CHAPTER 1 OVERVIEW

1.1 FEATURES

- ◆ Supports system-level navigation processing
- ◆ Supports DVD 1.0Video Object (VOB) bit streams
- ◆ Decodes MPEG-2 and MPEG-1 video in real time
- ◆ Display: NTSC (720x480 @ 60 fps) and PAL (720x576 @ 50 fps)
- ◆ Decodes Dolby AC-3, LPCM, or MPEG-1 (Layer 1 & Layer 2) audio samples and outputs in 2 channels
- ◆ Supports sub-picture decoding, DCC, Closed Caption, DSI, PCI, and HLI parsing
- ◆ Provides programmable OSD and Digest functions
- ◆ Provides letterbox, pan & scan, and unaided 3:2 pull-down
- ◆ Requires only 2MB SDRAM (1Mx16)
- ◆ Host interface supports 8-/16-bit microcontroller or ISA mode
- ◆ Supports Video CD, CD-I, and CD-DA
- ◆ Power management
- ◆ Direct interface to industry-standard NTSC/PAL encoders, audio DACs, DVD/CD-DSP, and SDRAM
- ◆ Package: 160-pin PQFP, TQFP
- ◆ Supports S/P DIF digital audio output
- ◆ PC Player development supported by reference designs and navigation software

1.2 GENERAL OVERVIEW OF TROIKA

The Troika is a cost-effective, single-chip solution for DVD and other consumer electronic systems that require MPEG decompression. The Troika provides MPEG-2 and MPEG-1 video decompression, incorporated with Dolby AC-3, MPEG-1, and Linear PCM (LPCM) audio decompression. The Troika fully supports ISO 13818-2 MPEG-2 Main Profile at Main Level (MP@ML), which specifies the coded representation of video data and the decoding process required to reconstruct pictures. The Troika also decodes ISO 13818-3 bit streams, which specify the coded representation of audio data. The Troika decodes audio for 5.1-channel Dolby AC-3, which is mapped into two channels (MPEG-1, Layer 1 & Layer 2) and downmixes up to eight LPCM channels into two channels.

Additionally, sub-picture decoding, on-screen display (OSD), automatic letterbox, automatic pan & scan, unaided 3:2 pull-down, display control commands (DCC), and Closed Caption parsing are supported. Sub-picture decoding is a run-length compressed bitmap that is overlaid on top of the MPEG reconstructed video and used for menus, subtitles, karaoke, and simple animation. The Troika also provides a 16-color (64K palette) programmable on-screen display (OSD) function that allows text and graphics overlay on full-motion video. OSD provides the user with a visual feedback to a currently running function such as disk play, fast forward, pause, or any other display desired by the system designer. Automatic letterbox provides vertical filters that scale a 16:9 coded video image onto a 4:3 display. Automatic pan & scan scales a 16:9 coded image onto a 4:3 display by selecting a centralized point of reference within a film sequence and using that point of reference as the center of the 4:3 display. Unaided 3:2 pull-down converts progressive film sequences to the display frame rate of interlaced television. DCC is used to set the start address in pixel data (PXD), set colors, set contrast, set sub-picture screen position, start/stop display, and set Change Color/ Contrast Command (CHG_COLCON) areas. A Closed Caption feature is included to aid the hearing impaired.

The Troika device is designed to minimize the system cost in two ways. First, cost is minimized by performing on-chip MPEG-2 Program stream parsing, MPEG-1 System stream parsing, audio/video decoding, audio/video synchronization, PLL/clock generation, sub-picture decoding, OSD, and scaling. Second, a glueless interface is provided. The Troika connects directly to the video encoder (CCIR-601 or CCIR-656), audio DAC, synchronous DRAM (SDRAM), CD-DSP, DVD-DSP, and MPEG-2 Program stream devices, thus eliminating any need for additional connective components. In the event an external multichannel sound processor is required for bypass audio, the Troika has on-board an IEC-958 (S/P DIF) interface to pass along the compressed audio bit stream.

The Troika requires only one 1-Mbit x16 SDRAM to support PAL and NTSC MPEG-2 decoding, and supports 67-MHz speed memory, the lowest cost SDRAM grade. An optional 256-Kbit x16 or 1-Mbit x16 SDRAM can be added for extended OSD graphics and color capabilities, as well as allowing an oversized video buffer verifier (VBV) buffer. The Troika parses both MPEG-2 Program and MPEG-1 System bit streams, decodes and synchronizes the video and audio bit streams, and plays back in real time in CCIR-601/CCIR-656 NTSC or PAL format. Selected image sizes below 720x480 (NTSC) and 720x576 (PAL) are scaled to meet CCIR-601 format requirements. The Troika consists of Oak Technology's proprietary 32-bit RISC engine core for chip sequence control and management, audio/video synchronization, error concealment handling, data flow control, and programmable features. Additional features include OSD free video outputs and an audio pass-through with an IEC-958 (S/P DIF) interface for glueless support to external multichannel sound processors.

The Troika supports seven different bus interfaces:

1. The host bus interface supports an 8-bit or 16-bit microcontroller interface and 16-bit ISA.
2. Supports the DVD-DSP interface or DVD-DSP.
3. Supports a CD-DSP or a DVD/CD-DSP interface.
4. Supports a video display bus of either one 16-bit, 13.5-MHz (CCIR-601) interface or two 8-bit, 27-MHz (CCIR-656) pixel interfaces.
5. Supports the Troika's on-chip SDRAM controller interfacing to one 1-Mbit x16 SDRAM (required), and offers optional expansion to an additional 256-Kbit x16 or 1-Mbit x16 SDRAM (Hitachi's HM5216165 or NEC's uPD4516161, for example).
6. Supports an audio DAC interface designed so that any of the popular audio DACs can be used through simple programming.
7. Supports an IEC-958 (S/P DIF) output. This IEC-958 interface provides all the digital format conversions on-chip so that the Troika can pass compressed audio, such as Dolby AC-3 or MPEG-2 multichannel audio, to an external processor.

The target applications for the Troika are consumer electronics applications, such as DVD playback devices.

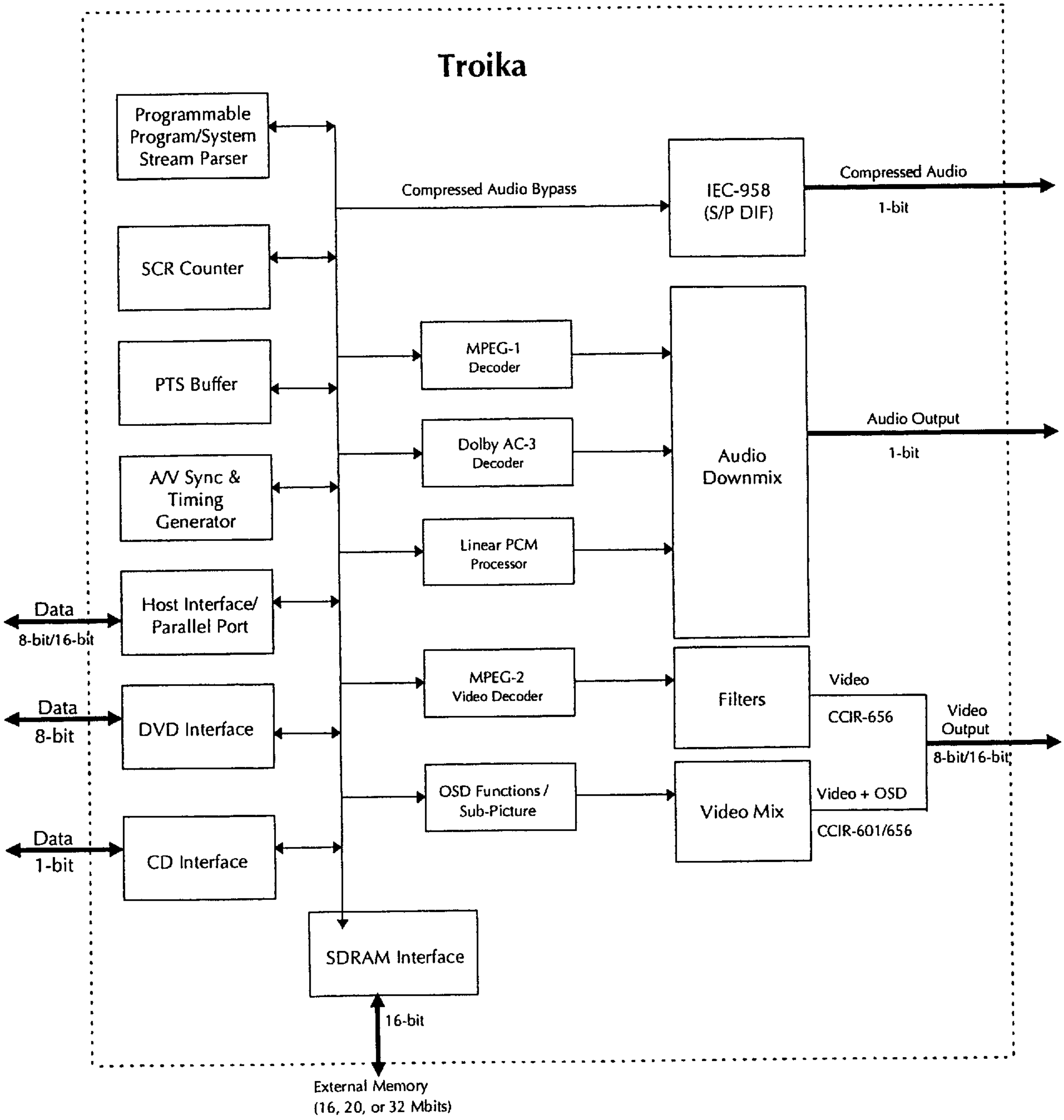


Figure 1-1: *Troika Functional Block Diagram*

1.3 FEATURE DESCRIPTIONS

1.3.1 ACCEPTS DVD VOB BIT STREAMS

- ◆ VOB refers to a video object and is the MPEG Program stream. This stream is comprised of the following elementary streams: video, audio, sub-picture, presentation control information (PCI), and data search information (DSI).

1.3.2 PROGRAM STREAM PARSER

- ◆ Provides MPEG-2 Program stream parsing
  - A Program stream is composed of one or more packetized elementary streams (PESs) with a common timebase (e.g., audio/video streams interleaved).
  - MPEG-2 Program streams are typically used in relatively error-free environments such as in a DVD video player.

- ◆ Provides MPEG-1 System stream parsing
  - For VideoCD applications.

- ◆ Parses auxiliary data/audio output for output via the IEC-958 port
  - Bypasses on-chip processing by the Troika and outputs the compressed stream to a separate processing chip. An example includes multichannel processing for AC-3.

1.3.3 AUDIO

- ◆ Decodes 5.1-channel AC-3 and downmixes to a 2-channel output.

- ◆ Decodes ISO 13818-3 audio bit streams (MPEG-1, Layer 1 & Layer 2)
  - ISO/IEC 11172-3 (MPEG-1 Audio)
  - In Layer 1, one finds the basic 32-band filtering with fixed segmentation and a psychoacoustic model that is used for allocating bits dynamically. Layer 1 was developed with low compression/decompression (CODEC) complexity in mind.
  - Layer 2 is similar to Layer 1, but achieves better performance through a few modifications. The bit allocation, the scale factors, and the sample values include additional coding. Typically used for broadcast audio.

- ◆ Downmixes up to 8 channels of Linear PCM data and provides a 2-channel output
  - Uncompressed LPCM audio data offers high-definition audio with eight audio channels.
  - Downmixing enables the user to decode 8-channel LPCM audio with equipment designed for 2-channel stereo.

- ◆ Full 24-bit audio data path
  - Provides high-quality audio decompression.

- ◆ Mono, dual mono, joint stereo, stereo
  - *Mono* is one-channel output
  - *Dual mono* has two separate channels and is used for functions such as bilingual programming
  - *Joint stereo* is a specific coding mode in Layer 2 audio coding in which the upper frequencies of a stereo signal are joined and coded as intensity stereo in order to conserve valuable bits and improve overall quality
  - *Stereo* has separate channels for the left and right channels.

- Serial PCM audio output is compatible to a standard audio DAC. The outputs have a bit clock, left/right clock, and over-sample clock to interface with an over-sampling DAC. Outputting audio test tones in a programmable test mode is also valuable.

- Supports a 384-Kbps bit rate and audio sampling rates of 32 KHz, 44.1 KHz, 48 KHz, and 96 KHz.

- The audio decoder will mute during illegal bit stream errors, channel errors, and reconstruction errors. It provides both soft and hardware mute.

- Provides compressed audio bypass
  - Via the IEC-958 output, any compressed signal (AC-3, audio MPEG-2, etc.) may be sent to a separate receiver for multichannel decoding.

- Playback of CD-DA
  - The standard format for digital audio compact discs.
  - Comprised of uncompressed PCM audio data.

1.3.4 MPEG-2 VIDEO

- Supports both NTSC & PAL resolutions
  - 720x480 @ 30 Hz for NTSC.
  - 720x576 @ 25 Hz for PAL.

- Decodes MPEG-2 ISO 13818-2 Main Profile/Main Level (MP@ML) video
  - Main Profile specifies the 4:2:0 chroma format. The color difference channels (Cb, Cr) have half the resolution in both the horizontal and vertical direction with respect to luminance (Y).
  - Main Level supports NTSC and PAL resolutions.

- Supports an 8-bit (CCIR-656) or 16-bit (CCIR-601) pixel output
  - Digital coding standards for television that are applicable to both the NTSC as well as PAL technologies.

- Supports I, P, and B frames
  - I-picture (intra-coded picture) is picture-coded using information only from itself. It does not use other pictures as a reference.
  - P-picture (predictive-coded picture) is a picture that is coded using motion-compensated prediction from past reference pictures.
  - B-picture (bi-directionally predictive-coded picture) is a picture that is coded using motion-compensated prediction from past and/or future reference pictures.

- Backward compatible to MPEG-1, ISO 11172
  - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbps
  - Necessary for VideoCD support.

- Supports continuous bit streams of 15-Mbps compressed video and 1-Mbps compressed audio as defined in the MPEG-2 specification.

- Supports 4:2:2 output (scaled)
  - Applications for the 4:2:2 format can be found in professional broadcasting, editing, and contribution-quality distribution environments.

- ◆ Unaided 3:2 pull-down, automatic letterbox, pan & scan
  - *3:2 Pull-Down.* Converts progressive film sequences to the display frame rate of interlaced television. Most film sources are captured at 24 fps. In order to achieve the necessary 30 fps required by a standard NTSC signal, 6 out of the 24 frames are repeated to achieve the necessary 30 frames for proper NTSC display. These six frames do not need to be encoded into the MPEG bit stream, which minimizes data requirements.
  - *Letterbox.* Vertical filters that scale a 16:9 coded video image onto a 4:3 display. This feature is necessary since the majority of movies are coded for the 16:9 aspect ratio, while most television displays have a 4:3 aspect ratio. You've seen this as black bars that shrink the height of the picture of a television screen while movie credits scroll.
  - *Pan & Scan.* Allows viewing of a 16:9 coded image on a 4:3 display by using the director's discretion to "chop off" parts of the left/right edges of the screen that is not significant to the story line.

- ◆ Supports 6:5 PAL-to-NTSC and 5:6 NTSC-to-PAL conversion
  - For PAL-to-NTSC conversion, 576 lines are converted to 480 lines, and 25 fps is converted to 30 fps
  - For NTSC-to-PAL conversion, 480 lines are converted to 576 lines, and 30 fps is converted to 25 fps.

- ◆ Extracts Closed Caption data from the video stream and provides the data to the external host

- ◆ Uses a proprietary B-picture regioning scheme to reduce SDRAM requirements, with <u>no loss</u> of data.

1.3.5 SUB-PICTURE DECODER

- ◆ Supports sub-picture decoding, display control commands (DCC), and Closed Caption
  - A sub-picture is a simple picture intended to be superimposed over the video. The display size may vary, but the size is bounded to parameters defined in CCIR-601 (720x480 for NTSC or 720x576 for PAL).
  - DCC can set the start address in pixel data (PXD), set colors, set contrast, set sub-picture screen position, start/stop display, and set Change Color/Contrast Command areas.
  - Closed Caption is used in video to aid the hearing impaired.

- ◆ Provides data search information (DSI), presentation control information (PCI), and highlight information (HLI) support
  - DSI packets contain navigation information that makes it possible to search for data while maintaining seamless playback of video. Typically used for fast forward/fast backward play.
  - PCI provides information about the timing and presentation (aspect ratio, angle, etc.) of a program
  - HLI enables the user to scroll through a menu bar and highlight an item of choice.

1.3.6 GRAPHICS OSD

- ◆ On-screen display (OSD) allows text and graphics to be overlaid on full-motion video.
  - OSD provides the user with a visual feedback to a currently running function such as disk play, fast forward, pause, or any other display desired by the system designer.

- ◆ Supports full-screen resolution graphics.

- ◆ 16 colors from a color palette that includes 64K choices.

- ◆ Provides digest function support
  - An interactive, on-screen menu system.
  - Provides a picture-based menu system.
  - Indexing provides the ability to navigate through the disk to various titles or chapters.
  - Discrete video scaling allows flexibility within the number of menu choices desired.

- Multi-level translucency and transparent color in programmable graphical bit-mapped OSD
  - Allows proprietary character design.

- Multi-level programmable fading and contrast.

1.3.7 DVD-DSP

- Supports an 8-bit parallel interface.

- Accepts the DVD-DSP system clock that can be used as the Troika system clock.

- Uses request/acknowledge scheme for data transfer.

- Glueless interface between the DVD-DSP and Troika
  - Provides simple design implementation and reduces overall system cost.

1.3.8 CD-DSP

- Supports a full CD-DSP interface including subcoding.

- Provides CD-ROM controller function with ECC (error correction code) and block decode.

- Supports 1-bit serial interface.

1.3.9 MISCELLANEOUS

- Requires only one 16-Mbit, 1-Mbit x16 SDRAM.

- Supports optional memory extensions for either one 1-Mbit x16 SDRAM or one 256-Kbit x16 SDRAM.
  - Provides extended OSD graphics and color capabilities, as well as allowing an oversized VBV buffer.

- Supports various SDRAM, 67 MHz and higher.

- Supports CCIR-601 & CCIR-656 video output
  - CCIR-601 describes how to sample the component signal and how to structure the data. It is the most common studio standard for digital video based on analog component 4:2:2 signals.
  - CCIR-656 is the standard describing how to interface to a CCIR-601 formatted signal.

- Power management mode
  - Saves energy when not in use.

- Supports 8-bit or 16-bit microcontroller interface (Motorola or Intel).

- Scales to 720x480 for NTSC or 720x576 for PAL from 352x240 or 352x288, respectively.

- Provides external timing for AC-3 audio bypass mode.

- Supports automatic error recovery.

- Internal PLL synched to external 13.5-MHz, 27-MHz, or 67.5-MHz clock.

- 0.35µ technology.

- 3.3-volt operation.

CHAPTER 2 PIN DESCRIPTIONS

2.1 PIN DIAGRAM

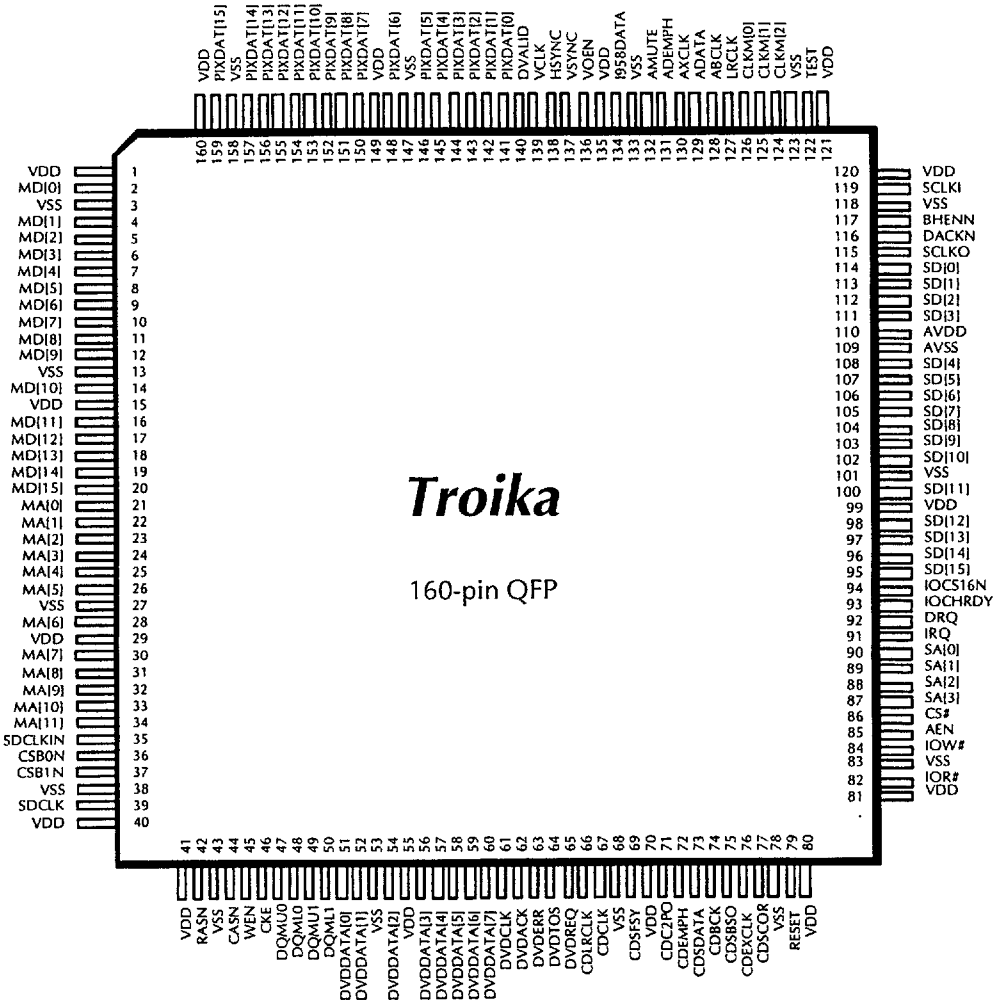


Figure 2-1: *Pin Diagram*

2.2 EXTERNAL INTERFACE

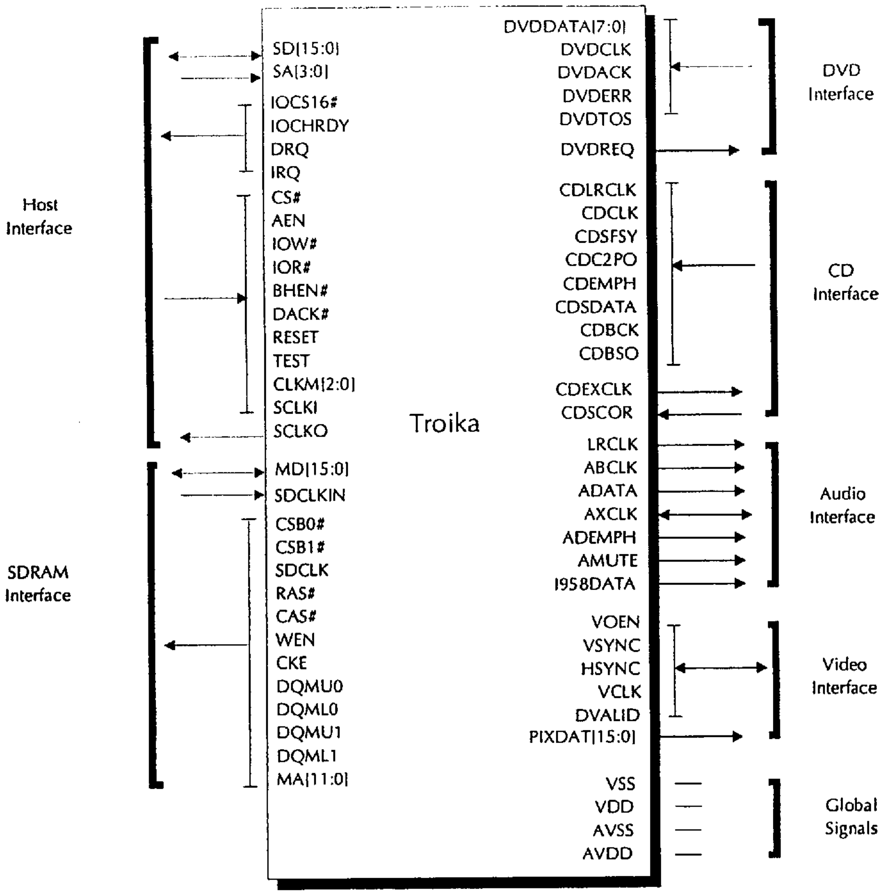


Figure 2-2: *External Interface*

2.3 PIN SUMMARY IN NUMERICAL ORDER

Pin Name	Pin Number	Pin Type	Pin Name	Pin Number	Pin Type
VDD	1	-	VDD	41	-
MD[0]	2	I/O	RAS#	42	O
VSS	3	-	VSS	43	-
MD[1]	4	I/O	CAS#	44	O
MD[2]	5	I/O	WEN	45	O
MD[3]	6	I/O	CKE	46	O
MD[4]	7	I/O	DQMU0	47	O
MD[5]	8	I/O	DQML0	48	O
MD[6]	9	I/O	DQMU1	49	O
MD[7]	10	I/O	DQML1	50	O
MD[8]	11	I/O	DVDDATA[0]	51	I
MD[9]	12	I/O	DVDDATA[1]	52	I
VSS	13	-	VSS	53	-
MD[10]	14	I/O	DVDDATA[2]	54	I
VDD	15	-	VDD	55	-
MD[11]	16	I/O	DVDDATA[3]	56	I
MD[12]	17	I/O	DVDDATA[4]	57	I
MD[13]	18	I/O	DVDDATA[5]	58	I
MD[14]	19	I/O	DVDDATA[6]	59	I
MD[15]	20	I/O	DVDDATA[7]	60	I
MA[0]	21	O	DVDCLK	61	I
MA[1]	22	O	DVDACK	62	I
MA[2]	23	O	DVDERR	63	I
MA[3]	24	O	DVDTOS	64	I
MA[4]	25	O	DVDREQ	65	O
MA[5]	26	O	CDLRCLK	66	I
VSS	27	-	CDCLK	67	I
MA[6]	28	O	VSS	68	-
VDD	29	-	CDSFSY	69	I
MA[7]	30	O	VDD	70	-
MA[8]	31	O	CDC2PO	71	I
MA[9]	32	O	CDEMPH	72	I
MA[10]	33	O	CDSDATA	73	I
MA[11]	34	O	CDBCK	74	I
SDCLKIN	35	I	CDSBSO	75	I
CSB0#	36	O	CDEXCLK	76	O
CSB1#	37	O	CDSCOR	77	I
VSS	38	-	VSS	78	-
SDCLK	39	O	RESET	79	I
VDD	40	-	VDD	80	-

Note: Pin specifications:

1. Input pins accept an input of 5V
2. Output pins output 3.3V. If your design requires a 5V output, the system designer must pull-up the voltage.

PIN SUMMARY IN NUMERICAL ORDER (Cont'd)

Pin Name	Pin Number	Pin Type	Pin Name	Pin Number	Pin Type
VDD	81	-	VDD	121	-
IOR#	82	I	TEST	122	I
VSS	83	-	VSS	123	-
IOW#	84	I	CLKM[2]	124	I
AEN	85	I	CLKM[1]	125	I
CS#	86	I	CLKM[0]	126	I
SA[3]	87	I	LRCLK	127	O
SA[2]	88	I	ABCLK	128	O
SA[1]	89	I	ADATA	129	O
SA[0]	90	I	AXCLK	130	I/O
IRQ	91	O	ADEMPH	131	O
DRQ	92	O	AMUTE	132	O
IOCHRDY	93	O	VSS	133	-
IOCS16#	94	O	I958DATA	134	O
SD[15]	95	I/O	VDD	135	-
SD[14]	96	I/O	VOEN	136	I/O
SD[13]	97	I/O	VSYNC	137	I/O
SD[12]	98	I/O	HSYNC	138	I/O
VDD	99	-	VCLK	139	I/O
SD[11]	100	I/O	DVALID	140	I/O
VSS	101	-	PIXDAT[0]	141	O
SD[10]	102	I/O	PIXDAT[1]	142	O
SD[9]	103	I/O	PIXDAT[2]	143	O
SD[8]	104	I/O	PIXDAT[3]	144	O
SD[7]	105	I/O	PIXDAT[4]	145	O
SD[6]	106	I/O	PIXDAT[5]	146	O
SD[5]	107	I/O	VSS	147	-
SD[4]	108	I/O	PIXDAT[6]	148	O
AVSS	109	-	VDD	149	-
AVDD	110	-	PIXDAT[7]	150	O
SD[3]	111	I/O	PIXDAT[8]	151	O
SD[2]	112	I/O	PIXDAT[9]	152	O
SD[1]	113	I/O	PIXDAT[10]	153	O
SD[0]	114	I/O	PIXDAT[11]	154	O
SCLKO	115	O	PIXDAT[12]	155	O
DACK#	116	I	PIXDAT[13]	156	O
BHEN#	117	I	PIXDAT[14]	157	O
VSS	118	-	VSS	158	-
SCLKI	119	I	PIXDAT[15]	159	O
VDD	120	-	VDD	160	-

Note: Pin specifications:

1. Input pins accept an input of 5V
2. Output pins output 3.3V. If your design requires a 5V output, the system designer must pull-up the voltage.

2.4 PIN SUMMARY IN REGISTER FORMAT

2.4.1 HOST BUS INTERFACE

Signal Name	Pin Number	Type	Function
SA[3:0]	87-90	I	System Address Bus
CS#	86	I	Chip Select, active LOW
AEN	85	I	Address enable, defines valid I/O address lines for DMA cycles
IOW#	84	I	I/O Write byte cycle command, active LOW
IOR#	82	I	I/O Read cycle command, active LOW
RESET	79	I	System Reset, required to be High at least 1ms before the reset function can be assured, active HIGH
TEST	122	I	Test signal. All output are Z-state when set HIGH.
CLKM[2:0]	124-126	I	Clock mode select
SCLKI	119	I	System clock input: 13.5, 27.0, or 67.5 MHz
BHEN#	117	I	Byte High Enable. When active, it indicates valid data on SD[15:8]. When inactive, it indicates 8-bit data width. 16-bit mode is active LOW.
DACK#	116	I	DMA Acknowledge. Used by the DMA controller to select an I/O resource that requested the DRQ, active LOW.
SCLKO	115	O	PLL Clock Output. For test purposes only.
SD[15:0]	95-98, 100, 102-108, 111-114	I/O	System Data Bus. The chip provides both 8-bit and 16-bit data modes, and can be programmed through the host bus unit register.
IOCS16#	94	O	I/O Chip Select. The Troika can be programmed to receive 8-bit or 16-bit control/data transfer, active LOW.
IOCHRDY	93	O	I/O Channel ready signal. This signal indicates to the bus owner that additional bus cycle time is required, active HIGH.
DRQ	92	O	DMA Request. Active when the Troika requests a DMA transfer, active HIGH
IRQ	91	O	Interrupt Request, active HIGH

2.4.2 DVD INTERFACE

Signal Name	Pin Number	Type	Function
DVDDATA[7:0]	51, 52, 54, 56-60	I	DVD MPEG-2 Program Data
DVDCLK	61	I	DVD DSP Clock (27 MHz)
DVDACK	62	I	Acknowledge data from DVD-DSP
DVDERR	63	I	Sector Error Indicator
DVDTOS	64	I	Top of Sector Indicator
DVDREQ	65	O	Data Request to DVD DSP

2.4.3 CD INTERFACE

Signal Name	Pin Number	Type	Function
CDLRCK	66	I	Left/Right Clock for CD
CDCLK	67	I	CD DSP Clock
CDSFSY	69	I	Write Frame Clock
CDC2PO	71	I	C2 Pointer Erasure Flag
CDEMPH	72	I	CD-DA Emphasis Flag
CDSDATA	73	I	CD Serial Data
CDBCK	74	O	CD Data Clock
CDSBSO	75	I	Subcode information for CD-G
CDEXCLK	76	I	Subcode clock for CD-G
CDSCOR	77	I	S0/S1 Subcode Blocksync

2.4.4 SDRAM INTERFACE (SDIF)

Signal Name	Pin Number	Type	Function
MD[15:0]	2, 4-12, 14, 16-20	I/O	SDRAM Data bus, preset Pull-up pads (MD[3:0]) used for chip initialization to configure selected interface
SDCLKIN	35	I	Fed back SDCLK clock
CSB0#	36	O	Bank select for first bank of 16-Mbit SDRAM, active LOW
CSB1#	37	O	Bank select for second bank of 4-/16-Mbit SDRAM, active LOW
SDCLK	39	O	SDRAM Master Clock
RAS#	42	O	Row Address Strobe, active LOW
CAS#	44	O	Column Address Strobe, active LOW
WEN	45	O	Write enable, active LOW
CKE	46	O	Clock enable, active HIGH
DQMU0	47	O	Upper data byte buffer control for first bank of 16-Mbit SDRAM, active LOW
DQML0	48	O	Lower data byte buffer control for first bank of 16-Mbit SDRAM, active LOW
DQMU1	49	O	Upper data byte buffer control for second bank of 4-/16-Mbit SDRAM, active LOW
DQML1	50	O	Lower data byte buffer control for second bank of 4-/16-Mbit SDRAM, active LOW
MA[11]	34	O	First bank 16-Mbit SDRAM internal bank select
MA[10]	33	O	First bank 16-Mbit SDRAM row address
MA[9]	32	O	First bank 16-Mbit SDRAM row address and Second bank 4-Mbit/16-Mbit SDRAM internal bank select
MA[8]	31	O	Second bank 4-/16-Mbit SDRAM row address
MA[7:0]	21-26, 28, 30	O	First and second bank SDRAM row/column address

2.4.5 AUDIO INTERFACE

Signal Name	Pin Number	Type	Function
LRCLK	127	O	Audio Left/Right channel clock
ABCLK	128	O	Audio data bit clock
ADATA	129	O	Audio serial data output
AXCLK	130	I/O	Audio clock (12.2880 MHz, 16.9344 MHz, 18.4320 MHz, 36.864 MHz)
ADEMPH	131	O	Audio DAC Emphasis Flag
AMUTE	132	O	Audio mute pin

2.4.6 VIDEO INTERFACE

Signal Name	Pin Number	Type	Function
VOEN	136	I/O	Video data output enable, active LOW
VSYNC	137	I/O	Vertical Synchronization
HSYNC	138	I/O	Horizontal Synchronization
VCLK	139	I/O	Video clock (13.5 or 27 MHz) from NTSC/PAL encoder, transport decoder, or internal
DVALID	140	O	Video data valid, active HIGH
PIXDAT[15:0]	141-146, 148, 150-157, 159	O	Video data bus 13.5 MHz mode: PIXDAT[15:8] = Y data PIXDAT[7:0] = CrCb data 27.0 MHz mode: PIXDAT[15:8] = OSD overlaid YCrCb PIXDAT[7:0] = YCrCb data

CHAPTER 3 FUNCTIONAL DESCRIPTIONS

3.1 DISK INTERFACE

The Troika supports the following modes of operation (as defined in Reg. x08b):

Disk Interface:

DIU_MOD[3:0]	Disk Interface Unit Function
0000	MPEG-1 System (Video CD)
0001	MPEG-1 Audio only
0010	MPEG-1 Video only
0011	CD-DA
0100	CD-ROM (mode 0)
0101	CD-ROM (mode 1)
0110	CD-ROM (mode 2 form 1)
0111	CD-ROM (mode 2 form 2)
1000	MPEG-2 Program (DVD)
1011	CD-DA with CD-G
Others	Reserved

3.2 HOST INTERFACE

3.2.1 OVERVIEW

The host bus interface unit (HBIU) has four primary functions:

- Troika chip housekeeping
- Troika chip control
- Communication ports for DVD and CD interfaces
- Highlight information presentation information from the external microcontroller.

For system control and data flow operations, the HBIU directly interfaces with an 8-/16-bit microcontroller bus or a 16-bit ISA bus. The external microcontroller performs chip initialization through the HBIU. Once initialized, however, the HBIU reports internal operational status and generates system interrupts. The Troika's internal registers, counters, memory, and SDRAM are accessible through the HBIU. Compressed MPEG-2 and MPEG-1 data streams are loaded into the Host FIFO module via an interrupt request (IRQ) or direct memory access (DMA) mode. Interfaces are through two basic methods: 1) through the host bus interface unit (HBIU) via a microcontroller or ISA bus or 2) through a DSP interface (CD-DSP or DVD-DSP). In the DVD mode, data search (DSI), presentation control (PCI), and highlight (HLI) information for sub-picture decoding is passed through the HBIU to the sub-picture processor within the Troika. The DSI and PCI information is buffered in the Troika for use by the external system.

There are several status registers that provide current information for the internal FIFOs. Depending upon whether the FIFOs are currently full or empty, the signals are logically "AND-ed" with internal IRQ and DRQ enabling registers to output an active IRQ or DRQ signal on the host bus.

The Troika host interface supports three interface protocols:

- 8-/16-bit Intel 803x/805x/808x microcontroller interface
- 8-/16-bit Motorola 680x microcontroller interface
- 16-bit ISA bus

3.2.2 HOST INTERFACE CONFIGURATION

The host interface is mainly used to provide communication and housekeeping access between the DVD system and the Troika chip. The other dedicated presentation ports are used by the Troika to receive presentation data or to provide data output from the Troika to the DVD system.

The host interface provides the following functionality:

- Read/write the internal registers for the Troika
- Read/write the external SDRAM for the Troika
- Load the microcode and initialization files for the Troika
- Input coded DVD, VideoCD, MPEG-2, MPEG-1, AC-3, LPCM, CD-I bit streams
- Input highlight commands for sub-picture processor

Using the MD[2:0] data pins (pins 2, 4, and 5 of Troika) on the SDRAM data bus during initialization, the host bus configuration can be set to any one of five different available states, as listed in the table on page 3-1.

Note: During initialization, MD[5:3] (pins 6-8) is also used. MD[5:4] defines whether an optional second bank of SDRAM is used, and MD[3] defines the SDRAM speed. For further details, please refer to the register definitions.

Host Bust Configuration:

MD[2:0]	Host Processor
000	16-bit ISA
001	Reserved
010	Reserved
011	Reserved
100	8-bit Motorola microprocessor
101	16-bit Motorola microprocessor
110	8-bit Intel microprocessor
111	16-bit Intel microprocessor

During Troika initialization, MD[2:0] and the SDRAM data bus (MD[15:0]) are configured using preset pull-down and pull-up resistors connected to the MD[2:0] data pins. The Troika latches this information during hardware or software reset cycles and internally self-configures to support the selected interface. Logical LOW values require a pull-down resistor, and logical HIGH values require a pull-up resistor. There is no default selection for the host processor since a "no connection" to MD[2:0] is not defined. **When an external resistor is used, the recommended pull-down and pull-up resistor value is 4.7kΩ.**

Motorola 8-/16-bit Microcontroller Interface

A sample connection of the Troika's 8-bit microcontroller interface to a Motorola style family is shown in Figure 3-1.

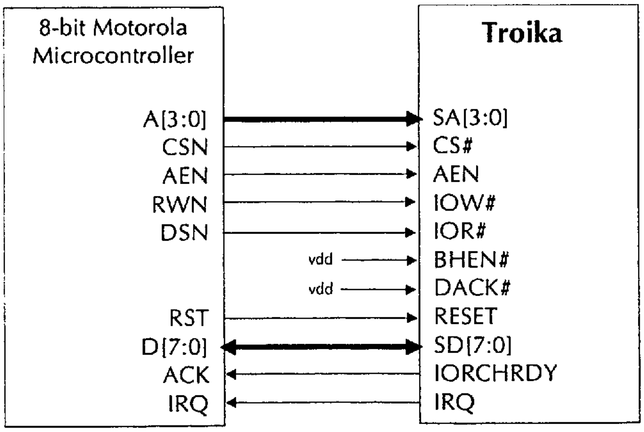


Figure 3-1: *Troika 8-bit Motorola Microcontroller Connection*

A sample connection of the Troika's 16-bit microcontroller interface to a Motorola style family is shown in Figure 3-2.

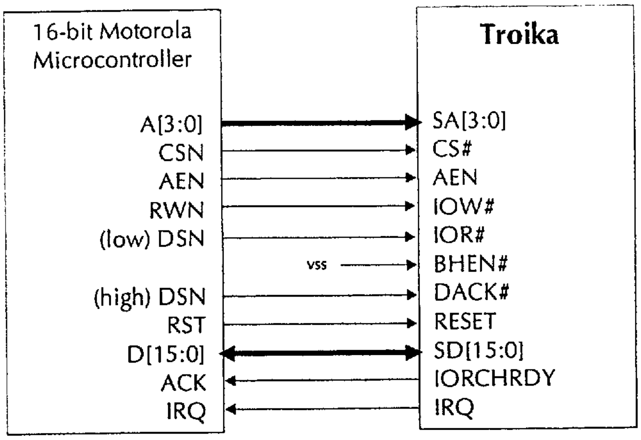


Figure 3-2: *Troika 16-bit Motorola Microcontroller Connection*

As shown in Figure 3-3, the Motorola microcontroller asserts the SA[3:0] address line to indicate the requested Troika host registers. The DSN (IOR#) and R/W* (IOW#) control signals provide data read and write commands to the Troika. The Troika asserts the DACK# (IOCHRDY) signal as the data acknowledge signal to the host processor.

During the Host Read Operation, the host processor asserts the destination address onto the SA[3:0] pins, and asserts the R/W* (IOW#) signal HIGH to indicate a read operation. Note that R/W* may already be set HIGH and no assertion will then be necessary. Then the host processor asserts the data strobe signal (DSN) IOR#. Once the Troika detects the DSN signal, the Troika monitors the R/W* control line and asserts the DACK# (IOCHRDY) signal to the host processor once the data is ready to transfer on the SD[7:0] bus. The host processor should latch in the data and de-assert the DSN signal. After DSN is inactive, the Troika will de-assert the DACK# signal to complete the process.

During the Host Write Operation, the host processor asserts the SA[3:0] address line, asserts the R/W* control signal LOW to indicate a Write operation, and asserts the DSN strobe signal. The Troika will respond with the DACK# signal when it is ready to latch in the data, and then the host processor will de-assert the DSN signal. At the DSN's rising edge, the Troika will latch in the data and then de-assert the DACK# signal to complete the write operation.

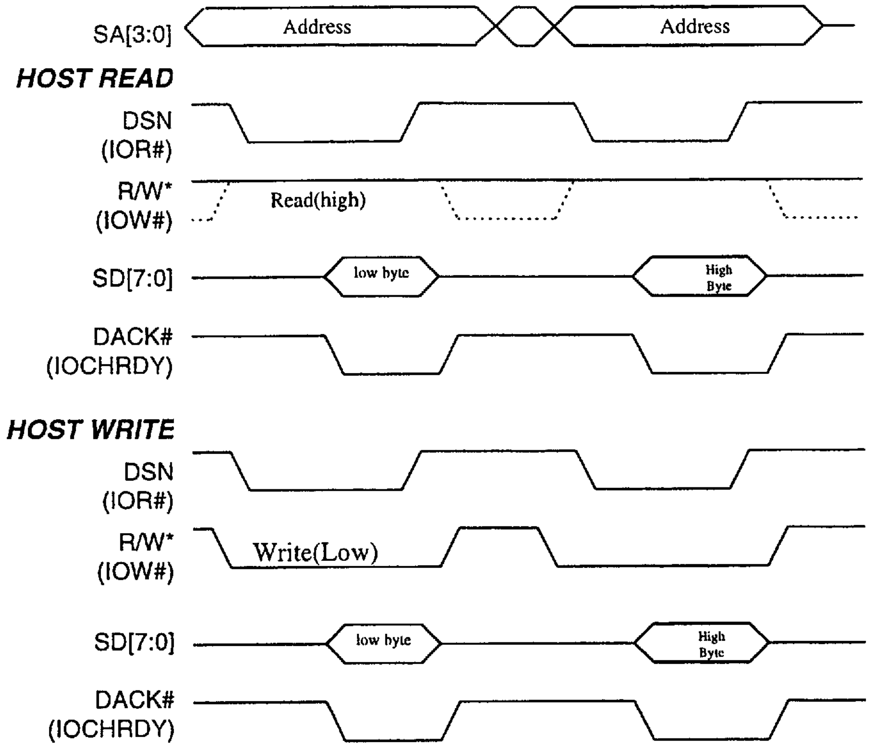


Figure 3-3: *Motorola 8-bit Microcontroller Timing Diagram*

Intel 8-/16-bit Microcontroller Interface

A sample connection of the Troika's 8-bit microcontroller interface to an Intel style family is shown in Figure 3-4.

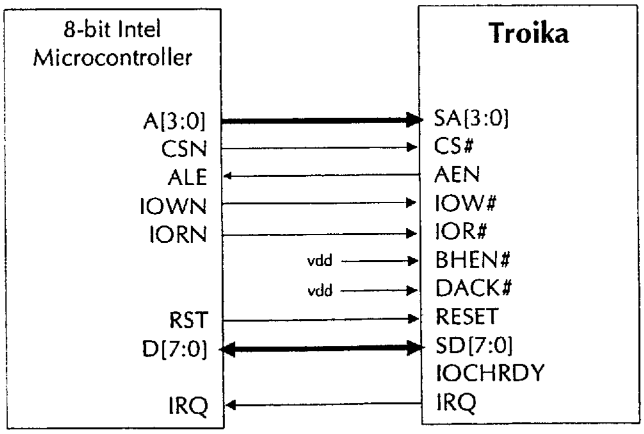


Figure 3-4: *Troika 8-bit Intel Microcontroller Connection*

A sample connection of the Troika's 16-bit microcontroller interface to an Intel style family is shown in Figure 3-5.

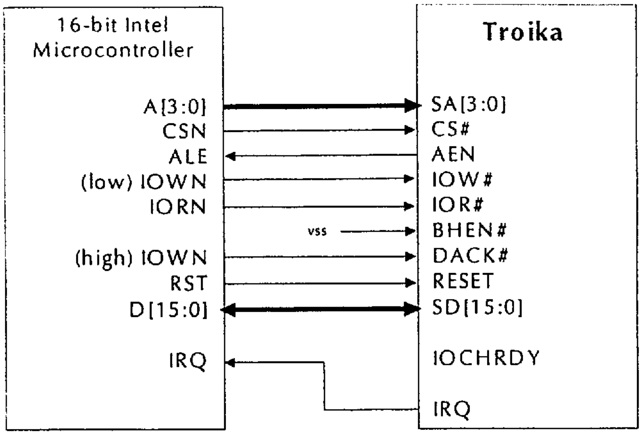


Figure 3-5: *Troika 16-bit Intel Microcontroller Connection*

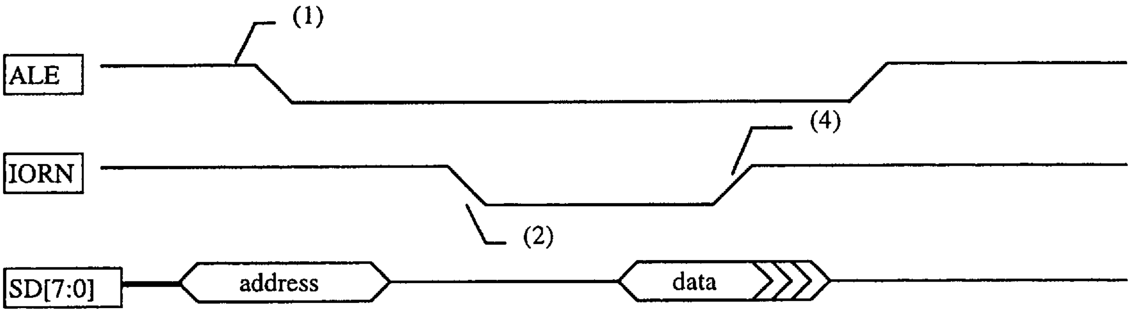


Figure 3-6: *Timing Diagram for 8-bit Intel Read Operation*

The sequence of events for Figure 3-6 is as follows:

1) The Host asserts ALE. In response to the assertion of ALE, the Troika latches the address on SD[7:0] at the falling edge of ALE.

2) The Host drives IORN LOW to indicate a read operation.

3) In response to the assertion of IORN, the Troika begins a "read" operation. The Troika holds "Data Ack" HIGH until the data is available. Note that the Host processor is not actually tied to the Data Ack signal. When data is ready, the Troika asserts Data Ack and also drives the data onto SD[7:0].

4) After a fixed time period, the Host de-asserts IORN.

5) In response to the de-assertion of IORN, the Troika releases Data Ack and SD[7:0].

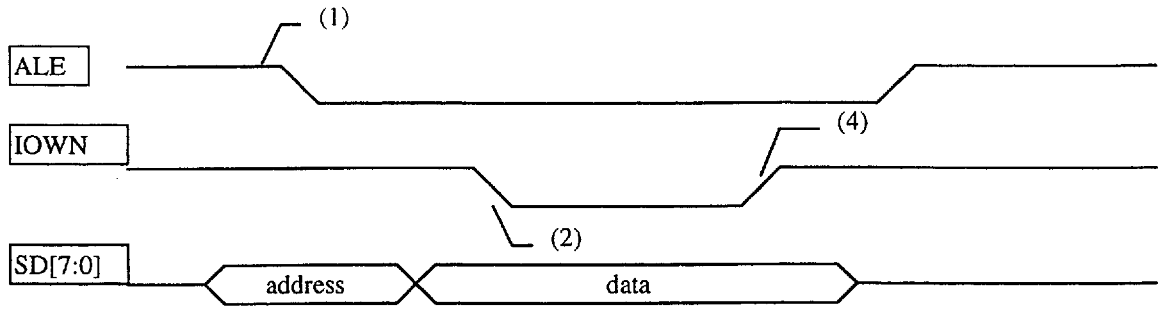


Figure 3-7: *Timing Diagram for 16-bit Intel Write Operation*

The sequence of events for Figure 3-7 is as follows:

1) The Host asserts ALE. In response to the assertion of ALE, the Troika latches the address on SD[7:0] at the falling edge of ALE.

2) The Host drives IOWN LOW to indicated a write operation.

3) In response to the assertion of IOWN, the Troika begins latching data and sometime later generates a "Data Ack," although the Host is not actually tied to the Data Ack signal. (The Troika latches the data when Data Ack goes low.)

4) After a fixed time period, the Host de-asserts IOWN.

5) In response to the de-assertion of IOWN, the Troika releases Data Ack and data is no longer written to the Troika.

ISA Bus Interface

The Troika ISA bus interface provides a direct connection to the industry-standard ISA bus. The ISA bus interface supports a 16-bit I/O operation. Both interrupt and DMA transactions are supported. This section will describe in detail the following issues associated with the ISA bus interface:

- Troika ISA I/O address allocation
- Troika ISA direct memory access (DMA) operation

A sample connection of the Troika's 16-bit ISA interface is shown in Figure 3-8.

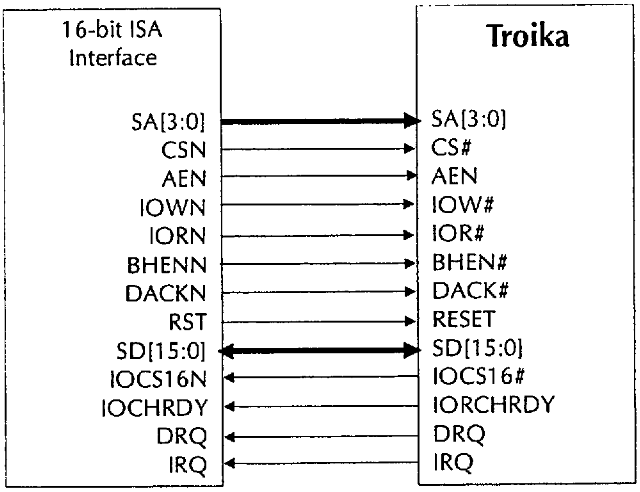


Figure 3-8: *Troika 16-bit ISA Connection*

Troika ISA I/O Allocation

The Troika's ISA I/O allocation is controlled by the chip select (CS1#) pin. If the CS1# signal is active, the Troika will claim this I/O operation cycle, and respond accordingly.

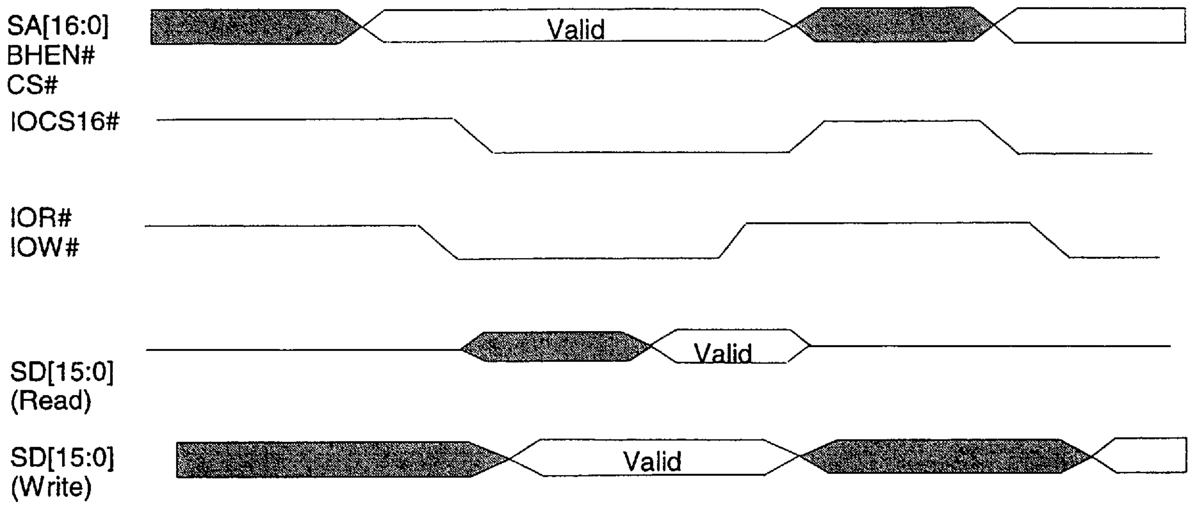


Figure 3-9: *ISA I/O Access Timing Diagram*

Troika ISA PIO (Programmable I/O) Operation

The Troika ISA PIO (programmable I/O) operation is based on the standard ISA bus protocol, and supports 16-bit PIO operation. Programming the Data Mode of HBIU Control Register 1 (Reg. x1c5 bit[1]) to HIGH enables 16-bit operation. Note that the following signals are only valid for the Troika at 16-bit operations:

- SD[15:8]
- BHEN#
- IOCS16#
- DRQ
- DACK#

Figures 3-10 and 3-11 show the PIO transaction.

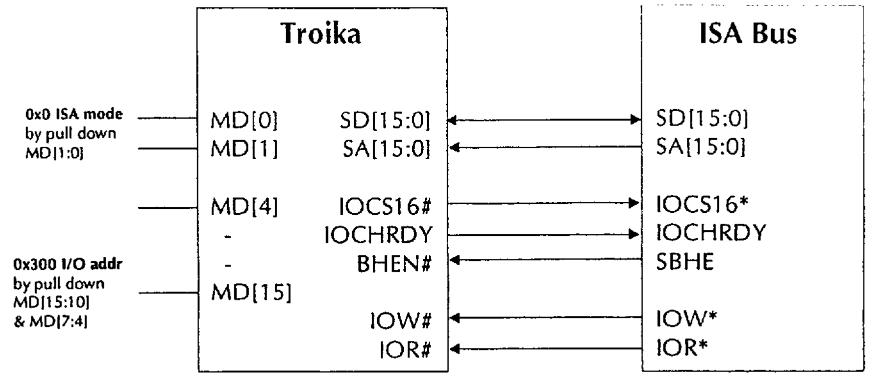


Figure 3-10: *Troika ISA Interface Block Diagram*

Note: Recommended pull-down resistor value: 4.7k ohm

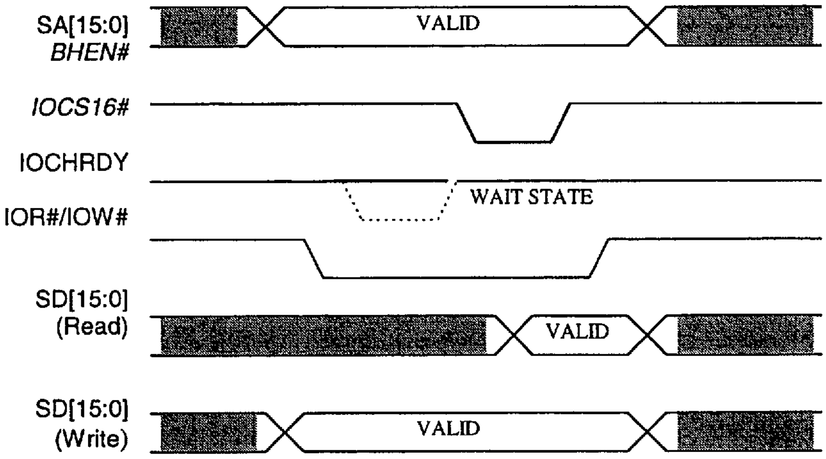


Figure 3-11: *Troika ISA PIO Timing Diagram*

Note: BHEN#, IOCS16#, and SD[15:8] are only valid with 16-bit operation

In the ISA I/O transaction cycle, the Troika monitors the SA[3:0] address line and the IOR#/IOW# signals. When the SA address matches the pre-set address, and IOR#/IOW# has been asserted, the Troika asserts IOCS16# and sends out data to the SD bus. The Host sends the data in the IOW# cycle and latches-in data from the SD bus at the IOR# cycle's rising edge.

The Troika will assert the IOCHRDY signal as a wait state to request more I/O cycles to complete the current I/O operation. In situations like reading or writing to the internal registers or sending data to the HFIFO (host FIFO) register, the Troika will need more I/O cycles to transfer data internally without interfering with the decoding process. It is recommended that during MPEG decoding, only host bus direct access registers be accessible, any access to internal registers or SDRAM contents should be held to a minimum. Otherwise, the decoding performance will be adversely affected.

Troika ISA DMA Operation

The Troika supports 16-bit standard DMA slave mode operations. Programming the Data Mode of HBIU Control Register 1 (Reg. x1c5 bit[1]) to HIGH enables 16-bit operation. Only write operations to the HFIFO are supported. All other access to internal registers or SDRAM content must use the PIO operation. The connection block diagram and timing diagram are shown in Figures 3-12 and 3-13.

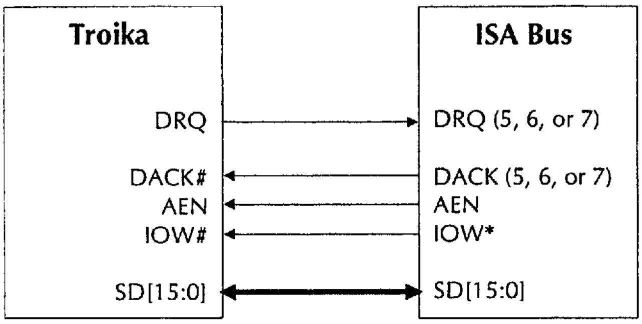


Figure 3-12: *Troika DMA Connection Diagram*

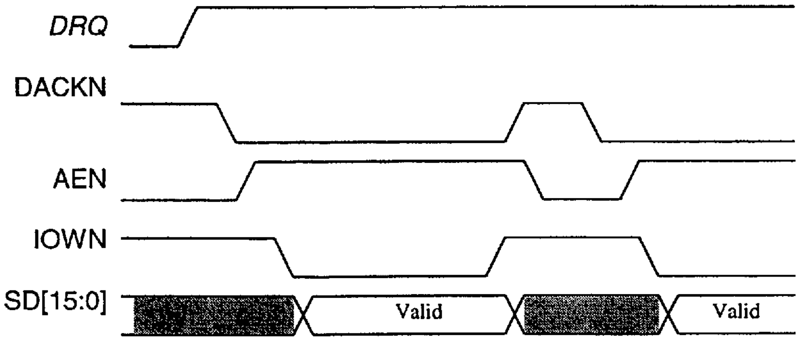


Figure 3-13: *Troika DMA Timing Diagram*

The Troika is programmed into the DMA mode, and the number of 16-bit words to be transferred should be set to the DMA Word Count Register (Reg. x1c13). Once the DMA transfer commences, the Troika will keep DRQ high and keep requesting more data from the host. This process only happens as the Troika's DMA word count is not zero and the HFIFO is not full. The host processor should assert AEN and IOW# with the SD[15:0] data line. The Troika will latch in data after IOW#'s rising edge. When the DMA word count register is zero, the Troika will de-assert the DRQ signal and assert the IRQ interrupt signal to inform the host processor that the DMA operation is completed.

3.3 PLL AND CLOCK INTERFACE

The Troika is a highly integrated MPEG-2 and MPEG-1 video/audio decoder that uses a clocking scheme based on a direct interface to a DVD-DSP, CD-DSP, or external clock synthesizer. For interface to a DVD-DSP or CD-DSP, a high-precision internal PLL circuit generates the clock for the video pixel and reference clocks. For interface to an external clock synthesizer, the video and reference clocks are generated via a direct connection to the external clock. Figures 3-14 and 3-15 illustrate the Troika clocking interface for both of these clocking schemes.

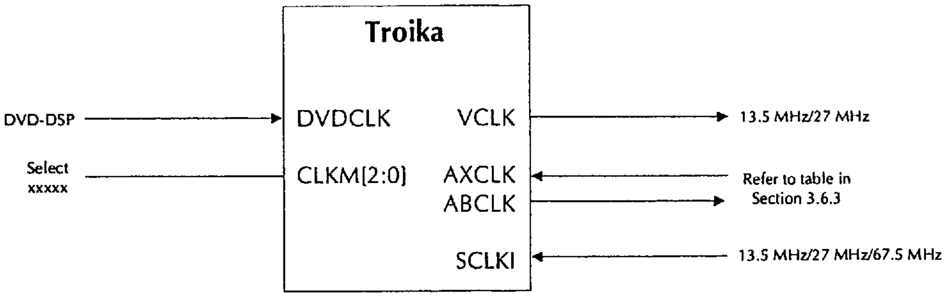


Figure 3-14: *Troika Clock Scheme Using Internal PLL*

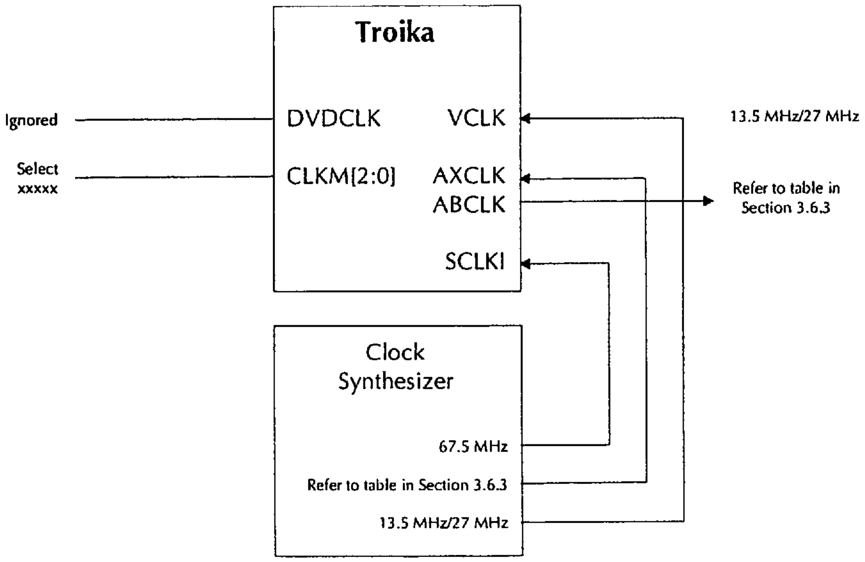


Figure 3-15: *Troika with an External Clock Synthesizer*

For the clocking scheme using the internal PLL, the Troika provides a direct interface to DVD-DSP and CD-DSP chips, and accommodates several clock sources such as system clock, video pixel clock, audio clock, DVD-DSP clock, and CD-DSP clock. In order to simplify the clocking scheme, the Troika inputs the clock source from either the DVD-DSP (DVDCLK), the video clock (VCLK), or the system clock (SCLKI), and uses the internal PLL to generate the internal system clock, the video pixel clock, and the internal 90-KHz MPEG reference clock. The Troika can use the video master mode to generate a video pixel clock from the system clock as long as the original clock source meets all accuracy requirements for NTSC/PAL encoding.

The Troika supports a DVD-DSP clock and several CD-DSP clocks. A single DVD-DSP clock is supported at 27 MHz for a direct interface to the DSP, and several CD-DSP clock rates are supported via programmable software so that many of the popular CD-DSP chips are compatible.

For NTSC/PAL video output (master mode) an external clock synthesizer is used, and the PLL can be used for internal system clock generation (for CLKM[2:0] set to 110 or 111). The video clock used by the encoder must be very accurate for proper NTSC/PAL video output, so an external clock synthesizer is used, which provides a video clock of 13.5 MHz or 27 MHz, accurate to within 50 ppm.

Troika Clocking Structure:

CLKM[2:0] (pins 124-126)	**PLL Source**	**PLL Output** SCLKO (pin 115)	**System Clock Source** (internal system clock)	**Video Clock Source** (internal video clock source)	**Master/Slave** VCLK (pin 139)
000	Not Used	Not Used	SCLKI (67.5 MHz)	VCLK (13.5 MHz)	Slave
001	Not Used	Not Used	SCLKI (67.5 MHz)	VCLK (27.0 MHz)	Slave
010	VCLK (13.5 MHz)	67.5 MHz	PLL (67.5 MHz)	VCLK (13.5 MHz)	Slave
011	VCLK (27.0 MHz)	67.5 MHz	PLL (67.5 MHz)	VCLK (27.0 MHz)	Slave
100	DVDCLK (27.0 MHz)	67.5 MHz	PLL (67.5 MHz)	VCLK (13.5 MHz)	Slave
101	DVDCLK (27.0 MHz)	67.5 MHz	PLL (67.5 MHz)	VCLK (27.0 MHz)	Slave
110	SCLKI (13.5 MHz)	67.5 MHz	PLL (67.5 MHz)	SCLKI (13.5 MHz)	Master
111	SCLKI (27.0 MHz)	67.5 MHz	PLL (67.5 MHz)	SCLKI (27.0 MHz)**	Master

Note: **In master mode, VCLK may also be selected as 13.5-MHz output by setting the appropriate global bus registers in the VBIU

3.4 SDRAM INTERFACE

This section describes the Troika's SDRAM interface bus, as shown in Figure 3-16. The following sections explain, in detail, the supported SDRAM configuration types and various PCB layout recommendations.

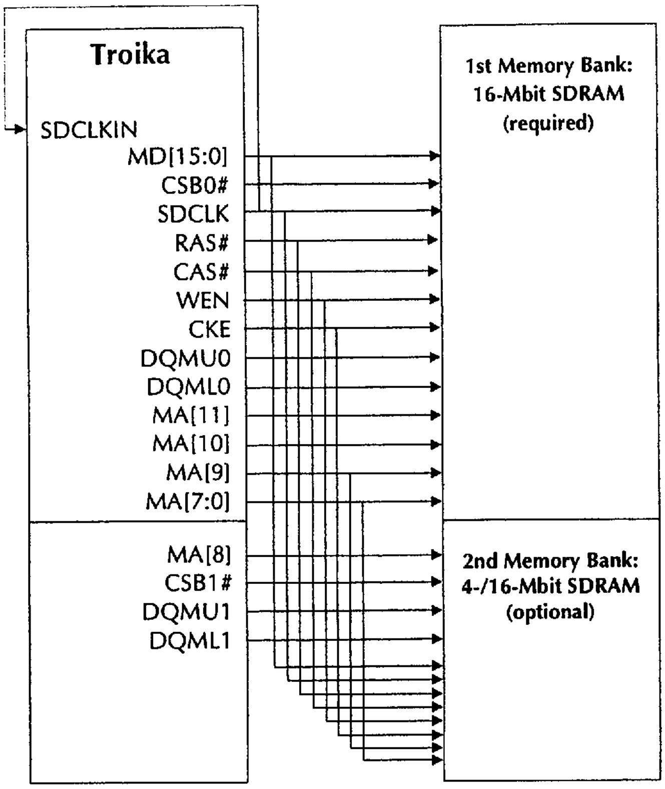


Figure 3-16: *Troika SDRAM Interface*

The SDRAM controller provides the following functionality as a basic mechanism:

- ◆ Address generation and control for the external 16-Mbit SDRAM and optional 4-Mbit or 16-Mbit additional SDRAM.
- ◆ Provides internal control of all hardware blocks, provides an interrupt scheme, and schedules the movement of data in and out of the Troika.

The minimum SDRAM memory size requirement for the Troika is 16 Mbits, but the SDRAM controller permits an additional 4 Mbits or 16 Mbits of SDRAM for extended OSD graphics and color capabilities, as well as allowing an oversized video buffer verifier (VBV) buffer.

In order to compensate for pad and wire delays and to improve timing accuracy during data transfer from the SDRAM to the Troika, the SDCLK signal is fed back into the Troika as SDCLKIN.

3.4.1 SDRAM CONTROLLER

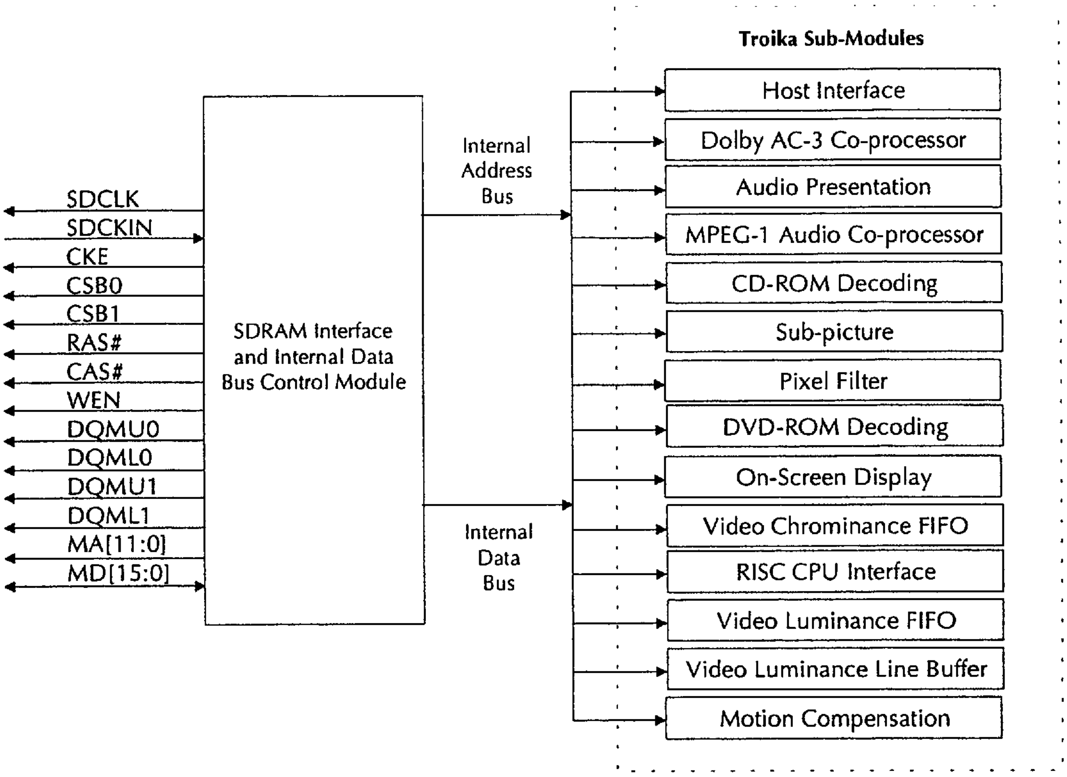


Figure 3-17: *Troika Internal Interface to SDRAM Controller*

3.4.2 SDRAM CONTROLLER OVERVIEW

The Troika requires a minimum of 135 MB/s SDRAM bandwidth. The main SDRAM controller frequency is specified at 67.5 MHz, which provides the minimum frequency that meets bandwidth requirements.

The Troika SDRAM controller implements several features that enhance the flexibility, performance, and power consumption of the player system. The Troika provides a memory interface clock input (SDCLKIN) that allows flexibility and control in system-level design to meet critical timing parameters such as tAC (clock to data access time). The Troika also supports 2- and 3-clock CAS latency with 3-clock CAS as the default. Provided as an option to the system designer, 2-clock CAS latency may be used either to increase performance or reduce performance. Performance can increase by running an SDRAM speed grade of 80 MHz or 100 MHz because the higher speed grade allows the user to use 2-CAS latency instead of 3-CAS latency used with an SDRAM speed grade of 70 MHz. To help reduce the system power consumption, the Troika supports the standard SDRAM power-down sequence.

The Troika requires 16 Mbits of SDRAM to operate properly. However, an optional second bank of 4-Mbit or 16-Mbit SDRAM may be added to increase OSD color depth or increase VBV/ABV buffer size.

Major Address Generation Blocks Within the SDRAM Controller

Pixel Filter (PF) Address Generation

Calculates the addresses of the video reconstruction reference data based on motion vectors derived from the video bit stream and current macroblock addresses. Luminance reference data is always 17x9 (WxH) and chrominance reference data is always 9x9 (WxH), which may be in field or frame format depending on picture type and motion type as specified by the bit stream. I and P frame buffers are stored in SDRAM in field format, and provide forward and backward references. This module also generates the reference buffer address within the pixel filter module.

Motion Compensation (MC) Address Generation

This module generates the macroblock storage address for the current reconstruction macroblock. Luminance block size is 16x8 (WxH) and chrominance block size is 8x8 (WxH), which may be in frame or field format depending on the picture type and picture structure. I and P frame macroblocks are SDRAM bank interleaved to improve data transfer and reduce SDRAM access latencies.

Video BufferVerifier (VBV) Address Generators

This module is responsible for generating SDRAM addresses for the input bit stream circular buffer. The input to this buffer is from C-FIFO (input code FIFO) after the system layer parsing. The output from this buffer goes to DFIFO, which supplies data for picture sequence layer decoding.

Video FIFO (VFIFO) Address Generator

This module provides the address generation for video display outputs, which is supported only in an interlaced (non-progressive) mode. Additionally, pan and scan horizontal filtering, letterbox vertical filtering, and video format conversion vertical filtering are performed by this module.

External Memory Content:

External Memory (1M x 16 bits SDRAM)
User Data & Closed Caption
I (P) Video Frame
P (I) Video Frame
B Video Frame
Video Buffer Verifier (VBV)
Sub-picture Buffer Verifier (SPBV)
Sub-picture
DSI Buffer Verifier (DSIBV)
On-Screen Display (OSD)
Audio Frame 0
Audio Frame 1
Audio Downmix
Audio Delay
Audio Buffer Verifier (ABV)

The SDRAM controller consists of a sophisticated memory access request mechanism to ensure sufficient SDRAM bandwidth. While the SDRAM is busy, new memory access requests are stored in queues. The priority of the next memory access is based upon the request types, which are prioritized according to the programmer.

3.4.3 SDRAM REQUIREMENTS AND CONFIGURATIONS

The Troika's SDRAM interface can support up to 4MB (32 Mbits) of SDRAM data address. Only 2MB (16 Mbits) of SDRAM are required to decode DVD bit streams for NTSC or PAL.

The Troika can support up to two banks of SDRAM addresses. The first bank is required and should be composed of one 1-Mbit x 16 SDRAM. The second bank offers two options, either a 256-Kbit x 16 or 1-Mbit x 16 SDRAM. The preference depends on how much the system designer wants to increase OSD color depth or to increase VBV/ABV buffer size.

3.4.4 SUPPORTED SDRAM TYPE

The Troika supports 1-Mbit x16 or 256-Kbit x16 SDRAM chips in the lowest speed grade available, 70 MHz. Both the refresh rate and the necessary SDRAM refresh services are programmable via the internal Refresh Cycle register (Reg. x143).

The Troika has a programmable CAS latency, which allows the Troika to interface with many SDRAM vendor parts, including:

- Fujitsu MB811161621A
- Hyundai HY57V16161TC
- Hitachi HM5216165
- IBM IBM0316169C
- NEC uPD4516161
- Samsung KM416S1120AT-6G
- Siemens 39SI6I60
- Toshiba TC59S1616A
- TI TMS626162

3.4.5 SDRAM MEMORY BUS TIMING DIAGRAM

The Troika is capable of using the lowest and most inexpensive speed grade of SDRAM, 70 MHz. The number of refresh services and cycles is programmable by the internal Refresh Cycle Register (Reg. x143). For detailed timing information, please refer to the SDRAM manufacturer's specification.

3.4.6 SDRAM DESIGN GUIDELINE

The following design guideline is recommended when designing the Troika SDRAM:

- The SDRAM should be placed as close as possible to the Troika. Do not place any signal lines under the SDRAM traces, especially the SDRAM data bus line signals, SDRAM clock, and control signals.
- As displayed in the following figure, the SDCLK that is fed back to the Troika should be connected as closely to SDRAM as possible because the Troika chip design has taken propagation delays into account for SDRAM data output to the Troika. A connection too close to the Troika (dotted line in Figure 3-18) may cause timing issues since the data may arrive faster to the Troika than expected.

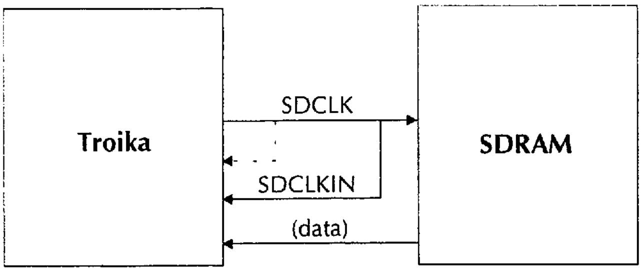


Figure 3-18: *SDRAM Design Guideline*

3.5 VIDEO INTERFACE

The video interface incorporates two types of video output: video slave mode and video master mode. In the most common usage, the Troika will output the uncompressed video pixel data, which is stored as a video image in SDRAM, to a NTSC/PAL video encoder for video output. The Troika's digital video data output format is the industry-standard 16-bit YUV 4:2:2 format which is widely used in NTSC/PAL video applications. For RGB applications, such as displaying graphics in the personal computer environment, an additional video overlay chip or a simple YUV/RGB DAC is necessary to encode MPEG video into an RGB format.

3.5.1 POSSIBLE CONFIGURATIONS OF VIDEO OUTPUT MODES

Video Slave Mode

Video slave mode (see Figure 3-19) is typically used in NTSC/PAL video applications.

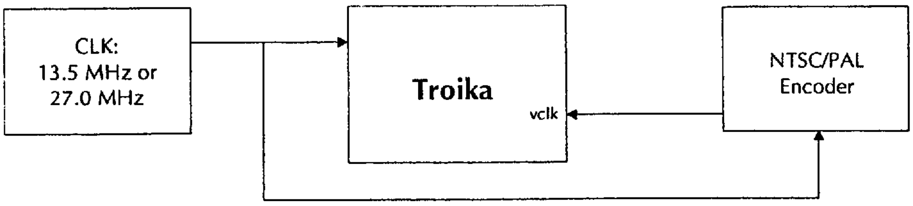


Figure 3-19: *Video Slave Mode Using PLL*

The Troika internal system clock is generated using the input clock from the external source, and passed through the Troika's PLL to generate a system clock of 67.5 MHz.

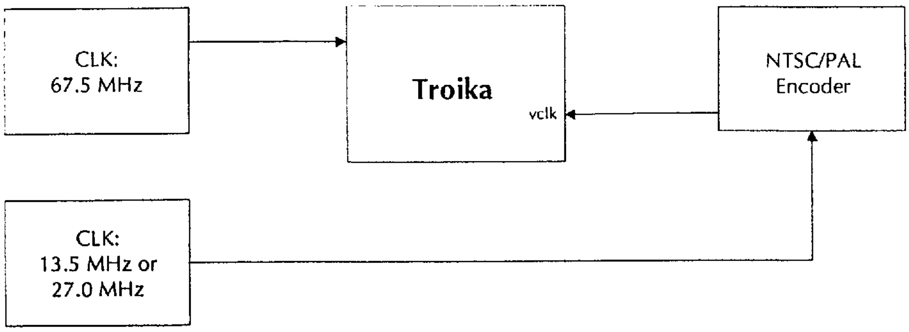


Figure 3-20: *Video Slave Mode Using External Clock*

Since the external clock source to the Troika is already 67.5 MHz (see Figure 3-20), the internal PLL does not need to generate another signal for the system clock.

Video Master Mode

Video master mode (see Figure 3-21) is typically used in computer graphic overlay applications (as in a PC environment), but may be used in NTSC/PAL video applications with certain restrictions.

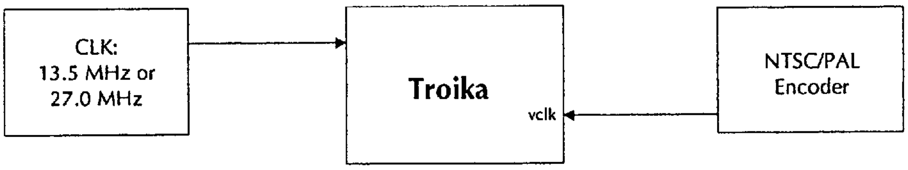


Figure 3-21: *Video Master Mode*

The Troika internal system clock is generated using the input clock from the external source and passed through the Troika's PLL to generate a system clock of 67.5 MHz.

Master and slave video modes are explained in more detail in later sections.

The Troika supports the following playback format output: 704/720 x 480 field mode at 60 fps (fields per second) for NTSC support or 704/720 x 576 field mode at 50 fps for PAL support. The Troika provides high-quality video by interpolating both horizontal and vertical lines by using an internal 7-tap horizontal filter and 2-tap vertical filter. The Troika also supports letterbox, pan & scan, upscaling of MPEG-1 images, and cross-conversion between NTSC and PAL. For the related register settings and programming examples, please refer to the Troika register definition document and application notes document.

Summary of video bus interface unit (VBIU) modules:

1. Host Command Interface
2. Video Data Buffer
3. Vertical and Horizontal Synchronization Control
4. Horizontal Filtering Control Functions for pan and scan, which provides up to 7-tap horizontal filtering function
5. Vertical Filter for letterbox and format conversion

3.5.2 SLAVE MODE VIDEO INTERFACE

Figures 3-22 and 3-23 show the Troika video interface in a standard video application connected to an NTSC/PAL video encoder.

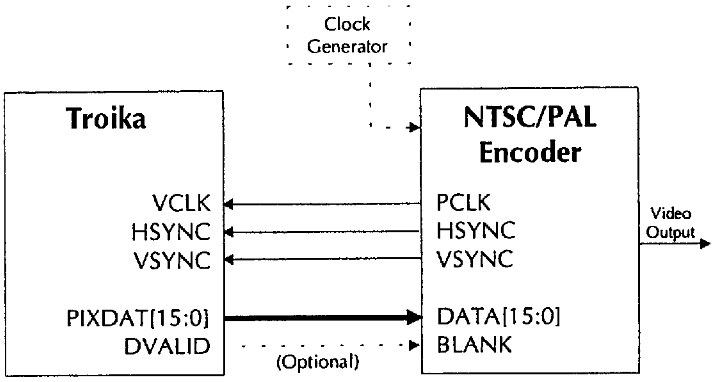


Figure 3-22: *Troika Video Interface (Slave Mode)*

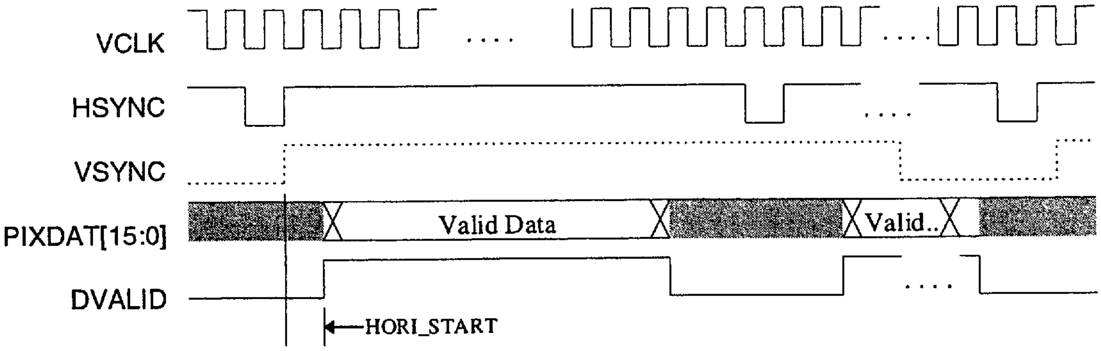


Figure 3-23: *Troika Video Interface Timing*

In the Troika slave mode interface, all the video control signals —VCLK, HSYNC, and VSYNC — originate from the video encoder chip, and the VSYNC and HSYNC signals must synchronize with VCLK, the video pixel clock. The polarity of VSYNC and HSYNC may be active HIGH or active LOW depending upon the programming of the Video Control register (Reg. x180 bits[6:5]). The Troika will output valid video data at the programmable Vertical Display Start register (VERT_START - Reg. x183) and Horizontal Display Start register (HORI_START - Reg. x184) pixel location and continue until the end of the valid video window that is programmed by the Vertical Display End register (VERT_END - Reg. x182) and Horizontal Display End register (HORI_END - Reg. x181). Valid data is indicated by a HIGH signal on DVALID.

In NTSC/PAL video applications, the pixel clock must be very precise, typically within 50 ppm (parts per million) accuracy. The video input clock to the Troika is 13.5 MHz or 27 MHz for both NTSC and PAL applications. Extra caution should be taken when using a crystal or oscillator clock source for video encoders, since the clock source accuracy may vary on multiple components.

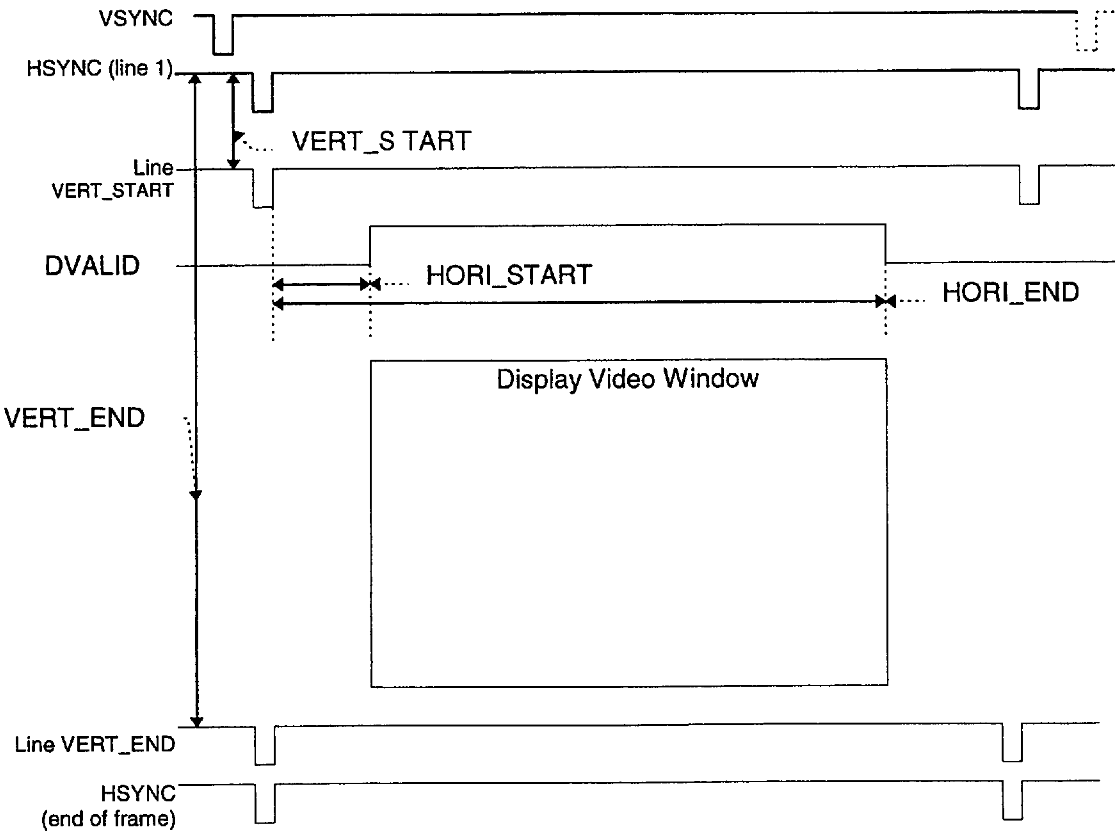


Figure 3-24: *Video Display Window Settings*

3.5.3 MASTER MODE VIDEO INTERFACE

Figure 3-25 shows a video master mode connection to a video overlay chip.

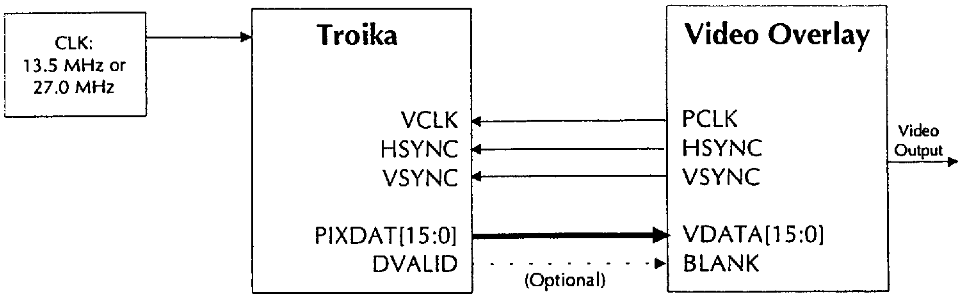


Figure 3-25: *Troika Master Mode Video Interface*

As displayed in the figure above, the Troika is responsible for generating VCLK and the control signals for the PIXDAT[15:0] bus and DVALID lines for data, but the synchronization signals, VSYNC and HSYNC, are generated by the NTSC/PAL encoder. The control signal VCLK is programmable in frequency, polarity, and signal width. VCLK can be either 13.5 MHz or 27 MHz.

If a 27-MHz or 13.5-MHz input clock is used for the Troika system clock, the Troika can be used as a video master provided that the 27-MHz or 13.5-MHz input clock source meets all accuracy requirements for the NTSC/PAL encoder.

For MPEG video playback in VGA overlay, a multimedia/graphics chip provides the processing necessary for color space conversion, re-sizing, and frame conversions.

The PIXDAT[15:0] and DVALID timing is identical to video slave mode, and all the video display range registers (HORI_START, HORI_END, VERT_START, VERT_END) are valid in the master mode application. Please refer to the Troika application notes for more detail regarding the Troika master mode programming.

3.6 AUDIO INTERFACE

The audio module interfaces to the RISC CPU and SDRAM controller modules, and its main function is to reconstruct Dolby Digital (AC-3), LPCM, and MPEG-1 (Layer 1 or Layer 2) audio samples, and to provide a two-channel output. In addition, the audio module provides an audio co-processor bypass in the event audio decoding is intended for off-Troika chip processing. The reconstruction block reconstructs the quantized version of the set of mapped samples, and inverse map transforms these mapped samples back into uniform PCM. Full 24-bit sample and data paths are maintained to ensure high-quality reconstruction. (Note: If error check is applied, the compressed bit stream unpacking and error detection in the encoder are done by the RISC CPU.)

The ABIU Control register (Reg. x196) is used to indicate the type of mute function applied by the Troika. Enabling bit[7] provides a software mute, which utilizes the programmed value in the software mute coefficient (bits[15:8]) to offer variable volume mute control. There are 256 possible mute values, ranging from completely muted all the way up to no mute at all. Enabling bit[1] provides a hardware mute, which has two states, on and off. This function is used to prevent the audio DAC from receiving an Troika output signal containing an audio error. Stopping the flow of improper data to the audio DAC prevents the user from hearing undesirable sounds generated from illegal bit streams. Popping sounds output by the speakers are one example.

In the MPEG audio decoding mode the emphasis output signal is driven with the emphasis indicated in the MPEG Audio stream. The type of de-emphasis used is contained in EPHSIS[1:0] (Reg. x199 bits[10:9]).

The audio master clock, AXCLK, is either 256, 320, or 384 times the sampling frequency, correlating to 16-, 20-, or 24-bit audio samples, respectively. The sampling frequencies supported by the ABIU include 32 KHz, 44.1 KHz, 48 KHz, and 96 KHz. The bit clock, ABCLK, is used for outputting the audio data sample bits. ABCLK is either 32, 40, or 48 times the sampling clock, which is necessary for two-channel output (Right/Left) for 16-, 20-, or 24-bit audio samples.

The Troika provides reconstructed PCM audio samples in bit-serial format directly to the external audio DACs, and the audio DAC interface is programmable so that most of the popular audio DACs can be used. The audio interface supports several serial data output modes that may be set from the host bus into the Audio Control Mode register (Reg. x199). In addition, the audio attenuation is controlled by the audio interface unit and the emphasis information is provided.

The Troika audio interface is a direct serial connection to the audio DAC as in the following diagram:

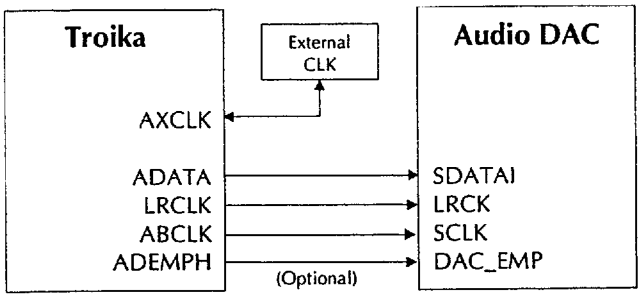


Figure 3-26: *Troika Audio Interface*

The audio interface can be configured to support several serial data output modes. Please refer to the Audio Control Mode register (Reg. x199).

3.6.1 AUDIO OUTPUT LEVEL CONTROL

The Troika has internal registers to control the audio output level. The ABIU Control register (Reg. x196) incorporates a mute function that controls the type of mute function that will be applied.

MUT_COEFF[7:0] of the ABIU Control register (Reg. x196 bits[15:8]) is used as the software mute coefficient. The default value is *0xff*, which indicates the coefficient used for non-attenuated volume (the highest possible volume). The minimum value is *0x0*, which refers to an audio mute. For any value between the range of *0xff* and *0x0*, the audio output level decreases proportional to the value decrease within the audio coefficient range. The output level control function is as follows:

Audio Output Level = Soft mute coefficient / 0xff * original volume

When ABIU_EN (Reg. x196 bit[0] of the ABIU Control register is set LOW, the audio output is disabled and the output attenuation on the PCM samples is set to negative infinity.

3.6.2 AUDIO INTERFACE CLOCKING MODES

The Troika supports all MPEG-1 audio sampling frequencies of 32 KHz, 44.1 KHz, 48 KHz, and 96 KHz. The main clock source is the AXCLK signal line, which will in turn generate the ABCLK (bit clock) and the LRCLK (the left/right channel clock). The exact frequency is selectable by PCM_BITOUT (Reg. x199 bits[5:4]. The ABCLK is generated by dividing AXCLK by 8 or 4, and can be either 48, 40, or 32 times the sampling clock. The polarity of LRCLK and valid data at ABCLK's rising or falling edge can be programmed by LRCK_INV (Reg. x199 bit[1]). The following figures show the relationship between AXCLK, ABCLK, and LRCLK.

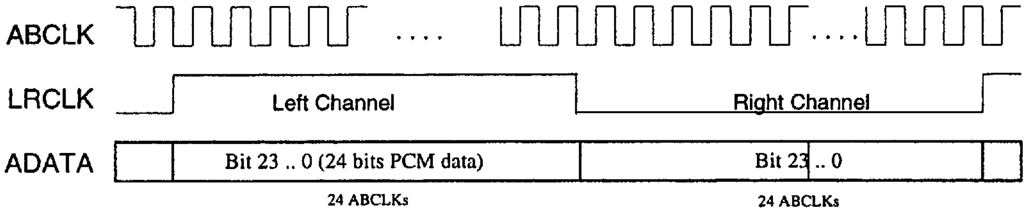


Figure 3-27: *ABCLK = 32 * Sampling Rate, Left Channel High, MSB First*

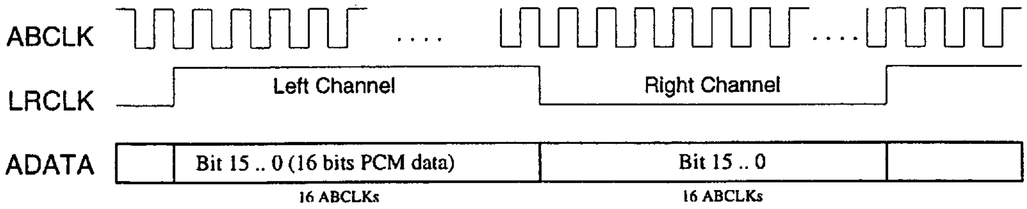


Figure 3-28: *ABCLK = 48 * Sampling Rate, Left Channel High, MSB First*

3.6.3 AUDIO CLOCK FREQUENCIES

Sampling Frequency (KHz)	AXCLK Divisor	Samples		
		256x (16-bit data path)	320x (20-bit data path)	384x (24-bit data path)
32 KHz	8	8.192 MHz	10.240 MHz	12.288 MHz
44.1 KHz	8	11.2896 MHz	14.112 MHz	16.9344 MHz
48 KHz	8	12.288 MHz	15.360 MHz	18.432 MHz
96 KHz	8	24.576 MHz	30.720 MHz	36.864 MHz
96 KHz	4	12.288 MHz	15.360 MHz	18.432 MHz

3.6.4 INPUT/OUTPUT AUDIO SAMPLES

The Troika offers flexibility for the sample size of input/output bit streams. For three input possibilities, there are a total of nine output possibilities for the Troika, depending on the input requirements of the audio DAC:

- 16-bit sample input
  - 16-bit output to audio DAC
  - 20-bit output to audio DAC
  - 24-bit output to audio DAC

- 20-bit sample input
  - 16-bit output to audio DAC
  - 20-bit output to audio DAC
  - 24-bit output to audio DAC

- 24-bit sample input
  - 16-bit output to audio DAC
  - 20-bit output to audio DAC
  - 24-bit output to audio DAC

3.6.5 POLARITY OF THE BIT CLOCK (ABCLK) AND LEFT/RIGHT CHANNEL CLOCK (LRCLK)

The polarity of the bit clock, ABCLK, and the left/right channel clock, LRCLK, is programmable via the Audio Control Mode register (Reg. x199). When BICK_H (bit[2]) is set HIGH, this indicates that the audio sample output changes on the clock's rising edge. When LRCK_INV (bit[1]) is set HIGH, this indicates that the left channel audio output is represented by LRCLK=1 and the right channel output is represented by LRCLK=0.

3.6.6 IEC-958 SUPPORT

The Troika supports the following IEC-958 outputs (two-channel only):

- Decoded PCM
- Linear PCM source
- MPEG compressed audio
- AC-3 compressed audio

Also, the Troika can output 96-KHz audio samples on the IEC-958 by providing downsampling of the 96-KHz audio sample to a 48-KHz audio sample.

3.7 DVD-DSP INTERFACE

The DVD-DSP interface provides direct connection to an 8-bit parallel DVD-DSP (see Figure 3-29).

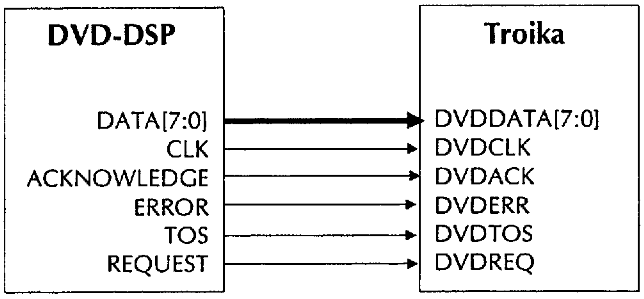


Figure 3-29: *Troika DVD-DSP Interface*

The Troika is designed to receive a compressed bit stream directly from a DVD-DSP. The Troika supports the following input formats: DVD VOBU, MPEG-2 Program or MPEG-1 System data, MPEG-1 video, and MPEG-1 audio. The DVD Interface Unit (DVDIU) is used as an 8-bit interface for a DVD-DSP, and supports an ACK/REQ mode (acknowledge/request) for the DVD-DSP.

In the DVD-DSP mode the ACK/REQ interface supports data burst rates up to 27 Mbits/second. If lower bit rates are anticipated, changing the memory partitions will conserve memory.

The DVD-DSP is programmable and offers flexibility to support the most common interfaces for DVD-DSP chips. The DVDTOS (Top of Sector) input signal is an optional signal that can be used to control the DVD input bit stream rate. For example, the Troika is programmable to input a single 2048-byte sector and then prevent further data input until another start signal is received. For more detailed information, refer to the register documentation (Reg. x091, x092, and x096).

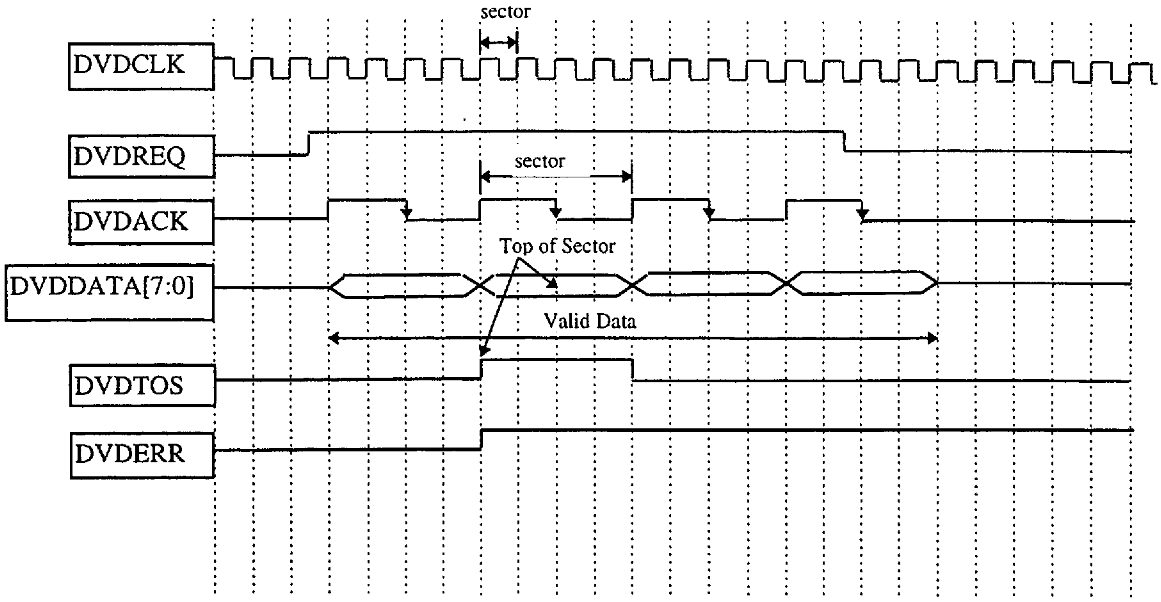


Figure 3-30: *Example of DVD-DSP Timing Diagram*

3.8 CD-DSP INTERFACE

The Troika's CD-DSP interface unit (CDIU) provides a direct interface to receive the bit-serial data output from the CD-DSP chip. The CD-DSP can output data streams for CD-DA (Audio PCM data), CD-ROM (ISO 9001 CD-ROM file format), and MPEG (Video CD, CD-I or Karaoke CD titles). In CD-ROM mode, the on-chip CD controller provides P/Q erasure corrections and EDC, and the on-chip registers are programmable to support various popular CD-DSP chips.

The following diagram shows the connection examples:

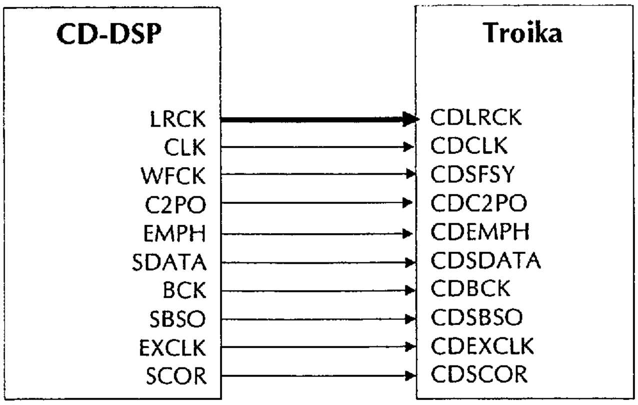


Figure 3-31: *Troika CD-DSP Interface*

In the CD-ROM decoding mode, the Troika handles Q/P erasure corrections and EDC. The serial data from the CD-DSP can be programmed to pass data with MSB or LSB first, and for added flexibility, the on-chip CD interface registers are programmable for several different input formats. This allows many of the popular CD-DSP chips to be supported.

The CDIU processes data in the following modes:

1. Seeking mode
   - Discards the user data
   - Only reports the header and sub-header information

2. CD-ROM decoding mode
   - Real-time processing of mode 0 and skipping to the next sector
   - Real-time parsing for mode 1 and mode 2-form 1 sectors (including the Q/P correction when not decoding MPEG)
   - Real-time parsing for mode 2-form 2 sectors

3. CD-DA mode
   - Ignores data error flag C2PO, and directly transfers data to the audio DAC interface unit

The Troika stores the header and sub-header information, which is accessible to the host. The host can set BLK_TXF_EN (Reg. x1ff bit[0]) to HIGH to read the disc information, play sequence descriptors, list ID offset tables, and read other information from SDRAM.

3.8.1 CD-ROM MODES SUPPORTED

The Troika can process the CD-ROM data in the following modes:

1. CD-DA mode (used for audio CD)
   - The Troika performs direct pass through to the audio interface of the CD-DA data, which also includes the CD-DA emphasis signal
   - The C2PO flag and all error checking are ignored

2. CD-ROM decoder mode (used in common CD-ROM files, CD-I, and Video CD applications)
   - Mode 2, Form 2 sectors: real-time parsing and decoding of MPEG Stream
   - Mode 2, Form 1 and mode 1 sectors: real-time parsing and performs ECC or EDC error checking if the Troika is not decoding MPEG streams
   - Mode 0 sectors: real-time processing

Besides the CD-DA mode, the Troika stores the sector's sub-header and system related information into internal registers for the host processor to access. Please refer to the Troika's register definitions for details regarding CDIU registers.

3.8.2 CD-DSP INTERFACE TIMING

The relationship between BCLK, CD-ROM speed, and data rates is as follows:

CD-DSP Interface Timing:

CD-ROM speed and data rates	CDBCK	CDBCKs per CDLRCK
Single (1.42 Mbits/s)	1.42 MHz	32
Single (1.42 Mbits/s)	2.13 MHz	48
Single (1.42 Mbits/s)	2.84 MHz	64
Double (2.84 Mbits/s)	2.84 MHz	32
Double (2.84 Mbits/s)	4.26 MHz	48
Double (2.84 Mbits/s)	5.84 MHz	64

The following diagram shows the CD-DSP interface timing. Data protocol and timing signals are programmable by the Disk Interface Unit (DIU) DSP Control register (Reg. x080). Data includes:

- CD-BCLK - Latch data at rising edge or falling edge
- CD-LRCK - Left or Right channel high
- CD-DATA - MSB first or LSB first, BCLKs per LRCK (64,48 or 32)
- CD-C2PO - MSB first or LSB first

Since the Troika is a passive device in the CD-DSP interface, please refer to the CD-DSP chip data sheet and program the CD-DSP register accordingly.

Signal Timing Example:

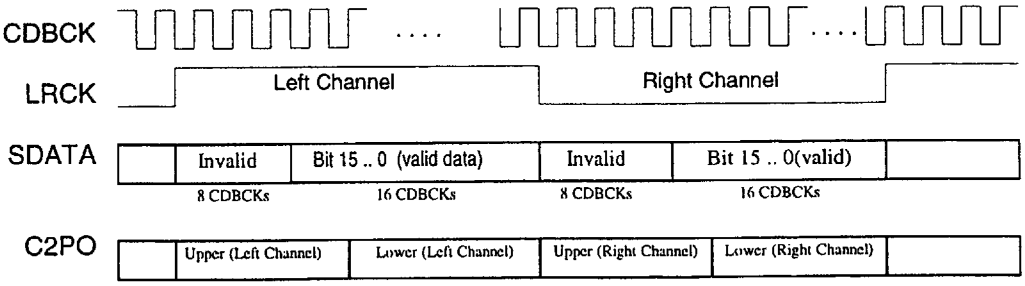


Figure 3-32: *24-bit BCLK, MSB First, Left Channel High, C2P0 MSB First*

3.9 VIDEO BUS INTERFACE UNIT (VBIU)

The video decompression co-processor contains four basic modules:

- VLD unit
- IQ/IDCT unit
- Pixel filter/motion compensation (PF/MC) unit
- RISC CPU

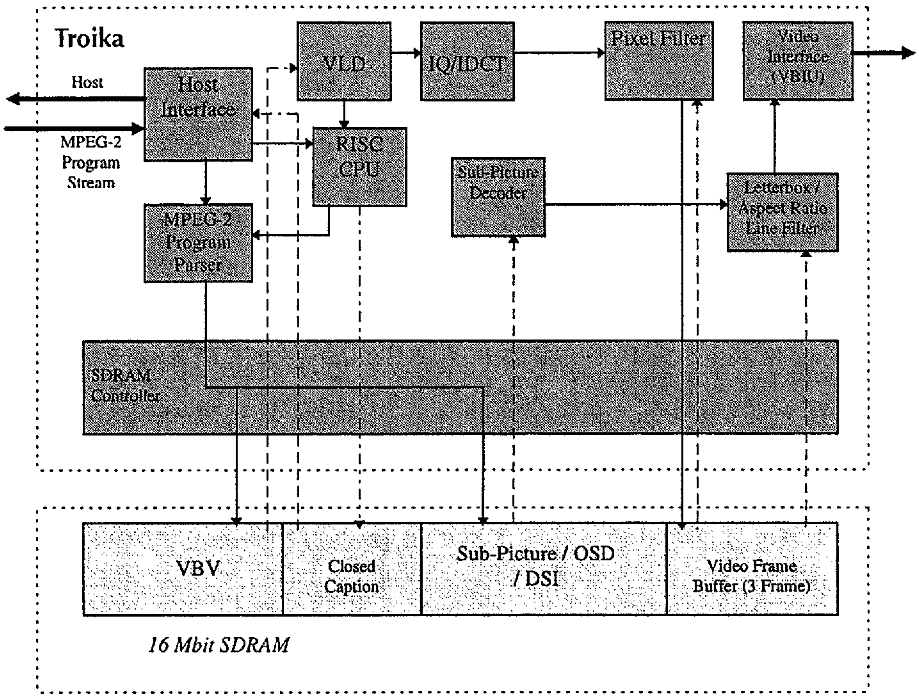


Figure 3-33: *Video Flow Diagram*

3.9.1 VARIABLE LENGTH DECODER (VLD) UNIT

VLD interfaces with SDRAM controller, IQ/IDCT, and the RISC CPU's functional units. The process begins when VLD receives 16-bit MPEG bit streams from SDRAM for decoding. Upon completion of an 8x8 block accumulation of quantized DCT coefficients, the coefficients are inverse scanned (zigzag to raster scan) and passed to IQ/IDCT. Along with the quantized coefficients and macroblock addresses, motion vectors associated with every macroblock are available to transfer to the CPU. The RISC CPU's microcode monitors handshaking signals between the VLD and CPU to complete the VLD operation.

Note that the VLD implementation of the Troika performs the decoding of MPEG-1 Video bit streams as well as detecting illegal bit streams and reporting detected errors to the CPU.

3.9.2 INVERSE QUANTIZER/INVERSE DISCRETE COSINE TRANSFORM (IQ/IDCT) UNIT

The IQ/IDCT module interfaces with the VLD, Pixel Filter, and RISC CPU modules. This unit receives 8-bit coefficients from the VLD and then performs the inverse quantization and inverse DCT operations. The IDCT arithmetic operation precision exceeds the IEEE standard, and provides better video quality. After the IDCT operation, 9-bit coefficients are output to the macroblock storage module.

The intra and non-intra quantizer coefficients may be loaded by the host or the CPU to a quantizer coefficient storage memory in the IQ/IDCT module. The coefficient must be loaded in zigzag scan order for proper quantizer operation.

3.9.3 PIXEL FILTER/MOTION COMPENSATION (PF/MC) UNIT

The PF/MC module interfaces to the CPU and SDRAM controller modules, and is mainly used to control the pixel reconstruction of decoded frames with the reference frames. Decoded differential data is stored in IQ/IDCT, and reference picture data is preprocessed through the PF module. The decoded differential data and reference picture data is added together in the MC module before the reconstruction data is stored into SDRAM.

3.9.4 RISC CPU

Functions running in parallel under the supervision of the on-chip RISC CPU:

- MPEG-2 Program/MPEG-1 System parsing/packet decoding
- Video above slice layer decoding
- Audio decoding
- Data flow control
- Video output control
- Audio output control
- Audio/video synchronization operation
- Error concealment handling
- Closed Caption extraction

3.10 AUDIO DECOMPRESSION CO-PROCESSOR

This module interfaces to the CPU and SDRAM controller modules. The main function of this module is to reconstruct the MPEG-1 (Layer 1 or Layer 2) audio samples. The reconstruction block reconstructs the quantized version of the set of mapped samples, and inverse map transforms these mapped samples back into uniform PCM (Note: the compressed bit stream unpacking and error detection are done by the RISC CPU). In general, the reconstruction operation is divided into the requantization of the sub-band sample and synthesis sub-band filtering. These include procedures for matrixing, windowing, and reconstructing PCM samples.

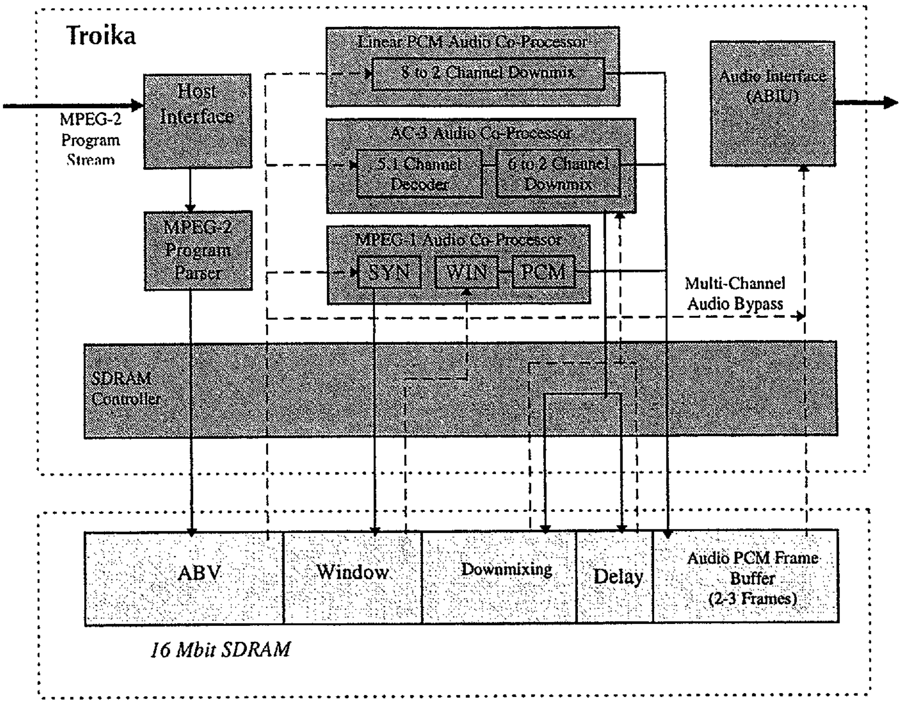


Figure 3-34: *Audio Flow Diagram*

AC-3 decoding is done through both microcode and a dedicated hardware module. First, microcode loads an AC-3 bit stream from the audio stream buffer and searches for an AC-3 sync word. After locating each sync frame, microcode will extract the frame header, including the bit rate and sampling frequency information. Microcode then turns to decode audio block data. Various coding parameters are extracted and stored in predefined global bus (GBUS) registers, whereas the coded data such as channel exponents, coupling coordinates, and delta bit allocation data are stored in data buffers located in the AC-3 audio co-processor on the chip. As soon as all fixed data is decoded, microcode will issue a signal to the hardware module to start calculating bit allocation based on the available information.

The bit allocation is done on the band-by-band basis for each audio channel (please refer to AC-3 specification for the definition of band). Upon finishing a band, hardware will send a signal to microcode to let it know the bit allocation data is ready. Microcode will then extract the packed mantissa data from the bit stream for that band based on the calculated bit allocation. It also does operations such as mantissa dequantization and denormalization. Such processed data is stored in a new data buffer in the AC-3 co-processor. After a number of such kinds of handshakings between microcode and co-processor, and until one whole channel is finished, the co-processor will start to do rematrixing, inverse MDCT, and downmixing whenever required for that channel. The downmixed data is stored in some buffer in external SDRAM.

The above process is repeated for each audio channel until all channels are processed for each audio block. Finally, the co-processor will perform de-windowing and put the final reconstructed data in the PCM buffer in SDRAM, thus finishing an audio block. With six audio blocks being done in the same way, the reconstruction for an audio frame is thus finished.

CHAPTER 4 TIMING DIAGRAMS

4.1 ISA I/O ACCESS

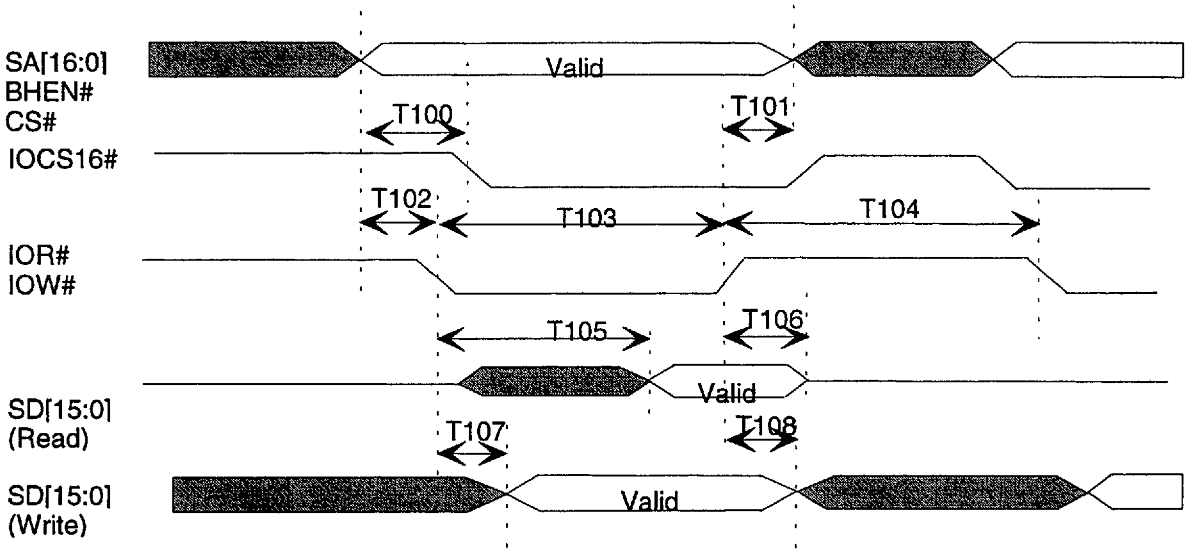


Figure 4-1: *I/O Access Timing*

Symbol	Parameter	Min (ns)	Max (ns)	Note
T100	IOCS16# active from valid SA[15:0]		20	
T101	SA[15:0], BHEN#, and CS# hold from I/O inactive	25		
T102	SA[15:0], BHEN#, and CS# setup to I/O	0		
T103	I/O command pulse width	60		
T104	I/O command inactive time	100		
T105	Valid read data from IOR# active		130	
T106	Read data hold from IOR# inactive	0		
T107	Write data valid from IOW# active	0		
T108	Write data hold from IOW# inactive	15		

4.2 ISA DMA

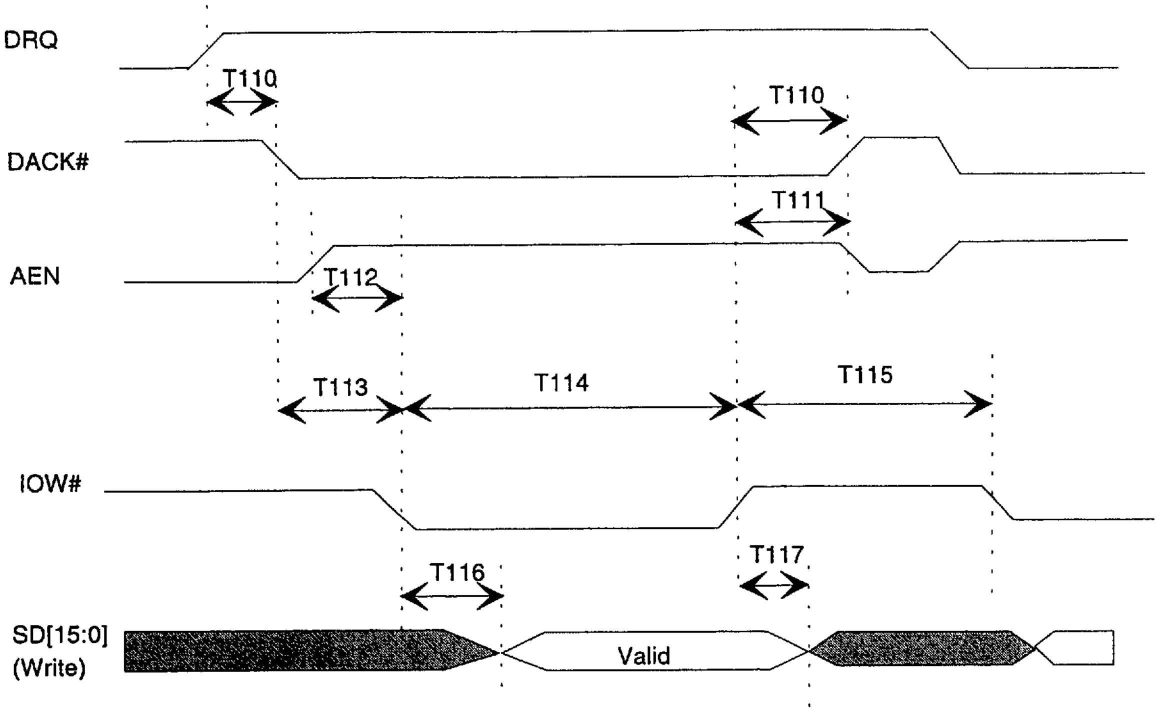


Figure 4-2: *ISA DMA Timing*

Symbol	Parameter	Min (ns)	Max (ns)	Note
T110	DACK# hold from IOW# inactive	0		
T111	AEN hold from IOW# inactive	15		
T112	AEN valid to IOW# active	0		
T113	DACK# active to IOW# active	0		
T114	IOW# pulse width	60		
T115	IOW# inactive time	100		
T116	Write data valid from IOW# active	0		
T117	Write data hold from IOW# inactive	15		

4.3 VBIU INTERFACE

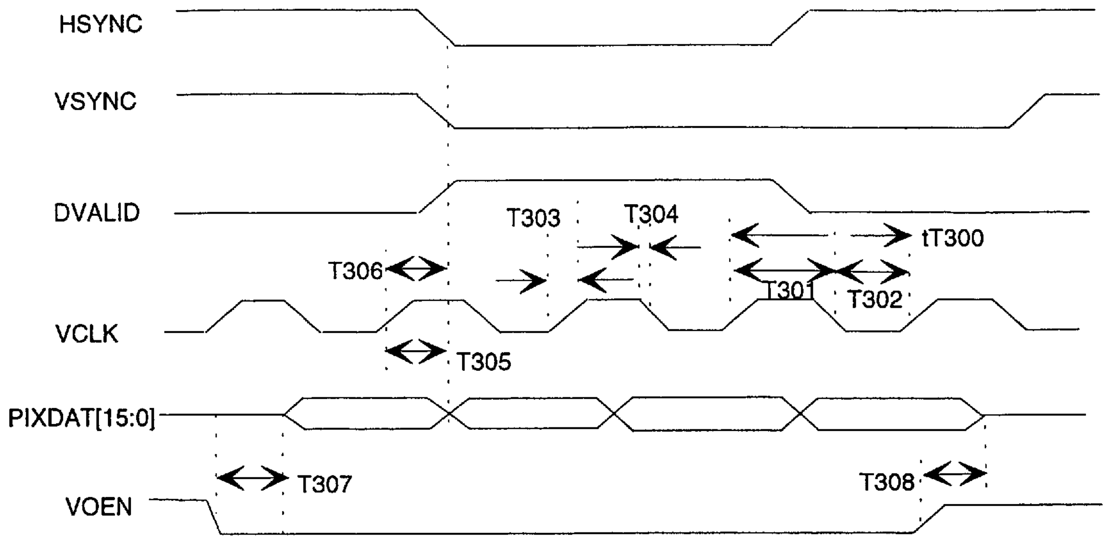
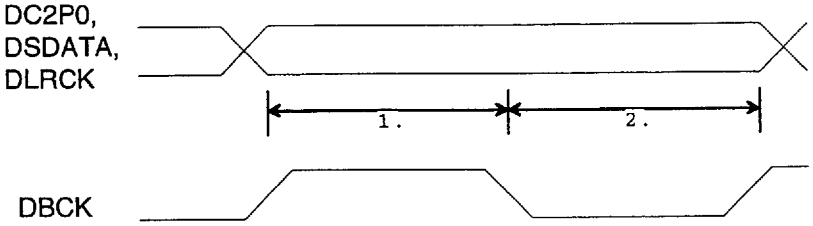


Figure 4-3: *VBIU Interface Timing*

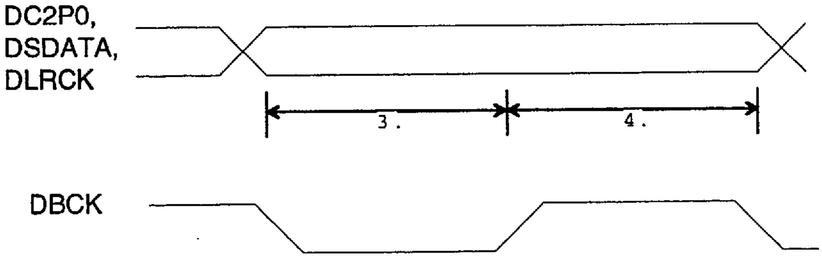
Symbol	Parameter	Min (ns)	Max (ns)	Note
T300	Master mode 10 MHz: VCLK period		100 nominal	
T300	Master mode 20 MHz: VCLK period		50 nominal	
T300	Slave mode 13.5 MHz: VCLK period		74 nominal	
T301	Slave mode 13.5 MHz: VCLK high period	30	44	40/60
T302	Slave mode 13.5 MHz: VCLK low period	30	44	40/60
T303	VCLK rise time (master & slave mode)		5.0	
T304	VCLK fall time (master & slave mode)		5.0	
T305	Pixel data output from rising edge of VCLK		12.5	c=50pf
T306	Control signal output from rising edge of VCLK		12.5	c=50pf
T307	VOE# to pixel data bus enable		12.5	
T308	VOE# to pixel data bus disable		12.5	

4.4 CD-DSP INTERFACE TIMING

Falling edge of clock DBCK selected (edge bit or register DSPSL set to 0)



Rising edge of clock DBCK selected (edge bit of register DSPSL set to 1)



No.	Description	Minimum Time	Maximum Time
1	DC2PO, DSDATA, DLRCK setup to DBCK fall edge (bit edge of register DSPSL set to 0)	20ns	N/A
2	DC2PO, DSDATA, DLRCK hold after DBCK falling edge (bit edge of register DSPSL set to 0)	20ns	N/A
3	DC2PO, DSDATA, DLRCK setup to DBCK rising edge (bit edge of register DSPSL set to 1)	20ns	N/A

CHAPTER 5 MICROCODE FUNCTION AND FEATURES DESCRIPTION

5.1 REGISTERS USED BY MICROCODE

The following ranges of register addresses are reserved for microcode designers:

x000-x01F
x040-x07F

5.2 MICROCODE FLOW CHART

The Level 0 Interrupt routine has the highest priority, followed by Level 1 and then Level 2. Therefore, the Level 2 Interrupt routine may be interrupted by Level 1 or Level 0, and the Level 1 Interrupt routine may be interrupted by Level 0, but not Level 2. The Level 0 Interrupt routine cannot be interrupted once the routine begins. Figure 5-1 shows the microcode flow chart.

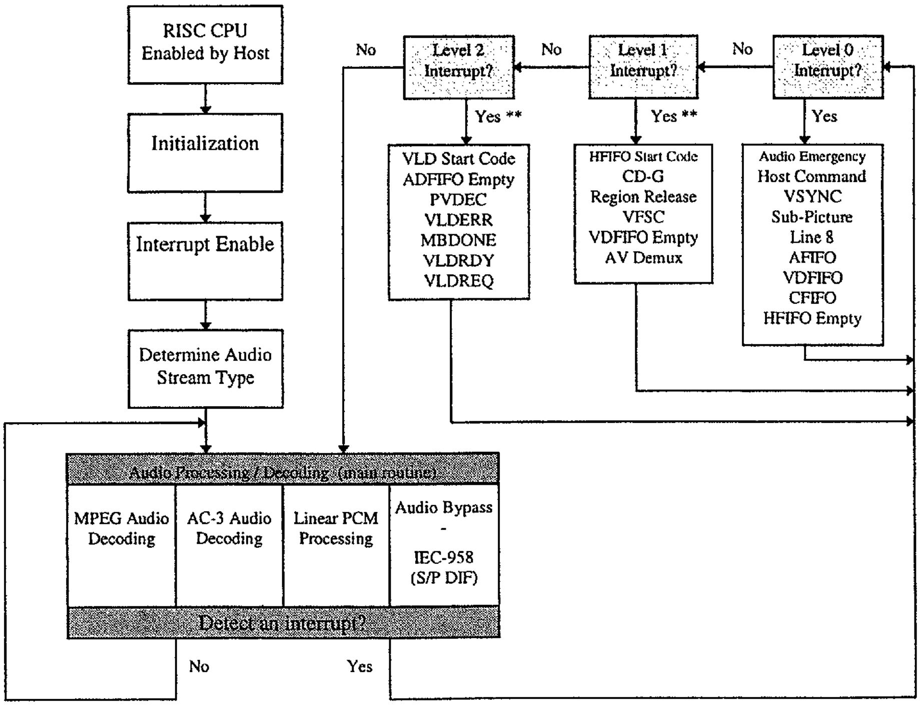


Figure 5-1: *Microcode Flow Chart*

After power up, the RISC CPU is in an idle mode. When CPUEN (Reg. x1f0 bit[0]) is set HIGH by the host processor, the CPU retrieves and executes instructions from IMEM address "0." This process initializes certain global bus registers (GBUS) or CPU internal registers to proper states and performs operations to initialize certain internal co-processors such as audio, SDRAM controller, etc.

After initializing the appropriate registers, the interrupt procedure starts as the program counter (PC) begins with the audio co-processor, the main routine. The microcode performs processing functions for AC-3 audio decoding, MPEG audio decoding, LPCM audio processing, and audio bypass. The architecture uses audio decoding as the main routine that can be interrupted by any one of the interrupt level routines.

There are three levels of interrupt sources, and within each interrupt level certain register bits are defined for masking control of each individual interrupt so that another interrupt of higher priority may pause a currently running routine. The masking control turns the masking function on or off for each individual interrupt source. The following are examples of the functions that are performed in the interrupt routines:

- MPEG-2 and MPEG-1 video decoding
- Various FIFO services
- Audio/video parsing for MPEG-2 Program stream and MPEG-1 System Layer decoding
- Output Control for video & audio
- Host Command interrupt handling

The microcode controls both the interrupt enable/disable and the mask on/off. Therefore, turning on the enable bit within microcode permits the interruption of a currently running interrupt routine as long as the other interrupt routine has a higher priority.

The various interrupt sources are summarized as follows:

Level 2 Interrupt - the lowest priority interrupt routine

VLD_SCS_REQ (Reg. x1fa bit[7]):
Setting this bit HIGH initiates the VLD Start Code Search request.

RESERV_2_INT (Reg. x1fa bit[6]):
Setting this bit HIGH initiates one of the two software programmable interrupt requests.

ADFIFO_EMPTY (Reg. x1fa bit[5]):
Setting this bit HIGH indicates the audio decoding FIFO (ADFIFO) is empty, which then initiates a request data transfer from the SDRAM audio buffer verifier (ABV) to the audio decoding FIFO (ADFIFO).

PVDEC (Reg. x1fa bit[4]):
Setting this bit HIGH indicates to the CPU to start the video picture layer above decoding. After power on, the microcode code initializes a process video decoding (PVDEC) request to have the CPU start the video stream decoding.

VLDERR (Reg. x1fa bit[3]):
Setting this bit HIGH indicates that there is a variable length decoder (VLD) syntax error detected by the VLD.

MBDONE (Reg. x1fa bit[2]):
Setting this bit HIGH indicates the PF/MC operation for the previous macroblock is done.

VLDRDY (Reg. x1fa bit[1]):
Setting this bit HIGH indicates that one macroblock's VLD decoding is finished and is ready to decode another macroblock.

VLDREQ (Reg. x1fa bit[0]):
Setting this bit HIGH indicates that a start code other than the slice start code is detected by the VLD.

Level 1 Interrupt - the second highest priority interrupt routine

HFIFO_SCS (Reg. x1f9 bit[6]):
Setting this bit HIGH initiates the HFIFO Start Code Search

CD-G (Reg. x1f9 bit[5]):
Setting this bit HIGH initiates the CD-G request from the CDIU (1 CD-G symbol is arrived in CDIU)

RESERV_1_INT (Reg. x1f4 bit[4]):
One of the two software programmable interrupt requests. Includes the data for 16 lines, regional release, and end of display, which are all internal microcode routines. For example, with regional_release the SDRAM controller generates a region_release request to tell the CPU that one region has finished displaying and the SDRAM memory space is available for use by the decoding function.

VFSC (Reg. x1f9 bit[2]):
SDRAM controller periodically generates this request to provide a timer to have the RISC CPU go back to the overwrite_checking routine to determine if an overwrite condition exists.

VDFIFO_EMPTY (Reg. x1f9 bit[1]):
Setting this bit HIGH indicates that the video decoding FIFO (VDFIFO) is empty.

AV_DEMUX (Reg. x1f9 bit[0]):
Setting this bit HIGH indicates that the host FIFO (HFIFO) is requesting that the CPU begin audio/video parsing.

Level 0 Interrupt - the highest priority interrupt routine

AUD_EMGCY (Reg. x1f8 bit[8]):
Setting this bit indicates an audio emergency request. This bit is set by the audio co-processor when the audio emergency time out counter expires before the audio co-processor's SDRAM request is served.

HOST_CMD (Reg. x1f8 bit[7]):
Setting this bit HIGH indicates that there is a host command interrupt initiated by the host. The microcode will need to read another register to know exactly what command has been set by the host.

VSYNC (Reg. x1f8 bit[6]):
Setting this bit HIGH indicates a vertical synchronization (VSYNC) interrupt is generated by the video bus interface unit (VBIU).

SP (Reg. x1f8 bit[5]):
Setting this bit HIGH indicates a Sub-picture request so that the sub-picture decoder is ready to begin decoding.

LINE8 (Reg. x1f8 bit[4]):
Setting this bit HIGH indicates a Line 8 request, which is used to inform the microcode when Line 8 of the video display has begun. During Line 8, the internal microcode will update all pointers and registers to switch to sub-picture decoding.

AFIFO (Reg. x1f8 bit[3]):
Setting this bit HIGH indicates that the audio FIFO (AFIFO) data level is below the threshold value.

VDFIFO (Reg. x1f8 bit[2]):
Setting this bit HIGH indicates that the video decoding FIFO (VDFIFO) data level is below the threshold value.

CFIFO (Reg. x1f8 bit[1]):
Setting this bit HIGH indicates that the parser control (CFIFO) data level is below the threshold value.

HFIFO_EMPTY (Reg. x1f8 bit[0]):
Setting this bit HIGH indicates that the host FIFO (HFIFO) is empty.

5.3 SUMMARY OF MICROCODE FUNCTIONS AND FEATURES

Audio Decoding

After power on initialization, the CPU loops at the audio decoding portion of the microcode. Audio decoding is the main routine, which performs various types of audio decoding functions depending on the incoming bit stream types. The main routine can be interrupted by any one of the three interrupt levels.

MPEG Audio Stream Decoding

The compressed audio data follows the path from the HFIFO, to the parser, to the CFIFO, to the ABV buffer (SDRAM), to the ADFIFO. The RISC CPU reads data from ADFIFO, decodes the data elements according to MPEG-1 Layer 1 and Layer 2 audio specifications, and passes the information to the audio co-processor. Simultaneously, the RISC CPU initiates the "start" signal so that the MPEG audio co-processor begins the audio reconstruction. The RISC CPU will handle the audio frame buffer management as well as the audio error recovery if a bit stream error occurs and is detected.

AC-3 Audio Stream Decoding

The RISC CPU reads data from the ADFIFO, which contains the AC-3 bit stream. The CPU extracts the bit stream information, encoding parameters, and the encoded data, and then sends the decoded data to the AC-3 audio co-processor, which handles the handshaking between the AC-3 co-processor and the BAND basis. The CRC checking is done by the hardware and the microcode to detect errors in the bit stream to implement a mute function until error recovery.

Linear PCM Audio Processing Mode

The data flow is HFIFO to the parser, to the CFIFO, to SDRAM ABV, to the ADFIFO. The microcode reads the data from the ADFIFO and passes the data to the AC-3 audio co-processor. The microcode will then send a start signal to the AC-3 co-processor to implement the downmixing procedure. The data will then be put back into the SDRAM PCM frame buffer.

Audio Bypass Mode

In the audio bypass mode, the data flow is through the HFIFO, to the parser, to the CFIFO, to the ABV buffer. The data is transferred to the AFIFO directly instead of the ADFIFO. The microcode is used to control particular registers in the bypass mode.

The incoming bit stream data goes to the HFIFO first. When the data level exceeds a programmable threshold level the HFIFO will generate the AV_DEMUX request to the RISC CPU. The CPU will enter the AV_DEMUX interrupt routine to handle the AV parsing and system layer bit stream decoding. The CPU can handle the MPEG-1 system layer and MPEG-2 program layer decoding until the packet layer. For the video or audio packets that match the target ID set by the host, the CPU will program the parser to transfer the packet data to the CFIFO. This data then goes to the VBV or ABV buffer in SDRAM. The microcode also supports the syntax decoding for the sub-picture packet, the DSI (data search information) packet, and the PCI (presentation control information) packet. For the other packets, the CPU will decode to the packet length (Reg. x054 information) and the program parser discards the unnecessary packet data.

MPEG-2/MPEG-1 Video Bit Stream Decoding Layer above the Syntax Decoding

The CPU reads data from the VDFIFO and decodes MPEG-2 and MPEG-1 video bit streams from the sequence header down to the slice layer. When the slice layer is detected, the CPU transfers the VDFIFO control to the VLD hardware and decodes the data to the block level in hardware.

When the VLD detects a start code other than a slice start code, the VLD sends a VLD request to the CPU. The CPU then takes control of the VDFIFO and handles the picture layer and above video decoding for the next picture.

Macroblock Decoding Control and the Handshaking with the VLD/SDRAM Controller

The CPU initiates a VLDGO (Reg. x1fb bit[0]) to have the VLD start the slice layer below decoding. Then, for each macroblock the CPU monitors the VLD_RDY (from VLD), which indicates that one macroblock's VLD decoding has finished. When the VLD_RDY is detected, the CPU issues a CCRP (Reg. x1fb bit[1]) signal which tells the VLD to start decoding for the next macroblock. The CPU then polls the MB_DONE signal from the SDRAM controller to determine if the PF/MC operation for the previous macroblock is complete. When the MB_DONE signal is detected, the CPU issues a PF_START (Reg. x1fc bit[1]) to have the PF/MC unit start the next macroblock's reconstruction.

The RISC CPU also performs the decoding overwrite_checking (internal microcode) in order to prevent the decoding process from overwriting the SDRAM frame buffer area where the display has not yet started processing.

The microcode also works with the SDRAM controller to support the B-frame regioning scheme, which uses a unique method of saving SDRAM area without any loss of data. The microcode is responsible for selecting the available regions for decoding and sets up the display region sequence correctly. The purpose of the B-regioning scheme is to save frame buffer SDRAM space, or in some cases to increase the decoding performance. The number of regions is host programmable.

FIFO Services and SDRAM Buffer Management

The normal bit stream data flow is described as follows. The data comes into the HFIFO, to the parser, and to the CFIFO. The CPU sends a CFIFO request (Reg. x1f8 bit[1]) and a request ID (Reg. x17b bits[11:8]) to the SDRAM controller to have the CFIFO data transferred to the VBV, ABV, SPV, DSIV, or PCIV buffer in the SDRAM. The data in the VBV will then be transferred to the VDFIFO. The data in the ABV will be transferred to the ADFIFO, and the data in the SPV will be transferred to the sub-picture decoder. The video reconstruction process (PF/MC) will dump the reconstructed data to the SDRAM video frame buffers. The audio reconstruction process performed by the audio co-processor will dump the audio data into the SDRAM audio frame buffers. The VBIU will get the data from the video frame buffers to the VFIFO and then shift the data out by the video clock (VCLK) to the display buffer. The ABIU is used to transfer data from the audio frame buffers to the AFIFO and then shift the data out to the audio DAC. The sub-picture information is processed by the sub-picture decoder and then passed through a line filter to integrate the sub-picture and video information.

The CPU is responsible for sending the CFIFO, VDFIFO, ADFIFO request to the SDRAM controller. The SDRAM controller performs the real data transfer. The VFIFO request goes directly to the SDRAM controller, but the CPU will need to perform the VFIFO flush and the display base address update at the VSYNC interrupt service routine.

The CPU manages the VBV, ABV, and SPV buffers, making sure that there is no data overwrite. Registers x149 and x14a define the start and end SDRAM address of the ABV, and registers x14b and x14c define the write pointer and the read pointer to the ABV. Registers x154 and x155 define the start and end SDRAM address of the VBV. Registers x156 and x157 define the write pointer and the read pointer to the VBV. Registers x1b4 and x1b5 define the start and end SDRAM address of the SPV. Register x1b7 defines the write pointer to the SPV, and the read pointer to the SPV is defined through registers x1b9 and x1bb.

Error Concealment

The video layer bit stream syntax error is detected by the CPU or the VLD. The RISC CPU performs the frame based error concealment operation when an error is detected, and also performs the audio bit stream syntax checking to detect the audio bit stream error. When an audio error is detected, the microcode initiates mute/ unmute operations to perform audio error concealment.

PAL/NTSC Support and Automatic Frame Rate Conversion

The Troika microcode supports NTSC and PAL display modes. The host writes the format to the input bit stream through PAL_FLG of the B-Frame Region Mode register (Reg. x053 bit[3]). The output format is determined by setting the Vertical Start register (VERT_START, Reg. x183) and the Vertical End register (VERT_END, Reg. x182). The microcode detects the bit stream source type with different frame rates and performs the frame rate conversion. The microcode is also responsible for setting up the correct VBIU vertical filtering mode (VERT_CTL, Reg. x18c) to eliminate the aspect ratio distortion that is caused by NTSC-to-PAL conversion.

CHAPTER 6 SUB-PICTURE

6.1 SUB-PICTURE DECODER MECHANISM

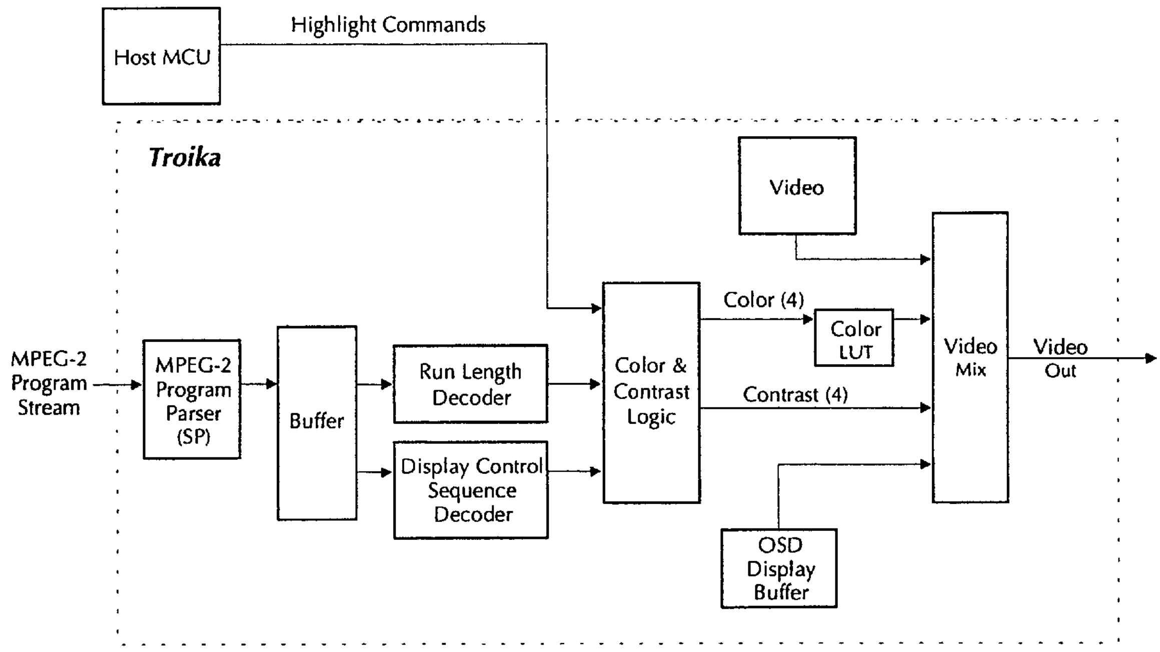


Figure 6-1: *Sub-Picture Decoder Block Diagram*

6.2 SUB-PICTURE DECODER

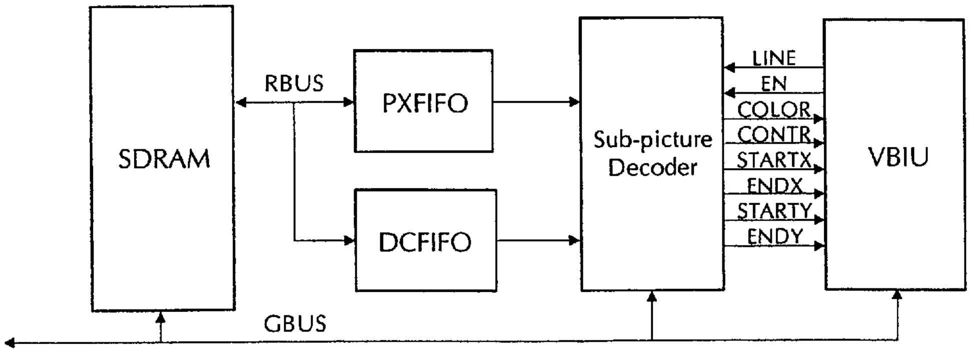


Figure 6-2: *Sub-Picture Decoder Block Diagram*

Two FIFOs are required for the sub-picture decoding operation. The PXFIFO buffers the sub-picture encoded pixel data (PXD), and the DCFIFO buffers the display command sequence tables (DCSQT). The PXD is the actual encoded image that is to be decoded and displayed, and the DCSQTs are the tables that manipulate the PXD to provide color information, contrast, and presentation formats.

Since two FIFOs are needed, the SDRAM controller must support a flexible division between the PXD and the DCSQT. This division is set by the microcode according to information provided in the sub-picture header. The RADR for the SDRAM access is shown in Figure 6-3.

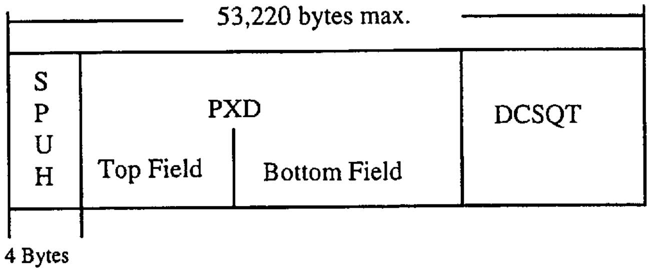


Figure 6-3: *Sub-Picture Unit*

Sub-Picture Unit Header (SPUH):

SPUH	Number of bits	Comments
SPU_SZ	16	Size of sub-picture unit
SP_DCSQT_SA	16	Start address of DCSQT

SPU_SZ:
Describes the size of the sub-picture unit in bytes. The maximum unit size is 53220 bytes. The size of an SPU in bytes shall be even. (When the size is odd, one (FFh) shall be added at the end of the SPU, to make the size even.) The size of SP_DCSQT in an SPU shall be equal to or less than half of the size of SPU.

SP_DCSQT_SA:
Describes the relative byte start address of the Display Control Sequence Table with respect to the first byte of the sub-picture unit.

Four different pixels are allowed. Each of the four pixels can have a different luminance value, chromance value, and contrast value. The four types of sub-picture pixel data are shown in the table at the top of page 6-3.

Sub-Picture Pixel Data:

Pixel Data	Value
Background Pixel	00
Pattern Pixel	01
Emphasis Pixel - 1	10
Emphasis Pixel - 2	11

Sub-picture pixel data is run-length encoded, as shown below for each line.

1. For run lengths of 1 to 3 pixels of the same pixel data.

d0	d1	d2	d3
Number of Pixels		Pixel Data	

2. For run lengths of 4 to 15 pixels of the same pixel data.

d0	d1	d2	d3	d4	d5	d6	d7
0	0	Number of Pixels				Pixel Data	

3. For run lengths of 16 to 63 pixels of the same pixel data.

d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11
0	0	0	0	Number of Pixels						Pixel Data	

4. For run lengths of 64 to 255 pixels of the same pixel data

d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
0	0	0	0	0	0	Number of Pixels								Pixel Data	

5. For run lengths to the end of the line of the same pixel data

d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Pixel Data

6. Each line of pixel data should be byte-aligned. If needed, add 4 "0" bits (0000b) to adjust.

7. The size of the run-length coded data within one line shall be 1440 bits or less.

Sub-Picture Display Sequence Table:

SP_DCSQT	Comments
SP_DCSQ #0	Display Control Sequence #0
SP_DCSQ #1	Display Control Sequence #1
-	-
-	-
SP_DCSQ #n	Display Control Sequence #n

Sub-Picture Display Control Sequence Table:

SP_DCSQ ( )	Number of bits	Comments
SP_DCSQ_STM	16	Display Control Start Time
SP_NXT_DCSQ_SA	16	Start Address of Next SP_DCSQ
SP_DCCMD #1	varies	Display Control Command #1
-	-	-
-	-	-
SP_DCCMD #n	varies	Display Control Command #n

SP_DCSQ_STM:

Describes the execution start time of Sub-Picture Display Control Commands. It is relative PTS with respect to the PTS described in the sub-picture packet (SP_PKT) header. Only bits 25 through 10 are used.

$$\text{Execution start time} = \frac{\text{(PTS of SP packet[32:0] + SP\_DCSQ\_STM[25:10] x 1024) seconds}}{90000}$$

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SP_DCSQ_STM[25:18]								SP_DCSQ_STM[17:10]							

SP_NXT_DCSQ_SA:

Describes the relative byte start address of the next Display Control Sequence with respect to the first byte of the sub-picture unit. If no other SP_DCSQs exist, describes the start address of this SP_DCSQ relative to the first byte of the sub-picture unit.

SP_DCCMD #n:

Describes one or more Display Control Commands executed in this SP_DCSQ. The same SP_DCCMD cannot be described more than once.

Sub-Picture Display Control CommandTable:

SP_DCCMD ( )	Value	Comments	Additional Bytes
FSTA_DSP	00h	Forcibly sets display start timing of pixel data	0
STA_DSP	01h	Sets display start timing of pixel data	0
STP_DSP	02h	Sets display stop timing of pixel data	0
SET_COLOR	03h	Sets luminance code of pixel data	2
SET_CONTR	04h	Sets contrast of pixel data	2
SET_DAREA	05h	Sets display area of pixel data	6
SET_DSPXA	06h	Sets display start address of pixel data	4
CHG_COLCON	07h	Sets change of color/contrast for pixel data	PCsize +2
CMD_END	FFh	End of Display Control Command	0

FSTA_DSP:

Command to forcibly start the display of an updated sub-picture unit, regardless of the on/off display state of the sub-picture. The command code is (00h).

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0

STA_DSP:

Command to start the display of an updated sub-picture unit. The command code is (01h). This command may be overruled by the OFF display state of the sub-picture.

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	1

STP_DSP:

Command to stop the display of a sub-picture unit. The command code is (02h).

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	1	0

SET_COLOR:

Command to set the color of each pixel. The command code is (03h).

b23	b22	b21	b20	b19	b18	b17	b16
0	0	0	0	0	0	1	1
b15	b14	b13	b12	b11	b10	b9	b8
Emphasis Pixel-2 color code				Emphasis Pixel-1 color code			
b7	b6	b5	b4	b3	b2	b1	b0
Pattern Pixel color code				Background Pixel color code			

Note: The color palettes used by SET_COLOR are defined either in theVideo Title Set Management Information asVTSSPPLT or in theVolume Information File Manager asVMMSPPLT.

SET_CONTR:

Command to set the mixture ratio between each pixel of the pixel data and the main picture. The command code is (04h).

The contrast is calculated as follows:

Contrast:

Main Picture = (16-k)/16
Sub-Picture (Pattern Pixel) = k/16
When SET_CONTR equal '0', k = described value
When SET_CONTR not equal '0', k = described value+1

b23	b22	b21	b20	b19	b18	b17	b16
0	0	0	0	0	1	0	0
b15	b14	b13	b12	b11	b10	b9	b8
Emphasis Pixel-2 contrast				Emphasis Pixel-1 contrast			
b7	b6	b5	b4	b3	b2	b1	b0
Pattern Pixel contrast				Background Pixel contrast			

SET_DAREA:

Command to set the rectangular display area and position of the pixel data. The upper left and bottom right corners of the display area are described. The command code is (05h).

<table>
<tr><td>b55</td><td>b54</td><td>b53</td><td>b52</td><td>b51</td><td>b50</td><td>b49</td><td>b48</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>b47</td><td>b46</td><td>b45</td><td>b44</td><td>b43</td><td>b42</td><td>b41</td><td>b40</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="6">Start X-coordinate (upper bits)</td></tr>
<tr><td>b39</td><td>b38</td><td>b37</td><td>b36</td><td>b35</td><td>b34</td><td>b33</td><td>b32</td></tr>
<tr><td colspan="4">Start X-coordinate (lower bits)</td><td colspan="2">Reserved</td><td colspan="2">End X-coordinate</td></tr>
<tr><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td></tr>
<tr><td colspan="8">End X-coordinate (lower bits)</td></tr>
<tr><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="6">Start Y-coordinate (upper bits)</td></tr>
<tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td></tr>
<tr><td colspan="3">Start Y-coordinate (lower bits)</td><td>0*</td><td colspan="2">Reserved</td><td colspan="2">End Y-coordinate (upper bits)</td></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td colspan="8">End Y-coordinate (lower bits)</td></tr>
</table>

Note: coord = coordinate
* Start Y-coordinate shall be even numbers, therefore, LSB is '0'

SET_DSPXA:

Command to set the head address of the data used for displaying. The command code is (06h).

<table>
<tr><td>b39</td><td>b38</td><td>b37</td><td>b36</td><td>b35</td><td>b34</td><td>b33</td><td>b32</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td></tr>
<tr><td colspan="8">Head address of the pixel data for top field (upper bits)</td></tr>
<tr><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td colspan="8">Head address of the pixel data for top field (lower bits)</td></tr>
<tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td></tr>
<tr><td colspan="8">Head address of the pixel data for bottom field (upper bits)</td></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td colspan="8">Head address of the pixel data for bottom field (lower bits)</td></tr>
</table>

CHG_COLCON:

Command to change the color and the contrast of pixel data at each video frame change during display. The command code is (07h). In the extended field, enter its size and the pixel control data that complies with the Pixel Control Data (PXCD).

Extended field size = (m-7) / 8 [bytes]

<table>
<tr><td>bm</td><td>bm-1</td><td>bm-2</td><td>bm-3</td><td>bm-4</td><td>bm-5</td><td>bm-6</td><td>bm-7</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>bm-8</td><td>bm-9</td><td>bm-10</td><td>bm-11</td><td>bm-12</td><td>bm-13</td><td>bm-14</td><td>bm-15</td></tr>
<tr><td colspan="8">Extended field size (upper bits)</td></tr>
<tr><td>bm-16</td><td>bm-17</td><td>bm-18</td><td>bm-19</td><td>bm-20</td><td>bm-21</td><td>bm-22</td><td>bm-23</td></tr>
<tr><td colspan="8">Extended field size (lower bits)</td></tr>
<tr><td>bm-24</td><td>bm-25</td><td>bm-26</td><td>bm-27</td><td>bm-28</td><td>bm-29</td><td>bm-30</td><td>bm-31</td></tr>
<tr><td colspan="8">Pixel control data (start)</td></tr>
</table>

<table>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td colspan="8">Pixel control data (end)</td></tr>
</table>

CMD_END:

Command to terminate Display Control Sequence. This command is required as the last part of the DCSQ.

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	1	1	1	1	1

CHAPTER 7 ON-SCREEN DISPLAY (OSD)

7.1 OSD OVERVIEW

The function of the video overlay unit is to provide the basic mechanism necessary to overlay graphics images on the video picture and include proper alpha-blending between video and graphics pixels. The major building blocks consist of one 128x16 OSD FIFO (OFIFO), one 256x16 Color Look-Up Table (CLUT), one alpha-blending unit, and corresponding hardwire control functions. OFIFO and CLUT utilize dual-port RAM. Alpha blending utilizes random logic.

7.2 OSD FUNCTIONAL DIAGRAM

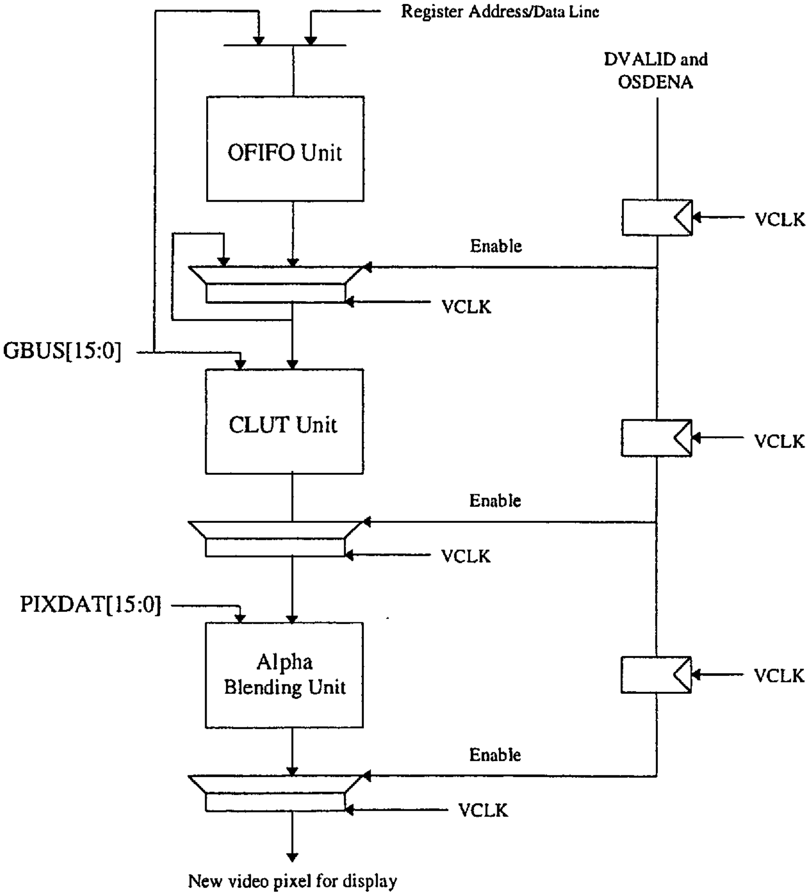


Figure 7-1: *OSD Functional Diagram*

7.3 OSD FUNCTIONAL DESCRIPTION

The OFIFO unit consists of 128x16 bits dual-port RAM, and functions as a FIFO storage for the OSD data that is filled by SDRAM. The OFIFO has a read pointer and a write pointer which operate at two asynchronous clocks, namely the system clock and the video clock (VCLK). The read and write pointers wrap around to a zero value whenever the pointers reach the last entry (i.e., the 127th entry) of the RAM storage. Since the read and write to the OFIFO occur asynchronously, proper care is taken to synchronize the read/write operations. Upon system reset or setting the OSD enable, OSDENA (Reg. x19a bit[11]), to LOW, both pointers to the OFIFO will be reset to a zero value.

When the OFIFO is less than half full and OSDENA (Reg. x19a bit[11]) is set HIGH, a request is sent to the SDRAM controller for a 64x16 block refill into the OFIFO. The request signal is asserted HIGH until OFIFO is at least half full.

When video pixel data from the VBIU is ready for display, the DVALID (pin 140) signal will be HIGH, accompanied by the corresponding pixel data information, PIXDAT (pins 141-146, 148, 150-157, 159). DVALID initiates the popping out of the OSD data residing in the dual-port RAM, and depending on the value in BIT_PER_PIXEL (Reg. x19a bit[10]), proper muxing control ensures the extraction of the OSD bits per pixel for mapping into the CLUT. When BIT_PER_PIXEL is set HIGH, 4 bits per pixel can be used for OSD. When BIT_PER_PIXEL is set LOW, then 2 bits per pixel can be used for OSD.

The Color Look-Up Table (CLUT) unit is a 256x16 dual-port RAM, and allows up to 8-bit address mapping into 256 entries of 16-bit color. The color content is assigned into 4-bit V, 4-bit U, and 8-bit Y starting from MSB. Entry 0 of the CLUT is strictly designated at the Transparent Color Location. This designation allows automatic translation of CLUT address zero into video bypass mode.

Reading and writing to the CLUT is accessed through the global address bus (GBUS). To write to the CLUT, first set up the address location by using the write to register, OSD_CLUT_ADD (Reg. x19b), and then writing the desired 16-bit color data to the CLUT write data, OSD_CLUT_OUT (Reg. x19c). Each GBUS write to OSD_CLUT_OUT will automatically increment the write pointer for the CLUT. For diagnostic purposes, during OSDENA set to LOW, the CLUT entries can be read onto GBUS through the same protocol by reading the data from OSD_CLUT_IN (Reg. x19d). To ensure proper operation, writes to CLUT should be performed only when OSD is disabled (OSDENA = 0).

The blend unit performs the alpha blending for the incoming pixel data and the OSD data. When the MSB of GBLEND (Reg. x19a bits[5:0]) is set, no attenuation is applied to the OSD data. The 5-bit local blending, LBLEND (Reg. x19a bits[4:0]), dictates the weighting average for the OSD data and incoming pixel data during the alpha-blending operation. The values vary from 0 to 16, with value 0 being all OSD and value 16 being all video.

The previous operations described are performed in three pipeline stages. The OFIFO unit operation is done in the first stage, followed by the CLUT operation in the second stage, and alpha blending in the final stage. The entire operation is initiated by the assertion of the DVALID signal from the VBIU. After 3 VCLK cycles, a new DVALID and new pixel data (PIXDAT) signal will be generated and will be sent to the I/O pad.

The Troika also includes the flexibility to incorporate a variable OSD window, which permits various window sizes and screen locations. Window sizes are programmable to allow an OSD window to vary in size. A minimal size OSD window creates the least amount of disruption for the viewer on the video screen, while a full-screen OSD window provides the maximum ease of viewing current user status. Variable OSD window screen locations are accomplished through programmable start and stop values for the X-coordinates and Y-coordinates in the OSD screen locations, which gives the system designer the flexibility to place the window in the screen location which provides the best viewing for the user.