DEVSEL Timing = Medium
		24	RO	Data Parity Error Detected
		23	RO/1b	Fast Back to Back Capable = Yes
		22	RO/0b	UDF Supported = No
		21	RO/0b	66 MHz Capable = No
	Command Reg.	9	RW/0b	Fast Back to Back Enable
		8	RW/0b	SERR# Enable
		7	RO/0b	Wait Cycle Control
		6	RW/0b	Parity Error Response
		5	RO/0b	VGA Palette Snoop
		4	RO/0b	Memory Write and Invalidate Enable
		3	RO/0b	Special Cycles
		2	RW/0b	Bus Master Enable
		1	RW/0b	Memory Space Enable
		0	RO/0b	IO Space Enable
D8h	Class Code	31:8	RO/048000h	Other Multimedia Device
	Revision ID	7:0	RO/00h	
0Ch	Latency Timer	15:8	RW/00h	This value loaded during configuration
10h	Base Address	31:7	RW/D00000h	This value loaded during configuration
		6:0	RO/00h	
3Ch	Max_Lat	31:24	RO/18h	6 µsec
	Min_Gnt	23:16	RO/09h	2.25 µsec
	Interrupt Pin	15:8	RO/01h	Selects INTA#
	Interrupt Line	7:0	RW/00h	This value loaded during configuration

14.1. The PCI Master

The PCI Master block is responsible for ensuring that all DMAs within the SAA7145 are serviced. It contains a modified round robin mechanism to assure the FIFO overflows and underflows do not occur. The order of hierarchy is as follows:

1. RPS
2. Audio
3. DEBI
4. Clipper
5. Video

The list is repeated until no DMAs are pending or until the latency timer runs out. It is important to note that priority flow is always downwards. The bus owner ignores requests from above if there is a request from below.
Example: if DEBI is the current owner and both RPS and Video have pending DMAs, the next owner will be the Video DMA. However if Video had not requested service then RPS would be the next owner.
Another way to look at this is that each of the five DMAs sees a different priority list. As below:

Owner:	RPS	Audio	DEBI	Clip	Video
highest	Audio	DEBI	Clip	Video	RPS
next	DEBI	Clip	Video	RPS	Audio
next	Clip	Video	RPS	Audio	DEBI
last	Video	RPS	Audio	DEBI	Clip

If Audio owns the bus, then DEBI has the highest priority to be the next owner. If DEBI has no request then it falls to the Clipper, and so on.
In the event of a retry cycle from the PCI target, arbitration will be locked until the retry is resolved.

15. The Event Manager

The SAA7145's event manager contains an integral interrupt and status monitoring system. It is capable of accumulating data for use by software to monitor system performance and provide system control.

15.1. Interrupt Control

The SAA7145 supports interrupts from many different sources. These sources are available as monitor bits in the Interrupt Monitor Register (IMR). Each interrupt can be masked in the Interrupt Enable Register (IER). If an interrupt is enabled and its monitor bit goes active the corresponding bit is set (1) in the Interrupt Status Register (ISR). If any bit of the ISR is active (1) the PCI INTA# pin is set to low.
The IMR and ISR are special registers in that writing to them has special effects. Writing a one to an IMR bit generates an interrupt if the corresponding enable is set. Writing a zero to an IMR bit has no effect on that bit. Writing a one to an ISR bit clears the corresponding interrupt. Writing a zero to this register has no effect on the status bit.
Some sources are provided not with the intention that they will generate interrupts but can be routed to event counters or GPIO pins. Reserved bits always read back as inactive (zero).
Interrupt sources are described below:

PPER: PCI Parity Error on Read. This bit is set when a PCI parity error is detected during a read of PCI data.
PPEW: PCI Parity Error on Write. This bit is set when a PCI parity error is reported by a the target of a PCI write.
PABO: PCI Access Abort. This bit is set when a PCI transfer is terminated with either a target – or master – abort.
DEBI_I: DEBI interrupt. This bit is set when the DEBI port's XIRQ pin goes active.
DEBI_TO: DEBI Time Out Error. This bit is set when a DEBI transfer times out.
DEBI_D: DEBI Done. This bit is set when a DEBI transfer completes.
IIC_E: I^2C Error. This bit is set when the I^2C port finishes a command.
ARI: Audio Range Interrupt. This bit is set by the Audio DMA controller dependent on the programming of the ARNG register.
UPLD: RPS Upload. This bit is set during the time an RPS upload is taking place. Setting the upload bit will force an upload to occur if RPS is not running.
RPS_PE: RPS Page error. This bit is set when RPS tries to write to an address outside the RPS page currently in the RPS page register.
RPS_L: RPS Late Error. This bit is set if RPS is unable to complete execution by the time the original start condition occurs again.
RPS_TO: RPS Time Out. This bit is set if RPS is in a wait state for a duration exceeding the time out value.
RPS_S: RPS Semaphore. This bit is set by software for use in synchronizing the video stream. It is cleared by RPS.
RPS_I: RPS Interrupt. This bit is set by the RPS command INTERRUPT.

VPE: Video DMA Address Protection Error. This bit is set if the Video DMA controller attempts to access an address beyond the FIFO protection address.
VFE: Video FIFO Empty. This bit is set if the video FIFO is empty.
VFO: Video FIFO Overflow. This bit is set if an overflow occurs in the Video FIFO.
AFO: Audio FIFO Overflow. This bit is set if an overflow occurs in the Audio FIFO.
CFU: Clipper FIFO Underflow. This bit is set when the Clip mask isn't loaded in time for use by the clipper.
FID: Field ID. This bit echoes the field ID of the field that is currently being processed.
FIDN: Field ID Not. This bit echoes the invert of the field ID of the field that is currently being processed.
VBI: Vertical Blank Interval. This bit is set when the video source enters vertical blanking.
HBI: Horizontal Blank Interval. This bit is set when the video source enters horizontal blanking.
GP0: GPIO 0 Status. This bit reflects the status of the GPIO0 pin.
GP1: GPIO 1 Status. This bit reflects the status of the GPIO1 pin.
GP2: GPIO 2 Status. This bit reflects the status of the GPIO2 pin.
GP3: GPIO 3 Status. This bit reflects the status of the GPIO3 pin.
EC1: Event Counter 1. This bit is set when event counter 1 reaches its threshold.
EC2: Event Counter 2. This bit is set when event counter 2 reaches its threshold.
MASTER: Master interrupt. In the ISR this bit reflects the sum of the enabled interrupt flags, it cleared automatically when no interrupt is present. In the IER this bit is used as the Master interrupt enable for all flags. This bit must be set for any of the monitor bits to cause an interrupt on INTA#. In the IMR this bit always reads back as zero.

ISR	Offset: 48/configure							
IER	Offset: 4C							
IMR	Offset: 50							
bit	31	30	29	28	27	26	25	24
name	PPER	PABO	PPEW	DEBI_D	DEBI_I	DEBI_TO	IIC_E	IIC_D
bit	23	22	21	20	19	18	17	16
name	ARI	UPLD	RPS_S	RPS_I	RPS_TO	RPS_L	RPS_PE	VFE
bit	15	14	13	12	11	10	9	8
name	VBI	HBI	FIDN	FID	AFO	CFU	VPE	VFO
bit	7	6	5	4	3	2	1	0
name	GP3	GP2	GP1	GP0	rsvd	EC2	EC1	MASTER

15.2. Event Monitors

The SAA7145 has the capability to provide statistical data on any of the interrupt sources whether they are enabled or not. The ECC is used to select the source of the counters ans also to clear and enable them. The Event Counter Selects bits (ECSEL) are used to select which source goes to the counter. ECSEL is loaded with the bit position of the interrupt to count.

A monitor bit can be fed into one of two counters int the Event Counter Register (ECR). The ECR contains two 10 bit counters. Control of the ECR is done in the Event Counter Control register (ECC). The Event Counter Threshold Register (ECT) sets a threshold value for each counter's the monitor bit. When this threshold is reached the corresponding monitor bit is set, the counter resets and continues counting.
The ENx bits enable each counter to begin counting. Writing a one in an ENx bit enables that counter. A zero stops the counter but does not clear it.
The CLRx bits allow the counters to be cleared independently of each other. Writing a one to one of the bit clears the counter to zero, writing a zero has no effect. The CLRx bit is a write only non-persistent bit, reads will always return a zero.
For GPIO counting, the edge being counted can be controlled via the GPIO register.
Note: Event FIDN is not available for counting.

ECC		Offset: 5C
Bits	**Name**	**Description**
15:11	ECSEL2	Interrupt bit position for counter 2
10	rsvd	Reserved
9	EN2	Enable counter 2
8	CLR2	Clear counter 2
7:3	ECSEL1	Interrupt bit position for counter 1
2	rsvd	Reserved
1	EN1	Enable counter 1
0	CLR1	Clear counter 1

ECR		Offset: 54
Bits	**Name**	**Description**
31:20	rsvd	Reserved
19:10	ECR2	Event Counter Register 2
9:0	ECR1	Event Counter Register 1

ECT		Offset: 58
Bits	**Name**	**Description**
31:20	rsvd	Reserved
19:10	ECT2	Event Counter Threshold 2
9:0	ECT1	Event Counter Threshold 1

16. General Purpose I/O Port (GPIO)

The GPIO port is as set of four general purpose/status I/O pins. These pins can be used to monitor any of the internal status bits in the IMR or to communicate status to or from another device on the board. The pins power up as inputs and have soft internal pull-up resistors. They are each programmed by a set of bits located in the GPIO Control Register.
The minimum pulse width of a signal on a GPIO pin must be at least two PCI clocks to ensure that the signal will be captured.

DIR: this bit is used to control the direction of the GPIO pin. A zero indicated the pin is an input.

IOS: this bit reports the status of the pin in input mode and is written to in output mode.

MIO: this bit is used to set up the bit as an interrupt monitor bit or a IO pin. A zero sets the pin to IO mode, a one will put the pin in monitor mode. If a pin is defined as an input and is in monitor mode, the data on the pin is ignored.

MSEL: In monitor mode, the 5 bit register is used to select which one of the Interrupts appears on the output pin. The number programmed represents the bit position of the interrupt to appear on the pin.
In IO mode, the MSEL[1:0] bits are used to determine the transition edge used to cause an event of interrupt:
00: level only (no recommended for interrupt use)
01: rising edge
10: falling edge
11: both edges

GPIOC	Offset: 50			
bits	31:27	26	25	24
name	MSEL3	MIO3	DIR3	IOS3
bits	23:19	18	17	16
name	MSEL2	MIO2	DIR2	IOS2
bits	15:11	10	9	8
name	MSEL1	MIO1	DIR1	IOS1
bits	7:3	2	1	0
name	MSEL0	MIO0	DIR0	IOS0

17. Pin List

The table below describes the pins and their functions.

Group	Pin Name	Type	Description
AUD	ACLK	Ip	Audio I^2S Bit Clock
AUD	ADAT	Ip	Audio I^2S Data Input
AUD	AWS	Ip	Audio I^2S Word Select
DEBI	AS_ALE	O	DEBI Port Address Strobe or Address Latch Enable
DEBI	DTACK_RDY	Ip	DEBI Port Data Transfer Acknowledge or Ready
DEBI	LDS_RDN	O	DEBI Port Data Transfer Control Signal
DEBI	RWN_SBHE	O	DEBI Port Data Transfer Control Signal
DEBI	XIRQ	I	DEBI Port Interrupt Request
GPIO	GPIO[3:0]	I/Oph	General Purpose Input/Output Pins
IIC	SCL	O	I^2C Serial Clock. Clock rate is determined by the value programmed int the IICCTL register
IIC	SDA	I/Opo	I^2C Serial Data
PCI	AD[31:0]	I/O	PCI Multiplexed address and data bus
PCI	C/BE#[3:0]	I/O	PCI Multiplexed Command and Byte Enables
PCI	CLK	I	PCI Clock
PCI	DEVSEL#	I/O	PCI DEVSEL# signal
PCI	FRAME#	I/O	PCI Transfer Control
PCI	GNT#	I	PCI Bus Arbitration Signal
PCI	IDSEL	I	PCI Transfer Control
PCI	INTA#	O	PCI Interrupt Line
PCI	IRDY#	I/O	PCI Transfer Control
PCI	PERR#	I/O	PCI Parity Error Signal
PCI	REQ#	O	PCI Bus Arbitration Signal
PCI	RST#	I	PCI Bus Global Reset
PCI	STOP#	I/O	PCI Transfer Control
PCI	TRDY#	I/O	PCI Transfer Control
PWR	VDD	-	Power pins. 5V
PWR	VSS	-	Ground pins. 0V
VP	FID	I/Oo	Field Identification Input
VP	HS	I/Oo	Horizontal Sync
VP	LLC	Ip	Line Locked Click. The video data clock
VP	PXQ	Ip	Pixel Qualifier. Valid data is defined by this pin high and a rising edge on LLC.
VP	VPI[15:8]	Ip	Upper eight bits of the video port data. Also these 8 bits are used to receive the D1 stream
VP	VPI[7:0]	Ip	Lower eight bits of the video ports data. These bits are ignored in D1 mode.
VP	VS	I/Op	Vertical Sync

I = Input Pin
O = Output Pin
I/O = Bi-direction Pin
p = weak Internal pull-up
h = high drive output (12mA), standard drivers are 6mA
o = Open Collector output

18. Pin Out

Pin #	Name	Group	Pin #	Name	Group	Pin #	Name	Group	Pin #	Name	Group
1	VDD	PWR	41	VSS	PWR	81	VDD	PWR	121	VSS	PWR
2	VDD	PWD	42	VSS	PWR	82	VDD	PWR	122	VSS	PWR
3	VPI[11]	VP	43	AD[24]	PCI	83	AD[10]	PCI	123	XAD[0]	DEBI
4	VPI[12]	VP	44	C/BE#[3]	PCI	84	AD[9]	PCI	124	XAD[1]	DEBI
5	VPI[13]	VP	45	IDSEL	PCI	85	AD[8]	PCI	125	XAD[2]	DEBI
6	VPI[14]	VP	46	AD[23]	PCI	86	C/BE#[0]	PCI	126	XAD[3]	DEBI
7	VPI[15]	VP	47	VDD	PWR	87	AD[7]	PCI	127	VDD	PWR
8	VDD	PWR	48	VSS	PWR	88	VDD	PWR	128	VSS	PWR
9	VSS	PWR	49	AD[22]	PCI	89	VSS	PWR	129	XAD[4]	DEBI
10	VS	VP	50	AD[21]	PCI	90	AD[6]	PCI	130	XAD[5]	DEBI
11	HS	VP	51	AD[20]	PCI	91	AD[5]	PCI	131	XAD[6]	DEBI
12	LLC	VP	52	AD[19]	PCI	91	AD[4]	PCI	132	XAD[7]	DEBI
13	PXQ	VP	53	VDD	PWR	93	AD[3]	PCI	133	VDD	PWR
14	VDD	PWR	54	VSS	PWR	94	VDD	PWR	134	VSS	PWR
15	VSS	PWR	55	AD[18]	PCI	95	VSS	PWR	135	SCL	IIC
16	FID	VP	56	AD[17]	PCI	96	AD[2]	PCI	136	SDA	IIC
17	AWS	AUD	57	AD[16]	PCI	97	AD[1]	PCI	137	rsvd	
18	ACLK	AUD	58	C/BE#[2]	PCI	98	AD[0]	PCI	138	GPIO[3]	GPIO
19	ADAT	AUD	59	FRAME#	PCI	99	XAD[15]	DEBI	139	GPIO[2]	GPIO
20	VDD	PWR	60	VDD	PWR	100	VDD	PWR	140	VDD	PWR
21	VSS	PWR	61	VSS	PWR	101	VSS	PWR	141	VSS	PWR
22	INTA#	PCI	62	IRDY#	PCI	102	XAD[14]	DEBI	142	GPIO[1]	GPIO
23	RST#	PCI	63	TRDY#	PCI	103	XAD[13]	DEBI	143	GPIO[0]	GPIO
24	VSS	PWR	64	DEVSEL#	PCI	104	XAD[12]	DEBI	144	VPI[0]	VP
25	CLK	PCI	65	STOP#	PCI	105	XAD[11]	DEBI	145	VPI[1]	VP
26	VDD	PWR	66	PERR#	PCI	106	VDD	PWR	146	VPI[2]	VP
27	VSS	PWR	67	VDD	PWR	107	VSS	PWR	147	VDD	PWR
28	GNT#	PCI	68	VSS	PWR	108	XAD[10]	DEBI	148	VSS	PWR
29	REQ#	PCI	69	rsvd	PCI	109	XAD[9]	DEBI	149	VPI[3]	VP
30	AD[31]	PCI	70	PAR	PCI	110	XAD[8]	DEBI	150	VPI[4]	VP
31	AD[30]	PCI	71	C/BE#[1]	PCI	111	RWN_SHBe	DEBI	151	VPI[5]	VP
32	VDD	PWR	72	AD[15]	PCI	112	VDD	PWR	152	VPI[6]	VP
33	VSS	PWR	73	VDD	PWR	113	VSS	PWR	153	VDD	PWR
34	AD[29]	PCI	74	VSS	PWR	114	AS_ALE	DEBI	154	VSS	PWR
35	AD[28]	PCI	75	AD[14]	PCI	115	LDS_RDN	DEBI	155	VPI[7]	VP
36	AD[27]	PCI	76	AD[13]	PCI	116	UDS_WRN	DEBI	156	VPI[8]	VP
37	AD[26]	PCI	77	AD[12]	PCI	117	DTACK_RDY	DEBI	157	VPI[9]	VP
38	AD[25]	PCI	78	AD[11]	PCI	118	XIRQ	DEBI	158	VPI[10]	VP
39	VDD	PWR	79	VSS	PWR	119	VDD	PWR	159	VSS	PWR
40	VDD	PWR	80	VSS	PWR	120	VDD	PWR	160	VSS	PWR

A.Appendix A: RPS Programmer's Reference

A.1.Programming Examples

RPS has many possible uses, below are couple of the more likely.
In general, an RPS list is composed of two sections although this is not requires. The first section is used for general configuration and setup. The first section is optional because the internal startup programming can also be accomplished by th host. The second section is the time dependent part. This section is where the registers are reprogrammed for the different tasks.
It is not recommended that the registers be accessed by the host while RPS is running, these accesses are permitted however and will not cause the chip to hang but may have unpredictable results. There is one exception to this rule: if a register is not used in the RPS list, it would be safe to reprogram it from the host during RPS execution.

A.2.RPS Basics

This RPS code segment should be used to cleanly switch video on after a move to a new location or a change in size, format or scaling factor. It is intended to be used in applications where multiple destinations and special video tasks are not requires.

```
      Ref:Hex Code  Op Code   Operand   Comment
1   where:50000100    LDREG   01,00     Load reg 0
2         FB000000            FB000000  00.Vid DMA Base
3         50000104    LDREG   01,04     Load reg 4
4         FB080000            FB080000  04.Vid DMA Prot
5   scale:5000001C    LDREG   01,0C     Load reg 0C
6         00000003            00000003  0C.Port Control
7         50000118    LDREG   01,18     Load reg 18
8         94000400            94000400  18.Scaling mode
9  winsiz:50000110    LDREG   01,10     Load reg 10
10        00880280            00880280  10.H Source Ctl
11        50000114    LDREG   01,14     Load reg 14
12        000D00F0            000D00F0  14.V Source Ctl
13 format:50000108    LDREG   01,08     Load reg 8
14        8CA01000            8CA01000  08.Vid DMA Ctl
15        00000080    CLEAR   EAW       Clear the EAW Flag
16 upload:80000080    CHECK   EAW       Wait for EAW
17   stop:30000000     STOP             Stop RPS
```

The above code assumes video is already running and should be used to (re)start the video DMA. To use this program to initialize the chip modify the list as shown below.

```
13 upload:80000000    CHECK             Upload
14   stop:30000000     STOP             Stop
```

A.3.Using RPS to send video to two locations

RPS can be used to divide an incoming video stream in to two field based streams with separate destinations. The following program sets the SAA7145 to send two NTSC fields to different locations. The off fields are sent to the frame buffer in RGB32 scaled to 200x150. The even fields are sent to system memory in YUV16 at 320x240. The list assumes a base address of 00400000.

```
      Ref: Hex Code  Op Code   Operand   Comment
1   setup: 5000030C    LDREG   01,0C     Load reg C
2          00000F03       DW   00000F03  0C.Vid Port Ctl
3          50000110    LDREG   01,10     Load reg 10
4          00800280       DW   00800280  10.H Source Ctl
5          50000114    LDREG   01,14     Load reg 14
6          001400F0       DW   001400F0  14.V Source Ctl
7          80000000    CHECK              Force Upload
8     odd: 50000100    LDREG   01,00     Load reg 0
9          A0200000       DW   A0200000  00.Vid DMA
10         50000108    LDREG   01,08     Load reg 8
11         F8801000       DW   F8801000  08.Vid Ctl
12         50000118    LDREG   01,18     Load reg 18
13         21400280       DW   21400280  18.Scaling Mode
14         00000008    CLEAR   ODD       Clear the odd flag
15         80000008    CHECK   ODD       Wait for odd field
16   even: 50000100    LDREG   01,00     Load reg 0
17         00600000       DW   00600000  00.Vid DMA
18         50000108    LDREG   01,08     Load reg 8
19         F8840140       DW   F8840140  08.Vid Ctl
20         50000118    LDREG   01,18     Load reg 18
21         22000400       DW   22000140  18.Scaling Mode
22         00000004    CHECK   EVEN      Clear the even flag
23         80000004    CHECK   EVEN      Wait for even field
24         40000000     JUMP              Jump always
25         0040001C       DW   odd       to odd
```

Explanation:
This RPS program is divided into three sections. Setup initializes the registers. Section Odd prepares the SAA7145 from transfer of the odd field to the frame buffer. Section Even prepare it for transfer of the even field to system memory.
Lines 1-6: load three registers beginning with register 0C. These three registers set up the Window Generator and the Video port. Since these things do not change between fields they are placed outside the main loop. Note that although three LOAD commands are performed here the values are only written to the Shadow RAM, not the working registers. If any registers in the shadow RAM had been previously used (by another application, etc.) they will still contain that information so it is always a good idea to completely reload the shadow RAM before running a new RPS list.
Line 7: this line contains the unconditional CHECK. This instruction moves the data in the shadow

RAM into the actual working registers. Alternatively, the entire setup part of the RPS list could be omitted and instead the initial configuration could be written to the Shadow RAM followed by an upload generated via the IMR Upload Bit.

Lines 8-11: this is the beginning of the main loop. The odd segment loads the DMA base address and configures the DMA for the odd field. These registers set up the DMA to the frame buffer and configure the pixel formatter for RGB data. Bit 31 of the Video DMA Control Register (VDMAC) is set to start the video DMA. It is important to note that this does not happen until the registers are uploaded into the working register from the Shadow RAM.

Lines 12-13: load register 18. This command loads the scaler control. It is configured for 200x150 with an FMODE of 2.

Lines 14-15: line 14 contains the CLEAR ODD which clears the flag. This is needed to ensure that video capture does not begin in the middle of a field. Line 15 contains the conditional CHECK ODD which waits until the odd field. When the odd field flag is set the Shadow RAM is uploaded into the working registers. This enables the SAA7145 to begin it's video DMA when the Active Video Window is reached. Execution of the RPS list continues immediately after the upload.

Lines 16-21: these commands load the shadow RAM with the operation values for the event field. This is 320x240 YUV data with an FMODE of 2.

Lines 22-23: this segment contains the CLEAR EVEN and the conditional CHECK EVEN. These lines are used to upload the shadow RAM into the working registers at the beginning of the even field. Execution of the RPS list continues immediately after the upload.

Lines 24-25: the JUMP command is the looping point of the list. The RPS pointer is moved to the address that follows this command, in this case the address of the odd field configuration.

B.Appendix B: Interfacing to Standard Philips Parts

B.1.Programming Hints

The most likely candidates for connection to the Video input port are the SAA7110 and SAA7111. The following tables indicate what to put in the registers for various inputs.

SAA7145 to SAA7111 Programming

	YUV16	**YUV16**	**D1**	**D1**	**?**
SAA7111	**NTSC**	**PAL**	**NTSC**	**PAL**	**?**
Subaddr	**CVBS***	**?**	**?**	**?**	**Svideo****
2	D8				DD
3	23				
4	0				
5	0				
6	F0				
8	A8				
9	12	2	12	2	
A	80				
B	47				
C	40				
D	0				
E	1				
10	48		C8	C8	
11	0C				
12	80				
SAA7145					
VPZ	1		0	0	
XSS	1		0	0	
SF	1		0	0	
FD[]	3		0	0	
XWS	88	AE	1	1	
XWZ	2D0	300			
YWS	0C	11	0D	12	
YWZ	F0	11F			

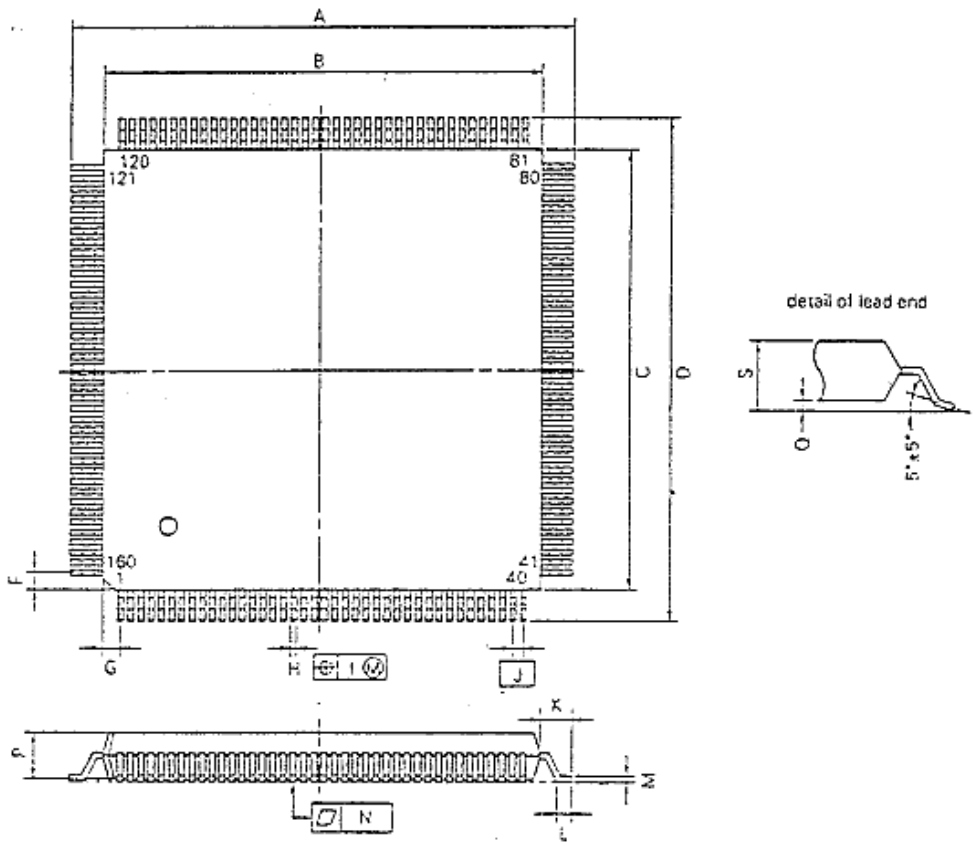
* Assumes Mode 0
* Assumes Mode 6

SAA7145 to SAA7110 Programming

SAA7145	NTSC	PAL
VPZ	1	1
XSS	1	1
SF	0	0
FD[]	0	0
XWS	88	AE
XWZ	280	300
YWS	0C	11
YWZ	F0	11F

C.Appendix C: Package Outline

160-PIN PLASTIC QFP (□28)



NOTE
Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	32.0±0.4	1.260±0.016
B	28.0±0.2	$1.102^{+0.009}_{-0.008}$
C	28.0±0.2	$1.102^{+0.009}_{-0.008}$
D	32.0±0.4	1.260±0.016
F	1.3	0.051
G	1.3	0.051
H	0.30±0.10	$0.012^{+0.004}_{-0.005}$
I	0.15	0.006
J	0.65 (T.P.)	0.026 (T.P.)
K	2.0±0.2	$0.079^{+0.009}_{-0.008}$
L	0.8±0.2	$0.031^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.12	0.005
P	3.7	0.146
Q	0.1±0.1	0.004±0.004
S	4.0 MAX.	0.157 MAX.