Hombre: Beyond Amiga

Dr. Edward L. Hepler

Director of System and VLSI Architecture

Commodore Business Machines

This document describes Hombre, a next generation family of high performance computer systems. Hombre's architecture and implementation are discussed. Section 1 introduces Hombre. Sections 2 through 4 describe the Hombre chip set in more detail. Appendices give detailed register definitions and detailed descriptions of the behavioral models used to simulate the chip set.

1.0 Hombre

1.1 Introduction

This document describes the architecture of the next generation Commodore Chip Set. The chip set consists of two custom chips and an ASIC which, when packaged and configured in various ways, produces a family of products. The chipset has been designed to be configurable from a low-cost two-chip configuration for "game machine" style consoles to a three chip midrange performance desktop workstation. Additionally, it can be configured to act as a high performance graphics and I/O peripheral to any PCI based system.

The decisions leading to the specification found herein have been motivated by:

- The need to provide a low cost, high performance, next generation game system.
- The need to support three dimension(3D), realtime rendered, graphics for game and high end systems in a cost effective yet dynamic way.
- The need to support an industry standard bus so that low cost, high volume, industry standard peripherals may be used.
- The need to directly support CD-ROM, MPEG, MPEG audio, and other technologies that are key to multimedia products.

After some preliminary investigations of a hardware 3D pipeline and some discussion with software developers, it was determined that the algorithms to be used for 3D rendering were not established well enough to commit their implementation to hardware. Further, it was determined that the way in which rendering was to be performed for interactive games at television resolutions was not

the same as the way in which rendering would performed in high resolution workstations. This led to the decision that a programmable graphics engine that was capable of performing 3D algorithms was necessary. The flexibility of being programmed allows changes in the future as new algorithms become available. Three levels of 3D support have been identfied:

- High resolution still or single frame, non-realtime animation generation. Here full 24 bit (8/8/8) full color images are rendered. Fixed point arithmetic in 16.16 or 8.8 determines pixel values. Hence, a single 64-bit operation can manipulate 2 pixels at 16.16 or 4 pixels at 8.8. Note that pixels here refer to a single primitive color (R, G, or B) and that all must be rendered to display a full image.
- High quality, realtime animation generation. Here, 16 bit (5/5/5) high color images are rendered. These pixels are generated using 5.5 fixed point arithmetic. Hence a single 64-bit operation can manipulate 2 full (R, G, and B) pixels at a time. This form of rendering is anticipated to be used for those portions of an animation which are slow moving and require more detail.
- Lower quality, realtime animation generation. Here, 16 bit (5/5/5) high color images are rendered. These pixels are generated using integer arithmetic. Hence a single 64-bit operation can manipulate 4 full (R, G, and B) pixels at a time. This form of rendering is anticipated to be used for those portions of an animation which are fast moving and require less detail.

Graphics assist instructions have been identified and added to enhance performance where possible. Details of the assist instructions are given in a later section.

The graphics engine is based around a RISC core. The selection of "which RISC" was based upon a number of factors and resulted in the selection of HP's PA-RISC. Some of the main points involved in this selection include:

- The performance of PA-RISC is very high.
- The code density of PA-RISC is enhanced by the fact that its instruction set is more powerful than that of most RISC processors at similar clock rates. This along with its nullification capability provide dense code and the elimination of some branches.
- The PA-RISC instruction set has built into it the ability to add software-assist instructions and co-processors. The assist instructions are used to implement many of the graphics operations.
- The potential of future cooperation with HP made the PA-RISC attractive.
- The high cost of development and support of a proprietary architecture made doing so prohibitive.
- The ability to include blitter and rendering hardware support in front of the processor MMU and cache enable a unified cache to display and system memory.

The Hombre will leverage the use of Commodore produced custom silicon by applying it where it can best enhance performance.

The proposed development strategy will insure that this product can be introduced in a timely fashion. The use of high level simulation and the development of a software emulators will permit the parallel development of hardware and software. The adherence to both hardware and software standards will insure the ease of application porting.

1.2 Configurations

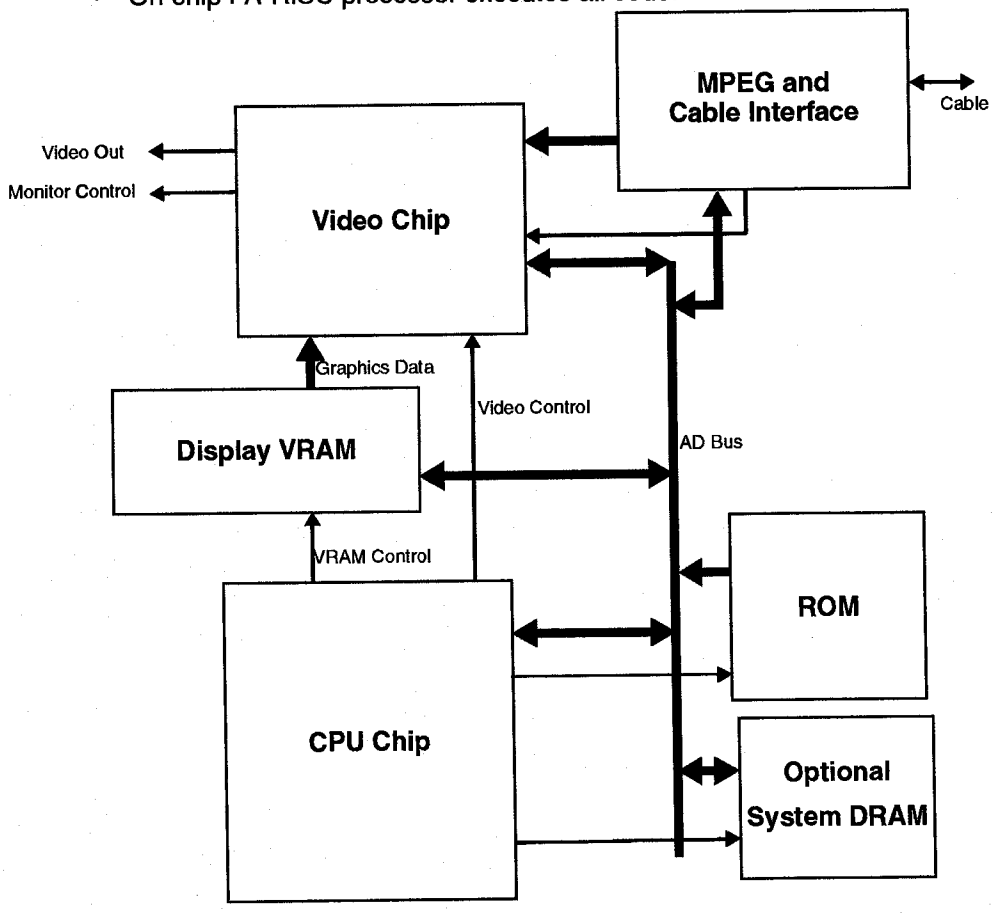
1.2.1 System Configurations

The partitioning of functionality among the chipset has been carefully chosen to permit a range of products to be created. In particular, all functions necessary for a low cost game system have been secured into the base two chips.

This chip set has been designed to extend the currently supported product line into a RISC processor based product line. As such, the following products may be supported:

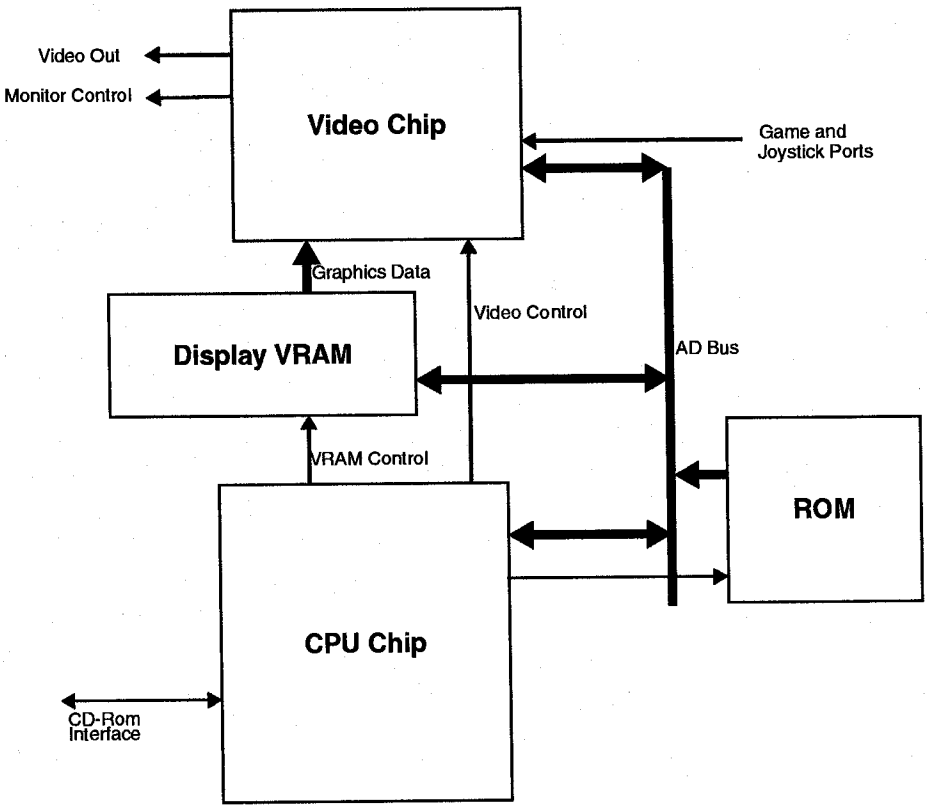
1.2.1.1 Cable box system

- Consists of CPU and VIDEO chips, display VRAM, and ROM.
- Uses low-cost chip configuration. Output is to an NTSC, PAL, or HDTV TV.
- On-chip PA-RISC processor executes all code.



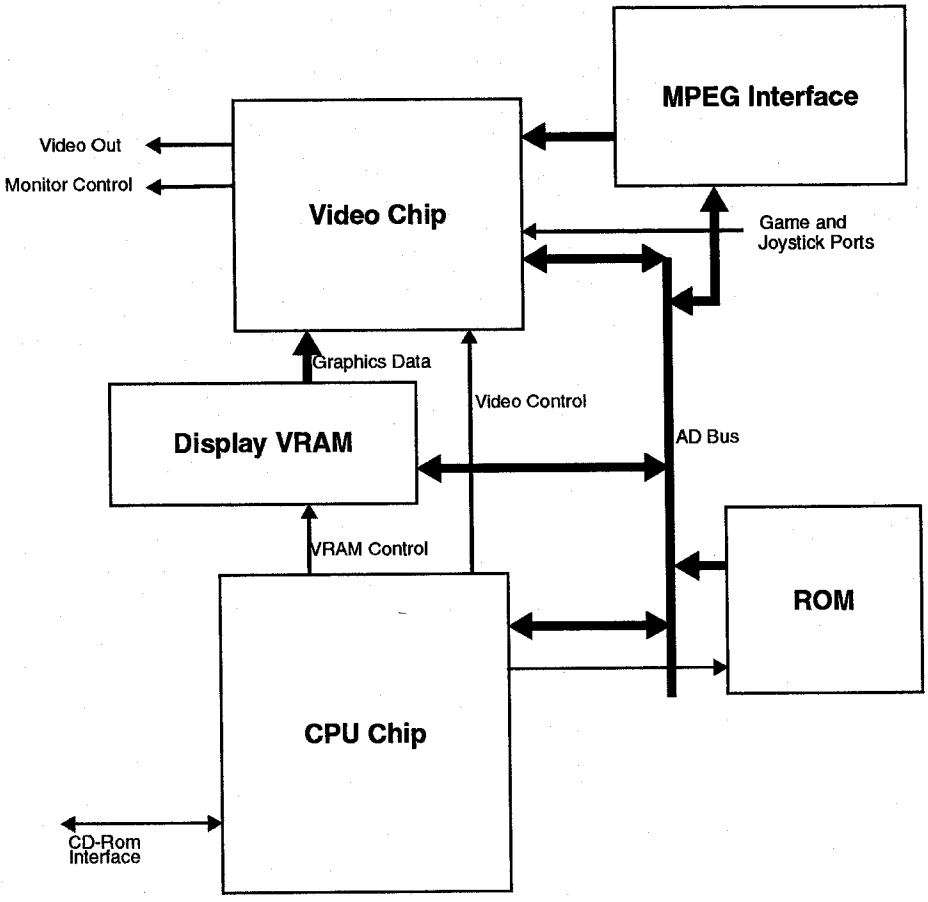
1.2.1.2 CD-game playing system:

- Consists of CPU and VIDEO chips, display VRAM, and ROM.
- Uses low-cost chip configuration. Output is an NTSC, PAL, or HDTV TV.
- Main storage device is the CD-ROM. On-chip PA-RISC processor executes all code.



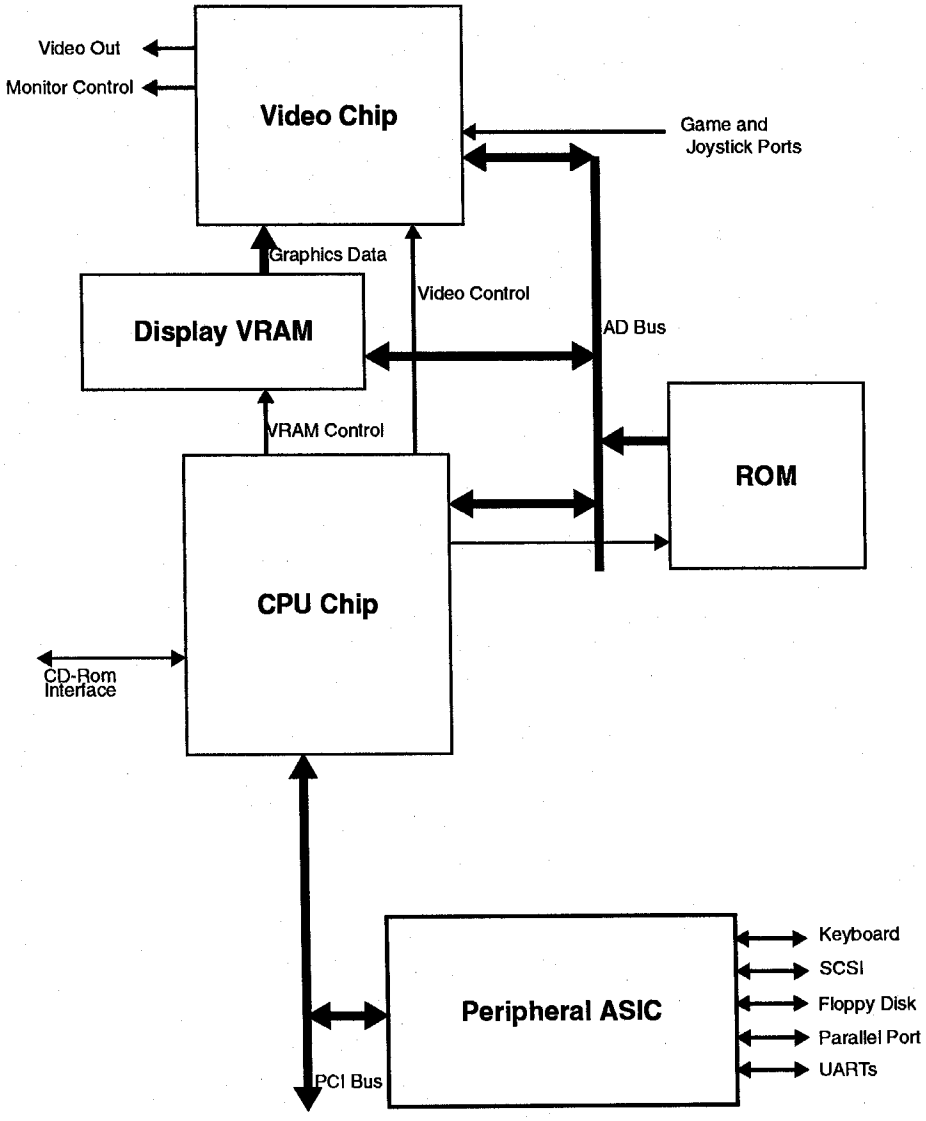
1.2.1.3 CD-game based MPEG player:

- Consists of CD-game playing system, plus:
- MPEG decoder hardware.
- RBG data from MPEG video decoder sent to VIDEO chip RGB inputs.
- MPEG audio data (CD format serial) sent to CPU chip serial audio input.



1.2.1.4 CD-game based home computer:

- Consists of CD-game playing system, plus:
- Base unit containing floppy disk, UART, parallel port, etc. ASIC
- Additional display and/or system memory.
- Peripherals (Floppy, HD, keyboard) provided via PCI based ASIC(s).
- NT capable

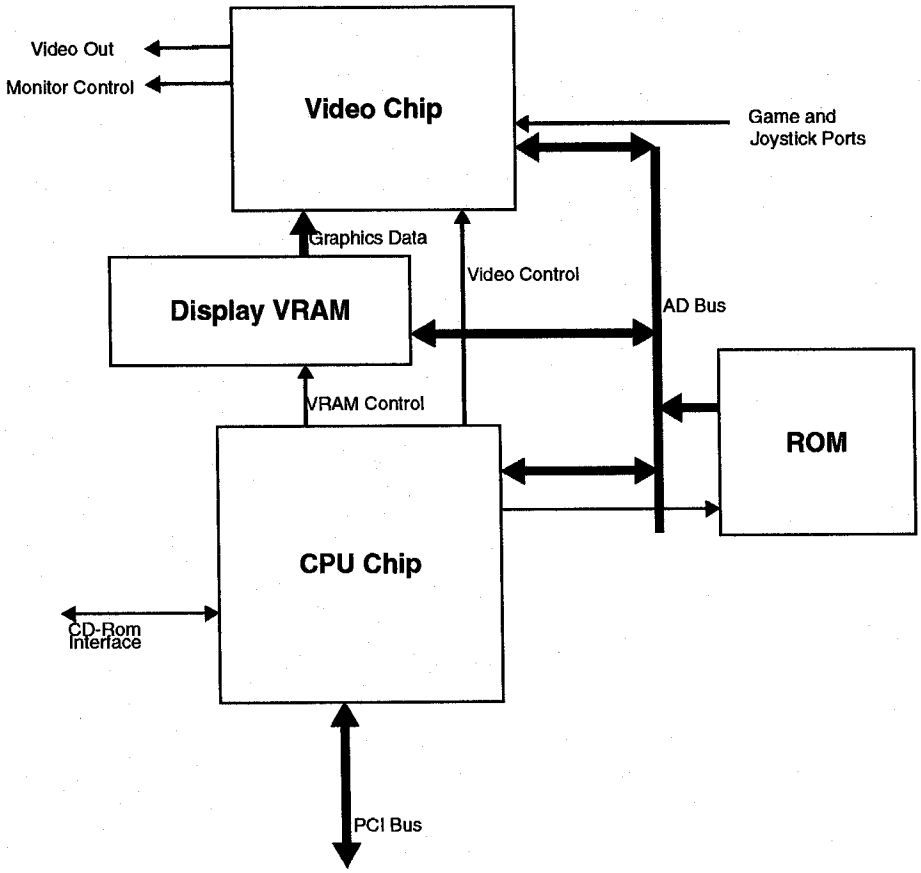


1.2.1.5 1200 class machine:

- Contains CD-game based home computer in a more traditional package.

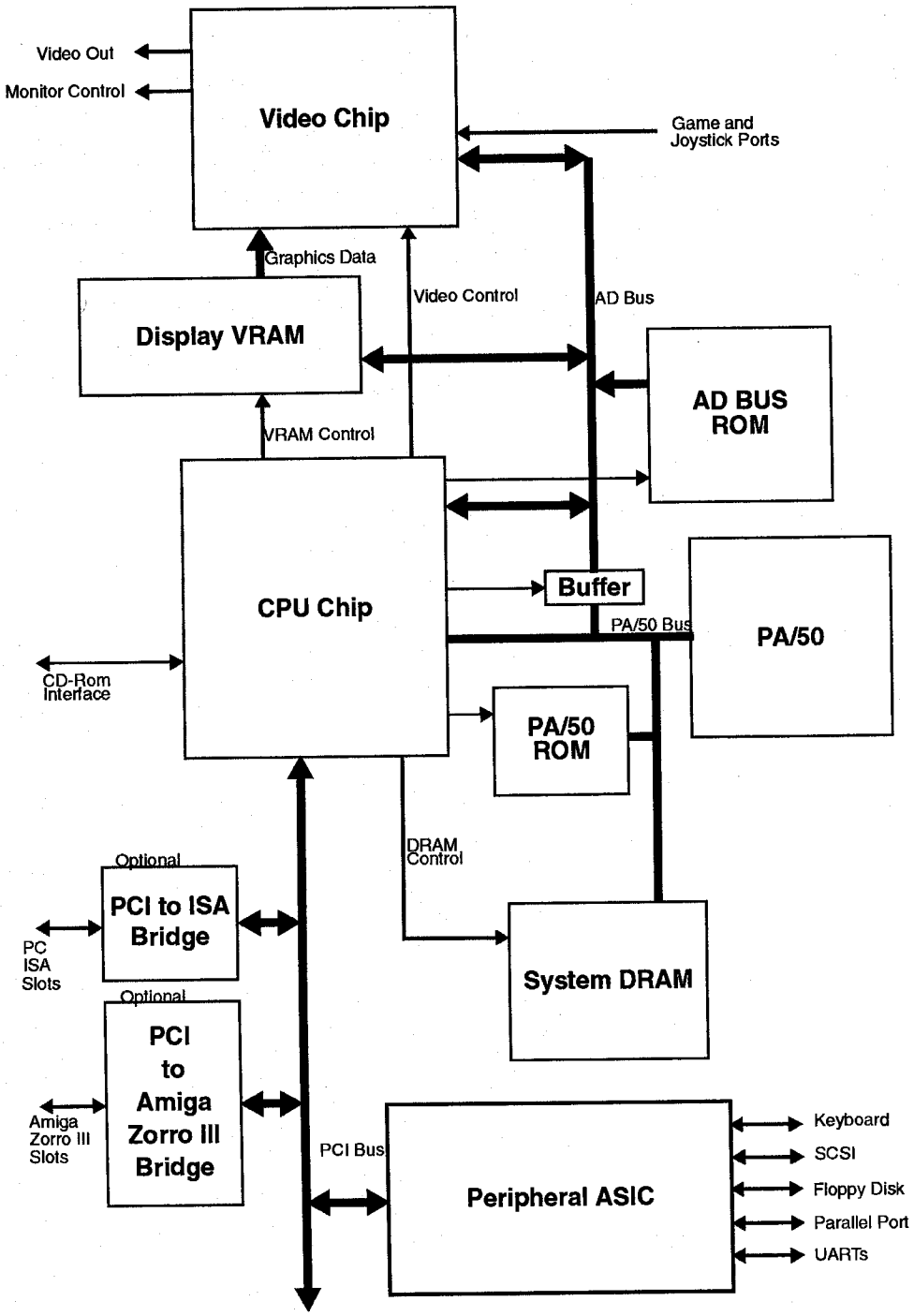
1.2.1.6 Multi-media graphics accelerator:

- Contains CD-game playing system chips on a plug-in card for any PCI based workstation or PC. Custom chip PCI interface becomes a slave serving the host system.



1.2.1.7 Desktop / Tower systems:

- Uses CPU and VIDEO chips as an IO processor.
- Commercial version of RISC processor (Hitachi PA/50) is used as system processor. System memory is used primarily by external RISC while display memory is used primarily by on-chip RISC core.



1.2.2 Chip Packaging Configurations

The CPU, VIDEO and PERIPHERAL chips will be packaged in two configurations:

1.2.2.1 Low cost configuration

The chips will be packaged in low-cost PQFP packages for game, cable TV, and home-computing applications. The parts packaged in this configuration will be limited in their frequency of operation to limit the power dissipation that must be supported.

A bonding opions will make the graphics data bus interface in this configuration be 32 bits in width. This will both allow a lower cost package to be used and permit lower cost systems to be built by reducing the minimum memory configuration. Internal data busses will remain at 64 bits and double page-mode fetches will be used to satisfy their needs.

This configuration will support up to 800x600x72x32 or 1024x768x72x8 resolution.

1.2.2.2 High performance configuration

Both the CPU and VIDEO chips will be packaged in more expensive, higher power dissipating packages. The operating frequency will be allowed to rise to the design limit permitting high resolution displays to be supported.

The full 64 bit graphics data busses will be brought to package pins.

This configuration will support up to 1280x1024x72x24 resolution.

1.3 Hombre Capabilities

This section summerizes the performance capabilities of the Hombre system.

1.3.1 Video Capabilities

1.3.1.1 Time Necessary for Accesses

Hombre uses VRAMs for display memory. This section discusses the time necessary to fetch the pixels required for one horizontal line of video. It assumes that the VRAMS used to implement the memory system have a 60ns page mode access and a 20ns serial mode access.

Further, it is assumed that VRAMS require eight 20 ns cycles to setup the serial access. After that, one access is possible every 20 ns cycle. Each access fetches 32 bits (four 8-bit pixels) or 64 bits (eight 8-bit pixels) depending upon the configuration (see section 1.3).

Assume that in a worst case, the word to be fetched first is on a page boundary. In this case, 8 cycles are used to setup VRAM, followed by a single cycle for the access.Eight additional cycles are needed to set up the VRAM for the next 512

accesses, etc. This is diagramed below where I represents a setup cycle and x represents a fetch cycle:

IIIIIIIIxIIIIIIIIxxxxxxxxxx...xxxxxxxIIIIIIIIxxxx...

The above example requires three notes:

- It assumes that video access requests have the highest priority.
- It also assumes that there is no delay between request and access start.
- Additional circuitry can be added to predict page crossings and eliminate many of the secondary overhead cycles.

Since any playfield may be scrolled, an additional access is required per playfield line. Given these facts and assumptions, Table 1 may be constructed.

Table 1:

Pixels/ playfield line	One 8-bit Playfield		Two 8-bit Playfields		Three 8-bit Playfields		Four 8-bit Playfields	
Bits/fetch	32	64	32	64	32	64	32	64
Overhead cycles	16	16	32	32	48	48	64	64
640	161/3.54	81/1.94	322/7.08	162/3.88	483/10.62	243/5.82	644/14.16	324/7.76
800	201/4.34	101/2.34	402/8.68	202/4.68	603/13.02	303/7.02	804/17.36	404/9.36
1024	257/5.46	129/2.90	514/10.92	258/5.80	771/16.38	387/8.70	1028/21.84	516/11.60
1280	321/6.74	161/3.54	642/13.48	322/7.08	963/20.22	483/10.62	1284/26.96	644/14.16

Here, each entry shows the number of memory accesses required to satisfy the pixel requirements for a single scan line followed by the number of microseconds required to perform them (including overhead) when the system clock is operating at 50 Mhz. Similarly, a table may be generated which shows the number of accesses and time required to satisfy sprite data fetches. This is shown in Table 2. Here, each entry shows the number of accesses, the number of over-

Table 2:

Sprites/line	Bits/Fetch	
	32	64
1	33/16/0.99	17/16/0.66
2	66/32/1.96	34/32/1.32
3	99/48/2.94	51/48/1.98

Table 2:

Sprites/line	Bits/Fetch	
	32	64
4	132/64/3.92	68/64/2.64
5	165/80/4.58	85/80/3.30
6	198/96/5.88	102/96/4.62
7	231/112/6.86	119/112/4.62
8	264/128/7.84	136/128/5.28

head cycles, and the number of microseconds required to fetch 8-bit pixels for 128 pixel sprites.

1.3.1.2 Time Available for Display

This section disccusses the time available to fetch one horizontal line of video data. It assumes that both video and sprites are fetched one line prior to use. The data is held in ping-pong buffers in the VIDEO chip. This allows the entire horizontal scan time (including sync) for data fetch.

Table 3 shows the horizontal line time for some common display sizes

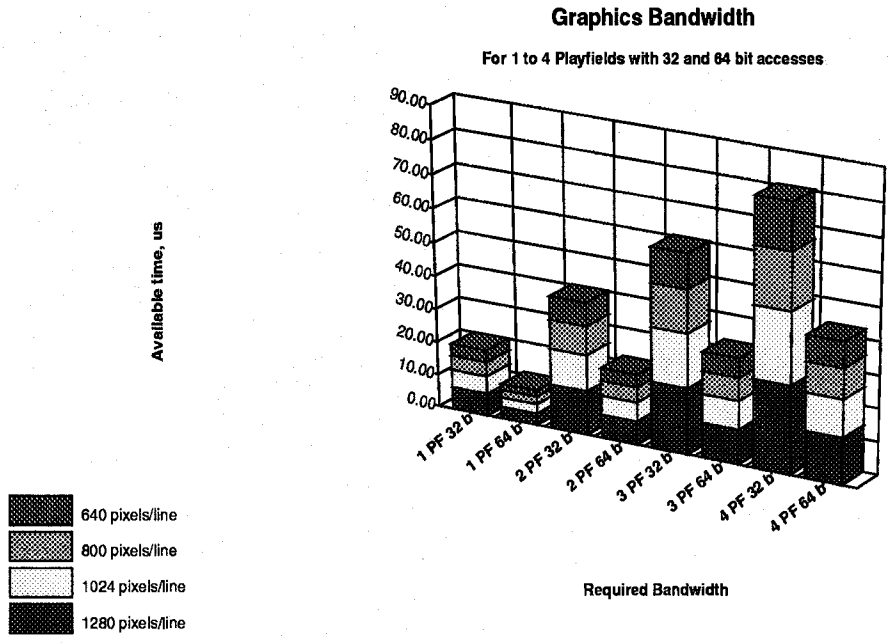
Table 3:

Screen Size in pixels	Pixel Rate Mhz	Pixel Period ns	Horizontal Line Period us
60 Hz Vertical Refresh			
640 x 480	25.125	39.722	31.78
800 x 600	40.000	25.000	26.40
1024 x 768	65.000	15.385	20.67
1280 x 1024	115.000	8,696	15.58
72 Hz Vertical Refresh			
640 x 480	31.500	31.746	26.41
800 x 600	50.000	20.000	20.80
1024 x 768[a]	75.000	13.333	17.71
1280 x 1024[b]	125.500	7.970	12.90

a. This is actually 70 Hz Vertical Refresh
b. Estimate

1.3.1.3 Supported Screen Resolutions

The data presented above may be plotted, as shown in Figure 1, to more easily determine which screen resolutions may be supported by each of the two memory configurations:



The vertical axis shows time in microseconds. The horizontal scan periods of the various screen sizes are plotted as horizontal lines at the appropriate scan period. Each horizontal line represents the available time for data fetches for that screen configuration.

The vertical bars represent data fetch activity for the number of play-fields indicated at the two memory configurations supported. The time required to fetch up to the eight possible sprites is also shown in the same format.

To determine if a particular screen configuration can be supported, one needs to determine if the vertical column representing the video data fetch activity falls under the line representing the time available for the screen configuration in question. If sprites are to be used, a column representing the appropriate number of sprites needs to be placed atop the data fetch column.

For example, a screen configuration of 1280x1024x72 with 3 play-fields and 3 sprites appears to be possible if the 64 bit memory configuration is used, while 1280x1024x72 with 2 play-fields and no sprites is not possible if the 32 bit memory configuration is used.

Note that this shows feasibility based on memory bandwidth only. Power considerations may cause some possible configurations to not be available in some package options.

1.3.2 Rendering Capabilities

Texture mapping capabilities

Shading capabilities

1.3.3 Audio Capabilities

1.4 Chip Set Features

This section presents an overview of the features of each of the primary blocks of each of the Hombre chips.

1.4.1 CPU chip

This chip will be implemented using structured custom techniques.

1.4.1.1 PA-RISC Core

- Implements PA-RISC 1.1 Architecture, Edition 3 at level 1.
- Supports Single issue integer instruction stream. Some floating point instructions supported in hardware.
- 8 (??) entry TLB.
- 4K (??) byte instruction cache. 2K (??) byte data cache.
- Internally has 64 bit datapath.
- Operates using 50 Mhz system clock for a (guess-timate) performance of 55 VAXMIPs.
- Augmented with Special Function Unit (SFU) instructions to support graphics.
- Graphics write buffer

1.4.1.2 Blitter

- 64 bit internal datapath.
- Uses burst mode to access display or system memory.
- Uses RMW cycles for destination to reduce memory activity.
- Performs traditional blits, line draw, shaded fills, and texture mapping.
- Performs functions on 8-bit and 16-bit pixels.
- Pixel addressed.

1.4.1.3 Display Memory Controller

- Configurable for 9, 10, or 11 bit multiplexed memory addresses.
- 120 ns memory cycle. (One word read: 120ns, Four word burst: 240ns)
- Burst across page boundary.
- Supports RMW cycles for individual byte writes.

- RAS decoding for 4 banks.
- VRAM access control.
- CAS before RAS refresh.

1.4.1.4 System Memory Controller

- Configurable for 9, 10, or 11 bit multiplexed memory addresses.
- 120 ns memory cycle. (One word read: 120ns, Four word burst: 240ns)
- Burst across page boundary.
- Supports individual byte write cycles.
- RAS decoding for 8 banks.
- CAS before RAS refresh.

1.4.1.5 DMA Controller

- Pointers for:
  4 Video sources
  16 Sprites (However, only 8 Sprite buffers may be supported initially)
  3 Blitter Sources
  12 Audio Sources
  Copper Source
  Memory Controller Refresh
  CD-ROM data and ECC
  MPEG source
- Understands 32 or 64 bit bus option.

1.4.1.6 CD-ROM interface

- Single and Double speed supported
- Error dectection in hardware, support for error correction.
- DMA access to data and error correction information

1.4.1.7 Audio Channels

- 12 AAA style, 16 bit, 44.1 Khz, CD quality channels.
- 2 CD formatted serial input lines. Each line is L+R for a total of 16 source channels.
- 1 CD formatted (L+R) serial output line.
- One of the serial input lines also serves as a sampling channel.

1.4.1.8 PCI Bus

- 32 bit, 25 Mhz option

1.4.1.9 Hitachi PA/50L Bus Interface

- Provides memory interface for PA/50.
- Provides PCI interface for PA/50.

1.4.1.10 Video synchronized Co-processor

- Executes lists of simple instructions including MOVE, JUMP, and WAIT.
- WAIT can stall for a specific video line or for a specified number of lines.
- One level of subroutine support provided
- MOVE can write to specific fullwords or to a block of contiguous double-words.

1.4.2 VIDEO chip

This chip will be implemented using structured custom techniques.

1.4.2.1 Video and Sprite Line buffer:

- Permits fetching during entire horizontal line time thereby increasing memory bandwidth.
- Horizontal scrolling of video data done on input side of line buffer.

1.4.2.2 Multiple ways to use the 4 video streams including:

- 1 to 4 individually scrolled 8-bit playfields.
- Any playfield may be HAM8.
- 3 combined to form 24 bit "true color".
- 3 combined to form 24 bit "true color" with 8 bit overlay.
- 1 or 2 16-bit playfields.
- 16-bit playfield with 1 or 2 8-bit playfields.

1.4.2.3 Sprites

- Architecture defines 16 sprites, 8 implemented in this chipset.
- Sprites are composed of 8 bit pixels.
- Sprites may be 8, 16, 32, 64, or 128 pixels in width.
- Sprites may be repeated or expanded in the vertical and horizontal directions.
- Each sprite has its own priority.

1.4.2.4 Color Lookup Table

- Dual 256x24 bit palletes
- Bit in sprite and video control registers determines which pallete to use.

1.4.2.5 Monitor Control

- Programmable registers for Sync., Blanking, and NTSC/PAL specific signals.
- Interlaced and Non-Interlaced modes
- Capable of producing Overscan displays
- Capable of HDTV formats

1.4.2.6 Joystick/Game Port Interface

- Support for IBM style analog joysticks

1.4.2.7 24 bit (8-8-8) RGB video input to support MPEG and Framegrabber devices.

- Permits external hardware MPEG decoder chip to easily share video DACs.
- Genlock capability permits merging of video input with computer generated graphics.

1.4.3 PERIPHERAL

This chip will be implemented using ASIC techniques.

1.4.3.1 Peripheral Component Interface (PCI)

- 32 bit, 25 Mhz option
- Bridges to other bus formats reside here (Zorro Bus, ISA Bus, etc.)

1.4.3.2 Floppy Disk Interface

- Support to 1/2/4/ Meg Floppies
- MFM/RLL encoding/decoding in hardware.
- Probably uses a commercially available MEGACELL.

1.4.3.3 Keyboard Interface

- Stripped down UART
- Support for IBM keyboards?

1.4.3.4 Parallel Port

1.4.3.5 UART

- 2 channels with modem control
- Probably uses a commercially available MEGACELL.

1.4.3.6 SCSI Interface

- SCS-II controller.
- Probably uses a commercially available MEGACELL.

1.5 Amiga Traditions...

While retaining many of the features which characterize an Amiga, this chipset departs from the object-code compatibility restraint of previous chipsets. A new, RISC based processor becomes the main engine of the system, with a core implementation of the processor resident in one of the custom chips.

Differences from the traditional / AAA Amiga and the new system include:

1.5.1 Video

- While the old Amiga encoded video data on a bit-by-bit (bitplane) basis, the new system will support 8-bit "chunky" pixels. 16-bit true color pixels are also supported.

- The old Amiga supported two independently scrollable playfields; the new system will support 4 independently scrollable playfields.
- Various modes of the 4 playfields will be supported including one where three of them represent R, G, and B for 24-bit true color with the fourth providing an 8-bit lookup overlay.
- Each playfield will have its own 8-bit priority.
- The actual display area will be defined by the blanking signal and window signals. Display_Data_Fetch has been eliminated..
- Data to be displayed will be fetched a line in advance of when it is needed. This provides higher bandwidth availability for fetching of data. Because of this and a wider data bus on the high-end configuration, higher display resolutions are supported. This is similar to the manner in which AAA fetches data.
- The original Amiga only allowed HAM images on a single playfield. HAM will be permitted on all four playfields.

1.5.2 Sprites

- Sprites are composed of chunky pixels (8-bits per pixel).
- In the old Amiga, sprite data was embedded into the sprite control stream; the new system will have a data pointer as part of the sprite control stream. The pointer will point to a canvas where sprite data will reside.
- Each sprite will have its own 8-bit priority.
- Sprites may be reused with no "dead line" between sprites.
- Sprite screen positioning is defined by a start position and size.
- Sprite collision detection has been removed.
- DMA pointers are provided for 16 sprites. Buffer space on the video chip will determine how many sprites may actually be used.

1.5.3 Copper

- Copper instructions are 64-bits in width.
- WAIT instructions can only wait relative to vertical video positions. No horizontal position waiting is provided.
- The position enable bits for wait matching have been eliminated.
- WAIT instructions may wait for specific vertical locations or for a specified number of vertical lines.
- A simple one-level subroutine mechanism is supported. No subroutines were available in previous Amiga's.
- Jump instructions contain their targets. In previous Amiga's, a jump was performed using two copper instructions: the first wrote a register with the target, the second initiated the jump.

1.5.4 Audio

- The old Amiga audio channels are replaced by AAA style Audio channels.

- Two CD formatted stereo serial inputs (L + R) are added. This adds 4 audio channels to the 12 computer generated channels for a total of 16 sources.
- One of the stereo serial inputs may be used to sample. 16 bit samples are stored via normal dma activity.
- Output is by a stereo serial channel only; no D/A converters are included on thechipset.
- Output serial channel may be looped back into the sampling serial input to capturecomposite audio data.

1.5.5 CD-ROM Interface:

- CD-ROM interface from CD-Game machine is included in the CPU custom chip.
- CD data is gathered and burst written into memory.

1.6 Target Technology

The chips described in this document have been specified such that they may be agressively implemented in the 0.8 micron technology that is currently available. However, given that these chips will not be produced for approximately 1 year, a more comfortable implementation technology would have the following characteristics:

- 0.6 micron CMOS technology
- 3 level metal with floating vias (level to level contacts which may, but need not be stacked).
- Support for high density, on-chip static ram. This may be achieved using in a number of ways including:

  a.) A second poly layer

  b.) A poly load arrangement with poly-substrate contacts
- Higher pin count packaging.

1.7 Target Costs

Even though the next generation system will have significantly more performance than the current generation chip set or the AAA chip set, it is important that the cost of the chips necessary to implement the lowest level system be competitive with the current costs. In order to achieve this, the combined cost of the CPU and VIDEO chips must be less than $40.

1.8 Development Strategy

The development of Hombre has been planned to allow the maximum amount of design parallelism possible. This is accomplished by the early definition of all system functions and development of software simulations of important hardware components.

All hardware elements will be captured and simulated using high level behavioral models (via Logic Lib and M models). These models will contain estimated timing information and will be kept current as later design details become available. Where appropriate, the behavioral models will be automatically transformed into layout. This will be accomplished via AutoLogic Synthesis and standard cell layout for random and control logic. Datapaths will be hand generated except where transformation via a datapath compiler is possible. It is expected that in the time frame of this project, most datapaths will be generated via compiler.

Software development will proceed in parallel with hardware development through the use of software emulators. These emulations differ from the traditional hardware simulations in that less emphasis is placed on how the operations are performed than on the fact that they get done and produce the same answer that the hardware will produce. An emulation of the processor chip is planned. Further speedup in emulation speed is possible through execution of the software on a PA-RISC workstation. Added graphics instructions will be emulated via emulation traps. This will allow software engineers to "run" large portions of code at hardware speeds.

Software tools are available for the PA-RISC.The GNU C compiler has been installed and the GNU assembler will be available. The assembler will be modified to include added instructions.

Figure xxx shows a rough schedule for the development of Hombre.

1.9 Current Status

Block diagrams of all chips have been generated. Interfaces to memory and between chips have been defined. Schematics of the CPU chip and Video chip have been started and various blocks have been simulated.

1.0 Interfaces

1.1 Pin Lists

The chip set consists of two custom chips and a system dependent ASIC. Table XXX shows the pin definitions of the three chips. Note that there are various packaging and bonding options to accomodate different product price points.

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
sys_clk	In	1	1	1	1	1	1	System Clock
sys_reset0	In	1	1	1	1	1	1	System Reset
pix_clk	In		1			1		Pixel Clock
uart_clk	In			1			1	Uart Clock
Section totals:		2	3	3	2	3	3	
s_ma1[10:0]	Out	11			11			System RAM Row/Col Addr
s_ras0[7:0]	Out	4			8			Per Bank RAS strobe
s_cas0[3:0]	Out	4			4			Per Byte CAS strobe
s_r1w0	Out	1			1			Read/Write Signal
s_buf_in1out0	Out	1			1			System RAM isolation buffer dir
s_buf_en0[1:0]	Out	1			2			System RAM isolation buffer en
Section totals:		22			27			
d_ma1[10:0]	Out	11			11			Display RAM Row/Col Addr
d_ras0[3:0]	Out	2			4			Per Bank RAS strobe
d_cas0	Out	1			1			CAS strobe
d_r1w0	Out	1			1			Read/Write Signal
d_dtoe0	Out	1			1			VRAM mode control
d_sc1	Out	1	1		1	1		VRAM shift clock
d_de0[3:0]	Out	2	2		4	4		VRAM shift enable
Section totals:		19	3		23	5		
ad[63:0]	In/Out	32	32		64	64		Multiplexed Address/Data Bus
ad_dle	Out-In	1	1		1	1		Data Latch Enable

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
ad_ale	Out-In	1	1		1	1		AD Address Latch Enable
ad_r1w0	Out-In	1	1		1	1		AD Read/Write Signal
gd[63:0]	In		32			64		Graphics Data Bus
vid_line	In-Out	1	1		1	1		Sync'd Line signal
Section totals:		36	68		68	132		
pa_ad[31:0]	In/Out				32			PA/50L Address Data
pa_adp[3:0]	In/Out				4			PA/50L Data Parity
pa_perr0	Out				1			PA/50L Parity Error
pa_byte_cntl[3:0]	In				4			PA/50L Byte Control
pa_cycle_start	In				1			PA/50L Cycle Start
pa_r1w0	In				1			PA/50L Read/Write
pa_rdy0	Out				1			PA/50L Ready
pa_sar0	In				1			PA/50L Same Address Region
pa_cycle_type[1:0]	In				2			PA/50L Cycle Type
pa_lock0	In				1			PA/50L Lock
pa_int[4:0]	Out				5			PA/50L Interrupts
Section totals:					53			
pci_ad[31:0]	In/Out	32		32	32		32	PCI Multiplexed Address/Data
pci_c1_be0[3:0]	In/Out	4		4	4		4	PCI Bus Cmd - Byte Enables
pci_par	In/Out	1		1	1		1	PCI Parity
pci_frame0	In/Out	1		1	1		1	PCI Cycle Frame
pci_trdy0	In/Out	1		1	1		1	PCI Target Ready
pci_irdy0	In/Out	1		1	1		1	PCI Initiator Ready
pci_stop0	In/Out	1		1	1		1	PCI Stop
pci_devsel0	In/Out	1		1	1			PCI Device Select
pci_id_sel	In/Out	1		1	1		1	PCI Initialization Device Select
pci_req0[5:0]	In	6		6	6		6	PCI requests to arbiter

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
pci_gnt[5:0]	Out	6		6	6		6	PCI grants from arbiter
pci_clk	Out	1		1	1		1	PCI clock
pci_rst	In/Out	1		1	1		1	PCI reset
Section totals:		57		57	57		57	
cd_data	In	1			1			CD-ROM drive data
cd_bclk	In	1			1			CD-ROM drive data clock
cd_lrck	In	1			1			CD-ROM left-right clock
cd_c2po	In	1			1			CD-ROM C2 error pointers
cd_intf	Out/In	6			6			
Section totals:		10			10			
aud_data_in[1:0]	In	2			2			Audio Serial Input Data
aud_lrck_in[1:0]	In	2			2			Audio Serial Left/Right Clock In
aud_bclk_in[1:0]	In	2			2			Audio Seriall Input Bclk
aud_data_out	Out	1			1			Audio Serial Output Data
aud_lrck_out	Out	1			1			Audio Serial Left/Right Clock Out
aud_bclk_out	Out	1			1			Audio Serial Output Bclk
Section totals:		9			9			
ad_rom_cs0	Out	1	1		1	1		Chip Select for AD rom
ad_video_cs0	Out-In	1	1		1	1		Chip Select for Video Chip
ad_other_cs0	Out	1			1			Chip Select for auxilliary chip
pa_rom_cs0	Out	1			1			Chip Select to PA/50 rom
ad_config_cs0	Out-In	1	1		1	1		Chip Select for Config Register
Section totals:		5	3		5	3		
vd_csync0	Out		1			1		Composite Sync
vd_vsync0	Out		1			1		Vertical Sync
vd_hsync0	Out		1			1		Horizontal Sync
vd_vblank0	Out		1			1		Vertical Blank
vd_color_burst0	Out		1			1		Color Burst

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
vd_lpen0	In		1			1		Light Pen
vd_clk_ext	In		1			1		External Clock Input
vd_clk_ext_en	In		1			1		External Clock Enable
vd_a_red	Out		1			1		Analog Red Video
vd_a_green	Out		1			1		Analog Green Video
vd_a_blue	Out		1			1		Analog Blue Video
vd_d_red	In/Out		8			8		Digital Red
vd_d_green	In/Out		8			8		Digital Green
vd_d_blue	In/Out		8			8		Digital Blue
Section totals:			35			35		
mouse_xmt	Out		1			1		Mouse transmit data (To)
mouse_rcv	In		1			1		Mouse receive data
mouse_rts	In		1			1		Mouse request to send
mouse_cts	Out		1			1		Mouse clear to send
mouse_dsr	In		1			1		Mouse data set ready
mouse_dtr	Out		1			1		Mouse data terminal ready
Section totals:			6			6		
joy0_fwd	In		1			1		Joy 0 Forward or Mouse V
joy0_back	In		1			1		Joy 0 Back or Mouse H
joy0_left	In		1			1		Joy 0 Left or Mouse VQ
joy0_right	In		1			1		Joy 0 Right or Mouse HQ
joy0_but_0	In		1			1		Joy 0 Pot X or Button 0
joy0_but_1	In		1			1		Joy 0 Pot Y or Button 1
joy0_but_2	In		1			1		Joy 0 Fire or Button 2
joy1_fwd	In		1			1		Joy1 Forward or Mouse V
joy1_back	In		1			1		Joy 1 Back or Mouse H
joy1_left	In		1			1		Joy 1 Left or Mouse VQ
joy1_right	In		1			1		Joy 1 Right or Mouse HQ

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
joy1_but_0	In		1			1		Joy 1 Pot X or Button 0
joy1_but_1	In		1			1		Joy 1 Pot Y or Button 1
joy1_but_2	In		1			1		Joy 1 Fire or Button 2
Section totals:			14			14		
scsi_ack0	In			1			1	SCSI handshake from initiator
scsi_req0	In			1			1	SCSI handshake from target
scsi_bsy0	In			1			1	SCSI busy
scsi_sel0	In			1			1	SCSI select device
scsi_i_o0	In			1			1	SCSI input/output
scsi_c_d0	In			1			1	SCSI command/data
scsi_msg0	In			1			1	SCSI message
scsi_atn0	In			1			1	SCSI attention
scsi_rst0	In			1			1	SCSI bus reset
scsi_d1[7:0]	In/Out			8			8	SCSI data
scsi_dbp0	In/Out			1			1	SCSI parity
Section totals:				18			18	
fd_rd0	In			1			1	Floppy read data
fd_wd0	Out			1			1	Floppy write data
fd_we0	Out			1			1	Floppy write enable
fd_rdy0	In			1			1	Floppy ready
fd_mtr0	Out			1			1	Floppy motor control
fd_sel0[1:0]	Out			2			2	Floppy select lines
fd_chng0	In			1			1	Floppy changed
fd_side0	Out			1			1	Floppy side select
fd_wprot0	In			1			1	Floppy write protect
fd_trk00	In			1			1	Floppy track 0 detect
fd_step0	Out			1			1	Floppy motor step control
fd_dir	Out			1			1	Floppy motor direction

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
fd_index0	In			1			1	Floppy index detect
fd_eject0	Out			1			1	Floppy eject disk (future)
Section totals:				15			15	
kb_clk	Out			1			1	Serial Keyboard clock
kb_data	In			1			1	Serial Keyboard data
kb_reset0	In						1	Serial Keyboard reset
Section totals:				3			3	
prll_d1[7:0]	In/Out			8			8	Parallel Port data
prll_ack0	In			1			1	Parallel Port acknowledge
prll_busy	Out			1			1	Parallel Port busy
prll_pout	Out			1			1	Parallel Port direction
prll_sel	In			1			1	Parallel Port select
prll_strb	In			1			1	Parallel Port strobe
Section totals:				13			13	
uart_rxd[1:0]	In			2			2	Receive data
uart_txd[1:0]	Out			2			2	Transmit data
uart_rts0[1:0]	Out			2			2	Request to send
uart_cts0[1:0]	In			2			2	Clear to send
uart_dsr0[1:0]	In			2			2	Data set ready
uart_cd0[1:0]	In			2			2	Carrier detect
uart_dtr0[1:0]	Out			2			2	Data terminal ready
uart_ri0[1:0]	In			2			2	Ring Indicator
Section totals:				16			16	
jtag_tclk	In	1	1	1	1	1	1	JTAG Clock
jtag_tms	In	1	1	1	1	1	1	JTAG Test Mode Select
jtag_ki	In	1	1	1	1	1	1	JTAG Data Input
jtag_do	Out	1	1	1	1	1	1	JTAG Data Output
Section totals:		4	4	4	4	4	4	

Table 1:

		Low Cost Package			Full Featured Package			Description
Name	Type	CPU	VID	PER	CPU	VID	PER	
power	vdd	10	9	10	20	20	10	
ground	vss	20	15	20	30	30	20	
Section totals:		30	24	30	50	50	30	
Chip totals:		194	160	159	304	302	159	

1.2 Clocks

The system clock will operate at 50 Mhz. All circuits in the CPU chip will be synchronous to this single clock. A separate pixel clock will be provided for the video chip. The rate of the pixel clock is a function of the screen resolution being supported. All transitions between pixel clock driven circuitry and circuitry driven by the system clock will be double flip-flop buffered.

1.3 Address Ranges

Table XX shows the address ranges as viewed by the three sources of memory

Table 2:

Area	Range	Size
With respect to Internal PA-RISC Processor and DMA Devices		
AD-ROM	0xf0000000 - 0xf1ffffff	32 Mbytes
Internal Registers	0xf2000000 - 0xf27fffff	8 Mbytes
Video Registers	0xf2800000 - 0xf2ffffff	8 Mbytes
Other AD Registers	0xf3000000 - 0xf3ffffff	16 Mbytes
Display VRAM	0xf4000000 - 0xfbffffff	128 Mbytes
PCI	0xfc000000 - 0xffffffff	64 Mbytes
System DRAM	0x000000 - 0xefffffff	4026 Mbytes
With respect to External PA-RISC Processor		
PA-ROM	0xf000000 - 0xf1ffffff	32 Mbytes
Display VRAM	0xf4000000 - 0xfbffffff	128 Mbytes
PCI	0xfc000000 - 0xffffffff	64 Mbytes
System DRAM	0x00000000 - 0xeffffff	4026 Mbytes
From the viewpoint of PCI Resident Objects		
Display VRAM		
System DRAM		
Internal Registers		
Video Registers		
Other AD Registers		

access. Both PA-RISC processors access location 0xf0000004 for the first instrcution after reset. Note that although the locations have the same numerical value they are decoded into their own individual ROMs.

1.4 Display VRAM Interface

Figure XXX shows the timing diagram for Display memory access. Read accesses always receive the entire 32/64 bit width of memory. Writes to individual bytes are supported via read-modify-write cycles.

The VRAM design requires the use of 60ns VRAMs which support page mode burst accesses and have a 20ns serial access.

Refresh is supported via a CAS-before-RAS access cycle.

1.5 System DRAM Interface

Figure XXX shows the timing diagram for System memory access. Read accesses always receive the entire 32 bit width of memory. Writes to individual bytes are supported via CAS-per-byte signals.

The DRAM design requires the use of 60ns DRAMs which support page mode burst accesses.

Refresh is supported via a CAS-before-RAS access cycle.

1.6 AD Bus Interface

The AD bus is a multiplexed address/data bus which interconnects the two custom chips and memory. Figure XXX shows the timing diagram for the AD bus. It has a synchronous protocol.

1.7 PCI Bus Interface

The PCI interface conforms to the Revision 1.0 Specification, June 22, 1992.

This needs to be updated to the new spec...

1.8 Audio Input/Output Format

Figure XXX shows the format of the serial audio inputs and output.

1.9 Video Request Format