OAK TECHNOLOGY

Multimedia Solutions in Silicon™

SPITFIRE

64-bit
Multimedia GUI Accelerator

OTI-64107/64105
Preliminary Specification

September 1994

Oak Technology, Inc. • 139 Kifer Court • Sunnyvale, CA 94086 •Phone (408) 737-0888 • FAX (408) 737-3838

TABLE OF CONTENTS

CHAPTER 1: OVERVIEW

1.1 FEATURES

- True 64-bit Architecture
- Screen resolutions:
  - 1280 x 1024, 256 colors @ 75Hz Non-interlaced
  - 1024 x 768, 16M colors @ 60Hz Non-interlaced
- 0.6-m CMOS technology
- 240-pin PQFP (0.5 mm lead pitch)

TRUE MULTIMEDIA PORT (OTI-64107 only)

- 16-bit data path
- Shared DRAM frame buffer for graphics and video
- 33MHz transfer rate, 66Mbytes/sec data bandwidth
- Video Masking (using a standard inexpensive RAMDAC) allows:
  - Text, Graphics, or Animation over video
  - Live video does not freeze under a pull-down menu
  - Arbitrarily shaped video windows
  - Multiple video windows
- Supports chroma keying with a special RAMDAC
- Hardware cursor on video
- Scaling: x 2, x 4, x 8
- Glueless live video support for the most popular video chips
- I^{2}C interface built-in
- Modular upgradability, live video can be added as an option

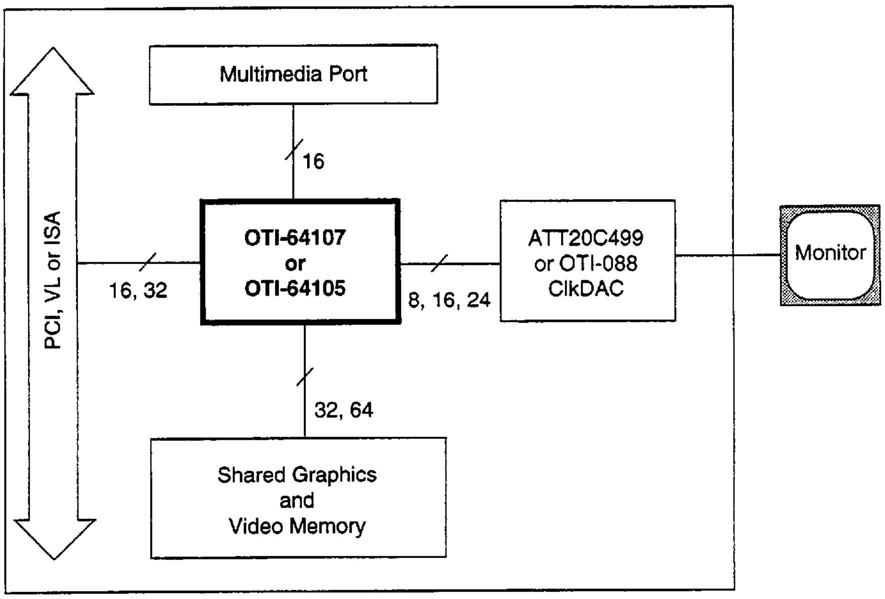
(Features continued on next page)

1.2 GENERAL DESCRIPTION

The Spitfire™, OTI-64107 and OTI-64105 are pin-compatible, high-performance 64-bit DRAM GUI Accelerators. The OTI-64107 also provides integrated multimedia support with a 16-bit interface and a shared display memory architecture. Spitfire's unique architecture allows the simultaneous display of multiple, overlapping video and graphics windows.

Both chips interface directly to the PCI bus as well as to the VL and ISA buses, with no external logic. Spitfire's 64-bit drawing engine and 64-bit DRAM interface provides high performance at a cost-effective price. Additionally, Spitfire supports Windows NT quaternary ROP's and Windows 3.x ternary ROP's providing dramatic performance improvement by executing typical software driver functions in hardware. On the PCI bus, Spitfire supports direct burst read from system memory and direct burst write to system memory for faster memory to screen and screen to memory transfers.

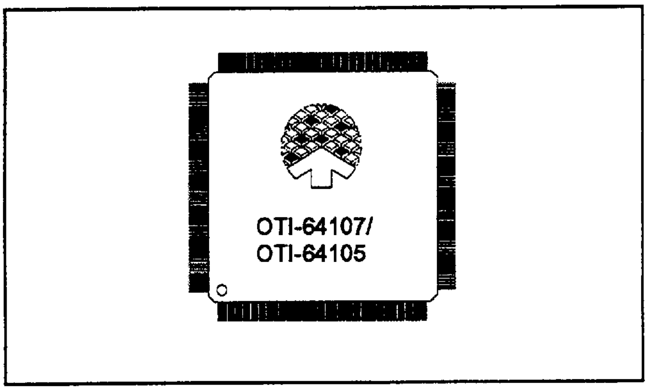
Combined with the OTI-088 Clock/DAC (24-bit pixel port) and DRAM, the Spitfire provides a complete, high-performance graphics solution with multimedia support. Using the OTI-64105, manufacturers can implement a cost effective GUI accelerator which can later be upgraded to add multimedia support by replacing the OTI-64105 with the OTI-64107 without the need for a board redesign.



Spitfire System Block Diagram

FEATURES (Cont.)

- High-Performance Drawing Engine:
  - Bitblt engine with color expansion/conversion and chipping
  - Supports four independent bitmaps (source, destination, pattern, mask)
  - Supports Windows NT quaternary ROPs and Windows 3.1 ternary ROPs
  - Line drawing, Area Fills, and CPU assisted drawing mode
  - 1, 8, 16, 32 bits/pixel
- Hardware Cursor:
  - 64 x 64 x 2 bits/pixel, at 16M colors
- Frequency of operation:
  - Pclk=110MHz
  - Mclk=66MHz
- Display Memory
  - Typical 2Mbytes of DRAM
  - Up to 8Mbytes
  - Supports 256K x 16, 512K x 8, 1M x 4, 256K x 8, and 256K x 4 DRAMS
  - 32/64-bit display memory data bus
  - Programmable display memory timing
- ISA/VL/PCI buses supported
  - Write buffer
  - Memory mapped I/O
- 4Gbytes memory addressing capability
- Glueless 32-bit VL-bus interface
- PCI bus:
  - Glueless 32-bit PCI bus interface
  - Master Mode support
  - Direct burst transfer to/from system memory
  - 4Gbytes memory addressing capability
- Output pixel port:
  - 8/16/24-bit
  - Directly inputs to OTI-088 24-bit ClkDAC
  - Allows multiple pixel packing
  - Supports 1280 x 1024 x 256 colors at 75Hz (135-MHz clock) by multiple pixel packing out of the OTI-64107
- EEPROM support for switchless implementation
- Fully integrated Feature Connector support, compliant with VESA VAFC Proposal 1.0 p



Spitfire Actual Package Size

CHAPTER 2: INTERFACE DESCRIPTION

This section describes the interfaces of the OTI-64105/107 to the other components of the graphics subsystem. The Display Memory interface and the interface to the Multimedia Port are described in subsequent sections. The OTI-64105 and the OTI-64107 are pin-compatible parts, except the OTI-64105 does not have the Multimedia Port. The Multimedia Port pins are No-Connects for the OTI-64105. The OTI-64105/107 Interface Diagram is shown below:

In the rest of this document, these two pin-compatible parts are referred to as the OTI-64107 or 64107.

The following interfaces have been described in the following sections:

2.1 System bus interface
2.2 DAC interface
2.3 Clock interface
2.4 ROM BIOS interface
2.5 Feature connector interface
2.6 EEPROM/Dipswitch interface

2.1 System Bus Interface

The OTI-64107 can be configured to interface directly to the three standard system buses in use today: VL, PCI and ISA. The chip configures itself to interface to a particular bus during hardware reset through the Hardware Configuration Register 1 (reg 3DF.07). The VL interface is compliant to the VESA VL-Bus Specification Version 2.0p, Revision 0.93p, dated 9/23/1993. The PCI interface is compliant to the PCI Local Bus Specification Revision 2.0, dated 4/30/1993. The ISA interface is compatible to the IEEE P996 standard for 8-bit and 16-bit ISA bus.

2.1.1 VL-Bus

The OTI-64107 supports both I/O and memory cycles on the VL-bus up to 50MHz. I/O cycles are minimally one wait state and programmable up to five wait states. Memory write cycles are designed to run at 0-2 wait states on cache hit cycles, depending on the speed of the bus. Strictly following the VL-bus specification of 4 ns setup time for address and data would force the controller to run at a minimum of one wait. Memory mapped I/O write cycles are minimally one wait state and programmable up to five wait states, while memory mapped I/O read cycles are minimally two wait states and programmable up to six wait states. As a general rule of thumb, the memory write cycles are zero wait state at a 25MHz bus speed, and one wait state at 33-50MHz bus speeds.

32-bit I/O is supported for Extended System Interface registers (2xxx). Standard VGA and Oak Extended registers at 3DE/3DF can only be supported with word or byte cycles. DAC and Auxiliary registers can only be supported with byte cycle. BIOS and drivers must take care of this.

OTI-64107 does not support Bus Mastering or Burst mode on the VL-bus.

See VESA VL-Bus Specification for more operational information on the VL-bus.

2.1.2 PCI Bus

The 32-bit PCI bus is supported up to 33MHz. I/O and Configuration cycles are minimally one wait state and programmable up to five wait states. Memory cycles, both read and write, are minimally one wait state and maximally three wait states. Memory mapped I/O write cycles are minimally one wait state and programmable up to five wait states, while memory mapped I/O read cycles are minimally two wait states and programmable up to six wait states.

32-bit I/O is supported for Extended System Interface registers (2xxx). Standard VGA and Oak Extended registers at 3DE/3DF are only be supported with word or byte cycles. DAC and Auxiliary registers are only supported with byte cycles.

Bus mastering and burst mode are supported for the PCI bus. However, physical memory address must be used in order to use master mode. Parity generation is supported but there is no parity checking.

See PCI Local Bus Specification (Rev 2.0) for more details.

2.1.3 ISA Bus

The OTI-64107 provides zero wait state memory operations and 1 wait I/O on ISA bus up to 12.5MHz. Faster bus systems should disable the zero wait state feature. DAC and Auxiliary registers can only be supported with byte cycle, IO16n will not be generated for these cycles.

Bus Mastering is not supported.

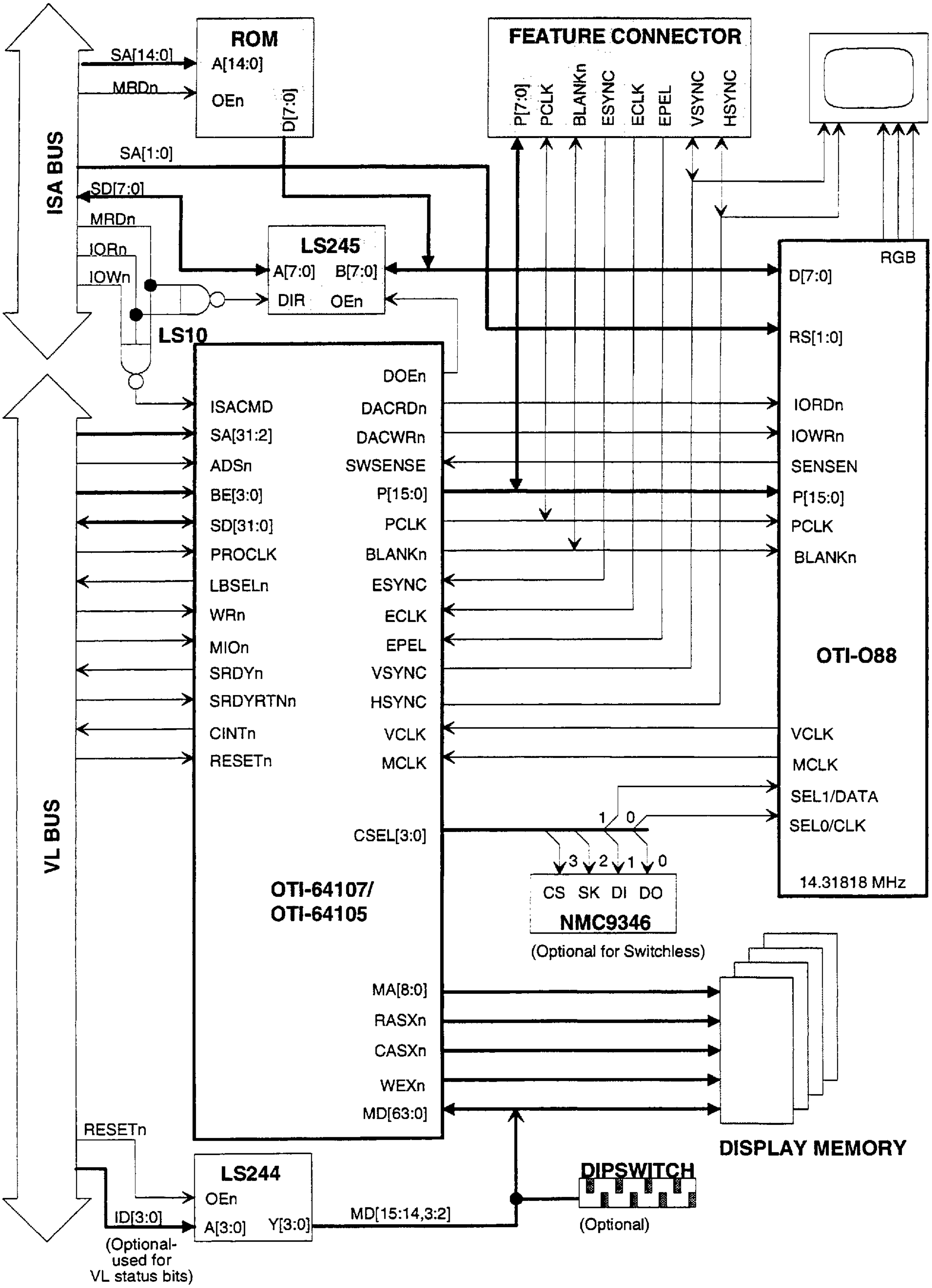


Figure 2.1 - VL Bus Block Diagram using OTI-088 ClkDAC

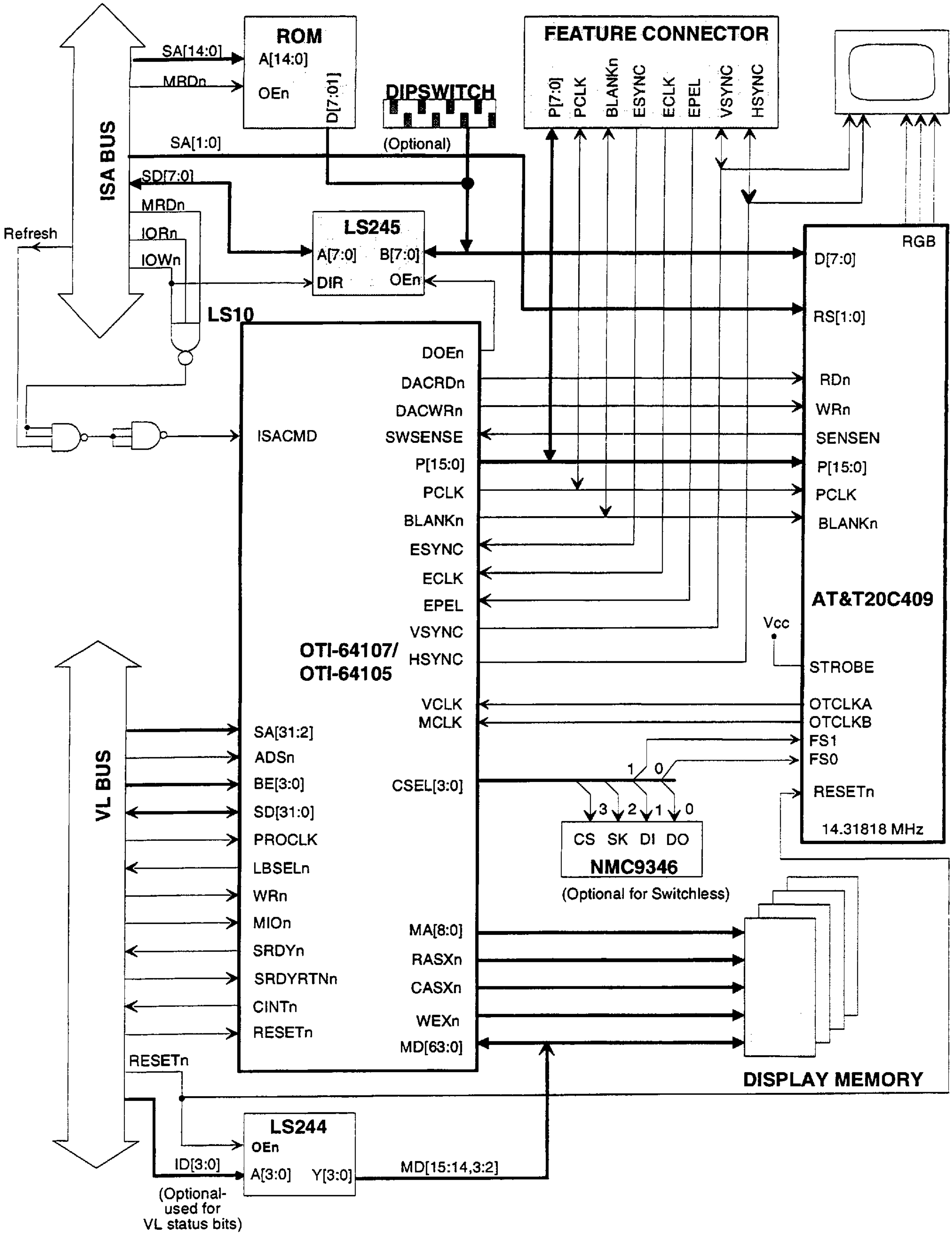


Figure 2.2 - VL Bus Block Diagram using AT&T20C409 Precision DAC™

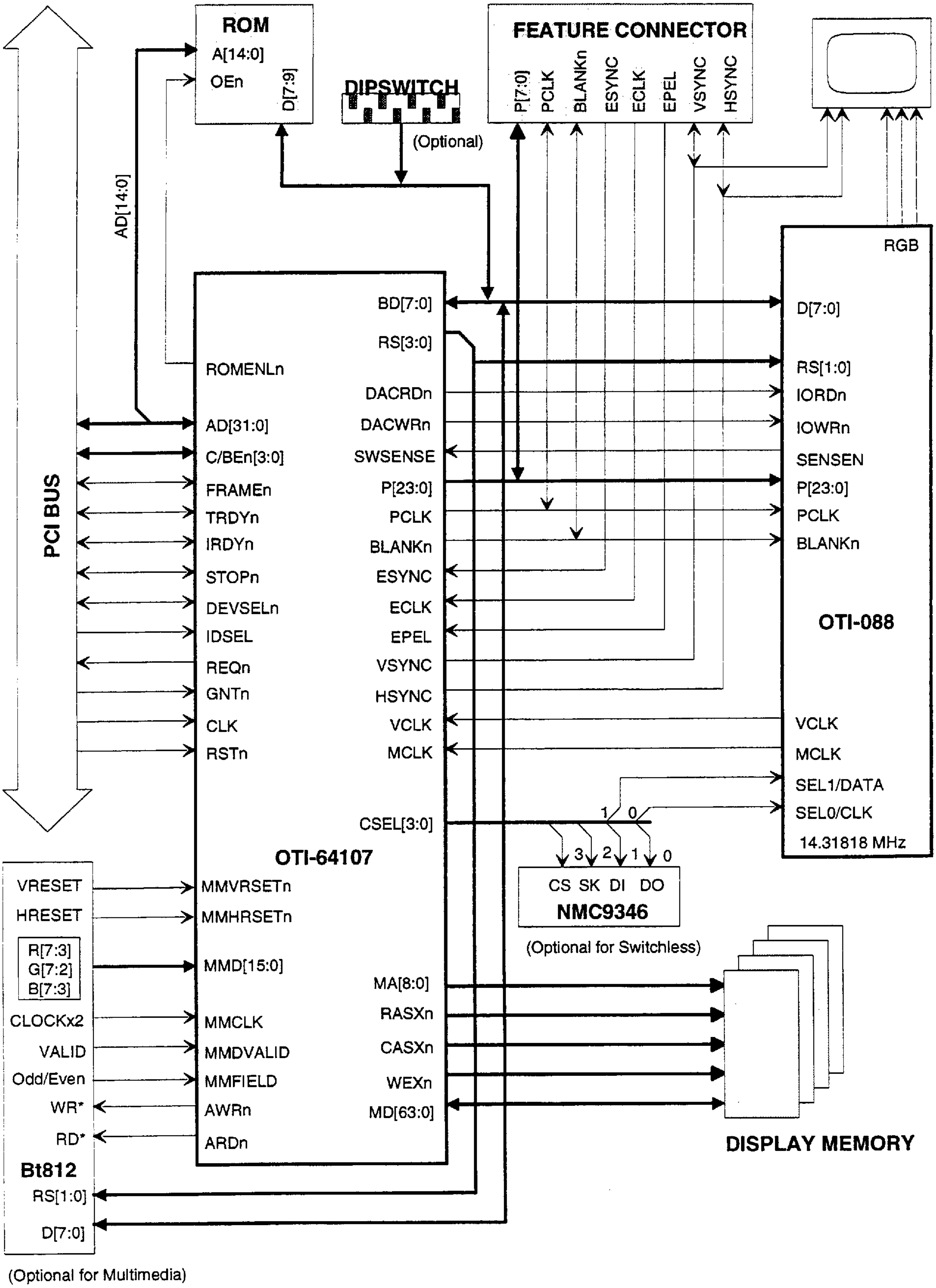


Figure 2.3 - PCI Bus Block Diagram (Glueless) with the Multimedia Port

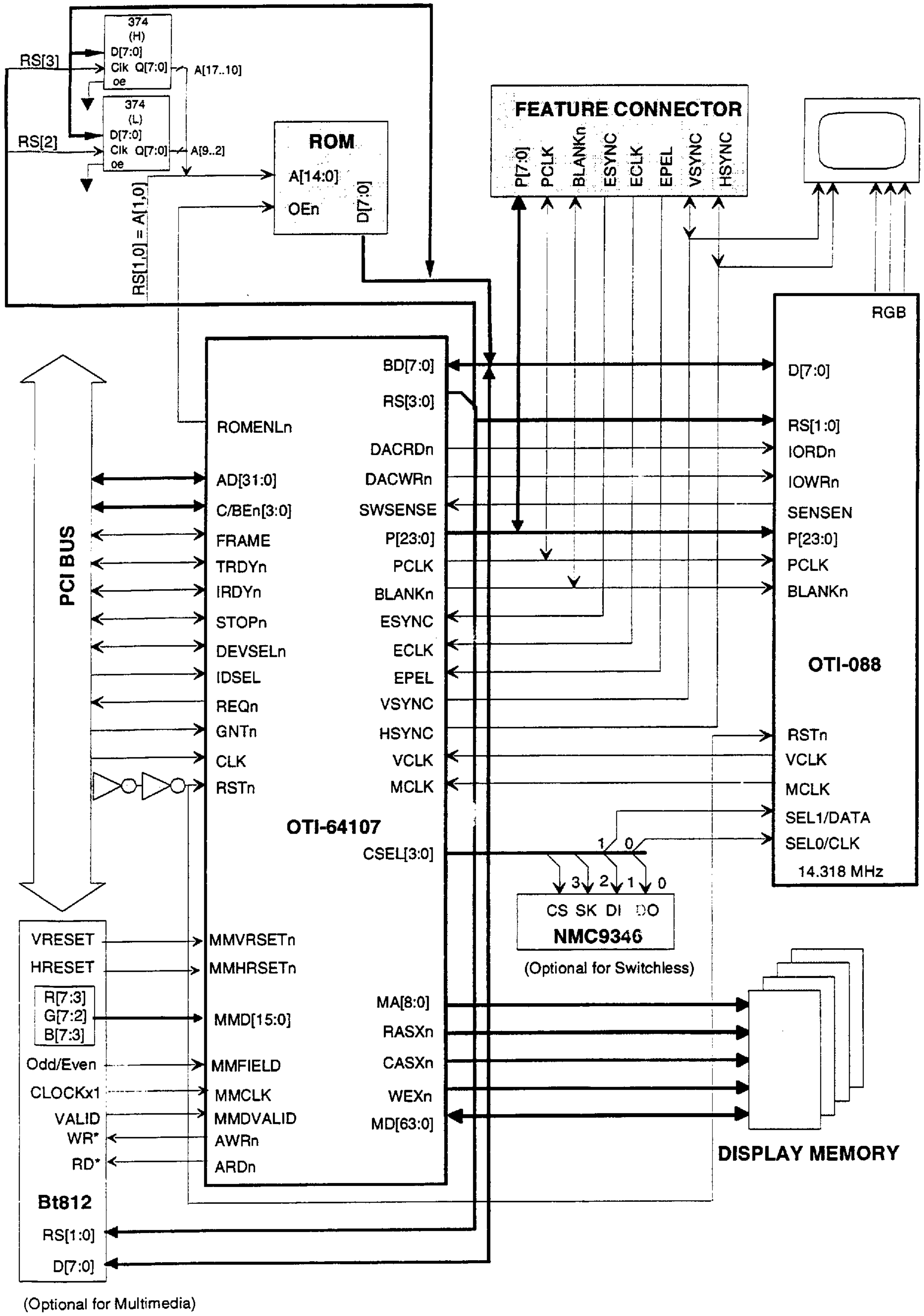


Figure 2.4 - PCI Bus Block Diagram (compliant) with the Multimedia Port

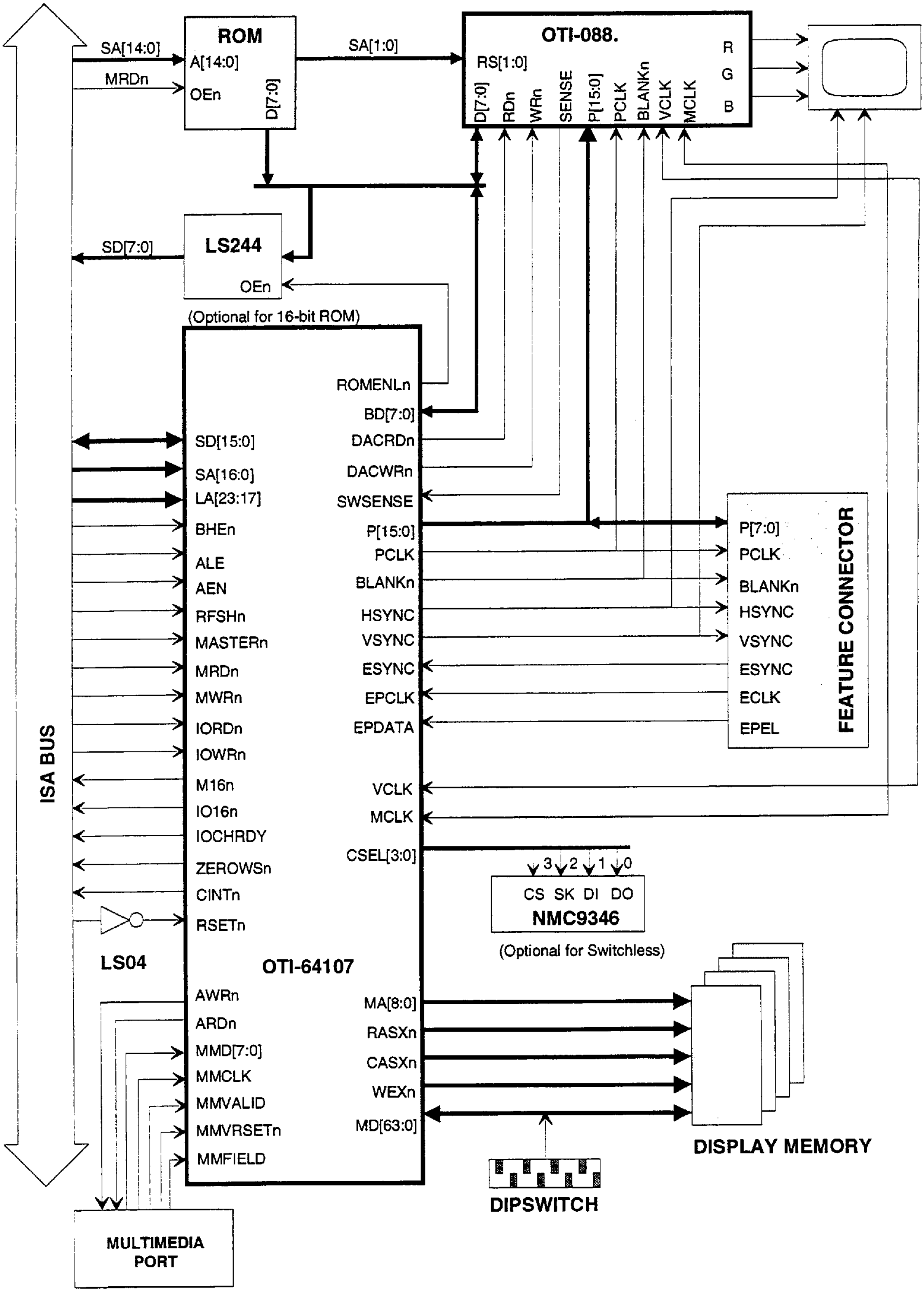


Figure 2.5 - ISA Bus Block Diagram

2.2 DAC Interface

DAC I/O only supports 8-bit transfers. For ISA or PCI bus implementation, DAC system data is routed through BD bus to the system bus. For VL bus implementation, DAC system data is routed to the ISA bus through external buffers. OTI-64107 will always decode address and status/command to generate DACRDn and DACWRn for the DAC I/O commands. DAC address space by default is 3C6-3C9. Extended DAC I/O space can be supported from 2x80-2x9F. The 'x' is programmable and defaults to 1. See system block diagrams for more information.

The OTI-64107 can be configured to be either 8-, 16-, or 24-bit pixel bus support. Hi-color and true-color support in 8-bit configuration is the same as 87X (double or triple frequency). Hi-color support in 16-bit pixel bus will allow the controller to run at regular frequency instead of doubling. True-color support in 16-bit pixel bus will require doubling frequency (with one byte wasted or with packed format), but not tripling the frequency. For high resolution (1280 x 1024) 256 color modes, there is an option to send two 8-bit pixels out at a time and consequently the pixel clock can be half the regular RAMDAC frequency.

24-bit pixel bus is supported for true color modes only. Due to pin limitations on the VL bus, pins P[23:16] are available only when the OTI-64107 is on either ISA or PCI buses. 24-bit pixel bus allows the OTI-64107 to support true color at higher resolution without having to double or triple the pixel clock frequency. See Pixel Interface register for more information on various modes supported at different pixel bus widths.

2.3 Clock Interface

Up to 16 external video clock frequencies can be selected by four programmable clock select output pins. Video clock frequencies can be supported up to 110MHz. These programmable pins can also be used as clock and data pins for serially programmable clock sources, allowing the OTI-64107 to support both VESA and conventional video frequencies without any hardware switches.

Memory clock is selectable through either software if the clock chip is programmable, or by hardware through jumpers if the clock chip has fixed frequencies. Memory clock can be supported up to 66MHz.

The dual clock synthesizers of the OTI-088 SynDAC have been designed for a glueless and flexible interface for the OTI-64107. The OTI-088 has two programmable pins which are used for clock and serial data inputs. The OTI-64107 sends out a 16-bit serial data stream to program each of the Video Clock and Memory Clock registers in the OTI-088. The register description of the Video Clock Select register (3DF index 6) explains how the OTI-64107 programs the OTI-088. Also refer to the OTI-088 data sheet for more details on the clock interface.

The OTI-64107 can also be used with the AT&T *Precision*DAC's AT&T20C409 (16-bit pixel port for VL bus) and AT&T20C499 (24-bit pixel port for PCI bus). The *Precision*DAC's are programmed over the parallel 8-bit data bus. Please refer to the appropriate datasheet for more details on the clock interface.

2.4 ROM BIOS Interface

For ISA bus configuration, the OTI-64107 can support either one (8-bit BIOS) or two ROMs (16-bit BIOS). The ROM data is routed back to the ISA bus through the BD bus and the external buffers if two ROM's are used.

For VL bus configuration, only a single ROM (8-bit BIOS) can be used. The BIOS ROM can only be on the ISA bus. ROM data is routed back to the ISA bus through an external buffer.

For PCI bus configuration, only a single ROM (8-bit BIOS) can be supported. The ROM address is connected to AD[14:0], while the data is connected to BD[7:0]. During ROM read cycles, the OTI-64107 latches the lower system address AD[14:0] to generate internal RA[14:0] and drives them out through AD[14:0], generates ROMENLn as a ROM output enable signal, and toggles the RA[1:0] appropriately depending on whether the current cycle is a byte, a word or a doubleword. If it is a word or a double word, the OTI-64107 toggles the RA[1:0] and latches in the ROM data through AD[7:0] bus, aligns the data, and sends it back out to AD[31:0]. Wait states are automatically asserted for the ROM access, assuming MCLK is 66MHz and the slowest ROM speed is 120ns.

ROM BIOS address space is assumed to be at C0000h. ROM BIOS support as described above can be enabled or disabled through the Hardware Configuration Register 2.

2.5 Feature Connector Interface

The OTI-64107 supports the feature connector in all configurations. The OTI-64107 feature connector support is compliant to the VESA Standard VGA Pass-Through Connector (VSVPC), and is compliant to the VESA Advanced Feature Connector (VAFC) with an appropriate DAC/Synthesizer like the OTI-088. For VAFC, base line compliance is readily achieved with OTI-64107 and any 16-bit pixel port DAC, with clock doubling modes, and 5:6:5 RGB format, such as the OTI-088. To be VAFC compliant in extended modes, a more advanced DAC such as the Bt885 is needed.

Pins EPDATA, EPCLK, and ESYNC are inputs to the chip to enable/disable P[23:0], PCLK, and BLANKn/HSYNC/ VSYNC respectively. Pins HSYNC & VSYNC have AC timing requirements with respect to PCLK to meet the VAFC specification.

Although PCLK can operate up to 110MHz, it is not meant to drive the feature connector at this speed. Feature connector operation should be limited to 40MHz or less.

Pin GRDY is currently not generated from the OTI-64107, but with appropriate timing, BLANKn can be used to generate GRDY for the feature connector.

2.6 EEPROM/Dipswitch Interface

A small EEPROM (1024x1) can be supported to enable the add-on card or the graphics subsystem to be switchless. Because the control pins for the EEPROM interface are muxed with clock select pins, the EEPROM should be programmed or read during POST only. See EEPROM Control Register (3DF index E) description for more details. For cost saving purposes, an 8-bit dipswitch can be supported in place of the EEPROM. The dipswitch support does not need external buffers. Dipswitch can be read at anytime through extended register 3DF index D.

CHAPTER 3: DISPLAY MEMORY INTERFACE

The OTI-64107 supports a wide variety of DRAM types. It supports 64K x 16, 256K x 4, 256K x 8, 256K x 16, 512K x 8, and 1M x 4 DRAMs. The OTI-64107 provides all the necessary control signals, address and data lines to access the display memory in fast page mode.

The wide variety of support for different DRAM types has been provided to take care of the contingency if availabilities of some DRAMs become scarce. The maximum display buffer size is 8Mbytes when used with 1M x 4 DRAMs, 4M bytes when used with 512K x 8 or symmetric 256K x 16 DRAMs, 2M when used with 64K x 16, 256K x 4 or asymmetric 256K x 16 DRAMs. The minimum configuration is 256Kbytes when used with 64K x 16 DRAM, 1Mbyte when used with 256K x XX or 512K x 8 DRAM, and 2Mbytes when used with 1M x 4 DRAM. See the table below for details on the different DRAMs supported.

Support for 256K x 16 includes the 10-bit row address, 8-bit column address type as well as the 9-bit row, 9-bit column address type. There is an option to convert WExn signals to CASxn signals, and vice versa, to support the myriad types of DRAMs available today.

Memory cycles can be programmed to match the different types of DRAM. The RAS precharge and pulse, CAS pulse width, and RAS-to-CAS delay are all programmable. Matching memory cycles with a programmable memory clock would virtually guarantee the most efficient memory interface for a given memory type. Memory clock can be supported up to 66MHz. This can drive a 45ns fast page mode DRAM.

The display buffer can be addressed through either a programmable linear address range above 1M up to 4G on VL and PCI buses, and up to 16M on ISA bus, or through the conventional graphics address range (A0000 to BFFFF) with the segment registers. Depending on the type of memory used, and the amount of memory installed, the chip can be configured to have either 32-, or 64-bit memory data bus. See memory configuration block diagrams for more details.

DRAM Types Supported

	(for 64 bit data bus)		Memory Size		Number of Control Signal Lines		
DRAM Type (address bits) Control bits	Memory Size	# of Memory Chips	Maximum Memory Size	Minimum Memory Size	RAS	CAS	WE
256K x 16 (9 RAS, 9 CA 1 CAS, 2 WE 2 CAS, 1 WE	2Mbytes	4	4Mbytes	1Mbyte	2 2	1 8	8 1
256K x 16 (10 RAS, 8 CAS) 1 CAS, 2 WE 2 CAS, 1 WE	2Mbytes	4	2Mbytes	1Mbyte	1 1	1 8	8 1
64K x 16 1 CAS, 2 WE 2 CAS, 1 WE	256Kbytes	4	2Mbytes	256Kbytes	2 2	2 8	8 2
256K x 4	2Mbytes	16	2Mbytes	1Mbyte	1	1	8
256K x 8	2Mbytes	8	4Mbytes	1Mbyte	1	8	1
512K x 8	4Mbytes	8	4Mbytes	2Mbytes	2	1	8
1M x 4	8Mbytes	16	8Mbytes	4Mbytes	1	1	8

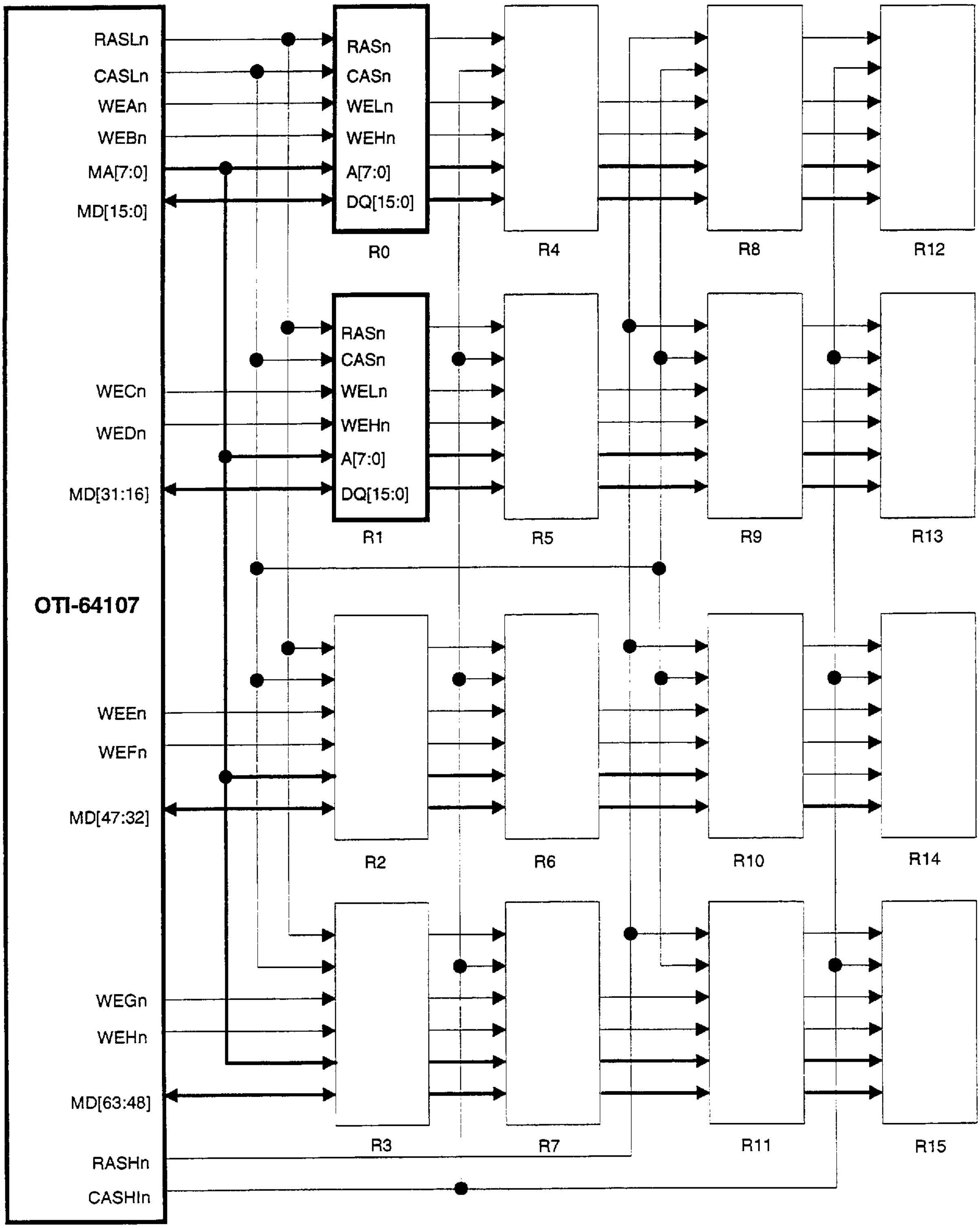


Figure 3.1 - 64K x 16 DRAMs (1 CASn, 2 WEn), MD32 & 64, 256K - 2Mbytes

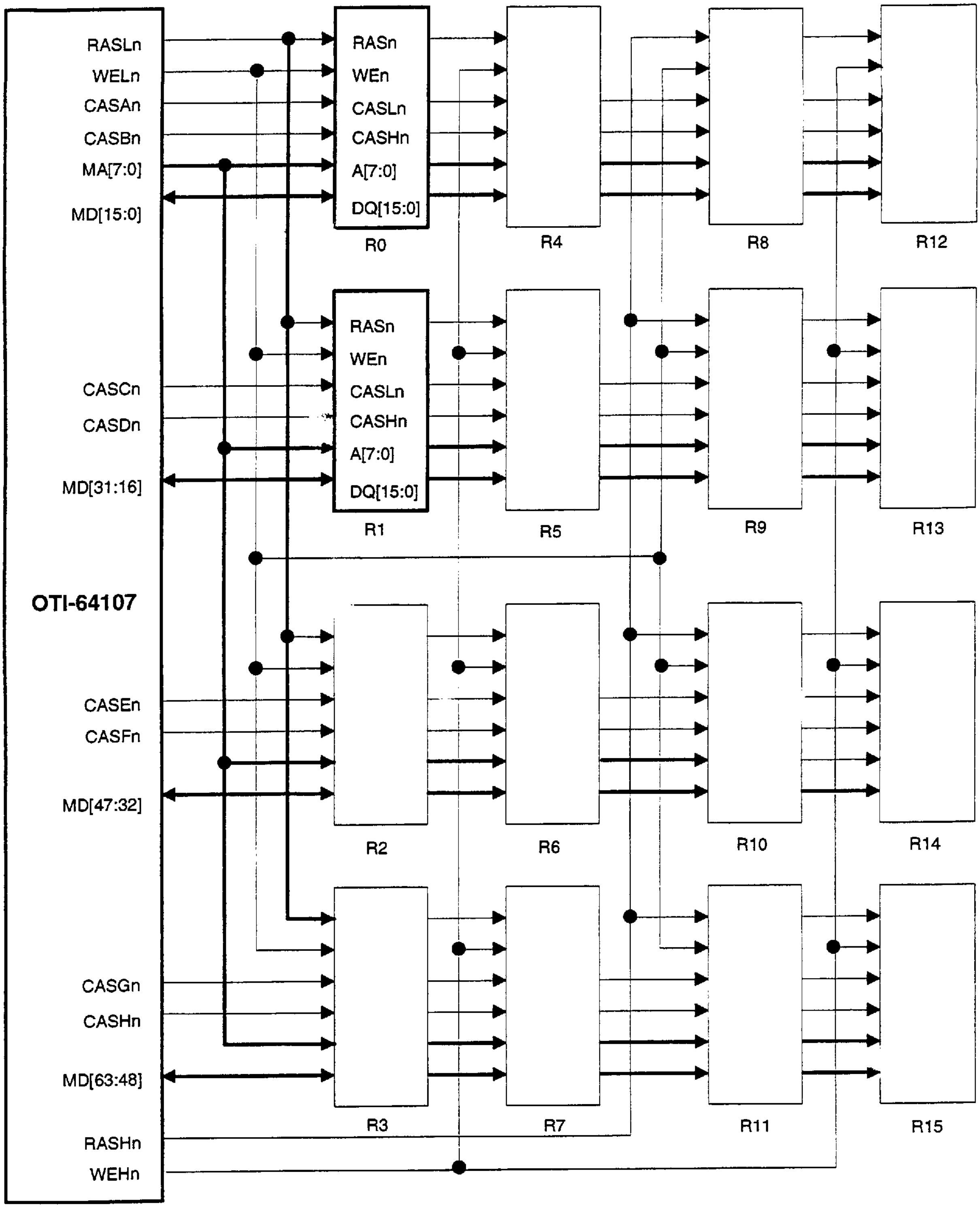


Figure 3.2 - 64K x 16 DRAMs (2 CASn, 1 WEn), MD32 & 64, 256K - 2Mbytes

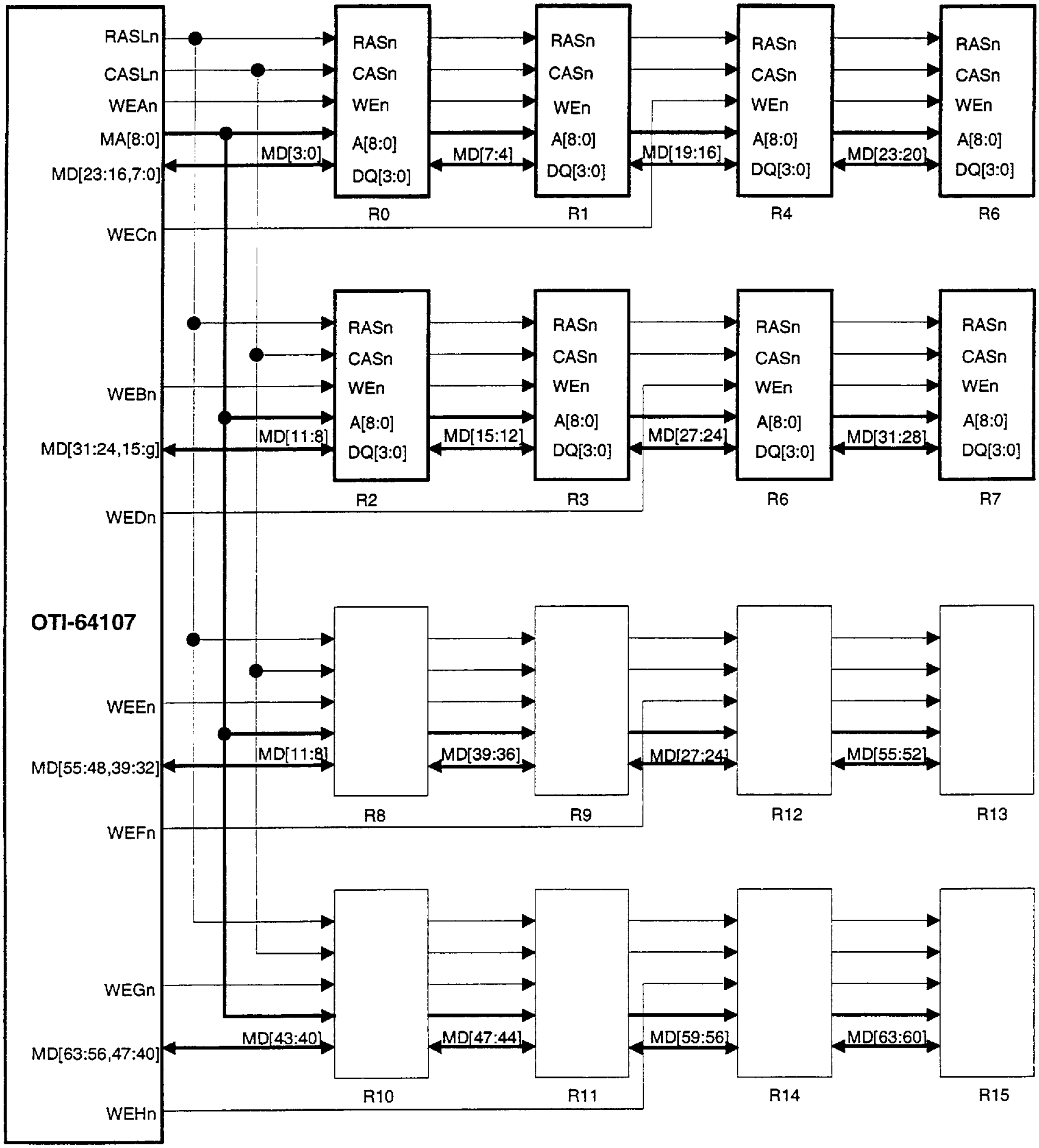


Figure 3.3 - 256K x 4 DRAMs, MD32 & 64, 1M - 2Mbytes

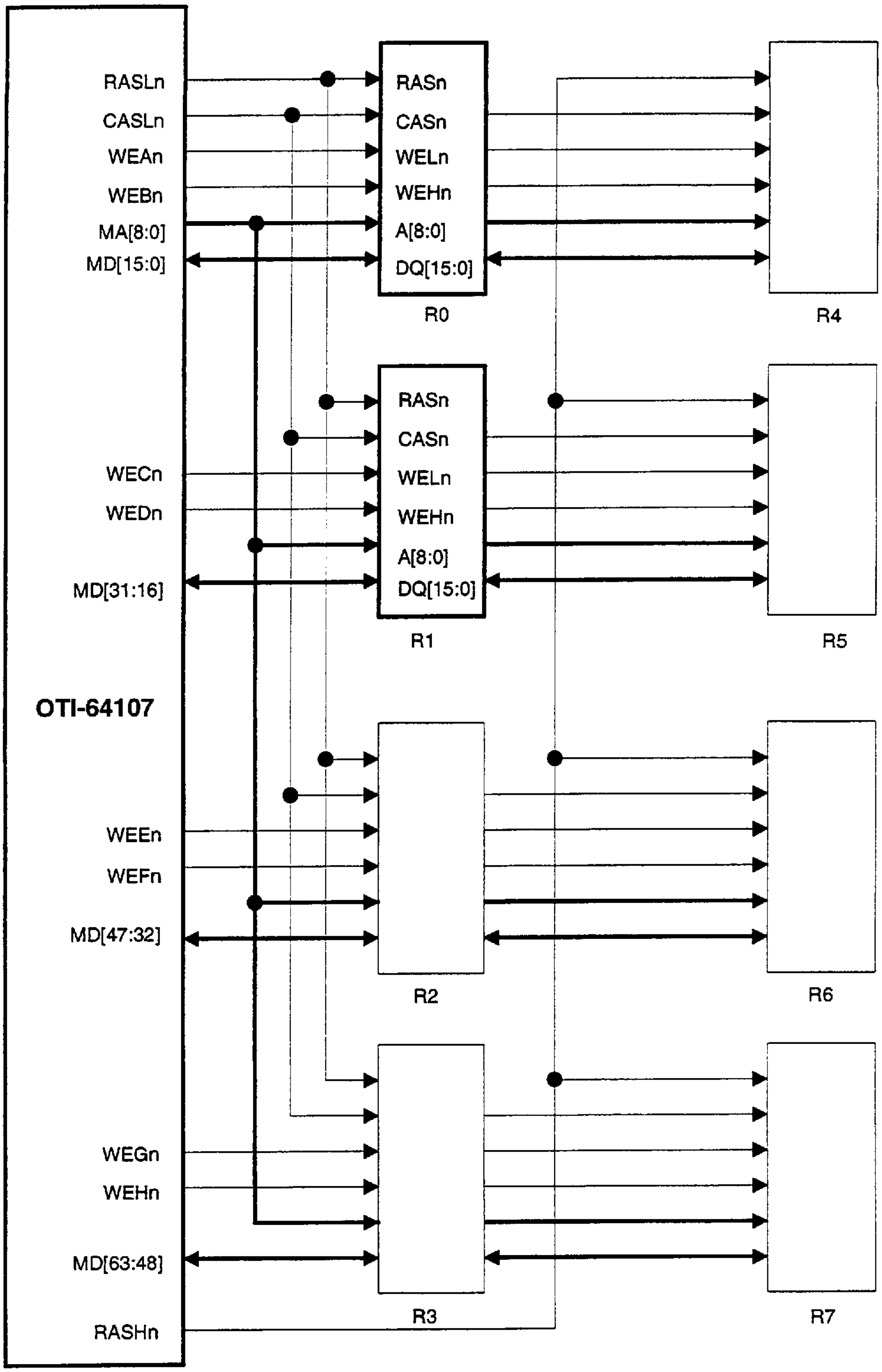


Figure 3.4 - 256K x 16 DRAMs (10 x 8, 1 CASn, 2 WEn), MD32 & 64, 1M - 4Mbytes

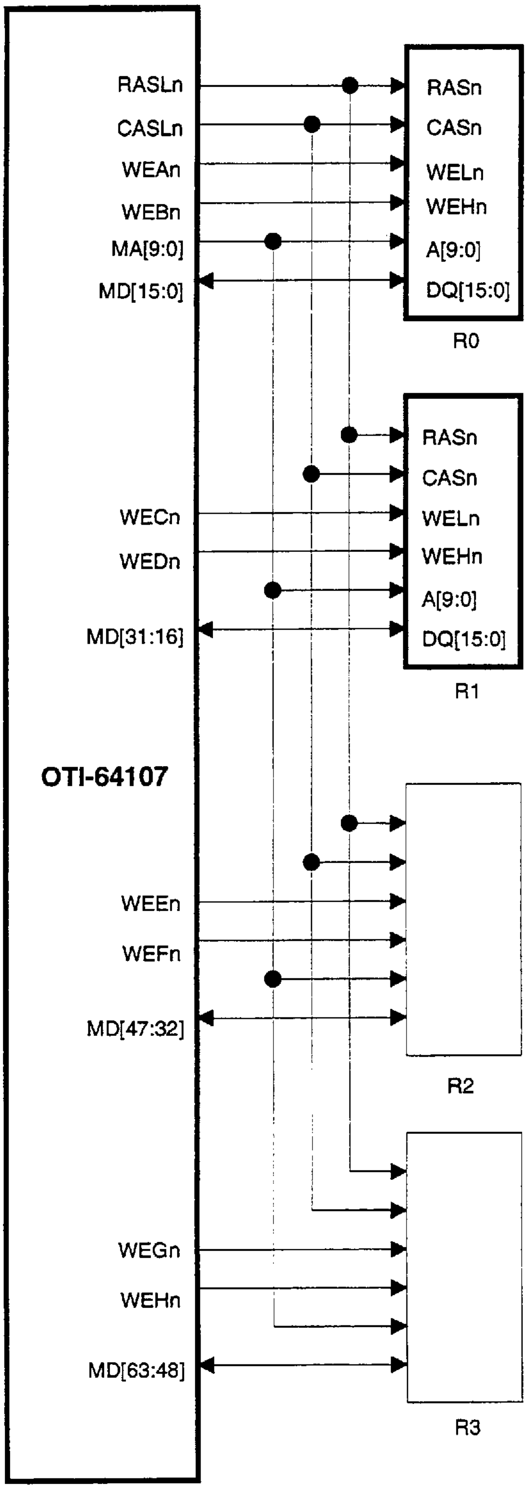


Figure 3.5 - 256K x 16 DRAMs (10 x 8, 1 CASn, 2 WEn), MD32 & 64, 1M - 2M bytes

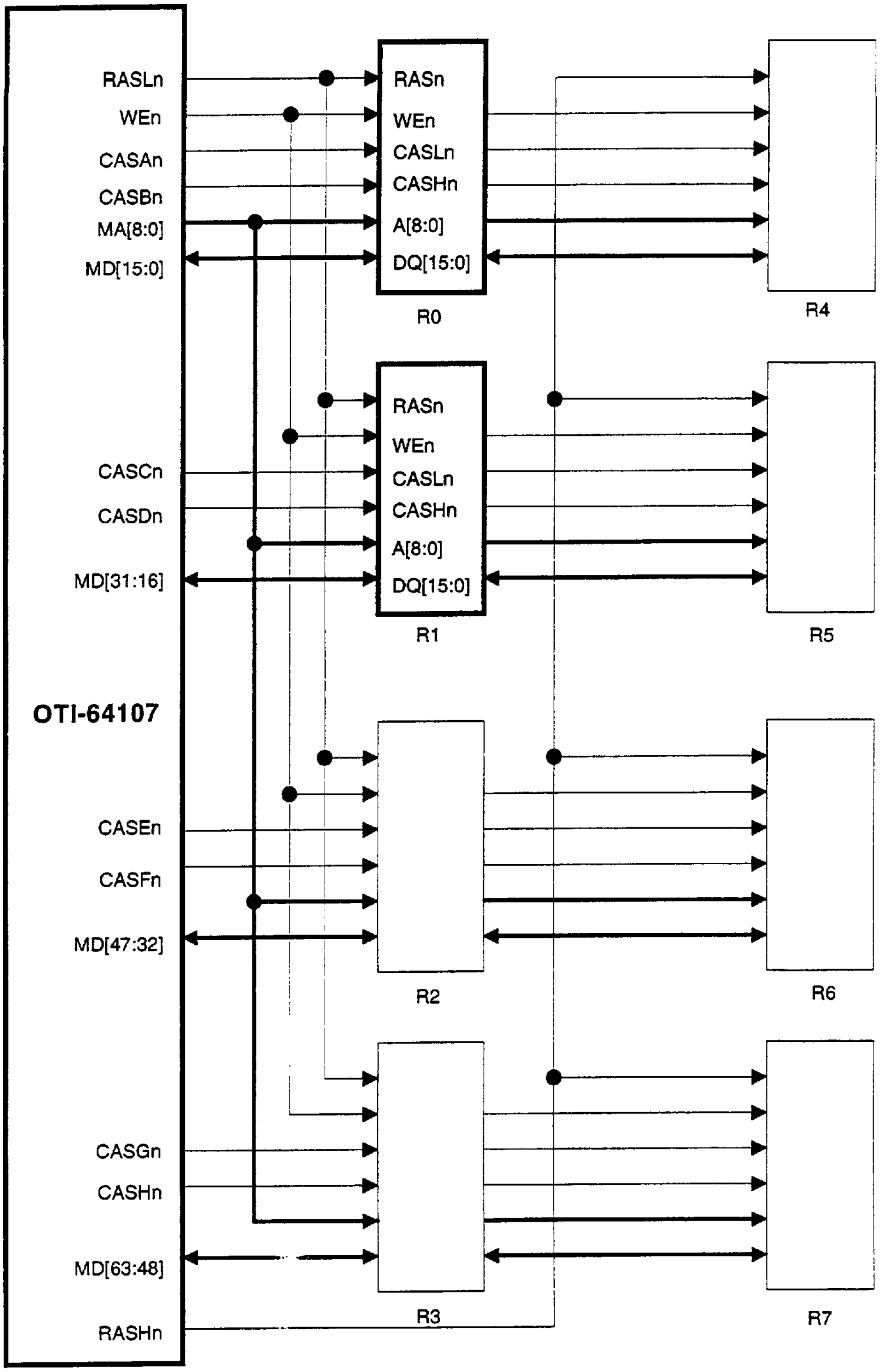


Figure 3.6 - 256K x 16 DRAMs (9 x 9, 2 CASn, 1 WEn), MD32 & 64, 1M - 4Mbytes

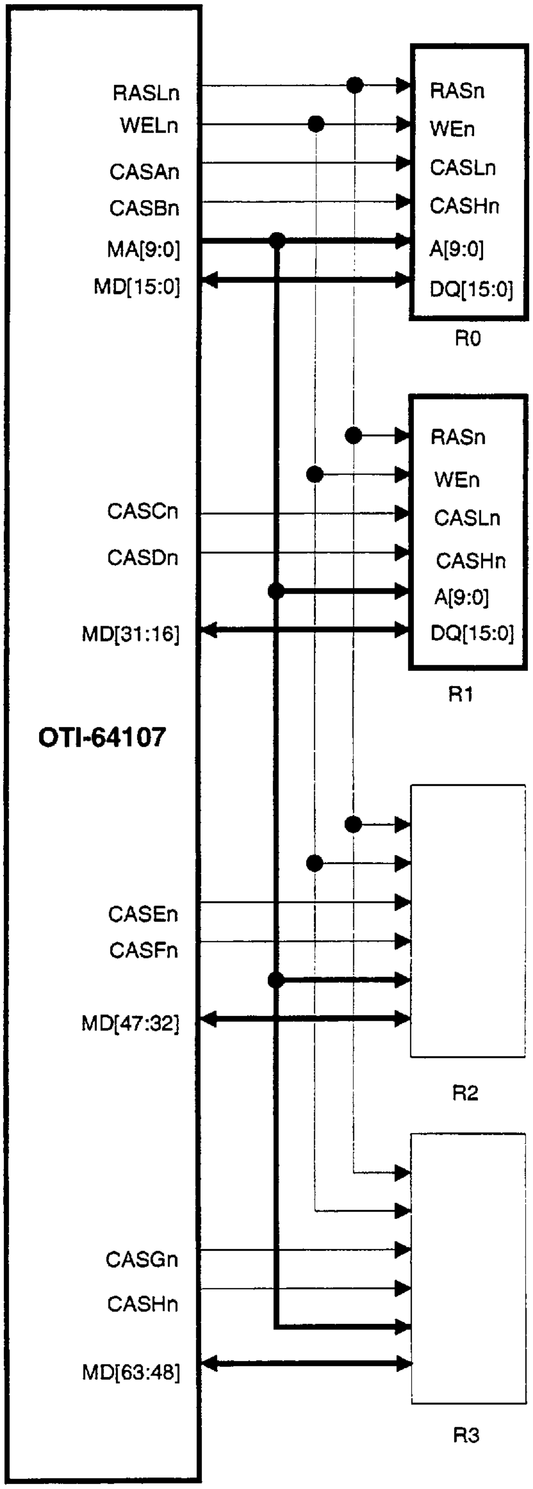


Figure 3.7 - 256K x 16 DRAMs (9 x 9, 2 CASn, 1 WEn), MD32 & 64, 1M - 2Mbytes

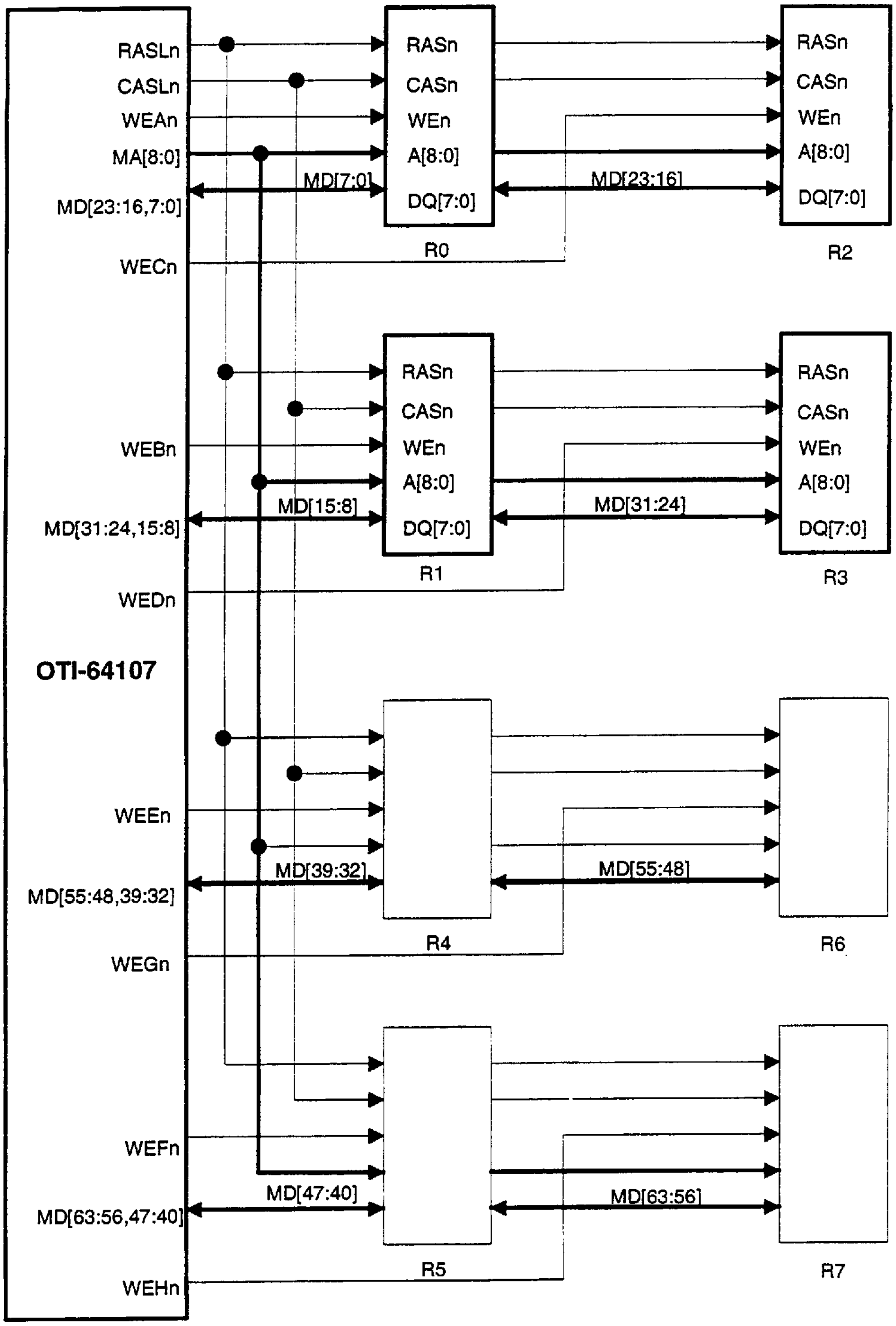


Figure 3.8 - 512K x 8 DRAMs, MD32 & 64, 2M - 4Mbytes

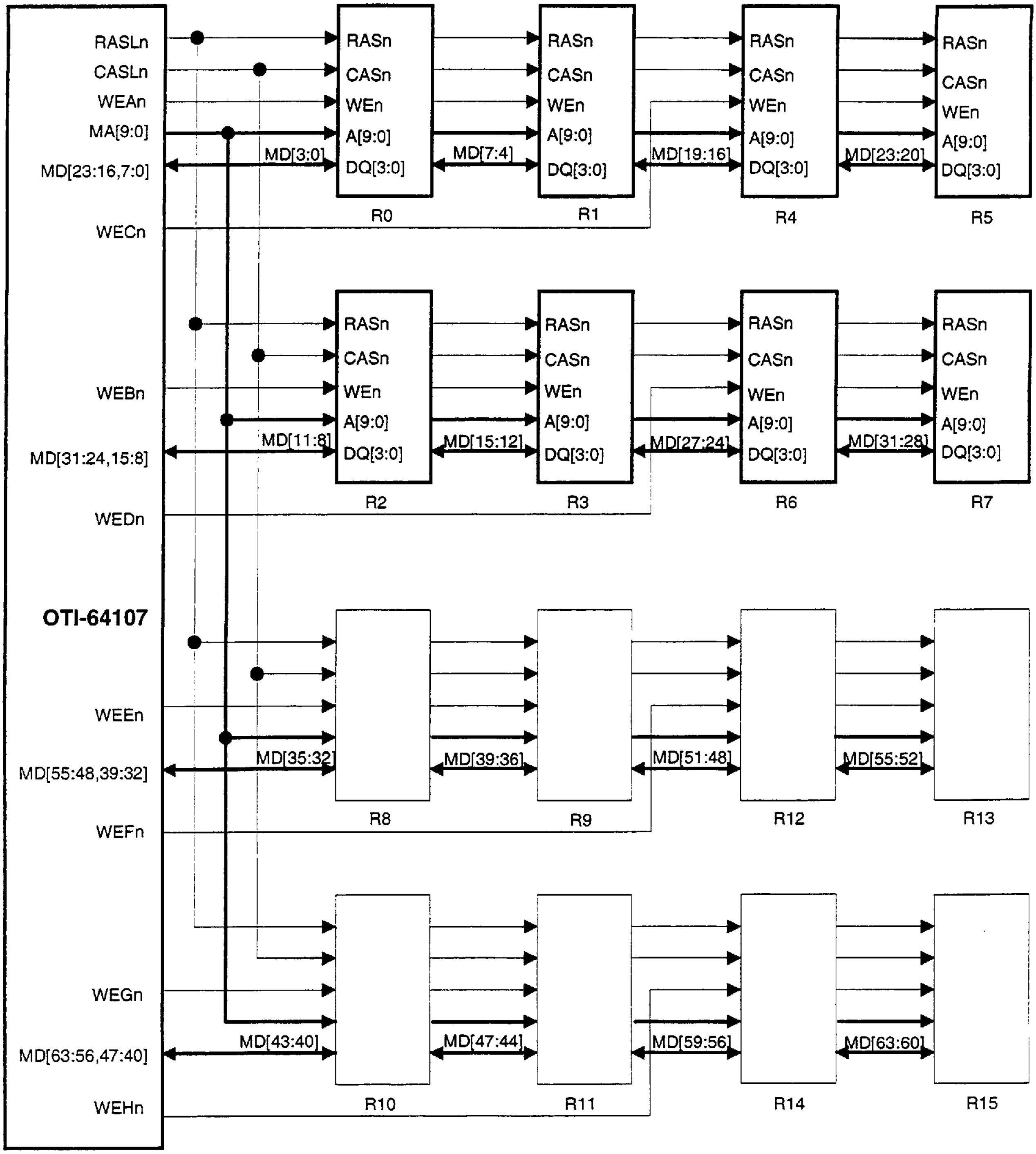


Figure 3.9 - 1M x 4 DRAMs, MD32 & 64, 4M & 8Mbytes

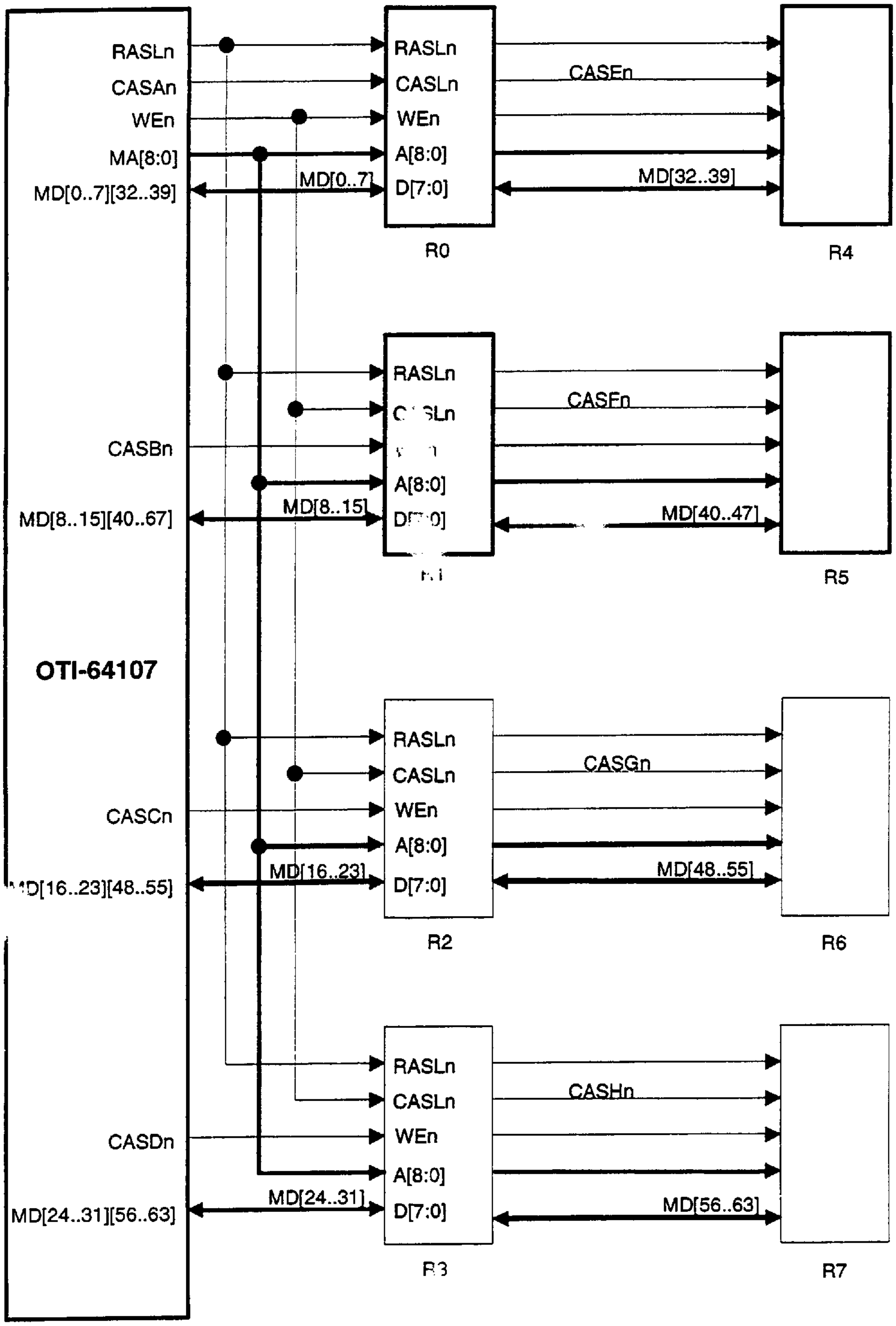


Figure 3.10 - 256 x 8 DRAMs, MD32 & MD64, 1M or 2Mbytes

CHAPTER 4: MULTIMEDIA INTERFACE

The Spitfire OTI-64107 has a 16-bit **multimedia port** which allows video data to be input to it's DRAM frame buffer. This **multimedia interface** consists of 5 control pins in addition to the 16-bit input data port. The multimedia section in OTI-64107 VGA chip consists of a **multimedia input (MMI) port** and a **multimedia output (MMO) port** pair connecting directly to the graphics frame buffer memory (also referred to as the display memory or video memory). The **MMI port** accepts live video data (i.e. a continuous input data stream from any external video source) and stores it into the graphics frame buffer, eliminating the need of a system bus data path and a dual-frame-buffer memory scheme. The video data coming into the MMI port can be either in an RGB or in a YUV data format. Once the video data is stored into the graphics frame buffer it can be displayed on the screen via the MMO port. The **MMO port** consists of a hardware window logic section that displays the stored video data (as opposed to graphics data) on the screen. The hardware window fetches the video data directly from the graphics frame buffer and sends the video data to the DAC.

There are two methods of displaying video data, which are described in the next two sections.

4.1 Chroma-keying

The first method called **"chroma-keying"** (also called color-keying) involves storing the video data into a non-displayed area of the graphics frame buffer. In this case, a '**key**' code is placed in the display area of the graphics frame buffer at the point at which the video data is to be displayed on the screen. As the data from the graphics frame buffer is sent to the DAC, the DAC scans the incoming data for the key code. When this key code is encountered the DAC multiplexes the data between graphics data (displayed area) and video data (non-displayed area) on a pixel boundary each time the key code is encountered. Chroma-keying uses a predefined (programmable) byte for the key and a set of multiplexing logic to facilitate the switch between data types. The switch is done on a pixel boundary. The key code cannot be used for a color on the screen, as it will cause video to be displayed at that pixel. Figure 1 shows the memory map for chroma-keying.

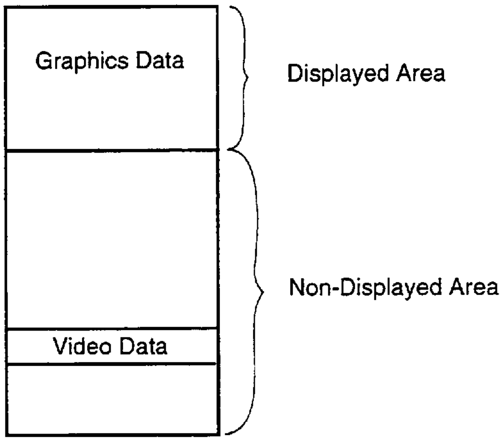


Figure 1: Memory Map for Chroma-Keying

The '**window**' size for the video data that is displayed on the screen is controlled in the following manner. For each scan line chroma-keys are placed in the graphics frame buffer to indicate the points at which the data multiplexing between the graphics and the video data is to occur. A key is placed in the display area to indicate the video image from begining to end, defining the window size of the video image. The advantage of chroma-keying is that the color depth of the video data (i.e. 4-, 8-, 16- or 24-bits per pixel) and format (RGB vs. YUV) can be different from that of the graphics data. While this scheme provides color depth flexibility it requires the use of an intelligent DAC such as Brooktree Bt885 and external support logic to multiplex between graphics and video data (the data bus to the frame buffer is 64-bits, whereas the video data port of the Bt885 is 32-bits).

As the Bt885 DAC performs the switch between displaying graphics data and video data, both data types must be transferred out of the graphics frame buffer. This requires a very high transfer bandwidth out of the graphics frame buffer memory; data must be fetched from two different places in the frame buffer, one from the graphics data area (the displayed area) and the other from the video data area (the non-displayed area). This is accomplished by sending the graphics data out during the display time and sending the video data out during the non-display time. This scheme effectively limits the size of the video window as the ratio of the non display time to the display time is typically only 20-25%. Another limit on the video window is the 800-byte video FIFO in the Bt885 (typically 400 pixels).

4.2 Video Masking

Another method called **"Video Masking"**, or **"Alpha Channel"** is implemented in the OTI-64107 to eliminate these problems. It involves storing live video data into the graphics frame buffer at the location at which it will be displayed on the screen. This places two requirements on the system: the color depth of the video data must be the same as the color depth of the graphics data, and the input must be in RGB format, as the graphics and video data are both in RGB format.

There are several advantages of video-masking over chroma-keying. **Any standard inexpensive DAC can be used which could result in significant cost savings. This scheme does not require additional graphics frame buffer memory bandwidth allowing the use of larger video windows. No external glue logic is required between the graphics DRAM memory and the DAC. Also, no special keying codes are required in video masking. This allows the use of all the colors, whereas in chroma-keying one color cannot be used as it is allocated to the key.** The contents of the graphics frame buffer are sent to the DAC without regard of data type. In order to support overlapping windows a Multimedia Mask Map (not to be confused with the Coprocessor Mask Map) is utilized to mask the video data that is stored in the graphics frame buffer. The Multimedia Mask Map is a 1bpp map used to prevent the incoming video data from updating current graphics data on the screen, thus preserving the content of any graphics data that overlaps the video data. In this manner text, graphics, or animation can be displayed over the video data. Each bit in the mask corresponds to one pixel in the displayed area of the screen. Only incoming pixels of the video data corresponding to set bits in the mask will be stored into the graphics frame buffer. Because the video data occupies a portion of the displayed area, the video data is sent to the DAC in the same manner as graphics data. The driver must determine the size and shape requirements of the video window and setup the Multimedia Mask Map to simulate window overlap. **A major advantage of video masking is that graphics or text data over video can be captured, unlike chroma-keying where text data or graphics annotated over the video cannot be captured. Video masking also allows arbitrarily shaped windows which can be used to create unique effects. Pull-down menus do not freeze the video.** Figure 2 shows the memory map for video masking.

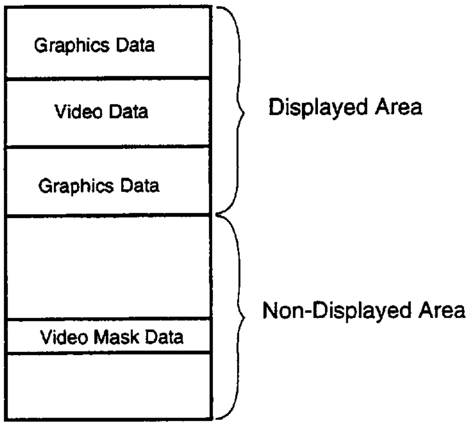


Figure 2: Memory Map for Video Masking

The Multimedia Mask Map is read and written as bytes in the graphics frame buffer memory address space; only whole bytes can be manipulated for each read or write to the mask. Since each bit in the mask corresponds to one pixel, and since the number of pixels that the mask operates on could be odd, the software driver must account for partial-byte reads or writes to the mask. A '1' would allow video data to update the screen, a '0' would prohibit updating. This map can be located anywhere in the non-displayed area and is programmable through the Multimedia Mask Map Start Address register. This map is assumed to be continuous and has no offset. There is no alignment hardware for this map. All unused bits should be filled with 0's, otherwise incorrect pixels might appear on the screen.

There are two sets of memory-mapped registers that affect this scheme. The first set of registers control the **video data window: HW Start Address** registers (offset[9C-9E]) and, **HW Address Offset** register (offset[9F]). The Start Address registers define the address within the graphics frame buffer at which the input video data will be stored. The Address Offset register defines the starting address of the next scan line of the video data. (The value in the Offset register is the scan line length of the displayed area of the screen which may not be the same as the total scan line length). Once these registers are setup, the video data is automatically written into the graphics frame buffer for each cycle on the MMI port. Or, in the case of the outputs, the video data is fetched from the frame buffer for each cycle on the MMO port.

A second set of registers control the **Multimedia Mask Map: HW Mask Map Start Address** registers 3DF[98-9A] and, **HW Control** register 3DF[96]. The Mask Map Start Address registers define the address within the frame buffer at which the Multimedia Mask Map begins. The Control register enables the use of the mask. In the case in which the video data is stored directly into the display area of the frame buffer, the Multimedia Mask Map, when enabled via the Control register, prevents live video data from updating the existing contents of the frame buffer whenever a 0 is encountered in the mask. The area defined by the video data window control registers must be greater than or equal to the area masked by the Multimedia Mask Map.

Figure 4.3 illustrates a display using the video masking scheme in the OTI-64107.

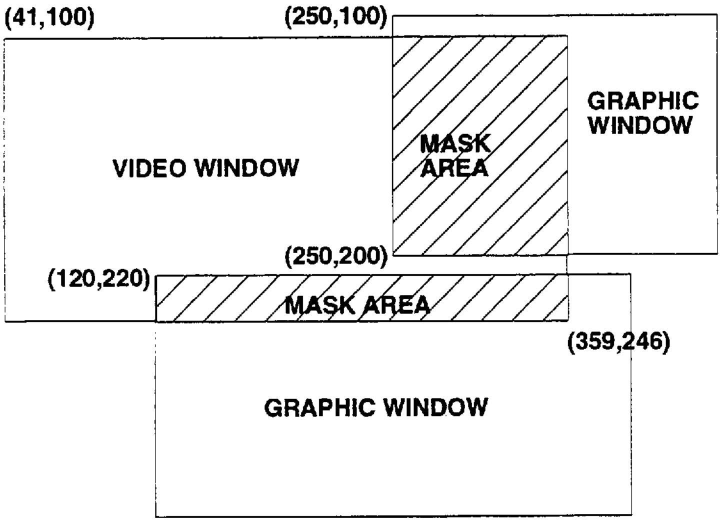


Figure 4.3 - Video Windows & Image Masking

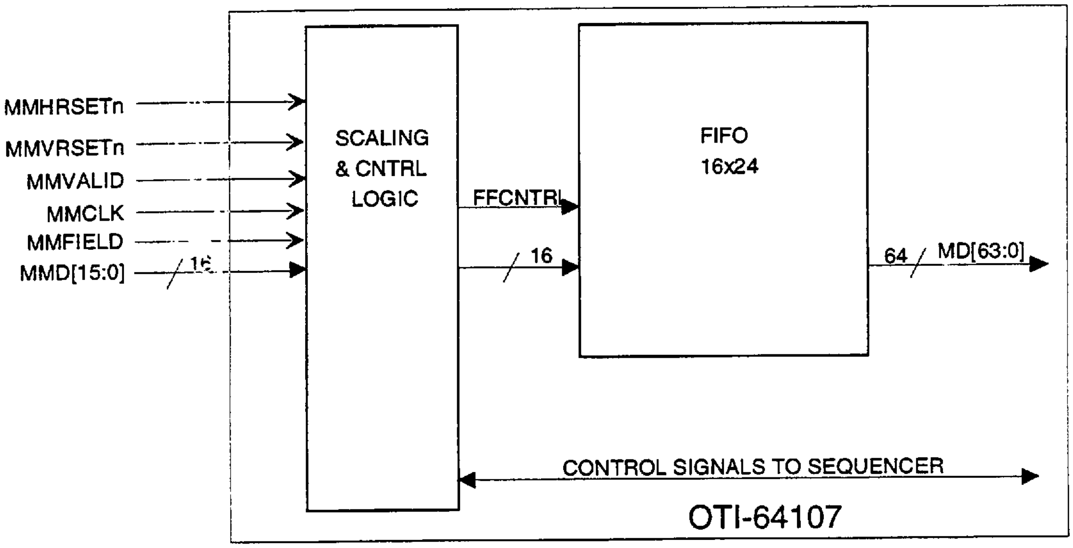
In the example, the display resolution is 1024 x 768 and the color depth is eight bits/pixel. The absolute diagonal coordinates of the video windows are x1=41 pixels, y1=100 pixels and x2=359 pixels, y2=246 pixels. The Multimedia Mask Map is typically placed in the upper, non-displayed area of the graphics frame buffer. For this resolution 98,304 bytes are required to store the mask data.

The Multimedia Mask Map will contain 1's for the following areas: 41,100 to 250,200; 41,201 to 359,219; and 41,220 to 120,246. The mask will contain 0's in all other pixel positions.

4.3 Multimedia Input (MMI) Port

The **MMI port** is an 16-bit synchronous data port that can receive data at rates up to 33MHz. In the Spitfire OTI-64107, **this input port is designed to interface gluelessly to devices that support the I²C bus or directly to programmable video decoders or scalers.** Devices that support the **I²C** protocol include video decoders and scalers from **Philips** (SAA7110, SAA7186, SAA7196, etc.), **ITT**, and **Sony**. Devices that have a **programmable interface** include video decoders from **Brooktree** (Bt812) and scalers from **Thesys** (Th6205). Almost all video chips that do not support the I²C bus, can be interfaced gluelessly with the OTI-64107 using the programmable interface. Support for the external video devices includes a chip select, system I/O command and system data control logic, and the I²C bus. In the future, the Spitfire family will be modified to interface directly to the VESA Media Channel, 16-bit mode and still maintain pin-to-pin compatibility with the current version. The MMI port takes video data into the OTI-64107 and stores it into the graphics frame buffer.

The input data can also be scaled down vertically and/or horizontally by the OTI-64107. Vertical scaling by displaying every 2, 4, or 8 lines is supported with an option to start skipping on odd or even scan lines. **Horizontal scaling** by displaying every 2, 4 or 8 pixels is supported in a similar manner. Scaling is particularly important to save memory space and bandwidth.



The input signals for the MMI Port are:

- MMVRSETn is used to indicate the beginning of a frame,
- MMHRSETn is used to indicate the end of a line,
- MMVALID is used to indicate that data is ready to be clocked in,
- MMFIELD is used to indicate even or odd fields,
- MMCLK is the input clock, and,
- MMD[15:0] is the 16-bit input data bus.

The registers that control the input port are:

- **HW Start Address** registers (offset[9C-9E]) define the address within the graphics frame buffer at which the input video data will be stored, and,
- **HW Address Offset** register (offset[9F]) defines the starting address of the next scan line of the video data. The value in the Offset register is the scan line length of the displayed area of the screen which may not be the same as the total scan line length..

Once these registers are setup,

- MMVRSETn would load the HW Start Address Register into the MMI address counter,
- MMHRSETn would add the HW Offset register to the current start address and load it into the MMI address counter, and,
- MMVALID active would initiate transferring data to the frame buffer.

4.4 Multimedia Output (MMO) Port

The **MMO port** is supported as a hardware window (very similar to the hardware cursor) and is designed to interface with the **Brooktree Bt885** with some additional logic (8 F374's when used with the 64-bit memory data bus, none with 32-bit memory data bus). The data path for the video data is connected to the Bt885 via MD[63:0]. A video window can be defined through the **Vertical Position Start**, **Width** and **Height** registers. In this case **chroma-keying** is selected. The MMO port as described above is used when video data is stored in the non-display area and a video DAC like the Bt885 is used. The Bt885 will display graphics data until it reads the chroma-key at which time it will start displaying video data in the window.

The hardware cursor is supported through the logic in the Bt885. A repeat line input signal is also available to assist the Bt885 to do vertical zooming. When this line is active, the memory controller would fetch the same video line again and cause zoom-in by pixel line replication.

The control signals for the MM Port are:

- VDVALID is use to indicate to the VideoDAC that the data is ready to be clocked in.
- MDMXn is used to mux MD(63:32) with MD(31:0)
- RPLINE is used to assist the Bt885 to do vertical zoom-in. The memory controller would fetch the same video line again and cause a zoom-in by pixel line replication.

The registers that control the output port are the Hardware Cursor control registers (registers 0-88).

When a standard ClkDAC like the OTI-088 or the AT&T20C499 are used, the hardware cursor logic is enabled in the OTI-64107 instead of the hardware window. In this case the video masking scheme must be used to display the video data window and the Multimedia Mask Map is used to control the size of the video window.

Video masking allows graphics (e.g., text or pull-down menus) to overwrite the video. It also allows arbitrarily shaped windows, and with some constraints, multiple video windows.

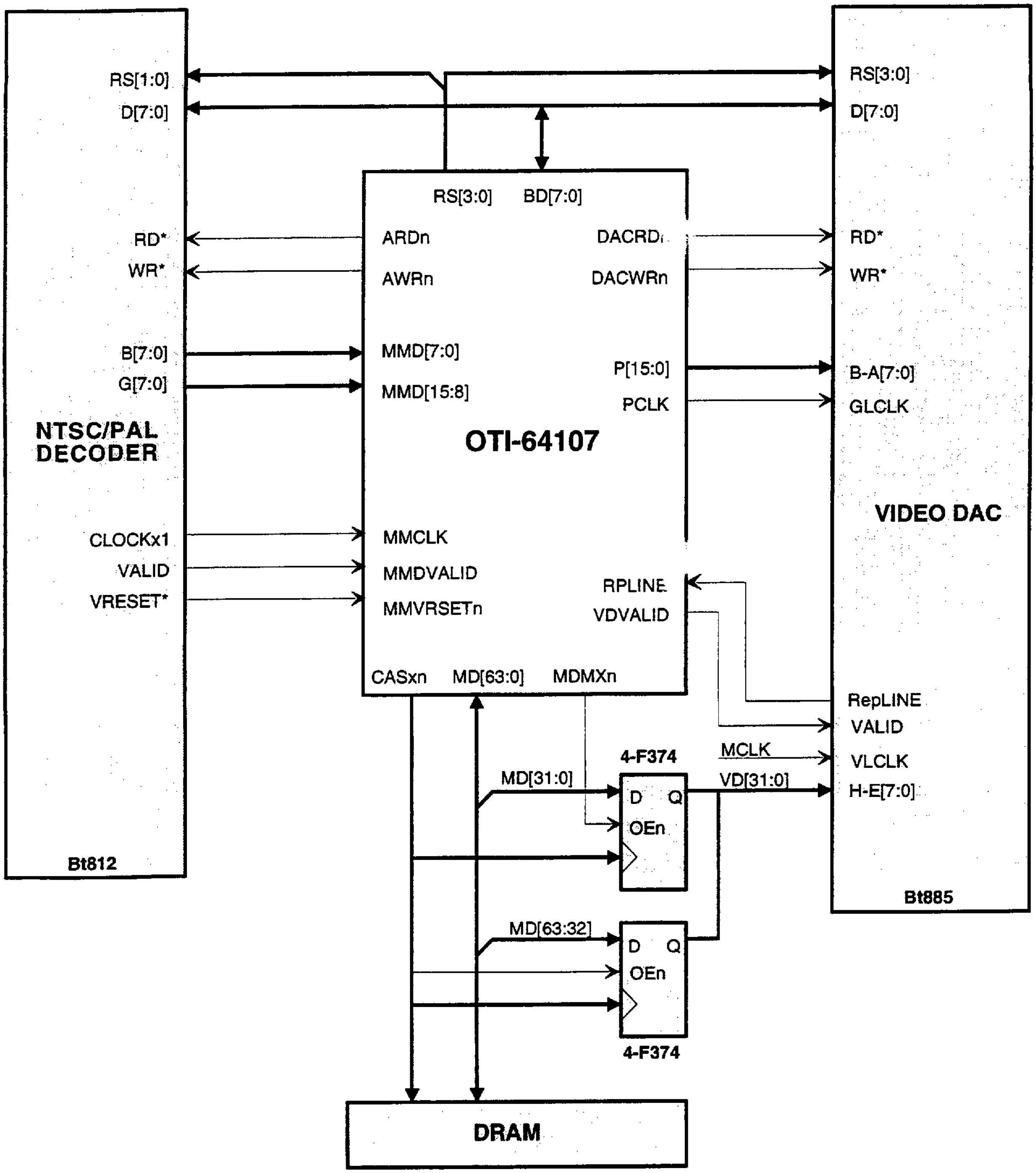


Figure 4.2 - Block Diagram of the Multi-Media Port with the Brooktree Bt885 Video DAC on the VL Bus

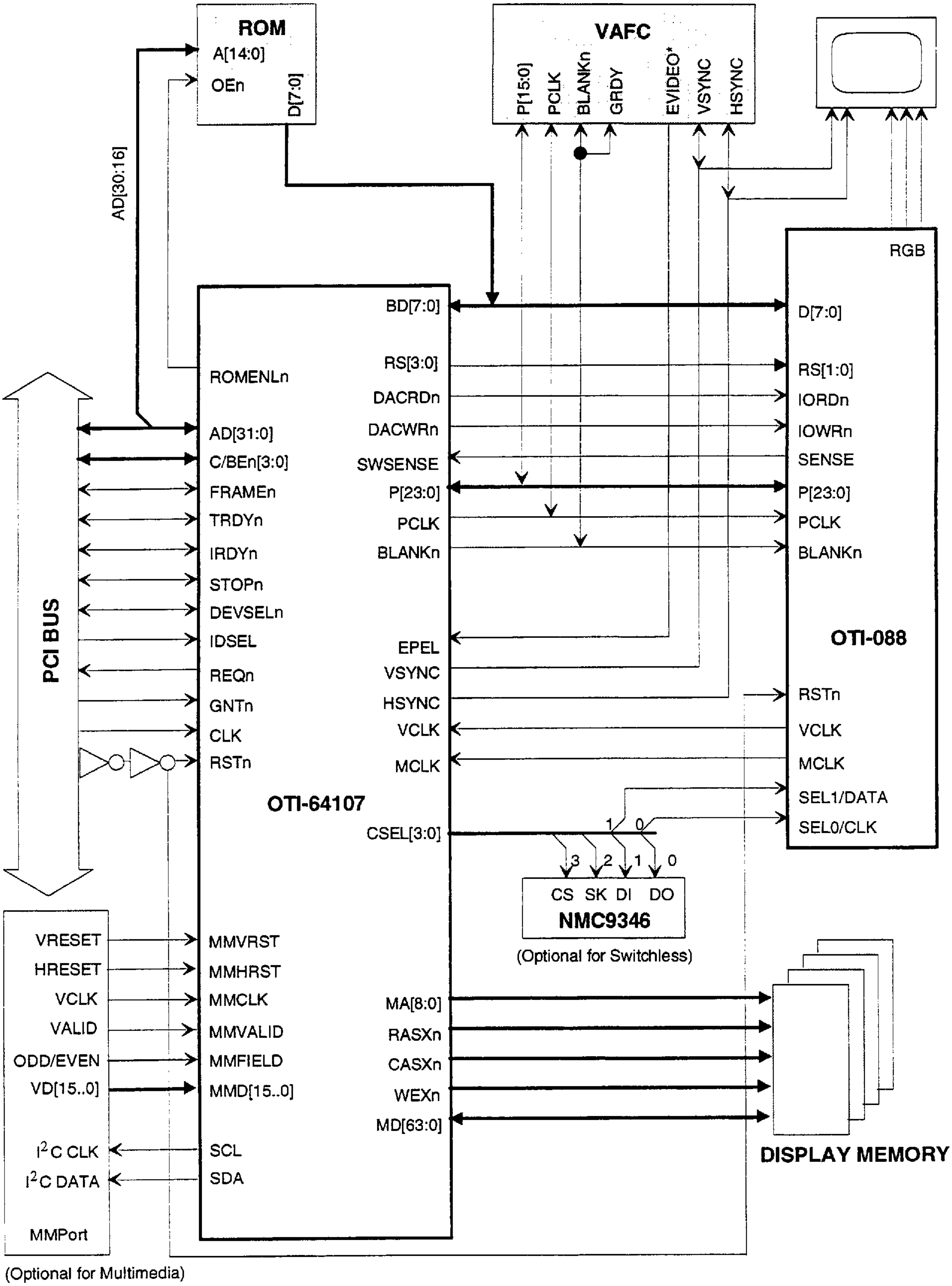
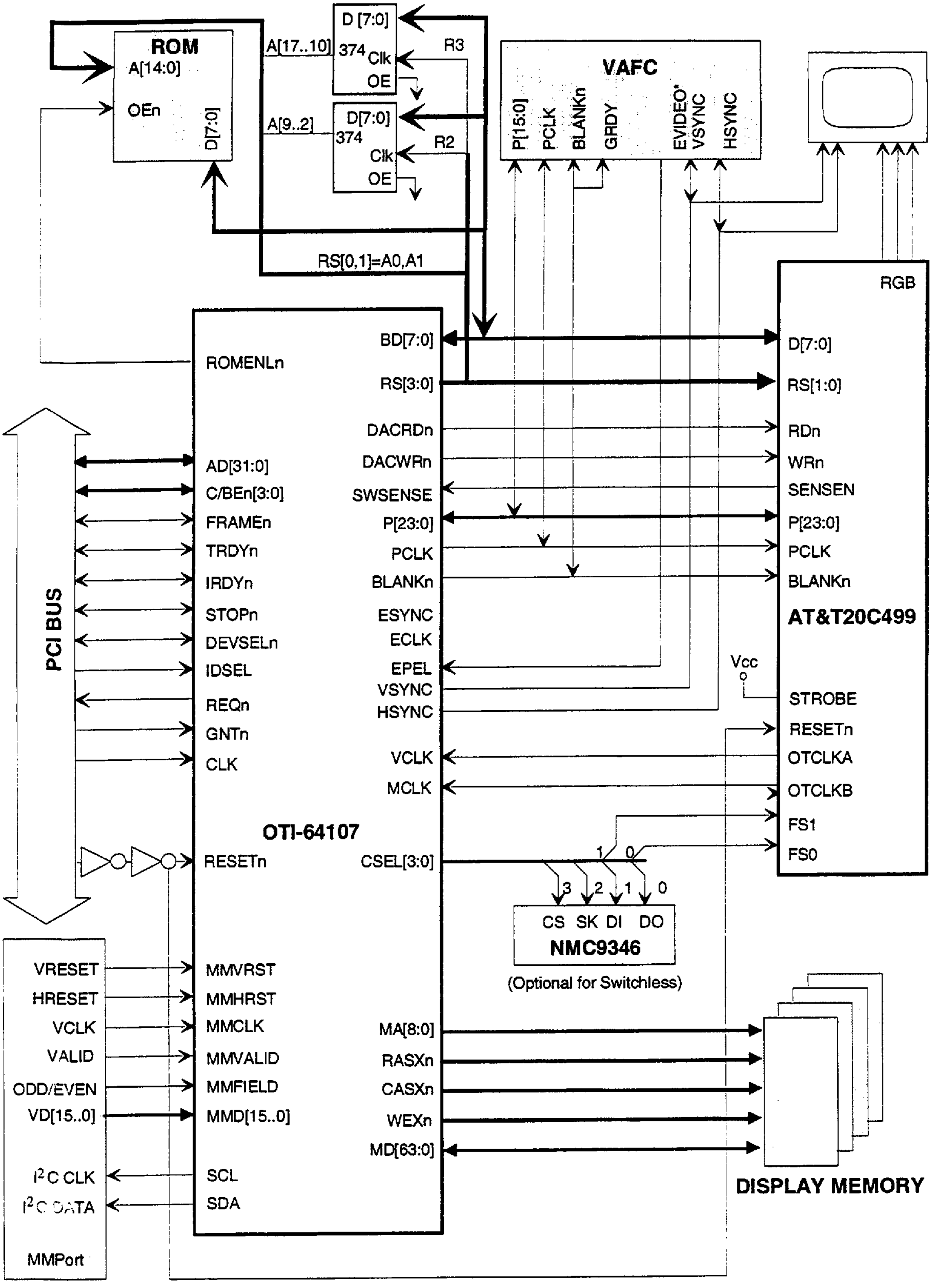


Figure 4.3 - Block Diagram of the Multi-Media Port and the VAFC using OTI-088 ClkDAC and PCI Bus (Glueless)



(Optional for Multimedia)

Figure 4.4 - Block Diagram of the Multi-Media Port and the VAFC using AT&T20C499 Precision DAC™ and PCI Bus (Glueless)

4.5 Support for External Video Chips

To save glue logic when external Video Chips (such as video decoders, image scalers, etc.) are required for the graphics subsystem, the OTI-64107 provides additional address decoding, address latching and data routing for these external components. For components that have an I^{2}C interface (like Philips or ITT decoders and scalers) instead of the standard host interface, the OTI-64107 can also interface with them through a combination of hardware (through the Multimedia Port) and software. On the PCI bus, support for external Video chips normally requires fairly extensive circuitry (probably a custom gate array). This includes the configuration registers, address latches and decoders, data latches and buffers, address counters, etc. The OTI-64107 has all of this circuitry built-in, allowing a very low cost and convenient interface to external Video chips.

4.5.1 I^{2}C Bus

The I^{2}C Bus is a serial data communications bus developed by Philips. It is a two-wire multi-master serial bus with a standard transfer rate of 100 Kbits/sec, and 400 Kbits/sec in fast-mode. The number of interfaces connected to the bus is solely dependent on the limiting bus capacitance of 400 pf.

The 2 wires, Serial Data (**SDA**) and Serial Clock (**SCL**), carry information between the devices connected to the bus. Each device is recognized by a unique address, and can operate as a **transmitter** or **receiver**, depending on the function of the device. A **master** is the device which initiates a data transfer and generates the clock signals to permit that transfer. Any device addressed is considered a **slave**. As the I^{2}C bus is a multi-master bus, more than one device capable of controlling the bus can be connected to it.

More than one master could initiate a data transfer at the same time. An **arbitration** scheme using **wired-AND connections** on all I^{2}C devices is used. Generation of the clock signals on the I^{2}C bus is always the responsibility of the master device; each master generates its own clock signals when transferring data on the bus. Bus clock signals can only be altered when they are stretched by a slow-slave device holding down the clock line, or by another master when arbitration occurs.

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW. A HIGH to LOW transition on the SDA line when SCL is HIGH indicates a **start** condition. A LOW to HIGH transition of the SDA line defines a **stop** condition. Start and stop conditions are always generated by the master.

Data is transferred with the MSB first. The addressing procedure for the I^{2}C bus is such that the first byte after the **start** condition usually determines which slave will be selected by the master. The first seven bits of the first byte make up the **slave address**, followed by the **direction bit (R/Wn)**. Each data byte is followed by an **acknowledge** bit. See the I^{2}C Bus Specification for more information.

4.5.2 I^{2}C Support

The OTI-64107 has the 2 programmable I^{2}C bus pins, **SRCK** (Serial Clock line) and **SRD** (Serial Data line), to support the I^{2}C bus. These two pins are controlled and can be read from the **I^{2}C Control** register (3DF index C). Writing 0's will drive the pins low, and writing 1's will tristate the pins. Software can control and read these pins in accordance with the I^{2}C bus protocol to program the external component's registers. A typical I^{2}C (Philips decoder or scaler) bus to ISA interface needs to be designed to accept video in a PC system. This interface requires at least four TTL parts and three PALs to be implemented. The OTI-64107 has this interface built-in, saving the parts and time to implement the I^{2}C bus interface.

4.5.3 Auxiliary I/O Support

For non-I^{2}C bus devices that have a 16-bit data bus, host read and host write signals (like the Bt812), the OTI-64107 provides all the necessary control signals to the address space to select and route data. The auxiliary I/O support is different for each system bus platform. On the VL-bus, auxiliary I/O can be supported by configuring the OTI-64107 to generate **DACCSn** (could be used by the Bt885) and **ACSn** (could be used by the Bt812) instead of DACRDn & DACWRn respectively. These chip select signals, can be ORed (LS32) with IOWRn and IORDn from the ISA bus to generate **DACRDn**, **DACWRn**, **ARDn**, and **AWRn**. The register select address can be connected directly to the SA bus of the ISA connector. Data routing is done through the same LS245 that is used for DAC & BIOS ROM routing.

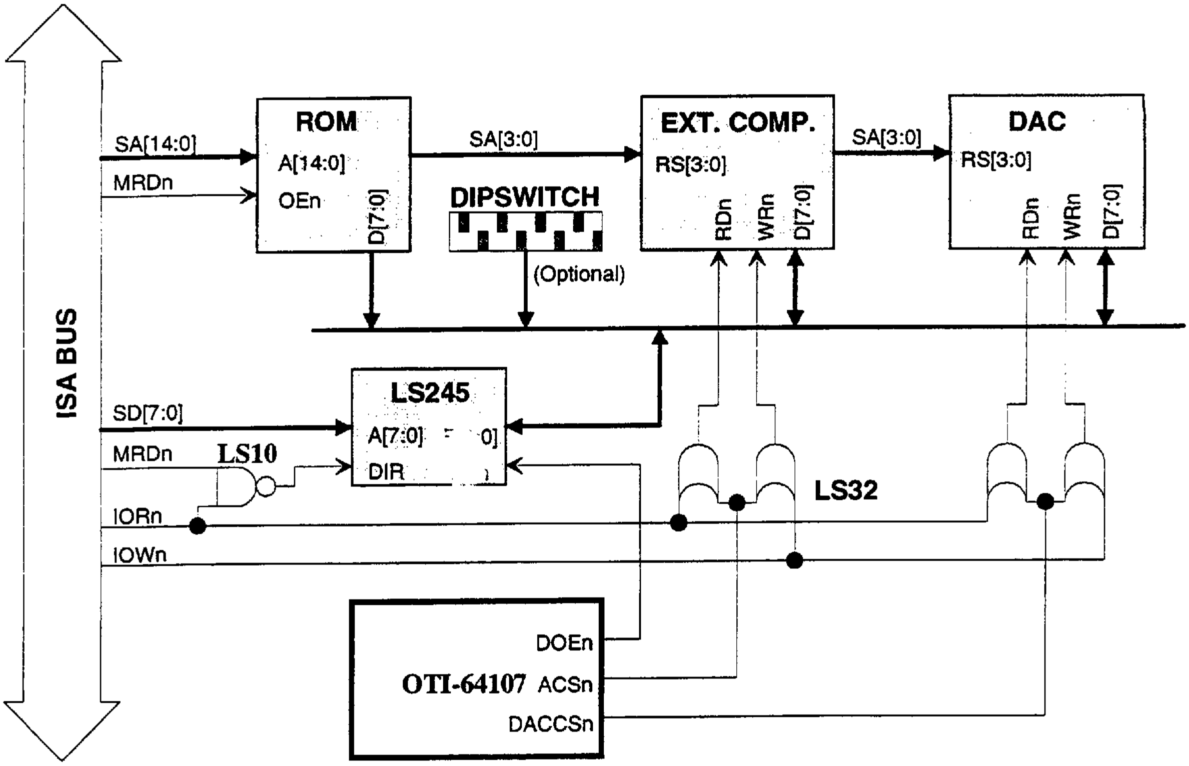


Figure 4.5 - Auxiliary I/O on VL Bus (via ISA Bus)

On the PCI bus, ARDn and AWRn are generated from the OTI-64107. ARS[3:0] are latched from the system address and sent out to the external components. Up to 16 address pins can be supported using external address latches. Data is routed through the BD[7:0] bus. See the PCI Bus block diagram for more information.

On the ISA bus, ARDn and AWRn are also generated, BD[7:0] is used to route data, but ARS[3:0] are not generated and the register select lines should be connected directly to the system bus just like on the VL-bus.

4.6 Video Bandwidth

Maximum video resolution and refresh rate is a function of memory bandwidth available to the MM port after screen refresh has been satisfied. Total memory bandwidth is a function of memory clock, memory bus width, and the programmed memory timing. The memory controller of the OTI-64107 reserves first priority for screen refresh, then MM port, and then other sources. Thus, the maximum video resolution that can be supported for a given graphics mode can be defined by the maximum bandwidth available to the MM port for the mode. The only restriction is caused by the bandwidth of the 48-byte FIFO which recieves the input data from the 16-bit MMdata port. the Maximum Video Bandwidth numbers listed in the table below are extremely conservative. In actuality, the bandwidth should be significantly higher. The following tables show the video bandwidth that can be supported for a given graphics mode with 64-bit memory bus and 66MHz MClk.

Table 1. OTI-088 Video Bandwidth (or Equivalent DACs the AT&T 20C409/20C499)

Graphics Resolution	bpp	Vertical Refresh	Maximum Video Bandwidth	Maximum MMClk	MClk	MD
1280 x 1024	8	75Hz	44Mbytes/sec	22MHz	66MHz	64
1024 x 768	8	75Hz	70Mbytes/sec	35MHz	66MHz	64
1024 x 768	16	75Hz	35Mbytes/sec	17MHz	66MHz	64
1280 x 1024	8	60Hz	47Mbytes/sec	23MHz	66MHz	64
1024 x 768	8	60Hz	70Mbytes/sec	35MHz	66MHz	64
1024 x 768	16	60Hz	44Mbytes/sec	22MHz	66MHz	64

Notes:

1. Standard Video Resolution is just an example of what standard resolution can be supported with the given available bandwidth. Standard Video Resolutions are 640 x 480, 320 x 240, and 160 x 120. The exact video resolution that can be supported is actually higher than the standard ones. For example, in the case of 1280 x 1024 8 bpp graphic mode, a video window of 640 x 480 at 30Hz and 8 bpp only requires 9.22Mbytes/sec, but the available bandwidth is 22Mbytes/sec.
2. Maximum MMClk is the maximum rate of transfer the MM port can sustain without losing any data. To run a higher rate than this, the internal scale down circuitry should be utilized.
3. Maximum video bandwidth is the maximum rate of updating the video screen.

For video windows using the Bt885 as the RAMDAC, the resolution that can be supported is limited not only by the bandwidth available to the MM port, but also the bandwidth available to MMO port, and the cache size (800 bites) of the Bt885.

Table 2. Bt855 (or equivalent RAMDAC) Video Bandwidth with the Bt885 DAC[1,2]

Graphics Resolution	bpp	Vertical Refresh	Maximum Video Bandwidth	Maximum Video Resloution	Maximum MMClk	MCLK	MD
1280 x 1024	8	75Hz	35Mbytes/sec	256 x XX @ 16 bpp	17MHz	66MHz	64
1024 x 768	8	75Hz	38Mbytes/sec	320 x XX @ 16 bpp	19MHz	66MHz	64
800 x 600	8	75Hz	38Mbytes/sec	512 x XX @ 16 bpp*	19MHz	66MHz	64
640 x 480	8	75Hz	38Mbytes/sec	640 x XX @ 16 bpp*	19MHz	66MHz	64
1280 x 1024	8	60Hz	38Mbytes/sec	384 x XX @ 16 bpp	19MHz	66MHz	64
1024 x 768	8	60Hz	38Mbytes/sec	512 x XX @ 16 bpp*	19MHz	66MHz	64
800 x 600	8	60Hz	38Mbytes/sec	640 x XX @ 16 bpp*	19MHz	66MHz	64
640 x 480	8	60Hz	38Mbytes/sec	640 x XX @ 16 bpp*	19MHz	66MHz	64

Notes:

1. Maximum Video Line Width is the maximum resolution that can be displayed by the MMO port at 16 bpp or 8 bpp video color depth.
2. Maximum Video Resolution - maximum video data that can be fetched to display per line.
3. * Although 512 and 640 pixels can be fetched by the OTI-64107, only 400 pixels (800 bytes) can be handled by the Bt885 per line. Thus, the maximum resolution when used with Bt885 is 400 x XX.

CHAPTER 5: POWER MANAGEMENT SUPPORT

The OTI-64107 has power saving modes that are compliant to the VESA DPMS Proposal version 1.0p. The four modes of power management are as follows:

State	HSYNC	VSYNC	BLANKn	P[23:0]	DPMS Requirement	Power Savings
On	Pulses	Pulses	Active	Active	Mandatory	None
Stand-by	No Pulses	Pulses	Blanked	Blanked	Optional	Minimal
Suspend	Pulses	No Pulses	Blanked	Blanked	Mandatory	Substantial
Off	No Pulses	No Pulses	Blanked	Blanked	Mandatory	Maximum

Active - means normal operation, signal is switching between active and inactive

No Pulses - signals remain at inactive state (HSYNC, VSYNC = 0)

Pulses - normal operation for HSYNC & VSYNC

Blanked - BLANKn is asserted (0), P[23:0] = 0

When in any of the power saving modes, the Memory Controller stops fetching data for display, but continues to refresh the DRAM. Power management is controlled by the Power Management Control register (3DF index F).

CHAPTER 6: PIN DESCRIPTION

6.1 ISA Bus Interface

Pin Name	Pin Number	Type	Description
MASTERn	60	I	MASTERn. This pin indicates that the current cycle is a master cycle when the controller is in add-on configuration. It enables the LA address to pass through during master cycle.
MWRn	62	I	MEMORY WRITE. Active low memory write strobe.
MRDn	63	I	MEMORY READ. Active low memory read strobe.
CINTn	66	Ood	CRT INTERRUPT REQUEST. Interrupt is asserted when vertical retrace occurs if it is enabled by bit 5 of the Vertical Retrace End (3?5 index 11) register. It is an active low open drain output.
RESETn	67	I	RESET. Active low system reset signal. This input signal will reset the VGA controller and initialize the configuration register based on the logic level on MD[15:0] pins at power-up reset time. This signal is inverted from the bus reset.
LA[23:17]	75-72,70, 65-64	I	UNLATCHED SYSTEM ADDRESS BITS 23:17. These bits are decoded to generate M16n. Bits 19:17 are latched by ALE to generate SA[19:17].
M16n	68	Ood	16-BIT MEMORY. Active low, open drain output signal used to indicate to the system that the present cycle is a 16-bit data transfer to video memory. This signal is derived from the decoding of LA23:LA17.
ZEROWSn	69	Ood	ZERO WAIT STATE. This pin is used to indicate the current cycle is a 0 wait state cycle.
IO16n	71	Ood	16-BIT I/O. This active low, open drain output signal is used to indicate to the system that the present data transfer is a 16-bit I/O cycle. It is derived from the decode of system address bits SA19-SA0.
BHEn	76	I	BYTE HIGH ENABLE. This active low input indicates that there is valid data on SD[15:8] bus. This signal and SA[0] together indicate to the OTI-107 whether an 8 bit or 16 bit cycle is being executed by the system.
ALE	78	I	ADDRESS LATCH ENABLE. This input is used to latch a valid address from the CPU in add-on configuration.
RFSHn	79	I	REFRESH. This input is used to qualify the video memory and I/O access from CPU. An active low indicates a system memory refresh cycle.
IORn	80	I	I/O READ. This is an active low I/O read strobe.
IOWn	81	I	I/O WRITE. This is an active low I/O write strobe.

Pin Name	Pin Number	Type	Description
IOCHRDY	82	Ood	IO CHANNEL READY. An open drain active high output to signal processor that it ready for memory access. This signal is used to add wait states to the bus cycle during display memory accesses.
SA[16:0]	101-92,90-85,83	I	LATCHED SYSTEM ADDRESS BITS 16:0.
SD[15:0]	102-108,110, 112-115,117-120	I/O	SYSTEM DATA BUS BITS 15:0.
AEN	111	I	ADDRESS ENABLE. This input is used to qualify the video I/O access from CPU. When it is active high, the DMA controller has control of the address bus, data bus, and command lines.

Total: 55 pins

6.2 VL Bus Interface

Pin Name	Pin Number	Type	Description
SA[31:2]	3-10,72-69, 65-62,60-47	I	SYSTEM ADDRESS BUS bits 31:2. This bus should be connected to the ADR[31:2] bus of the VL connector.
CINTn	66	Ood	CRT INTERRUPT REQUEST. Interrupt is asserted when vertical retrace occurs if it is enabled by bit 5 of the Vertical Retrace End (register 3?5 index 11) register. It is an active low open drain output.
RESETn	67	I	MASTER RESET. This reset signal is used to reset all internal state machines and some default registers. During Reset active, all bidirectional buses are tri-stated. This signal is also used to latch-in the configuration register values. This reset is active low and should be connected the VL bus RESETn signal instead of the ISA bus active high RESET.
PROCLK	68	I	PROCESSOR CLOCK. Processor clock input used to sample CPU status and address. This clock is 1X for VL bus but can be configured to be 2X. This signal should be connected to the LCLK pin of the VL connector.
SD[31:0]	73-76,78-81, 85-90,92-93, 102-108,110, 112-115,117-120	I/O	SYSTEM DATA BUS bits 31:0. This bus should be connected to the DAT[31:0] bus of the VL connector.
BEn[3:0]	82,94,101,111	I	BYTE ENABLES. Used to indicate which of the 4 bytes of the 32-bit data bus are involved with the current transfer cycle. These signals should be connected to the BE[3:0] pins of the VL connector.
LBSELn	83	Ood	LOCAL BUS SELECT. Active low simulated open drain output signal used to indicate to the system controller chipset that the current cycle is a video local bus cycle and the chipset should not respond to the CPU. This signal should be connected to the LDEVn pin of the VL connector.

Pin Name	Pin Number	Type	Description
ADSn	95	I	ADDRESS STATUS. Active low input used to indicate a valid address is on the bus. This signal should be connected to the ADSn pin of the VL connector.
SRDYIn	96	I	SYSTEM READY INPUT. Input from the chipset to indicate termination of a cycle. This signal should be connected to the RDYRTNn pin of the VL connector. For system without RDYRTNn signal, SRDYn should be routed back to SRDYIn.
SRDYn	97	Osod	SYSTEM READY. Tristateable active low output used to indicate the termination of a bus cycle. This signal should be driven high for one Proclk before being released. This signal should be driven during the first T2 state only when in Fast Write configuration. For regular write configuration and all read cycles, this signal should only be driven from the second T2 state onward. This signal should be connected to the LRDYn pin of the VL connector.
ISACMD	98	I	ISA COMMAND. This is a NAND of IORn, IOWn, MRDn. This signal is used to generate DACRDn, DACWRn, and DOEn.
WRn	99	I	WRITE/READ. Signal used to distinguish between a write (WRn high) or a read (WRn low) transfer. This signal should be connected to the W/Rn pin of the VL connector.
MIOn	100	I	MEMORY or I/O STATUS. Input from the bus to indicate the current cycle is a memory (MIOn high) transfer or an IO (MIOn low) transfer. This signal should be connected to the M/IOn pin of the VL connector.

Total: 76 pins

6.3 PCI Bus Interface

Pin Name	Pin Number	Type	Description
CINTn	66	Ood	CRT INTERRUPT REQUEST. Interrupt is asserted when vertical retrace occurs if it is enabled by bit 5 of the Vertical Retrace End (register 3?5 index 11) register. It is an active low open drain output.
RESETn	67	I	RESET. This signal is used to reset all internal state machines and some default registers. During RST active, all bidirectional buses are tri-stated. RESET signal is also used to latch-in the configuration register values. This signal is active low.
CLK	68	I	CLOCK. Used to provide timing for all trasactions on PCI. ALL other PCI signals are sampled on the rising edge of CLK, and all other timing parameters are defined with respect to this edge.

Pin Name	Pin Number	Type	Description
PAR	70	I/O	PARITY. Active high even parity across AD[31:0] and C/BEn[3:0]
GNTn	71	I	GRANT. Input from the bus arbiter to indicate that the request has been granted.
REQn	72	O	REQUEST. Output to the bus arbiter to request for the bus.
AD[31:0]	73-76,78-81 85-90,92-93 102-108,110 112-115,117-120	I/O	ADDRESS/DATA BITS 31:0. Address and Data are multiplexed. During the first clock of transaction AD[31:00] contain a physical byte address (32-bits). During subsequent clocks, AD[31:0] contain data.
C/BEn[3:0]	82,94,101,111	I/O	BUS COMMAND/BYTE ENABLES. Bus Command and Byte Enables are multiplexed. During the address phase of transaction, C/BEn define the bus command. During the data phase C/BEn are used as Byte Enables. The Byte Enables determine which bytes carry meaningful data. This bus is an input during slave mode and an output during master mode.
IDSEL	83	I	INITIALIZATION DEVICE SELECT. Active high chip select in lieu of the upper 24 address lines during configuration read and write transactions.
FRAMEn	95	I/O	CYCLE FRAME. Used to indicate the beginning and duration of an access. This signal is an input during slave mode and an output during master mode.
IRDYn	96	I/Osod	INITIATOR READY. IRDYn used to indicate the initiating agent's ability to complete the current data phase of the transaction. It is used in conjunction with TRDYn. This signal is an input during slave mode and an output during master mode.
TRDYn	97	I/Osod	TARGET READY. Used to indicate the target agent's ability to complete the current data phase of the transaction. TRDYn is used in conjunction with IRDYn. A data phase is completed on any clock when both TRDYn and IRDYn are asserted. During a read TRDYn indicates that valid data is present on AD[31:0]. During a write it indicates the target is prepared to accept data. Wait cycles are inserted until both IRDYn and TRDY are asserted together. This signal is an output during slave mode and an input during master mode.
DEVSELn	98	I/Osod	DEVICE SELECT. When driven active, indicates the driving device has decoded its address as the target of the current access. As an input it indicates whether any device on the bus has been selected. This signal is an output during slave mode and an input during master mode.
STOPn	99	I/Osod	STOP. Active low signal used by the current slave to request the current Master to stop the current transaction. This signal is an output during slave mode and an input during master mode.
LOCKn	100	I/Osod	LOCK. Active low signal used to indicate an atomic operation that may require multiple transactions to complete.

Total: 49 pins

6.4 BIOS ROM Interface

Pin Name	Pin Number	Type	Description
DOEn/ ROMENLn	45	O	DATA OUTPUT ENABLE. VL bus configuration, this is an active low signal to enable the low byte of BIOS data, DAC, and DIP SWITCH data to ISA data bus. Depending on the cycle, the selected device will drive the output data or will receive the input data. ROM ENABLE: PCI Bus configuration, this is an active low signal to enable the the ROM read. ISA bus 16-bit ROM configuration, this signal is used to enable the external buffer which routes the upper byte out to the system bus.

Total: 1 pin

6.5 Clock Interface

Pin Name	Pin Number	Type	Description
VCLK	2	I	VIDEO CLOCK. This is the master input pixel clock.
CSEL[3]	11	O	CLOCK SELECT LINE 3. Clock select lines are used to select the appropriate video clock frequency. This pin can be programmed through register 3DF index 6.
CSEL[2]	12	O	CLOCK SELECT LINE 2. Clock select lines are used to select the appropriate video clock frequency. This pin can be programmed through register 3DF index 6.
CSEL[1]	13	O	CLOCK SELECT LINE 1. Clock select lines are used to select the appropriate video clock frequency. This pin can be programmed through register 3DF index 6 or register 3C2.
CSEL[0]	14	O	CLOCK SELECT LINE 0. Clock select lines are used to select the appropriate video clock frequency. This pin can be programmed through register 3DF index 6 or register 3C2.
MCLK	196	I	MEMORY CLOCK. This is the input clock used for display memory timing.

Total: 6 pins

6.6 Feature Connector Interface

Pin Name	Pin Number	Type	Description
ESYNC	216	Ipu	ENABLE SYNC. This active high input is used to enable the BLANKn, HSYNC and VSYNC output signals.
EPCLK	220	Ipu	ENABLE PCLK. This active high input is used to enable the PCLK output to the SynDAC.
EPDATA	222	Ipu	ENABLE PDATA. This active high input is used to enable the P[23:0] Pixel Data output pins.

Total: 3 pins

6.7 DAC Interface

Pin Name	Pin Number	Type	Description
P[23:16]	3-10	O	PIXEL DATA. Upper 8-bits of pixel data. This bus exists on PCI & ISA buses only. It is not applicable for the VL-bus. These pins can be three-stated.
P[15:0]	224-229, 231-240	O	PIXEL DATA. Output data bus interfaces to the external synDAC chip for color mapping during active CRT display time. These pins can be three-stated.
PCLK	221	O	PIXEL CLOCK. Pixel clock output to the SynDAC to latch the pixel data. It is derived from the dot clock of the operating mode.
BLANKn	217	O	BLANK. Active low output signal to RAMDAC to blank the pixel data for the display monitor.
DACWRn	215	O	RAMDAC WRITE. An active low I/O write signal generated for writing external color palette registers. For VL bus configuration, this signal is also used to control data flow from ISA bus to and from the DAC, ROM, and DIP SWITCH. When DACWRn is high, data is output to the ISA bus, when DACRDn is low, data is input from the ISA bus.
DACRDn	2[illegible]	O	RAMDAC READ. An active low I/O read signal generated for reading external color palette registers.
SWSENSE	15	I	SWITCH SENSE. An input signal used to auto detect color or monochrome monitors.

Total: 29 pins for PCI and ISA buses, and 21 pins for VL bus

6.8 Monitor Interface

Pin Name	Pin Number	Type	Description
VSYNC	218	O	VERTICAL SYNC. Vertical synchronization pulse to the display monitor. The polarity of the pulse is determined by bit 7 of the Miscellaneous Output Register (register 3C2).
HSYNC	219	O	HORIZONTAL SYNC. Horizontal synchronization pulse to the display monitor. The polarity of the pulse is determined by bit 6 of the Miscellaneous Output Register (register 3C2).

Total: 2 pins

6.9 Auxiliary Bus Interface

Pin Name	Pin Number	Type	Description
ARDn	49	O	AUXILIARY READ. This pin is decoded from the system address and ORed with the I/O read command internally to generate register read for some other device in the graphic subsystem. This pin is valid for PCI & ISA buses only.
AWRn	50	O	AUXILIARY WRITE. This pin is decoded from the system address and ORed with the I/O write command internally to generate register write for some other device in the graphic subsystem. This pin is valid for PCI & ISA buses only.
BD[7:0]	51-58	I/O	AUXILIARY DATA BUS 7:0. Data bits 7-0 of BIOS high byte data in 16-bit BIOS ISA bus configuration. Data bit 7-0 in 8-bit BIOS ISA or PCI bus configuration. Can also be used for buffering DAC data and dipswitch data or other devices in the subsystem. The OTI-107 routes data from/to the BD bus out to/from SD/AD bus during read/write cycles. The BD bus is available in ISA and PCI bus interfaces only.
ARS[3:0]	59-60,62-63	O	AUXILIARY REGISTER SELECT 3:0. These pins are available for PCI bus only. System address bits 3:0 are latched from the PCI bus and sent out for devices in the graphic subsystem. The primary function for these pins is for the color palette, but they can be used for other devices as well. These pins are valid on PCI bus only.

Total: 14 pins for PCI bus, 10 pins for ISA, and 0 pin for VL bus

6.10 I^2C Interface

Pin Name	Pin Number	Type	Description
SRCK	16	I/O	SERIAL CLOCK.
SRD	44	I/O	SERIAL DATA.

Total: 2 pins. (These pins will not exist when the VMC bus is supported.)

6.11 Bt885 Support

Pin Name	Pin Number	Type	Description
VDVALID	46	O	VIDEO DATA VALID. Output used to indicate to the Video DAC that data is valid and ready to be clocked in.
MDMXn	47	O	MEMORY DATA MUX. Active low signal to mux MD[63:32] with MD[31:0] for the Video DAC. This pin exists for the PCI and ISA buses only. For VL bus, CASxn can be inverted to generate MDMXn.
RPLINE	48	I	REPEAT LINE. Active high signal used to indicate to the OTI-107 to repeat the previous line for video window. This is used for vertical zooming. This pin exists for the PCI and ISA buses only.

Total: 3 pins for PCI and ISA buses, and 1 pin for VL bus.

6.12 Display Memory Interface

Pin Name	Pin Number	Type	Description
MA[7:0]	170-163	O	MEMORY ADDRESS. Memory address bits 7:0 for all DRAM configurations.
RASLn	161	O	ROW ADDRESS STROBE LOW. Active low output signal used for all DRAM configurations.
RASHn/ MA[9]	172	O	ROW ADDRESS STROBE HIGH. Active low output signal used in 64-bit memory bus, 64Kx16 and 256KxXX DRAM configurations. MEMORY ADDRESS 9. For 512Kx8 and 1Mx4 DRAMs, this pin is memory address bit 9, and should be connected to all maps and all banks of DRAMs.
CASLn/ WELn	162	O	COLUMN ADDRESS STROBE LOW. For x4, x8 and x16 DRAMs with 1 CASn and 2 WEn, this pin is configured to be CASLn. WRITE ENABLE LOW. For x16 DRAMs with 2 CASn and 1 WEn, this pin is configured to be WELn.
CASHn/ WEH MA[8]	171	O	COLUMN ADDRESS STROBE HIGH. Used for 64-bit memory bus, 64Kx16 DRAM, 1 CASn, 2 WEn only. WRITE ENABLE HIGH. Used for 64-bit memory bus, 64Kx16 DRAM with 2 CASn & 1 WEn only. MEMORY ADDRESS 8. For 256KxXX, 512Kx8, and 1Mx4 DRAMs, this pin is memory address bit 8, and should be connected to all maps and all banks of DRAMs.

Pin Name	Pin Number	Type	Description
WEAn/ CASAn	122	O	WRITE ENABLE A. Active low write enable to memory map 0 and 1 in 16-bit MD configurations (4 256Kx4, 2 512Kx8, 1 256Kx16 w/ 1 CASn, 2 WEn), write enable for map 0 in 32-bit MD configuration and 64-bit MD configuration when MA=XXx000. COLUMN ADDRESS STROBE A. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASAn controlling the same maps as WEAn.
WEBn/ CASBn	131	O	WRITE ENABLE B. Active low write enable pulse to memory map 2 and 3 in 16-bit MD configurations (4 256Kx4, 2 512Kx8, 1 256Kx16 w/ 1 CASn, 2 WEn), write enable for map 2 in 32-bit MD configuration and 64-bit MD configuration when MA=XXx001. COLUMN ADDRESS STROBE B. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASBn controlling the same maps as WEBn.
WECn/ CASCn	141	O	WRITE ENABLE C. Active low write enable pulse to memory map 1 in 32-bit MD configuration and 64-bit MD configuration when MA=XXx010. COLUMN ADDRESS STROBE C. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASCn controlling the same maps as WECn.
WEDn/ CASDn	150	O	WRITE ENABLE D. Active low write enable pulse to memory map 3 in 32-bit MD configuration and 64-bit MD configuration when MA=XXx011. COLUMN ADDRESS STROBE D. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASDn controlling the same maps as WEDn.
WEEn/ CASEn	173	O	WRITE ENABLE E. Active low write enable to memory map 0 in 64-bit MD configuration when MA=XXx100. COLUMN ADDRESS STROBE E. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASEn controlling the same maps as WEEn.
WEFn/ CASFn	183	O	WRITE ENABLE F. Active low write enable to memory map 1 in 64-bit MD configuration when MA=XXx101. COLUMN ADDRESS STROBE F. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASFn controlling the same maps as WEFn.
WEGn/ CASGn	193	O	WRITE ENABLE G. Active low write enable to memory map 2 in 64-bit MD configuration when MA=XXx110. COLUMN ADDRESS STROBE G. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASGn controlling the same maps as WEGn.
WEHn/ CASHn	204	O	WRITE ENABLE H. Active low write enable to memory map 3 in 64-bit MD configuration when MA=XXx111. COLUMN ADDRESS STROBE H. For x16 DRAMs with 2 CASn, 1 WEn, this pin is CASHn controlling the same maps as WEHn. Note that this is WEH or CASH as opposed to WE HIGH or CAS HIGH.

Pin Name	Pin Number	Type	Description
MD[63:0]	213-207,205, 203-200,198-197 195-194,191-184 182,180-174, 159-154, 152-151,149-142, 139-132,130-123	I/O	MEMORY DATA. Memory data bits 63-0. MD[23:0] are also used for the configuration register during hardware reset. MD[7:0] correspond to bits 7:0 of Configuration Register 1, MD[15:8] correspond to bits 7:0 of Configuration Register 2, and MD[23:16] correspond to bits 7:0 of Configuration Register 3.

Total: 84 pins

6.13 EEPROM Interface

Pin Name	Pin Number	Type	Description
EEPRD	14	I	EEPROM READ DATA. Data can be read from the EEPROM through the data read bit in the register 3DF index 18.
EEPWD	13	O	EEPROM WRITE DATA. Data can be written to the EEPROM through the data bit in the register 3DF index 18.
EEPSK	12	O	EEPROM SHIFT CLOCK. This clock can be toggled through register 3DF index 18.
EEPCS	11	O	EEPROM CHIP SELECT. This signal is used to enable the serial EEPROM for read and write operations.

Total: 0 pins. (pins EEPCS, EEPRD, EEPWD, and EEPSK are muxed with CSEL[3:0])

6.14 Multimedia Interface Option 1

Pin Name	Pin Number	Type	Description
MMVRSETn	17	I	MULTIMEDIA VERTICAL RESET. Active low input used to indicate the beginning of a new frame.
MMHRSETn	18	I	MULTIMEDIA HORIZONTAL RESET. Active low input used to indicate the beginning of a new line.
MMDVALID	21	I	MULTIMEDIA DATA VALID. Active high input used to indicate that data is ready to be latched in.
MMCLK	23	I	MULTIMEDIA DATA CLOCK. Input clock used to clock-in MMD[15:0].
MMD[15:0]	43-42,40-27	I	MULTIMEDIA PORT DATA. Multimedia port input data bits 15:0. This data is latched in at the rising edge of MMCLK.
MMFIELD	22	I	MULTIMEDIA FIELD. For odd or even fields.

Total: 21 pins

6.15 VESA™ MediaChannel-Multimedia Interface Option 2 (future version)

Pin Name	Pin Number	Type	Description
SAn	16	I/O	SERIAL LINE A - Serial I/O line used to link devices together in a chain. It is configured during the device ID writing phase, and is used to ensure that each device receives a unique ID.
BSn[1:0]	18-17	I/O	MEDIA CHANNEL BUS SIZE. These two signals are used to downsize the bus from 32 to 16-bits.
SNRDYn	20	Ood	SLAVE NOT READY - Active low open drain output used to indicate that the OTI-107 is not ready to receive data.
CNTRL	21	I	MEDIA CHANNEL CONTROL. Used to indicate that a Control Cycle rather than a Data transfer is taking place on the next transfer.
MCRESETn	22	I	MEDIA CHANNEL PORT RESET - Active low reset signal generated from the Media Channel Bus.
MCCLK	23	I	MEDIA CHANNEL CLOCK. Input clock used to clock-in MMD[15:0]
MASK[1:0]	25-24	I	MEDIA CHANNEL MASK BIT 0 - Input used to indicate whether the accompanying pixel should be displayed. This signal is combined with the internal Multimedia Mask Map to form the final mask. This signal is also used with SA & SB during the configuration process to assign IDs.
MCD[15:0]	43-42,40-27	I	MEDIA CHANNEL PORT DATA. Multimedia port input data bits 15:0. This data is latched in at the rising edge of MMCLK.
SBn	44	I/O	SERIAL LINE B - Serial I/O line used to link devices together in a chain. It is configured during the device ID writing phase, and is used to ensure that each device receives a unique ID.

Total: 26 pins

6.16 Power and Ground

Pin Name	Pin Number	Type	Description
VSSO0	1	GNDO	EXTERNAL GROUND
VSSO1	19	GNDO	EXTERNAL GROUND
VSSO2	41	GNDO	EXTERNAL GROUND
VSSO3	61	GNDO	EXTERNAL GROUND
VSSO4	91	GNDO	EXTERNAL GROUND
VSSO5	109	GNDO	EXTERNAL GROUND
VSSO6	121	GNDO	EXTERNAL GROUND
VSSO7	140	GNDO	EXTERNAL GROUND
VSSO8	160	GNDO	EXTERNAL GROUND
VSSO9	181	GNDO	EXTERNAL GROUND
VSSO10	206	GNDO	EXTERNAL GROUND
VSSO11	230	GNDO	EXTERNAL GROUND
VSSI0	77	GNDI	INTERNAL GROUND
VSSI1	192	GNDI	INTERNAL GROUND
VDD0	26	VDDB	EXTERNAL & INTERNAL POWER
VDD1	84	VDDB	EXTERNAL & INTERNAL POWER
VDD2	116	VDDB	EXTERNAL & INTERNAL POWER
VDD3	153	VDDB	EXTERNAL & INTERNAL POWER
VDD4	199	VDDB	EXTERNAL & INTERNAL POWER
VDD5	223	VDDB	EXTERNAL & INTERNAL POWER

Total: 20 pins

Pin Count for VL Bus:

System Interface	76
Memory Interface	84
DAC and Monitor	23
Feature Connector	3
ROM Interface	1
EEPROM Interface	0
Clock Interface	6
Auxiliary Bus	0
Multimedia	27
Power and Ground	20
Total	**240**

Pin Count for PCI Bus:

System Interface	49
Memory Interface	84
DAC and Monitor	31
Feature Connector	3
ROM Interface	1
EEPROM Interface	0
Clock Interface	6
Auxiliary Bus	14
Multimedia	29
Power and Ground	20
Total	**237**

Pin Count for ISA Bus:

System Interface	55
Menory Interface	84
DAC and Monitor	31
Feature Connector	3
ROM Interface	1
EEPROM Interface	0
Clock Interface	6
Auxiliary Bus	10
Multimedia	29
Power and Ground	20
Total	**239**

Pin Type:

I	Input
I_{pu}	Input with internal pull-up
O	Output
I/O	Input/Output
O_{od}	Open Drain output
O_{sod}	Simulated open drain output-output should be driven high for one system clock before it is released.

6.17 Pin Cross Reference Table

Pin Number	Type	Drive (DC)	Pad Name	ISA	PCI	VL	Future Version * (VMC Option)
1	GNDO		PV0A	VSSO0	VSSO0	VSSO0	
2	I		PT5D01	VCLK	VCLK	VCLK	
3-10	I/O	4mA	PT5B02	P[23:16]	P[23:16]	SA[31:24]	
11	O	2mA	PT5O01	CSEL[3]/ EEPCS	CSEL[3]/ EEPCS	CSEL[3]/ EEPCS	
12	O	2mA	PT5O01	CSEL[2]/ EEPSK	CSEL[2]/ EEPSK	CSEL[2]/ EEPSK	
13	O	2mA	PT5O01	CSEL[1]/ EEPWD	CSEL[1]/ EEPWD	CSEL[1]/ EEPWD	
14	I/O	2mA	PT5B01	CSEL[0]/ EEPRD	CSEL[0]/ EEPRD	CSEL[0]/ EEPRD	
15	I		PT5D01	SWSENSE	SWSENSE	SWSENSE	
16	I/O	4mA	PT5B02	SRCK	SRCK	SRCK	SAn
17	I		PT5D01	MMVRSETn	MMVRSETn	MMVRSETn	BSn[0]
18	I		PT5D01	MMHRSETn	MMHRSETn	MMHRSETn	BSn[1]
19	GNDO		PV0A	VSSO1	VSSO1	VSSO1	
20	Ood	4mA	PT5T02	NC	NC	NC	SNRDYn
21	I		PT5D01	MMDVALID	MMDVALID	MMDVALID	CNTRL
22	I		PT5D01	MMFIELD	MMFIELD	MMFIELD	IRESETn
23	I		PT5D01	MMCLK	MMCLK	MMCLK	
24-25	I		PT5D01	NC	NC	NC	MASK[0:1]
26	VDDB		PVDF	VDD0	VDD0	VDD0	
27-40	I		PT5D01	MMD[0:13]	MMD[0:13]	MMD[0:13]	
41	GNDO		PV0A	VSSO2	VSSO2	VSSO2	
42-43	I		PT5D01	MMD[15:14]	MMD[15:14]	MMD[15:14]	
44	I/O	4mA	PT5B02	SRD	SRD	SRD	SBn
45	O	2mA	PT5O01	ROMENLn	ROMENLn	DOEn	

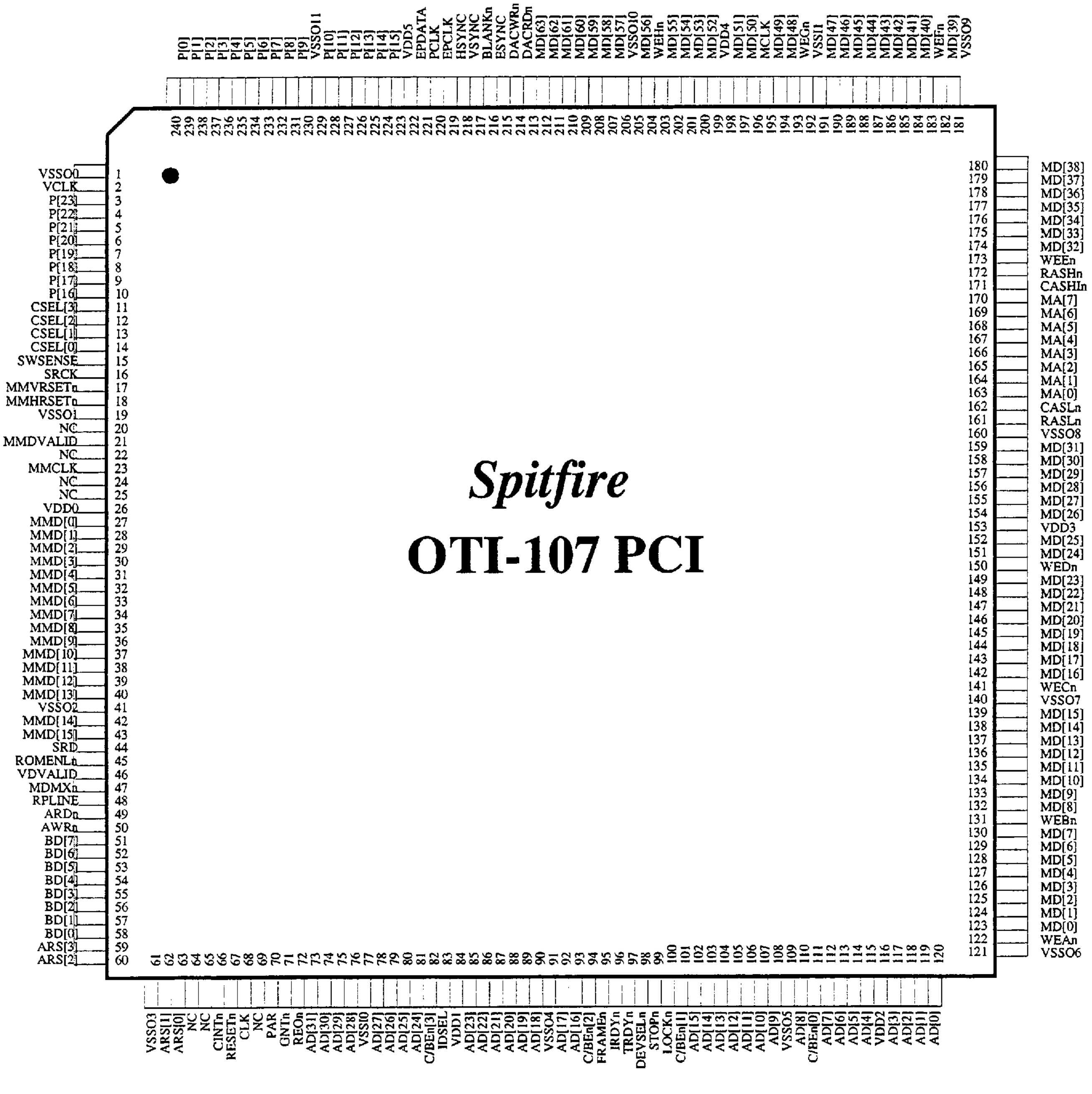
Pin Number	Type	Drive (DC)	Pad Name	ISA	PCI	VL	Future Version * (VMC Option)
46	O	2mA	PT5O01	VDVALID	VDVALID	VDVALID	
47	I/O	2mA	PT5O01	MDMXn	MDMXn	SA[2]	
48	I		PT5D01	RPLINE	RPLINE	SA[3]	
49	I/O	2mA	PT5B01	ARDn	ARDn	SA[4]	
50	I/O	2mA	PT5B01	AWRn	AWRn	SA[5]	
51-58	I/O	2mA	PT5B01	BD[7:0]	BD[7:0]	SA[6:13]	
59	I/O	2mA	PT5B01	NC	ARS[3]	SA[14]	AALH
60	I/O	2mA	PT5B01	MASTERn	ARS[2]	SA[15]	AALL
61	GNDO		PV0A	VSSO3	VSSO3	VSSO3	
62	I/O	2mA	PT5B01	MWRn	ARS[1]	SA[16]	NC
63	I/O	2mA	PT5B01	MRDn	ARS[0]	SA[17]	NC
64	I		PT5D01	LA[17]	NC	SA[18]	
65	I		PT5D01	LA[18]	NC	SA[19]	
66	O	24mA	PT5T04	CINTn	CINTn	CINTn	
67	I		PT5D01	RESETn	RESETn	RESETn	
68	I/O	24mA	PT5B04	M16n	CLK	PROCLK	
69	I/O	24mA	PT5B04	ZEROWSn	NC	SA[20]	
70	I/O	8mA	PT5B03	LA[19]	PAR	SA[21]	
71	I/O	24mA	PT5B04	IO16n	GNTn	SA[22]	
72	I/O	8mA	PT5B03	LA[20]	REQn	SA[23]	
73	I/O	8mA	PT5B03	LA[21]	AD[31]	SD[31]	
74	I/O	8mA	PT5B03	LA[22]	AD[30]	SD[30]	
75	I/O	8mA	PT5B03	LA[23]	AD[29]	SD[29]	
76	I/O	8mA	PT5B03	BHEn	AD[28]	SD[28]	
77	GNDI		PV0B	VSSI0	VSSI0	VSSI0	
78	I/O	8mA	PT5B03	ALE	AD[27]	SD[27]	
79	I/O	8mA	PT5B03	RFSHn	AD[26]	SD[26]	

Pin Number	Type	Drive (DC)	Pad Name	ISA	PCI	VL	Future Version * (VMC Option)
80	I/O	8mA	PT5B03	IORn	AD[25]	SD[25]	
81	I/O	8mA	PT5B03	IOWn	AD[24]	SD[24]	
82	I/O	24mA	PT5B04	IOCHRDY	C/BEn[3]	BEn[3]	
83	I/O	8mA	PT5B03	SA[0]	IDSELn	LBSELn	
84	VDDB		PVDF	VDD1	VDD1	VDD1	
85	I/O	8mA	PT5B03	SA[1]	AD[23]	SD[23]	
86	I/O	8mA	PT5B03	SA[2]	AD[22]	SD[22]	
87	I/O	8mA	PT5B03	SA[3]	AD[21]	SD[21]	
88	I/O	8mA	PT5B03	SA[4]	AD[20]	SD[20]	
89	I/O	8mA	PT5B03	SA[5]	AD[19]	SD[19]	
90	I/O	8mA	PT5B03	SA[6]	AD[18]	SD[18]	
91	GNDO		PV0A	VSSO4	VSSO4	VSSO4	
92	I/O	8mA	PT5B03	SA[7]	AD[17]	SD[17]	
93	I/O	8mA	PT5B03	SA[8]	AD[16]	SD[16]	
94	I/O	8mA	PT5B03	SA[9]	C/BEn[2]	BEn[2]	
95	I/O	8mA	PT5B03	SA[10]	FRAMEn	ADSn	
96	I/O	8mA	PT5B03	SA[11]	IRDYn	SRDYIn	
97	I/O	8mA	PT5B03	SA[12]	TRDYn	SRDYn	
98	I/O	8mA	PT5B03	SA[13]	DEVSELn	ISACMD	
99	I/O	8mA	PT5B03	SA[14]	STOPn	WRn	
100	I/O	8mA	PT5B03	SA[15]	LOCKn	MIOn	
101	I/O	8mA	PT5B03	SA[16]	C/BEn[1]	BEn[1]	
102-108	I/O	24mA	PT5B04	SD[15:9]	AD[15:9]	SD[15:9]	
109	GNDO		PV0A	VSSO5	VSSO5	VSSO5	
110	I/O	24mA	PT5B04	SD[8]	SD[8]	SD[8]	
111	I/O	8mA	PT5B03	AEN	C/BEn[0]	BEn[0]	
112-115	I/O	24mA	PT5B04	SD[7:4]	AD[7:4]	SD[7:4]	
116	VDDB		PVDF	VDD2	VDD2	VDD2	
117-120	I/O	24mA	PT5B04	SD[3:0]	AD[3:0]	SD[3:0]	

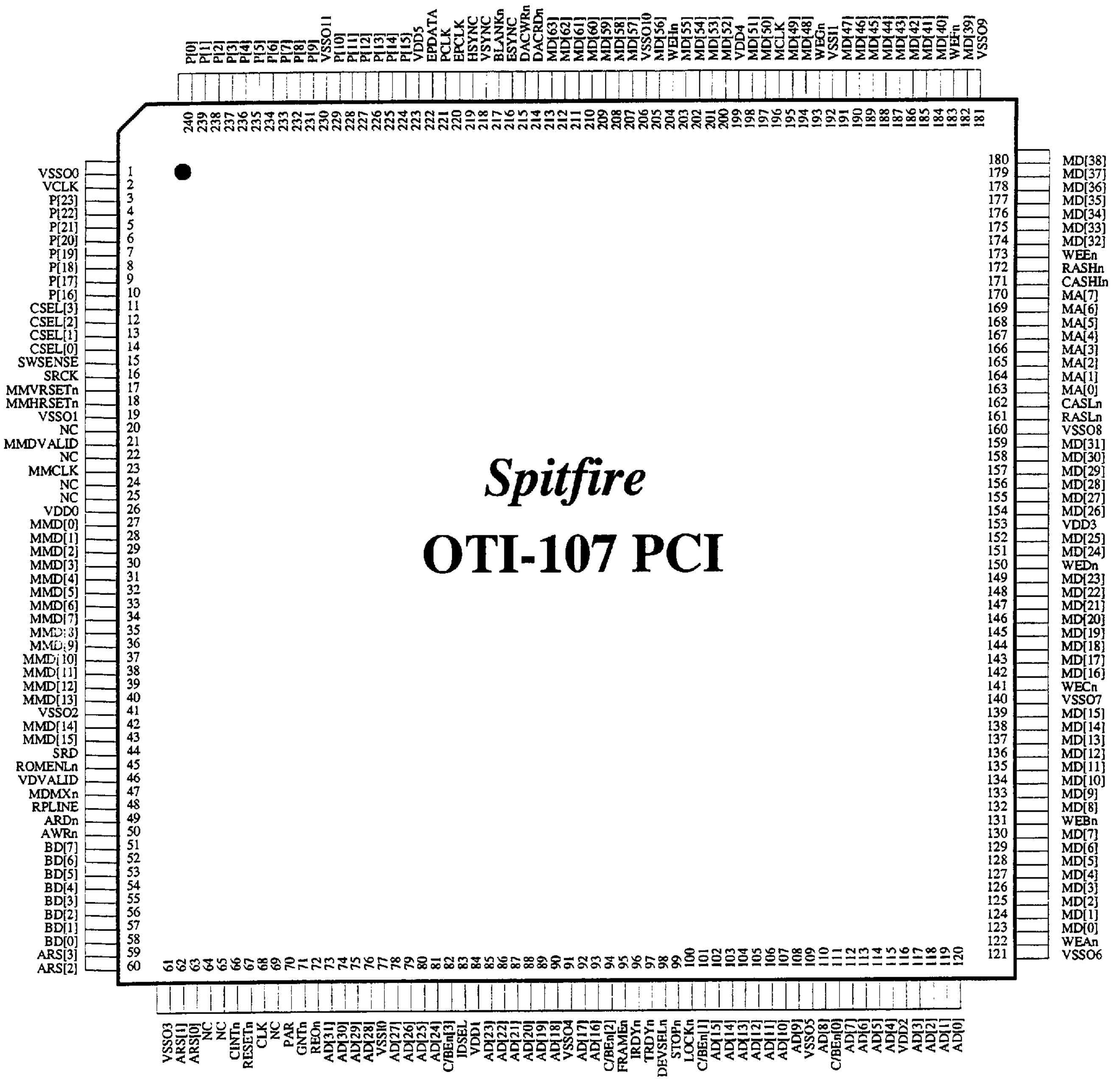
Pin Number	Type	Drive (DC)	Pad Name	ISA	PCI	VL	Future Version * (VMC Option)
121	GNDO		PV0A	VSSO6	VSSO6	VSSO6	
122	O	4mA	PT5O02	WEAn/CASAn	WEAn/CASAn	WEAn/CASAn	
123-130	I/O	2mA	PT5B01	MD[0:7]	MD[0:7]	MD[0:7]	
131	O	4mA	PT5O02	WEBn/CASBn	WEBn/CASBn	WEBn/CASBn	
132-139	I/O	2mA	PT5B01	MD[8:15]	MD[8:15]	MD[8:15]	
140	GNDO		PV0A	VSSO7	VSSO7	VSSO7	
141	O	4mA	PT5O02	WECn/CASCn	WECn/CASCn	WECn/CASCn	
142-149	I/O	2mA	PT5B01	MD[16:23]	MD[16:23]	MD[16:23]	
150	O	4mA	PT5O02	WEDn/CASDn	WEDn/CASDn	WEDn/CASDn	
151-152	I/O	2mA	PT5B01	MD[24:25]	MD[24:25]	MD[24:25]	
153	VDDB		PVDF	VDD3	VDD3	VDD3	
154-159	I/O	2mA	PT5B01	MD[26:31]	MD[26:31]	MD[26:31]	
160	GNDO		PV0A	VSSO8	VSSO8	VSSO8	
161	O	8mA	PT5O03	RASLn	RASLn	RASLn	
162	O	8mA	PT5O03	CASLn/WELn	CASLn/WELn	CASLn/WELn	
163-170	O	8mA	PT5O03	MA[0:7]	MA[0:7]	MA[0:7]	
171	O	8mA	PT5O03	CASHIn/ WEHIn/MA[8]	CASHIn/ WEHIn/MA[8]	CASHIn/ WEHIn/MA[8]	
172	O	8mA	PT5O03	RASHn/MA[9]	RASHn/MA[9]	RASHn/MA[9]	
173	O	4mA	PT5O02	WEEn/CASEn	WEEn/CASEn	WEEn/CASEn	
174-180	I/O	2mA	PT5O01	MD[32:38]	MD[32:38]	MD[32:38]	
181	GNDO		PV0A	VSSO9	VSSO9	VSSO9	
182	I/O	2mA	PT5B01	MD[39]	MD[39]	MD[39]	
183	O	4mA	PT5O02	WEFn/CASFn	WEFn/CASFn	WEFn/CASFn	
184-191	I/O	2mA	PT5B01	MD[40:47]	MD[40:47]	MD[40:47]	
192	GNDI		PV0B	VSSI1	VSSI1	VSSI1	
193	O	4mA	PT5O02	WEGn/CASGn	WEGn/CASGn	WEGn/CASGn	
194-195	I/O	2mA	PT5B01	MD[48:49]	MD[48:49]	MD[48:49]	

Pin Number	Type	Drive (DC)	Pad Name	ISA	PCI	VL	Future Version * (VMC Option)
196	I		PT5D01 + PC5C01	MCLK	MCLK	MCLK	
197-198	I/O	2mA	PT5B01	MD[50:51]	MD[50:51]	MD[50:51]	
199	VDDB		PVDF	VDD4	VDD4	VDD4	
200-203	I/O	2mA	PT5B01	MD[52:55]	MD[52:55]	MD[52:55]	
204	O	4mA	PT5O02	WEHn/CASH	WEHn/CASH	WEHn/CASHn	
205	I/O	2mA	PT5B01	MD[56]	MD[56]	MD[56]	
206	GNDO		PV0A	VSSO10	VSSO10	VSSO10	
207-213	I/O	2mA	PT5B01	MD[57:63]	MD[57:63]	MD[57:63]	
214	O	2mA	PT5O01	DACRDn	DACRDn	DACRDn	
215	O	2mA	PT5O01	DACWRn	DACWRn	DACWRn	
216	I		PT5D01U	ESYNC	ESYNC	ESYNC	
217	O	4mA	PT5T02	BLANKn	BLANKn	BLANKn	
218	O	8mA	PT5T03	VSYNC	VSYNC	VSYNC	
219	O	8mA	PT5T03	HSYNC	HSYNC	HSYNC	
220	I		PT5D01U	EPCLK	EPCLK	EPCLK	
221	O	8mA	PT5T03	PCLK	PCLK	PCLK	
222	I		PT5D01U	EPDATA	EPDATA	EPDATA	
223	VDDB		PVDF	VDD5	VDD5	VDD5	
224-229	O	8mA	PT5T03	P[15:10]	P[15:10]	P[15:10]	
230	GNDO		PV0A	VSSO11	VSSO11	VSSO11	
231-240	O	8mA	PT5T03	P[9:0]	P[9:0]	P[9:0]	

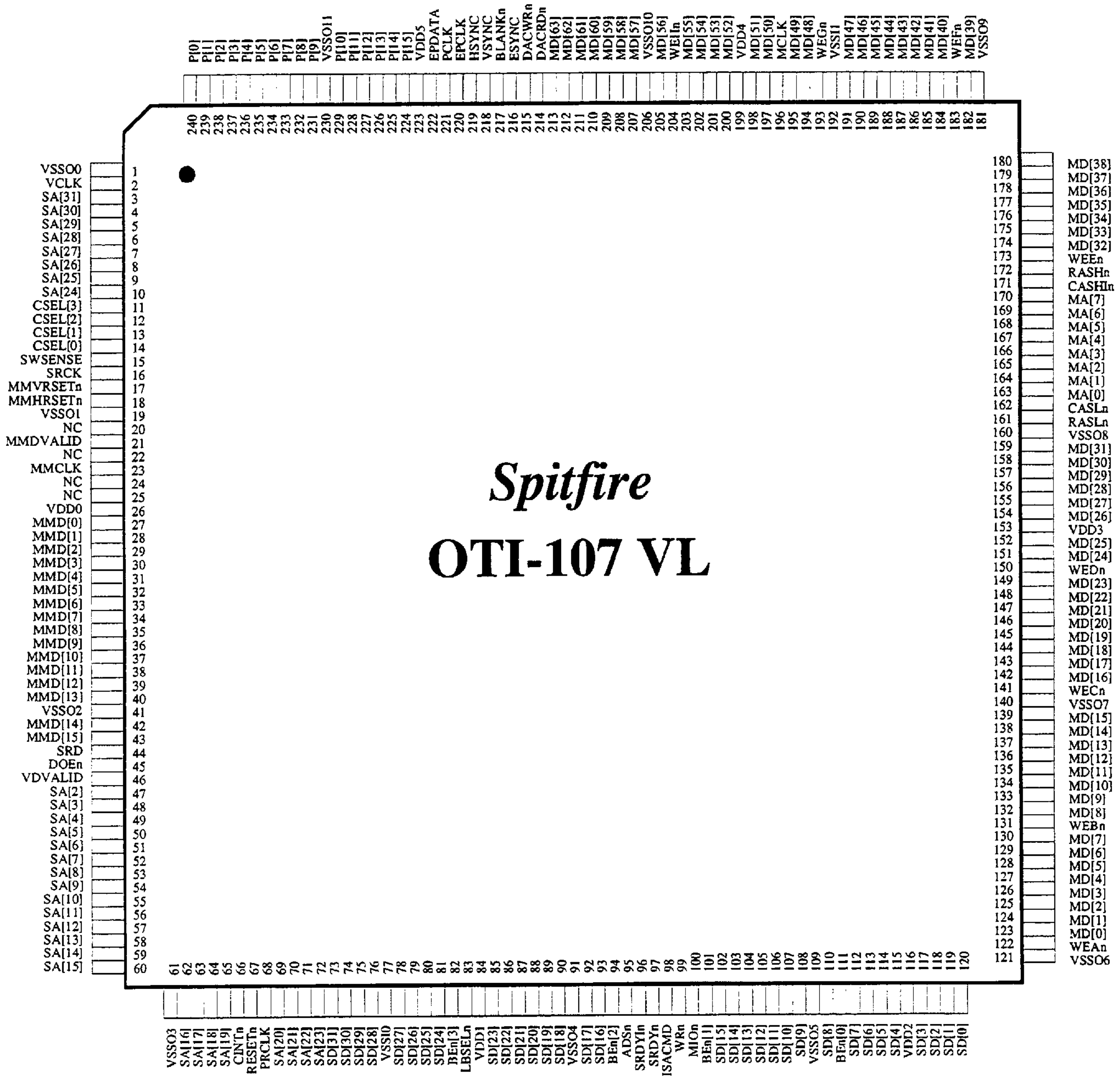
6.18 ISA Bus Pin Diagram - 240 pins



6.19 PCI bus Pin Diagram - 240 pins



6.20 VL Bus Pin Diagram - 240 pins



6.21 Multimedia Connector

There are two versions of the multimedia port: 1) Multimedia Port point-to-point input only and 2) VMC connector which will be supported for future versions of the Spitfire OTI-64107 family of parts. The connector is a 68-pin high density type, utilizing 0.050" pin spacing. The part number for AMP is 5-175473-8 or equivalent.

Multimedia Port	Pin Number	VMC Connector/OTI-64107 Pins
SRCK	1	SAn / SAn
NC	2	NC
NC	3	NC
GND	4	GND
MMVRSETn	5	BSn[0]/BSn[0]
MMHRSETn	6	BSn[1]/BSn[1]
GND	7	GND
NC	8	SNRDYn/SNRDYn
MMDVALID	9	CONTROL/CNTRL
GND	10	GND
NC	11	RESETn/IRESETn
GND	12	GND
MMCLK	13	DCLK/IMCLK
GND	14	GND
NC	15	NC
GND	16	GND
NC	17	M0/MASK[0]
NC	18	M1/MASK[1]
GND	19	GND
MMD[0]	20	P0/MMD[0]
MMD[1]	21	P1/MMD[1]
GND	22	GND
MMD[2]	23	P2/MMD[2]
MMD[3]	24	P3/MMD[3]

Multimedia Port	Pin Number	VMC Connector/OTI-64107 Pins
GND	25	GND
MMD[4]	26	P4/MMD[4]
MMD[5]	27	P5/MMD[5]
GND	28	GND
MMD[6]	29	P6/MMD[6]
MMD[7]	30	P7/MMD[7]
GND	31	GND
NC	32	P8
NC	33	P9
GND	34	GND
NC	35	P10
NC	36	P11
GND	37	GND
NC	38	P12
NC	39	P13
GND	40	GND
NC	41	P14
NC	42	P15
GND	43	GND
MMD[8]	44	P16/MMD[8]
MMD[9]	45	P17/MMD[9]
GND	46	GND
MMD[10]	47	P18/MMD[10]
MMD[11]	48	P19/MMD[11]
GND	49	GND

Multimedia Port	Pin Number	VMC Connector/OTI-64107 Pins
MMD[12]	50	P20/MMD[12]
MMD[13]	51	P21/MMD[13]
GND	52	GND
MMD[14]	53	P22/MMD[14]
MMD[15]	54	P23/MMD[15]
GND	55	GND
NC	56	P24
NC	57	P25
GND	58	GND
NC	59	P26
NC	60	P27
GND	61	GND
NC	62	P28
NC	63	P29
GND	64	GND
NC	65	P30
NC	66	P31
GND	67	GND
SRD	68	SBn

CHAPTER 7: OTI-64107 REGISTER DEFINITION

7.1 Conventions

The naming convention for the registers is as follows:

readport/writeport register name index read/write

Default - some registers or register bits should be reset or preset to the default value at boot-up time. Hardware reset (comes from the reset pin, as opposed to software reset which is generated from the Sequencer Reset register) should be used to set or preset these registers. Registers with no default values are not initialized during reset, and should be initialized by software before reading their contents.

Reserved - reserved bits should be implemented with tristate buffers only to save gates, and should always return 0 when read.

AR - Attribute Controller registers: 3C0, 3C1

CR - CRT Controller registers: 3B4, 3B5, 3D4, 3D5

CFR - Configuration registers for PCI bus: 0-3C

ER - Extended registers: 3DE, 3DF

GR - Graphics Controller registers: 3CE, 3CF

HR - Hardware Cursor registers: memory mapped

PR - Co-Processor registers: memory mapped

SR - Sequencer registers: 3C4, 3C5

R/W - register is readable and writeable

RO - register is read only

WO - register is write only

xRii[xx] - xR stands for register name, ii stands for index, [xx] represents bit number. For example, SR1[2:5,7] stands for Sequencer register index 1 Bits 2 through 5 and Bit 7.

3?X - the "?" here stands for "B" if in monochrome emulation modes, and "D" if in color emulation modes as determined by Bit 0 of Miscellaneous Output Register.

Register Order:

Address(last 2bits)	00	01	10	11
	Byte 0	Byte 1	Byte 2	Byte 3

Example: Pixel Map n Base Address Register

14	15	16	17
Least significant byte			Most significant byte

7.2 Enable Registers

There are two registers that control the enable/disable mechanism of the OTI-64107 product on ISA and VL bus. One register controls access to the on-board video subsystem and the other register controls access to the add-on video subsystem. The OTI-64107 chip contains both add-on and on-board registers. The register that controls on-board operation may also reside in the system chipset or on the motherboard. The addresses for both register sets are:

Add-on configuration:	46E8	enable/disable (in OTI-64107)
On-board configuration:	3C3	enable/disable (in OTI-64107 and in some system chipsets)

The setup/enable register sets 46E8/102 and 94/102 will not be implemented in this chip because of potential I/O conflicts with some system chipsets. Only enable registers 46E8 and 3C3 are implemented.

46E8 Add-on Enable Register **R/W**

This register serves as the POS (power on setup) equivalent on MCA machines.

Bit	Description
2-0	Reserved
3	0: Disable read/write access to OTI-64107 registers, DAC registers, and video memory 1: Enable read/write access to OTI-64107 registers, DAC registers, and video memory (normal operational mode)
7-4	Reserved

Default: 0h

3C3 On-board Enable Register **WO**

This register may also reside inside some system chipsets. To avoid possible conflict, this register is written in one clock (zero wait state), and signals LBSELn and SRDYn are not asserted. SRDYn should be asserted by the system chipset because either it has 3C3 or it has to pass this I/O cycle to the ISA bus.

Bit	Description
0	0: Disable read/write access to OTI-64107 registers, DAC registers, and video memory 1: Enable read/write access to OTI-64107 registers, DAC registers, and video memory (normal operational mode)
7-1	Reserved

Default: 0h

For the PCI bus, enable/disable is done through the Configuration Register 4 bits 1:0 for both add-on boards and on the mother board. Therefore, 46E8[3] and 3C3[0] have no effect when the chip is configured to be on the PCI bus.

7.3 Extended Register Summary

The OTI-64107 extended registers are NOT downward compatible with any of the previous chips from OAK. New registers are defined in the OTI-64107 to support enhanced functions and features which provide flexibility for different VGA configurations. A summary of these extended registers is given below.

7.3.1 Extended General Registers

Register Port	R/W	Port	Index	Bits	Block
Extension Address Register	R/W	3DE	-	8	BI
Backward Compatible ID Register	RO	3DF	0	8	BI
Backward Compatible Chip Version Register	RO	3DF	1	8	BI
Status Register	R/W	3DF	2	3	BI
OTI Test Register 1	R/W	3DF	3	8	BI
OTI Test Register 2	R/W	3DF	4	0	BI
Clock Select Register/Scratch	R/W	3DF	6	8	BI
Configuration Register 1	RO	3DF	7	8	BI
Configuration Register 2	RO	3DF	8	8	BI
Configuration Register 3	RO	3DF	9	0	BI
Interrupt Control Register	R/W	3DF	B	8	BI
I2C Control Register	R/W	3DF	C	4	BI
Dip Switch Read	RO	3DF	D	8	BI
EEPROM Control Register	R/W	3DF	E	4	BI
Power Saving Register	R/W	3DF	F	3	CRT
Scratch Register 0	R/W	3DF	F0	8	BI
Scratch Register 1	R/W	3DF	F1	8	BI
Scratch Register 2	R/W	3DF	F2	8	BI
Scratch Register 3	R/W	3DF	F3	8	BI
Scratch Register 4	R/W	3DF	F4	8	BI
Scratch Register 5	R/W	3DF	F5	8	BI
Scratch Register 6	R/W	3DF	F6	8	BI
Scratch Register 7	R/W	3DF	F7	8	BI

7.3.2 Extended System Interface Registers

There are two groups of Extended System Interface registers:

PCI compatible registers that locate at Configuration Register (CFR) space when on PCI bus, and at 2x00-2x3C when on ISA or VL bus. Some of the bits in these registers are good for all buses, and some are for PCI only. The "x" in the register address is reconfigurable through ER19b7:4 during hardware reset.

OTI registers reside at 3DF index 10-1F for all bus configurations. (N/A stands for not applicable for that bus configuration.)

Register Port	R/W	Port/Index (ISA,VL)	CFR (PCI)	Bits	Block
Vendor ID Register	RO	2x00	0	16	BI
Device ID Register	RO	2x02	2	16	BI
System Bus Command Register	R/W	2x04	4	6	BI
System Bus Status Register	R/W	N/A	6	6	BI
Revision ID Register	RO	2x08	8	8	BI
Programming Interface Register	RO	N/A	B-9	24	BI
Cache Line Size Register	RO	N/A	C	0	BI
Latency Timer Register	R/W	N/A	D	0	BI
Header Type Register	RO	N/A	E	8	BI
Built-In Self Test Register	R/W	N/A	F	0	BI
Memory Mapped I/O Base Address Register	R/W	2x10	10	32	BI
Memory Base Address Register	R/W	2x14	14	32	BI
Auxiliary & DAC I/O Base Address Register	R/W	N/A	18	32	BI
Base Address Registers 5-3	R/W	N/A	27-1C	0	BI
Reserved Registers	RO	N/A	2F-28	0	BI
BIOS ROM Base Address Register	R/W	2x30	30	32	BI
Reserved Registers	RO	N/A	3B-34	0	BI
Interrupt Line Register	R/W	2x3C	3C	8	BI
Interrupt Pin Register	RO	N/A	3D	8	BI
Min_Gnt Register	RO	N/A	3E	8	BI
Max_Lat Register	RO	N/A	3F	8	BI
Local Bus Control 1	R/W	3DF/10	N/A	7	BI

Register Port	R/W	Port/Index (ISA,VL)	CFR (PCI)	Bits	Block
ISA Bus Control	R/W	3DF/13	N/A	3	BI
Memory Mapping Register	R/W	3DF/14	N/A	7	BI
Memory & Memory Mapped I/O Enable DAC & Auxiliary Command Control Register	R/W	3DF/15	N/A	2	BI
Configuration/Auxiliary/DAC Address Range	R/W	3DF/19	N/A	6	BI

7.3.3 Extended Sequencer Registers

Register Port	R/W	Port	Index	Bits	Block
Compatible Segment Register	R/W	3DF	11	8	SEQ
FIFO Depth Register	R/W	3DF	20	6	SEQ
Mode Select Register	R/W	3DF	21	7	SEQ
Feature Select Register	R/W	3DF	22	3	SEQ
Extended Read Segment Register	R/W	3DF	23	7	SEQ
Extended Write Segment Register	R/W	3DF	24	7	SEQ
Extended Common Read Write Register	R/W	3DF	25	7	SEQ
RASn Control Register	R/W	3DF	26	3	SEQ
CASn Control Register	R/W	3DF	27	5	SEQ
Refresh Control Register	R/W	3DF	28	3	SEQ
Hardware Window Arbitration Register	R/W	3DF	29	2	SEQ

7.3.4 Extended CRT Controller Registers

Register Port	R/W	Port	Index	Bits	Block
OTI CRT Overflow Register	R/W	3DF	30	4	CRTC
CRT Start Address Hi Register	R/W	3DF	31	7	CRTC
HSYNC/2 Start Register	R/W	3DF	32	8	CRTC
CRT Address Compatibility Register	R/W	3DF	33	1	CRTC

7.3.5 Extended Attribute Controller Registers

Register Port	R/W	Port	Index	Bits	Block
Pixel Interface Register	R/W	3DF	38	6	ATR
Extended Overscan Color Register 1	R/W	3DF	39	8	ATR
Extended Overscan Color Register 2	R/W	3DF	3A	8	ATR

7.3.6 Hardware Cursor/Hardware Window Registers

Register Description	R/W	Offset	Bits	Block
HC/HW Horizontal Position Start	R/W	81,80	11	HC
HC/HW Vertical Position Start	R/W	83,82	11	HC
HC Horizontal Preset/HW Width Low	R/W	84	8	HC
HW Width High	R/W	85	3	HC
HC Vertical Preset /HW Height Low	R/W	86	8	HC
HW Height High	R/W	87	3	HC
HC Start Address	R/W	8A-88	24	SEQ
HC Color 0	R/W	8F-8C	32	ATR
HC Color 1	R/W	93-90	32	ATR
HC Control	R/W	94	6	HC
IP/HW Status	RO	95	2	HC
IP/HW Control 1	R/W	96	8	HC
IP/HW Control 2	R/W	97	8	HC
HW Control	R/W	96	8	HC
HW Imaging Mask Map Start Address	R/W	9A-98	24	SEQ
HW Start Address	R/W	9C-9E	24	SEQ
HW Address Offset	R/W	9F	8	SEQ

Note: All Hardware Cursor/Hardware Window and Co-Processor registers are memory mapped only - the Memory Mapped I/O Base Address register is located at 2x10.

7.3.7 Co-Processor Registers

Register Description	R/W	Offset	Bits	Block
Co-Processor Status	RO	10	7	CP
Co-Processor Control	R/W	11	5	CP
Pixel Map Select	R/W	12	2	CP
Pixel Map n Base Pointer	R/W	17-14	32	CP
Pixel Map n Width	R/W	19, 18	16	CP
Pixel Map n Height	R/W	1B, 1A	16	CP
Pixel Map n Format	R/W	1C	8	CP
Bresenham Error Term	R/W	21, 20	16	CP
Bresenham K1	R/W	25, 24	16	CP
Bresenham K2	R/W	29, 28	16	CP
Direction Steps	R/W	2F-2C	32	CP
ROP	R/W	48	8	CP
Destination Color Compare Condition	R/W	4A	3	CP
Destination Color Compare Value	R/W	4F-4C	32	CP
Pixel Bit Mask	R/W	53-50	32	CP
Foreground Color	R/W	5B-58	32	CP
Background Color	R/W	5F-5C	32	CP
Operation Dimension 1	R/W	61, 60	16	CP
Operation Dimension 2	R/W	63, 62	16	CP
Mask Map Origin X Offset	R/W	6D, 6C	16	CP
Mask Map Origin Y Offset	R/W	6F, 6E	16	CP
Source X Pointer	R/W	71, 70	16	CP
Source Y Pointer	R/W	73, 72	16	CP
Pattern X Pointer	R/W	75, 74	16	CP
Pattern Y Pointer	R/W	77, 76	16	CP
Destination X Pointer	R/W	79, 78	16	CP
Destination Y Pointer	R/W	7B, 7A	16	CP
Pixel Operations	R/W	7F-7C	32	CP

7.4 Extended Register Description

7.4.1 Extended General Registers

3DE Extension Address Register **R/W**

Bit	Description
7-0	8-bit index pointer to the extended data registers

The contents of this register needs to be programmed before the data register can be accessed.

3DF Status Register **Index = 2** **R/W**

Bit Description

0 Reserved

3-1 Memory size

Bits 3-1	Memory Available
000	256K bytes
001	512K bytes
010	1M bytes
011	2M bytes
100	4M bytes
101	8M bytes

These bits are initially set to 'hA at power-up, then set to the proper value by the BIOS, depending on how much memory is detected.

7-4 Reserved

Default: 0Ah

3DF OTI Test Register 1 **Index = 3** **R/W**

Bit Description

2-0 Test mode - These bits define the test modes of the controller. Test mode is turned on by Bit 7.

Bits 2-0 Test mode

0 CRT Counter testing. When in this mode, CRT counters are muxed out to pins SD[15:0]. An internal 3-bit counter is used to mux out the CRT counters content.

Internal Count	Tested Counter
000	Vertical Counter
001	Upper three bits of Vertical counter and Row Scan Counter[4:0]
010	Vertical Counter
011	CRT address counter bits [7:0]
100	CRT address counter bits [15:8]
101	CRT address counter bits [22:16]
110	Horizontal Counter [7:0]
111	Horizontal Counter [15:8]

1 Reserved

2 Reserved

3 Reserved

4 Scan Test for Bypass mode

5 Scan Test Attribute Controller. This bit forces the font data and attribute data to be replaced by the system data bus [15:8] and [7:0] respectively in text mode. In graphics mode, this bit replaces the APA data with system data bus 15:8, to the inputs of the shifters in the Attributes Controller.

6 Reserved

7 Reserved

3 Reserved

4 Flush write cache. This bit is used for chip testing.
0: Normal cache operation
1: Flush write cache immediately after each memory write cycle

5 VSync Test. This bit will cause the VSync to toggle when it is changed from 1 to 0 or from 0 to1

6 Enable Debug mode. When Debug mode is enabled, internal signals would be routed out to pins P[23:16]. The muxing selection of internal signals are defined by Bits 2:0 of this register.

0: Normal operation

1: Enable Debug mode

Bits 2-0	Test mode
000	CRT address for text mode (Font)
001	CRT address for graphics mode
010	CPU interface to Sequencer
011	Co-processor interface to Sequencer
100	Hardware Window/Cursor interface to Sequencer
101	Refresh address
110	Multimedia interface to Sequencer
111	Reserved

The pin muxing is as follows:

P[23:21]	P[20:18]	P[17]	P[16]
Address source	Internal states	Frame signal	Command to Sequencer

7 Enable test mode. Test mode is used for chip testing only.

0: Normal operation

1: Enable global test modes. This bit must be 0 during normal operation.

Default: 00h

3DF OTI Test Register 2 Index = 4 R/W

Bit	Description
0	Enable software speed test mode 0: Normal operation 1: Co-processor will skip all functions and always stay ready
7-0	Reserved for co-processor testing

3DF Video Clock Select Register Index = 6 R/W

Bit	Description
3-0	Video Clock Select. The state of these four bits are reflected in the pins CSEL[3:0]. Bits 1-0 of this register are the images of Bits 3-2 of register 3C2, Bit 2 of this register is the image of Bit 5 of extended register D. See frequency tables for the OTI-088.
7-4	Scratch bits for BIOS used.

Frequency Table for OTI-088:

The OTI-088 is a SynDAC, with programmable frequencies for the dual clock synthesizers. The CSEL3 and CSEL2 input pins of the OTI-088 are don't cares. Under the column CLOCK in the OTI-088 frequency table, the frequency shown after Programmable/ is the default frequency at power-on. After power-on, the frequencies should be programmed as shown (in the column called "Program To") in the table below. Refer to the OTI-088 datasheet for video and memory clock programming details. The OTI-088 power-on default memory clock frequency is 40 MHz.

CSEL3	CSEL2	CSEL1	CSEL0	CLOCK (MHz)	Program to (MHz)
X	X	0	0	Programmable/25.2	Default (25.175)
X	X	0	1	Programmable/25.2	28.332
X	X	1	0	Programmable/25.2	Video Clock Reg 2
X	X	1	1	Programmable/25.2	Video Clock Reg 3

Frequency Table for ATT20C409 and ATT20C499:

The ATT20C409 and ATT20C499 are 16-bit and 24-bit *Precision*DACs respectively, with programmable frequencies for the dual clock synthesizers. The CSEL3 and CSEL2 of the ATT20C409/ATT20C499 are don't cares. Under the **"CLOCK"** column in the ATT20C409 and ATT20C499 frequency tables shown below, the frequency is the default frequency on Power-on. Refer to the ATT20C409 and ATT20C499 specifications for details on programming the the memory clock frequencies.

Video Clock (VClk) = ATT20C409/ATT20C499 ClockA (OTClkA)
Memory Clock (MClk) = ATT20C409/ATT20C499 ClockB (OTClkB)

Frequency Table for ATT20C409:

CSEL3	CSEL2	CSEL1	CSEL0	CLOCK (MHz)	Access	Program to (MHz)
X	X	0	0	25.235	-	Default (25.175)
X	X	0	1	28.338	-	Default (28.322)
X	X	1	0	50.114	R/W	ClockA Set C Regs
X	X	1	1	75.170	R/W	ClockA Set D Regs

Frequency Table for ATT20C499:

CSEL3	CSEL2	CSEL1	CSEL0	CLOCK (MHz)	Access	Program to (MHz)
X	X	0	0	25.235	R	Default (25.175)
X	X	0	1	28.338	R	Default (28.322)
X	X	1	0	50.114	R/W	ClockA Set C Regs
X	X	1	1	75.17	R/W	ClockA Set D Regs

Software reset must be executed each time this register is updated, whenever a mode change requires a new pixel (or video) clock frequency.

Default: x0h

3DF Hardware Configuration Register 1 Index = 7 RO

Bit Description

0 This bit is used by the hardware to determine the ROM data width on the board and route the data out to the system bus appropriately. This bit is used with ISA and PCI buses only. See ROM BIOS Interface for detailed descriptions.

0: 8-bit BIOS (1 ROM)
1: 16-bit BIOS (2 ROMs)

1 On-board/add-on configuration

0: on-board configuration. Enable chip through 3C3.
1: add-on configuration. Enable chip through 46E8.

4-2 Bus Types. These bits define the system bus configuration of the controller. Bits 3&2 should be derived from VL bus ID pins 1&0, respectively.

Bits 4:2	Bus Type
000	Reserved for VL bus
001	386 VL bus
010	486 VL bus
011	486 VL bus
100	PCI bus
101	Reserved
110	Reserved
111	ISA bus

7-5 DRAM type. These bits define the type of DRAM used.

Bits 7-5	DRAM type
000	64KxXX
001	256Kx16$_{10x8}$
010	256Kx4, 256Kx16$_{9x9}$
011	512Kx8
100	1Mx4
111	Reserved

The content of this register is loaded from MD[7:0] during hardware reset.

Note: For compatibility with future revisions, all reserved bits should be tied low through resistors.

3DF Hardware Configuration Register 2 Index = 8 RO

Bit Description

0 Pixel bus width status. Used by BIOS to determine the pixel bus width of the external DAC.

0: 8-bit pixel bus
1: 16-bit pixel bus

1 Enable Auxiliary I/O support on VL bus. This bit does not apply to PCI or ISA bus.

0: normal operation. OTI-64107 generates DACRDn and DACWRn.
1: enable auxiliary I/O. OTI-64107 generates DACCSn & ACSn instead of DACRDn & DACWRn.

Pixel bus width status for PCI and ISA buses:

0: 8 or 16-bit ClkDAC port, as defined by bit 0
1: 24-bit ClkDAC port. The 24-bit ClkDAC port is not applicable for the VL bus.

2 Enable decoding for ROM BIOS

0: Disable decoding for ROM BIOS
1: Enable decoding for ROM BIOS (default address is C0000).

4-3 ROM BIOS size

Bits 4-3	ROM BIOS size
00	32Kbytes
01	64Kbytes
10	128Kbytes
11	256Kbytes

5 CASxn/Wexn select. For x16 DRAM's, either CASn & WEHn & WELn, or CASHn & CASLn & WEn, both can be supported. This bit is used to select which type is selected.
0: CASn, WEHn & WELn
1: CASHn, CASLn & WEn

6 Fast Write Capable. This bit is used to indicate whether the motherboard can handle zero wait state writes. This bit should be connected to ID[2] of the VL bus through a LS244 buffer. This bit can also be read at ER10[1]. ER10[1] can be written over, while this bit is read only.
0: motherboard cannot handle zero wait state Data is latched at the end of the second T2, or one clock after IRDYn.
1: zero wait state for the cycle

7 Bus speed. This bit is for BIOS to determine the bus speed, and therefore determine the appropriate wait states to insert. This bit should be connected to ID[3] of the VL bus through a LS244 buffer.
0: bus speed is greater than 33MHz
1: bus speed is less than or equal to 33MHz

The content of this register is loaded from MD[15:8] during hardware reset.

Note: For compatibility with future revisions, all reserved bits should be tied low through resistors.

3DF Hardware Configuration Register 3 Index = 9 RO

Bit Description

0 DEVSELn speed indication. OTI-64107 can only be medium or slow response, but never fast.
0: forces CFR[10:9] to 01, medium response
1: forces CFR[10:9] to 10, slow response

1 Multi-function device selection for PCI bus. This bit can also be read at CFRE[7].
0: single function device (forces CFRE[7] to 0)
1: multiple function device (forces CFRE[7] to 1)

2 Enable alternate PCI ROM address generation.
0: compatible with revision A, B, B1. No external logic required, two loads on AD bus.
1: single load on AD bus. Required external latches. Available from rev C on.

7-3 Reserved

The content of this register is loaded from MD[23:16] during hardware reset.

Note: For compatibilit_ with future revisions, all reserved bits should be tied low through resistors.

3DF I^{2}C Control Register Index = C R/W

Bit Description

0 Controls pin SRCK
0: Drives pin SRCK low
1: Drives pin SRCK high

1 Controls pin SRD
0: Drives pin SRD low
1: Drives pin SRD high

3-2 Reserved

4 Status of pin SRCK. This bit is read only.

5 Status of pin SRD. This bit is read only.

7-6 Reserved

Default: 00h

3DF Dip Switch Register Index = D RO

Bit Description

7-0 Dip switch status register. For BIOS use.

3DF EEPROM Control Register **Index = E** **R/W**

Bit	Description
0	EEPROM Data. This bit is the data line between the serial EEPROM and the VGA controller. The read data for this bit comes from CSEL[0]. Write data is sent to CSEL[1]. CSEL[0] and CSEL[1] are connected to the EEPROM data out and the EEPROM data in pins respectively.
1	EEPROM CS. This bit is used as the chip select control for the EEPROM. It should be set to 1 for the VGA controller to access the EEPROM.
2	EEPROM Function Enable. This bit selects the function of the CSEL bus. When this bit is 1, CSEL[2:0] are used to interface with the EEPROM. When this bit is 0, the CSEL[2:0] functions as clock select signals.
3	EEPROM Clock(SK). The value of this bit which acts as the shift clock for the serial EEPROM is reflected on CSEL[2]. To program the EEPROM, this bit is programmed to toggle between 1 and 0 every 4 us.
7-4	Reserved

Default: 00h

3DF Power Management Control Register **Index = F** **R/W**

Bit	Description
1-0	Display Power Management Modes. When in any of the power saving modes, the Memory Controller stops fetching data for display (same as screen-off bit SR1b5) but continues to refresh the DRAM.

Bits 1-0	Modes
00	On - HSync, VSync, BLANKn, P[23:0] operate normally
01	Stand-by - VSync operates normally, HSync, BLANKn & P[23:0] off
10	Suspend - HSync operates normally, VSync, BLANKn & P[23:0] off
11	Off - HSync, VSync, BLANKn & P[23:0] off

Bit	Description
2	Enable Multimedia clock 0: Disable multimedia clock 1: Enable multimedia clock
7-3	Reserved

Default: 00h

3DF Scratch Registers 0-7 **Index = F0-F7** **R/W**

Bit	Description
7-0	Eight scratch bits. These scratch registers are defined and reserved for internal use. Application programs should not use them.

7.4.2 System Interface Registers

All 2xxx registers are PCI Configuration registers that can be used for other buses. On the ISA or VL buses, these registers are accessed at 2xxx, but they can only be accessed through configuration read/write on the PCI bus. The PCI Configuration register does not have 2xxx address and only responds to configuration cycles.

2x00 Vendor ID Register **CFR = 1,0** **RO**

Bit	Description
15-0	Vendor Identification used with PCI bus. The value is **104E.**

2x02 Device ID Register **CFR = 3,2** **RO**

Bit	Description
15-0	Device Identification

Bits 15-0	Product
0107h	OTI-64107

2x04 System Bus Command Register **CFR = 5,4** **R/W**

Bit Description

0 Enable I/O accesses. This bit is used for the PCI bus only.
0: Disables all I/O space accesses
1: Enables I/O space accesses

1 Enable memory and memory mapped I/O accesses. Both this bit and 3C2[1] need to be 1's to enable memory accesses to the controller.
0: Disables all memory space accesses
1: Enable memory space accesses

2 Enable bus master mode. This bit is used for the PCI bus only.
0: Disable bus master mode
1: Enable bus master mode

4-3 Reserved

5 Enable bus snooping for palette registers
0: Respond to palette register writes like ordinary I/O
1: Complete all palette register writes without asserting DEVSELn (LBSELn), allowing the cycle to propagate out to the standard ISA bus for other devices to shadow these registers.

6 Reserved

7 Enable address/data stepping. This bit is used for the PCI bus only.
0: Disable address/data stepping
1: Allow address/data to be driven out in more than one clock

8 Reserved

9 Back-to-back cycle capable. This bit is used when the controller is in Master mode and on the PCI bus only. Back-to-back cycles are transactions from one master to the same target without IDLE cycles in between.
0: Do not issue back-to-back cycles
1: All target devices can accept back-to-back cycles

15-10 Reserved

Default: 00A0h

System Bus Status Register **CFR = 7,6** **R/W**

Reads to this register behave normally. Writes are different in that writeable bits can only be reset, but not set. A bit is reset whenever a 1 is written to that particular bit. For example, to clear Bit 14 and not effect any other bits, write the value 4000h to the register.

Bit Description

6-0 Reserved

7 Fast Back-to-Back Cycle Capable. This bit is read-only and defaults to 1.
0: Not capable of fast back-to-back cycles
1: Capable of fast back-to-back cycles

8 Reserved

10-9 DEVSELn timing. These two bits are read only and are configurable through MD[16] during hardware reset. MD[16]=0 => CFR6[10:9]='b01, MD[16]=1 => CFR6[10:9]='b10. MD[16] state during reset can be read through ER9[0].

Bits 10-9	DEVSELn Timing
00	Fast
01	**Medium**
10	Slow
11	Reserved

11 Signaled Target-abort status. This bit is set whenever the OTI-64107 while being a target device, terminates a transaction with target-abort.

12 Received Target-abort status. This bit is set whenever the OTI-64107 while being a master, has its transaction terminated with target-abort.

13 Received Master-abort status. This bit is set whenever the OTI-64107 while being a master, terminates its transaction (except for Special cycle) with master-abort.

15-14 Reserved

Default:00000xx010000000b

2x08 Revision ID Register CFR = 8 RO

Bit Description

7-4 Chip functional revision. Chip revision only changes when there is a functional difference. The first revision is A.

Bits 7-4	Revision Letter
0000	**A**
0001	B, B1, B2
0010	C

3-0 Chip foundry revision. Chip foundry revision changes for each foundry tape out and/or library changes.

Bits 3-0	Revision Number
0000	**0 - TSMC OASC672**

Class Code Register CFR = B,A,9 RO

Bit Description

23-0 Register class encoding for backward compatibility

Bits 23-0	Register class
000000h	All currently implemented devices except for VGA compatible devices.
030000h	**VGA compatible devices**

Cache Line Size Register CFR = C RO

Bit Description

7-0 Reserved

Latency Timer CFR = D R/W

Bit Description

7-0 Specifies, in units of PCI bus clocks, the value of the Latency Timer for OTI-64107 while being a master. Bits 7-3 are read/writeable. Bits 2-0 are read only.

Default: 0h

Header Type CFR = E RO

Bit Description

6-0 Always set to zero to indicate current offset for the configuration register.

7 Multi-function device indicator. This bit is read only, but configurable through MD[17] during reset. Reset is used to latch in the value of MD[17] and data is routed to this bit.
0: single function device
1: multi function device

Built-In Self Test Register CFR = F RO

Bit Description

7-0 Reserved

2x10 Memory Mapped I/O Base Address (0) Register CFR = 13-10 R/W

Bit Description

0 Memory or I/O indication. This bit is read only.
0: Memory mapped
1: I/O mapped

2-1 Locatable area indication. These two bits are read only.

Bits 2-1	Locatable area
00	**Locate anywhere in 32-bit address space**
01	Locate below 1M
10	Locate anywhere in 64-bit address space
11	Reserved

3 Prefetchable indication. This bit is read only.
0: Memory is not prefetchable (not cacheable)
1: Memory is prefetchable (cacheable)

7-4 Always set to zero to occupy 256 bytes of address space. These bits are read only.

31-8 Upper 24 bits of the base address for memory mapped I/O. These bits are read/writeable.

2x14 Graphic Memory Base Address (1) Register CFR = 17-14 R/W

Bit Description

0 Memory or I/O indication. This bit is read only.
0: Memory mapped
1: I/O mapped

2-1 Locatable area indication. These two bits are read only.

Bits 2-1	Locatable area
00	**Locate anywhere in 32-bit address space**
01	Locate below 1 Meg
10	Locate anywhere in 64-bit address space
11	Reserved

3 Prefetchable indication. This bit is read only.
0: Memory is not prefetchable (not cacheable)
1: Memory is prefetchable (cacheable)

19-4 Always set to zero to occupy at least 1Mbyte of memory address space. These bits are read only.

22-20 Always set to zero and read only on PCI bus to occupy a block of 8Mbytes of address space. On VL and ISA buses, however, these bits are read/writeable so that the display memory can be located at 1Mbyte increments.

31-23 Upper nine bits of the base address for graphics memory. These bits are read/writeable.

Extended I/O Base Address (2) Register CFR = 1B-18 R/W

Bit Description

0 Memory or I/O indication. This bit is read only.
0: Memory mapped
1: I/O mapped

1 Reserved

3 Prefetchable indication. This bit is read only.
0: Memory is not prefetchable (not cacheable)
1: Memory is prefetchable (cacheable)

7-4 Always set to zero to occupy minimum of 256 bytes of address space. These bits are read only.

31-8 Upper 24 bits of the base address for extended I/O registers. These bits are read/writeable.

Base Address 5-3 CFR = 27-1C RO

Bit Description

31-0 Reserved

Reserved Registers CFR = 3B-34,2F-28 RO

Bit Description

31-0 Reserved

2x30 BIOS ROM Base Address Register CFR = 33-30 R/W

Bit Description

0 BIOS ROM address decode enable. This bit is read/writeable.
0: Disable address decode for BIOS ROM
1: Enable address decode for BIOS ROM

10-1 Reserved

14-11 Always zero's to indicate that the BIOS ROM is minimally 32 Kbytes. These bits are read only.

17-15 These bits are used to indicate how big the ROM BIOS is. These bits are either reset to 0's and read only, or read/writeable depending on Configuration Register 2, Bits 4-3.

ER8[4:3]	ROM size	RO or R/W
00	32Kbytes	Bits 17-15 are read/writeable
01	64Kbytes	Bits 17-16 are read/writeable, Bit 15 is always 0 and RO
10	128Kbytes	Bit 17 is read/writeable, Bits 16-15 are always 0's and RO
11	256Kbytes	Bits 17-15 are always 0's and RO

31-18 Upper 14 bits of the base address for BIOS ROM

Default: 000C0001h for ISA and VL bus
000C0000h for PCI bus

2x3C Interrupt Line Register CFR = 3C R/W

Bit Description

7-0 This is a scratch pad for POST software to write the interrupt line routing information during initialization and configuration of the system (PCI bus). Device drivers and OS's can use this information to determine priority and vector information. Non PCI BIOS may use this register as a general purpose scratch register in addition to registers 3DF index 9, A & B.

Interrupt Pin Register CFR = 3D RO

Bit Description

7-0 This register indicates which interrupt pin is used by the OTI-64107. The value of this register should be 01. This interrupt is for CINTn.

Min_Gnt Register CFR = 3E RO

Bit Description

7-0 This register (Minimum Grant) is used to indicate to the system how long a burst period the OTI-64107 needs. The value of this register should be FF (maximum allowed).

Max_Lat Register CFR = 3F RO

Bit Description

7-0 This register (Maximum Latency) is used to indicate to the system how often the OTI-64107 needs to get access to the PCI bus. The value of this register should be 01 (very often).

3DF Local Bus Control Register 1 Index = 10 R/W

Bit Description

0 Wait state control. This bit delays internal ADS/FRAME to push all cycles back by one clock. This bit affects all I/O, memory mapped I/O and memory cycles. This bit works independently from other wait state bits. The total wait states for a cycle is equal to the individual wait state plus this wait.
0: No additional wait state asserted
1: One wait state asserted

1 Reserved

2 Memory write wait state. This bit delays SRDYn/TRDYn by one clock on memory write cycles.
0: no additional wait state for the cycle
1: one wait state is inserted for the cycle

3 Reserved

5-4 Memory mapped I/O wait state. These bits delay SRDYn/TRDYn by the specified amount of clocks on memory cycles for the memory mapped I/O.

Bits 5,4	Read Wait State	Write Wait State
00	1	0
01	2	1
10	3	2
11	4	3

7-6 I/O wait state. These bits delay SRDYn/TRDYn by the specified amount of clocks on I/O cycles.

Bits 5,4	Wait State
00	1
01	2
10	3
11	4

Default: F5h

3DF ISA Bus Control Register Index = 13 R/W

Bit Description

2-0 Reserved

3 0: Disable zero wait state ISA bus operation
1: Enable zero wait state ISA bus operation

4 Reserved

5 Reserved

6 0: 8-bit I/O access
1: 16-bit I/O access

7 0: 8-bit memory access
1: 16-bit memory access

Default: 00h

3DF Video Memory Mapping Register Index = 14 R/W

Bit Description

0 Enable OTI address mapping
0: VGA memory mapping at A0000 and B8000 as dictated by Graphics Register 6, Bits 3 & 2
1: OTI memory mapping as dictated by register 2x14

1 Reserved

4-2 Memory address aperture select. These bits define the memory address aperture that OTI-64107 will respond to. These bits have no effect if Bit 0 = 0.

Bits 4-2	Memory address aperture
000	x00000-x3FFFF: 256K aperture
001	x00000-x7FFFF: 512K aperture
010	x00000-xFFFFF: 1M aperture
011	x00000-1FFFFF: 2M aperture
100	x00000-3FFFFF: 4M aperture
101	x00000-7FFFFF: 8M aperture
11x	Reserved

7-5 Reserved

Default: 00h

3DF Memory & Memory Mapped I/O Enable Index = 15 R/W

Bit Description

5-0 Reserved

6 Enable graphic memory response. This bit and Bit 7 of this register are needed in case the graphics memory base address is the same or in the same range as the memory mapped I/O base address. If the memory spaces are not the same, then both bits should be 1's.
0: disable access to graphics memory
1: enable access to graphics memory

7 Enable memory mapped I/O response. This bit and Bit 6 of this register are needed in case graphics memory base address is the same or in the same range as memory mapped I/O base address. If the memory spaces are not the same, then both bits should be 1's.
0: disable access to memory mapped I/O
1: enable access to memory mapped I/O

Default: 40h

3DF Configuration/DAC/Auxiliary Register Range Index = 19 R/W

Bit Description

3-0 These bits define the I/O address range for the PCI compatible Configuration registers on the ISA and VL buses, DAC register address range in addition to the IBM VGA DAC address range 3C6-3C9, and auxiliary register address range for other devices in the graphics subsystem. The I/O range can be programmed to be from 2x00-2xBF where x is defined by these four bits. The I/O space is divided among the three sources as follows:

Source	Address Range
Configuration Registers	2x00-2x3C
DAC Registers	2x80-2x9F
Auxiliary Registers	2xA0-2xBF

4 Enable additional DAC address space
0: DAC address space is at 3C6-3C9 only
1: DAC address space is at both 3C6-3C9 and 2x80-2x9F

5 Enable auxiliary address space. This will enable decoding for signal ACSn when on the VL bus, or outputs ARDn and AWRn when on the PCI or ISA buses.
0: Disable decoding for auxiliary registers
1: Enable decoding for auxiliary registers

7-6 Reserved

Default: 01h

7.4.3 Extended Sequencer Registers

3DF Compatible Segment Register Index = 11 R/W

Bit Description

3-0 Read Segment for CPU memory read

7-4 Write Segment for CPU memory write
These two 4-bit Segment registers are used to extend the CPU address for display memory size greater than 256K. Bits 3-0 are used to address the memory read operation. Bit 7-4 are used to address the memory write operation. Bits 3-0 are the image of Index Register 23, Bits 3-0. Bits 7-4 are the image of Index Register 24, Bits 3-0.
This register is provided for compatibility with OTI-077, 67, and 37 only. New software development should use registers 23, 24 & 25.

Default: 00h

3DF Display FIFO Depth Register Index = 20 R/W

Bit Description

3-0 Graphics mode display FIFO depth control. This register defines the level of the display FIFO at which the CRT refresh will reclaim the memory bus during graphics modes. Note, this register has a different meaning than 67/77/87SX because of the new arbitration scheme.

5-4 Reserved

7-6 Text mode display FIFO depth control. This register defines number of consecutive memory accesses for CRT refresh during text modes.

Bits 7-6	Consecutive accesses
00	6
01	8
10	10
11	12

Default: 81h

3DF Mode Select Register **Index = 21** **R/W**

Bit Description

0 Enable doubling horizontal timing. This bit is used for high color or true color modes. When this bit is enabled, all horizontal CRT parameters are multiplied by two. See Extended Attribute register for more information.

1 Enable tripling horizontal timing. This bit is used for true color modes. When this bit is enabled, all horizontal CRT parameters are multiplied by three. See Extended Attribute register for more information.

2 Extended graphics mode selection. These bits are used to control memory mapping.
0: VGA modes - 256K memory only
1: OTI packed pixel modes

3 Reserved

4 CClk Character Clock frequency
0: normal frequency as required by resolution (CClk = VCLK/8, VClk/9).
1: 1/2 normal frequency as required by resolution (CClk = VCLK/4).

6-5 Shift/Load frequency

Bits 6,5	Shift/Load Frequency
00	As specified by SR1[4,2]
01	One shift/load every two VCLKs
10	One shift/load every four VCLKs
11	One shift/load every six VCLKs *

7 Enable synchronous mode (not implemented for revision A)
0: Asynchronous mode. Internal memory clock is MClk, system interface clock is PROClk, internal commands are generated with PROClk and resynchronize with MClk.
1: Synchronous mode. Internal memory clock is PROClk, system interface clock is PROClk or PROClk/2, internal commands are generated with PROClk.

Default: 0h

This register can only be updated during software reset. The correct sequence to update this register is as follows, assuming all I/O cycles are 16-bit transfers:

```
3C4 <= 0100h
3DE <= xx21h
3C4 <= 0300h
```

Updating this register on the fly will hang the system.

* For Attribute Mode 3 (24bpp/16-bit bus), SHFT/LD is generated once every three VCLKs.

3DF Feature Select Register Index = 22 R/W

Bit Description

1-0 Reserved

2 Enable Command Buffer. When enabled, all memory mapped I/O writes from offset 0 to 7Fh (coprocessor registers) would be routed to the Command FIFO. In addition, after the coprocessor has been set up to do memory to screen bit block transfer (BBLT) in CPU assisted mode, all incoming memory writes would also be routed through the Command FIFO.
0: Disable command FIFO
1: Enable command FIFO

3 Enable write cache - This bit enables the write cache in the controller. It should be set to 1 for high-performance operations.

4 Graphics Latch Width
0: 32-bit - IBM compatible
1: 64-bit - Used with extended modes

5 Enable Write mode 4. This write mode is used in packed Pixel Mode only. In this mode, system data is routed directly to the memory, by-passing barrel shifter and ALU. This mode is meant to improve straight Memory to Screen Source Copy.
0: Write mode as controlled by GR5[1:0]
1: Write mode 4

6 Enable read mode 4. This read mode is used in packed pixel mode only. In this mode, memory data is routed directly to the system bus, by by-passing the internal VGA data path. This mode is meant to improve read performance.

7 Reserved

Default: 0h

3DF Extended Read Segment Register Index = 23 R/W

Bit Description

6-0 These seven bits correspond to the CPUADR[22:16] for CPU read operation. They are used to extend the 64 K video memory space (A0000-AFFFF). Bits 3:0 are the image of ER11[3:0]. Updating Bits 3:0 will also update Bits 3:0 of ER11.

7 Reserved

Default: 0h

3DF Extended Write Segment Register Index = 24 R/W

Bit Description

6-0 These seven bits correspond to the CPUADR[22:16] for CPU write operation. They are used to extend the 64 K video memory space (A0000-AFFFF). Bits 3:0 are the image of ER11[7:4]. Updating Bits 3:0 will also update Bits 7:4 of ER11.

7 Reserved

Default: 0h

3DF Extended Common Read Write Register Index = 25 R/W

Bit Description

6-0 This 6-bit register is a write port for both Register 23 and 24. A write to this register is equivalent to writing into both index Registers 23 and 24. A read to this register is equivalent to reading the Extended Write Segment register.

7 Reserved

Default: 0h

3DF RASn Control Register Index = 26 R/W

Bit Description

1-0 RASn precharge width. RASn precharge consists of st1 and st2, these bits control the number of st2s inserted. This timing is t_{M1}.

Bits 1,0	Precharge Width
00	2 MClk (st1, st2)
01	3 MClk (st1,st2,st2)
10	4 MClk (st1,st2,st2,st2)
11	5 MClk (st1,st2,st2,st2,st2)

2 Reserved

3 Enable half clock option for RASn

0: RASn precharge is as defined by bits 1:0

1: RASn precharge is decreased by half clock, and RASn pulse width is increased by half clock.

7-4 Reserved

Default: 01h (3 MClk for precharge)

Note: It is not necessary to program RASn pulse width, this parameter [illegible] equal to RAS-to-CAS delay plus CASn precharge plus CASn pulse width.

3DF CASn Control Register Index = 27 R/W

Bit Description

0 CASn precharge width. This is to control number of st4s. This timing is t_{M2}.

0: 1 MClk precharge width

1: Reserved

1 Enable half clock option for CASn

0: CASn precharge and pulse width as defined by bit 0 and Bit 2 respectively

1: CASn precharge is reduced by half clock, and CASn pulse width is increased by half clock

2 CASn pulse width. This is to control number of st5s. This timing is t_{M3}.

0: 1 MClk pulse width (st5)

1: 2 MClk pulse widths (st5,st5)

3 Reserved

5-4 CASn delay from RASn. This is to control number of st3s. This timing is t_{M4}. Note that this is not the regular delay from RASn going low to CASn going low, but rather from RASn going low to CASn going low minus CASn precharge. Thus, DRAM specification $t_{RCD} = t_{M4} + t_{M3.}$

Bits 5,4	Delay
00	1 MClk period
01	2 MClk periods
10	3 MClk periods
11	4 MClk periods

7-6 Reserved

Default: 10h (1 MClk for precharge and pulse width, 2 MClks for delay)

3DF Refresh Control Register **Index = 28** **R/W**

Bit Description

1-0 Number of refresh cycles per scan line. This register will override CR11[6].

Bits 1-0	Number of refresh
00	Reserved
01	1 refresh cycle per line
10	2 refresh cycles per line
11	Follow CR11[6]

2 Reserved

3 Type of refresh cycle
0: CAS-before-RAS refresh
1: RAS only refresh

7-4 Reserved

Default: 03h

3DF Hardware Window Arbitration **Index = 29** **R/W**

Bit Description

1-0 Number of consecutive accesses each time during display fetch. Each access can be either 32-bit or 64-bit depending on the memory bus width at the time.

Bits 2-0	Consecutive Accesses
000	4
001	8
010	16
011	32

7-2 Reserved

Default: 04h

7.4.4 Extended CRT Controller Registers

3DF OTI CRT Overflow Register Index = 30 R/W

Bit	Description
0	Vertical Total Bit 10
1	Vertical Blank Start Bit 10
2	Vertical Retrace Start Bit 10
3	CRT address Offset Bit 8
6-4	Reserved
7	Enable interlaced display 0: Non-interlaced display 1: Interlaced display

Default: 00h

3DF CRT Start Address Hi Index = 31 R/W

Bit	Description
6-0	High order start address Bits 22-16. Start address Bits 15-0 are from CR0C and CR0D
7	Reserved

3DF HSync Divided by Two Start Register Index = 32 R/W

Bit	Description
7-0	This 8-bit value indicates when the vertical retrace will start in every odd frame during interlaced mode. The unit of this value is in character clocks.

3DF CRT Address Compatibility Register Index = 33 R/W

Bit	Description
0	Address wrapping for compatibility with IBM's VGA. This is useful when an application assumes that there are only 256K of memory, and expecting the display address to be these 256K only (JDOS is an example). 0: No wrapping, display complete memory available 1: Wrap around the first 256 K of memory
7-1	Reserved

Default: 00h

7.4.5 Extended Attribute Controller Registers

3DF Pixel Interface Register **Index = 38** **R/W**

Bit Description

3-0 Pixel Mode - Only in mode 0 is memory data routed through the Attribute Controller datapath. For Modes 1-7, memory data is routed directly to P[23:0] as shown below, bypassing the internal palette and PEL panning logic. The table below describes how to program the input clock (VCLK), the output clock (PCLK), and internal clocks (CCLK, SHIFT/LD, HCLK), which can be programmed in ER21.

X - represents 8-bit P data

Bits 3-0 Mode	Bpp	P Bus Width	P Bus Ordering P[23:0]	VCLK	PCLK	CCLK	SHFT/LD	HCLK
0	VGA	8	XXP[7:0]	Normal	VCLK	8/9 VCLK	8/16/32 VCLK	CCLK
1	4	8	XX P1[3:0]P0[3:0]	1/2	VCLK	4 VCLK	4 VCLK	CCLK
2	8 16 24	8	XX P[7:0] XX P0[7:0] XX P0[15:8] XX P0[7:0] XX P0[15:8] XX P0[23:16]	Normal Double Triple	VCLK VCLK VCLK	8 VCLK 8 VCLK 8 VCLK	8 VCLK 8 VCLK 6 VCLK	CCLK CCLK/2 CCLK/3
3	24	16	XP0[15:0] XP0[23:16]P1[7:0]	3/2	VCLK	4 VCLK	6/2 VCLK*	CCLK/3
4	8	16	XP1[7:0]P0[7:0]	1/2	VCLK	4 VCLK	4 VCLK	CCLK
5	16	16	XP[15:0]	Normal	VCLK	8 VCLK	4 VCLK	CCLK
6	32	16	XP0[15:0] XP0[31:16]	Double	VCLK	8 VCLK	4 VCLK	CCLK/2
7	24	24	P[23:0]	Normal	VCLK	8 VCLK	2 VCLK	CCLK
8	32	24	P[23:0]	Normal	VCLK	8 VCLK	2 VCLK	CCLK
9	24	16	XP0[15:0] XP0[7:0]P0[23:16]	3/2	VCLK	4VCLK	6/2 VCLK	CCLK/3

3 Reserved

4 Color ordering - only applies to 16 bpp and 24 bpp modes

Bit 4	P[23:0]/Byte[2:0]	Bits/pixel
0	GGGRRRRR$_1$ BBBBBGGGGGRRRRR$_0$	16bpp
0	BBBBBBBBGGGGGGGGRRRRRRRR	24bpp
1	GGGBBBBB$_1$ RRRRRGGGGGBBBBB$_0$	16bpp
1	RRRRRRRRGGGGGGGGBBBBBBBB	24bpp

6-5 Bits per pixel. These bits are used in conjunction with Bit 4 for RGB swapping.

Bits 6,5	Bits/pixel
00	8 bpp
01	16 bpp
10	24 bpp
11	32 bpp - RGB swapping only involves the lower three bytes, with the most significant byte left alone. For Pixel Mode 8 above, the lower three bytes are sent out to the DAC and the most significant byte is dropped. Thus, Pixel Mode 8 is really a 24 bpp mode, but uses four byte address space to store three bytes of information.

7 Reserved
Default: 0h

* SHFT/LD is programmed to be six VCLKs , but with this mode, it is actually three VCLKs.

3DF **Extended Overscan Color Register 1 Index = 39 R/W**

Bit Description
7-0 This byte is the mid byte of the three bytes defining the overscan color. This byte is used only in hi and true color modes. The low byte resides at AR11, the high byte resides at ER40.

3DF **Extended Overscan Color Register 2 Index = 3A R/W**

Bit Description
7-0 This byte is the high byte of the three bytes defining the overscan color. This byte is used only in true color modes. The low byte resides at AR11, the high byte resides at ER39.

7.4.6 Hardware Cursor/Hardware Window Registers

These registers are shared between Hardware Cursor and Hardware Window because both of these can not exist simultaneously. These registers are memory mapped only.

HC/HW Horizontal Position Start Register Offset = 81, 80 R/W

Bit Description
15-0 Horizontal starting position of the HC/HW relative to the start of the display area in pixel units. This number can be negative and programmed in two's complement format. (0,0) is at the top left corner.

HC/HW Vertical Position Start Register Offset = 83, 82 R/W

Bit Description
15-0 Vertical starting position of the HC/HW relative to the start of the display area in scan line unis. This number can be negative and programmed in two's complement format. (0,0) is at the top left corner.

HC Horizontal Preset/HW Width Low Register Offset = 84 R/W

Bit Description
7-0 For Hardware Cursor, this register defines the starting horizontal position of the HC within the 64 x 64 area in pixel units. The HC always ends at position 63 (i.e., no wrapping).
For Hardware Window, this is the eight lower-order bits of the Hardware Window Width register in double word units.

HW Width High Register Offset = 85 R/W

Bit Description
7-0 For Hardware Window, this is the eight higher-order bits of the Hardware Window Width in double word units.

HC Vertical Preset/HW Height Register Low Offset = 86 R/W

Bit Description

7-0 For Hardware Cursor, this register defines the starting vertical position of the HC within the 64 x 64 area in scan line units. The HC always ends at position 63 (i.e., no wrapping).
For Hardware Window, these are the lower eight bits of the Hardware Window Height in pixel units.

HW Height Register High Offset = 87 R/W

Bit Description

7-0 For Hardware Window, this is the higher eight bits of the Hardware Window Height in pixel units.

HC Start Address Register Offset = 8A-88 R/W

Bit Description

23-0 Linear starting address of the buffer within the display memory. The value to be programmed is the system linear address divided by four when in planar modes and divided by eight when in packed pixel modes.

HC Color 0 Register Offset = 8F-8C R/W

Bit Description

31-0 HC color 0. Only the corresponding number of bits-per-pixel in the display mode are required in this register. For example, if the display mode is 8 bits-per-pixel, only the eight low order bits of this register are used.

HC Color 1 Register Offset = 93-90 R/W

Bit Description

31-0 HC color 1. Only the corresponding number of bits-per-pixel in the display mode are required in this register. For example, if the display mode is 8 bits-per-pixel, only the eight low order bits of this register are used.

HC Control Register Offset = 94 R/W

Bit Description

0 Color HC control. Power-up default to 0
0: Disable color HC
1: Enable color HC

1 HC display selection
0: HC is under overscan
1: HC is displayed over overscan

2 HC data format
0: Intel format, Bit 0 is the first pixel
1: Motorola format, Bit 7 is the first pixel

3 HC blink enable. Power-up default to 0
0: Disable HC blinking
1: Enable HC blinking

5-4 HC blink rate control. Power-up default to 01 (eight frames on and eight frames off).

Bits 5,4	Blinking Rate
00	4 frames on and off
01	8 frames on and off
10	16 frames on and off
11	32 frames on and off

7-6 Reserved

Default: 10h

Multimedia Port/HW Register Offset = 95 R/W

Bit Description
0 MMVRSETn status. Pin MMVRSETn is routed to this bit for software monitoring.
1 Field status. Pin MMFIELD is routed to this bit for software monitoring.
5-2 Reserved
6 Enable byte swapping for MMD
0: No byte swapping
1: MMD [15:8] is routed to MMD[7:0] and vice versa
7 Enable word swapping for MMD
0: No word swapping
1: Swaps the first set of MMD [15:0] with the second set of MMD [15:0]

HW Control Register **Offset = 96** **R/W**

Bit Description
0 Enable Hardware Window display. This bit will automatically disable Hardware Cursor.
0: Disable Hardware Window display
1: Enable Hardware Window display. Disable HC
2-1 Horizontal scaling factor for input data

Bits 2-1	Scaling factor
00	No scaling
01	Reduced by two
10	Reduced by four
11	Reduced by eight

4-3 Vertical scaling factor for input data

Bits 4-3	Scaling factor
00	No scaling
01	Reduced by two
10	Reduced by four
11	Reduced by eight

5 Select odd/even lines for vertical scaling
0: Select even lines for vertical scaling. Scaling by two would result in keeping all even lines. Scaling by four or eight would result in keeping lines 0, 4, 8 ... or 0, 8, 16 ..., respectively.
1: Select odd lines for vertical scaling. Scaling by two would result in keeping all odd lines. Scaling by four or eight would result in keeping line 1, 5, 9 ... or 1, 9, 17 ..., respectively.
6 YUV/RGB selection for horizontal scaling
0: Incoming data from the Multimedia port is in YUV format
1: Incoming data from the Multimedia port is in RGB format
7 Enable mask map for Multimedia Port. When enabled, the Multimedia Port update is masked by the Multimedia Mask Map (not to be confused with the Co-processor Mask Map).
Default: 00h

Hardware Control Register 2 **Offset=97** **R/W**

Bit Description
0 MDMXn inversion. This signal is default to be active low
0: Signal output as is
1: Signal is inverted before output
1 VDVALID inversion. This signal is default to be active high
0: Signal output as is
1: Signal is inverted before output
2 MMFIELD inversion. This signal is default to be active low
0: Signal used as is
1: Signal is inverted
3 MMVRSETn inversion. This signal is default to be active low.
0: Signal used as is
1: Signal is inverted
4 MMHRSETn inversion. This signal is default to be active low.
0: Signal used as is
1: Signal is inverted
5 MMCLK inversion. This signal is default to be active low.

0: Signal used as is
1: Signal is inverted

6 MMDVALID inversion. For proper operation, this signal should be active high.
0: Signal used as is
1: Signal is inverted

7 Interlaced input data.
0: Incoming data stream is non-interlaced
1: Incoming data stream is interlaced

Default: 00h

HW Mask Map Start Address Register Offset = 9A-98 R/W

Bit Description

23-0 Linear starting address of the multimedia mask map buffer within the display memory. The value to be programmed is the system linear address. The address must be on word boundary (Bit 0 = 0).

Multimedia Mask Map Offset Refister Offset = 9B R/W

Bit Description

7-0 Offset address for the multimedia mask map in quad word unit. Used to calculate the address of the next mask line in the video window. This register is used for input only. The value to be programmed can be calculated as follows:

Width /64 + 1 where Width=width of the video window in the pixel units. For example, a 320 pixel video window would have a multimedia mask map offset of 5. A 160 pixel video window would have a multimedia mask map offset of 3.

HW Start Address Register Offset = 9E-9C R/W

Bit Description

23-0 Linear starting address of the buffer within the display memory. This register is used for both the input port and the output port. The value to be programmed is the system linear address. The address must be on word boundaries (Bit 0 = 0).

HW Address Offset Register Offset = 9F R/W

Bit Description

7-0 Offset address in word unit. Used to calculate the address of the next line in the window.

Video Window Width Register Offset=A1-A0 R/W

Bit Description

15-11 Reserved

10-0 Video window width in transfer cycle unit, with each transfer cyle is 16 bits wide. This register is used to specify the maximum number of transfer cycle allowed within a HResetn period. The programmed value is one less than the actual value. The value to be programmed can be calculated as follow: x = Width * (bpp/16) where Width=width of the video window in a pixel unit, bpp=color depth of the video window. For example, a 320-pixel wide video window at 16 bpp would have 320 transfer cycles, and the programmed value would be O13Fh. A 320-pixel wide video window at 32 bpp would have programmed value of O27Fh.

Video Window Height Register Offset=A3,A2 R/W

Bit Description

15-11 Reserved

10-0 Video window height in pixel (scan line) unit. This register is used to specify the maximum number of lines within a VResetn period. The programmed value is one less than the actual value. for example, a 240-pixel high video window would have a programmed value of OOEFh.

7.4.7 Co-Processor Registers

Most of the Co-Processor registers are read/writeable so that register save/restore can easily be accomplished. All the Co-Processor registers are Memory Mapped only.

Co-Processor Status Register **Offset = 10** **RO**

Bit Description

1-0 Indicates which map needs data next. They are used in CPU assisted mode, where driver can find out which map the co-processor would need data to be fetched (memory to screen) by the CPU, or which map would need data to be written to (screen to memory).

Bits 1-0	Map
00	Pattern
01	Mask
10	Source
11	Destination

2 Indicates the above map needs to be written to screen memory or read back to system memory
0 = read back to system memory
1 = write data to screen memory

3 Map status valid. This bit is read only. This bit is used to indicate whether Bit 1-0 values are valid or not. When not valid, software should reread Bits 1-0 until this bit is valid.
0 = Bits 1-0 are invalid
1 = Bits 1-0 are valid

4 Indicates that the address of the next access needs to be reset to the beginning of the map
0 = Increment address for next access
1 = Reset address to beginning of the map

5 Advance to next line indication
0 = remain on current line
1 = advance to next line

6 Reserved

7 Co-Processor busy status bit. Reading this bit does not stop the Co-Processor.
0 = Co-Processor is idle
1 = Co-Processor is busy

Co-Processor Control Register **Offset = 11** **R/W**

Bit Description

0 Enable Co-Processor operation complete interrupt. Interrupt is always generated but is routed out to pin CINTn only when this bit is enabled.
0 = VGA interrupt is routed out to CINTn
1 = Co-Processor operation complete, interrupt is routed out to CINTn

1 Enable Master mode. This bit should be used with the PCI bus only.
0 = CPU assisted mode operation
1 = Master mode operation

3-2 Reserved

4 Interrupt status. This bit can be read to find out the interrupt status. Writing a zero to this bit will clear the interrupt. Writing a one has no effect.

5 Terminate Co-Processor operation. This bit is automatically reset to zero after the Co-Processor has stopped internally.
0 = Allow Co-Processor to finish the operation
1 = Terminate Co-Processor operation

6 Enable turbo Co-Processor data-path. Used with slower MCLK frequencies (50 MHz or less).
0 = default datapath delay
1 = delete an extra clock for datapath delay

7 Reserved. Enable fast Co-Processor address calculation. Used with slower MCLK frequencies (<50MHz)

Default: 0h

Pixel Map Select Register **Offset = 12** **R/W**

Bit Description

1-0 Index to indicate which pixel map registers are being accessed

Bit 1,0	Pixel Map Select
00	Mask Map
01	Pixel Map A
10	Pixel Map B
11	Pixel Map C

7-2 Reserved

Co-processor Control Register 2 Offset = 13 RO

Bit Description

3-0 Number of entries left in the Command FIFO. These four bits indicate the number of doubled word entries available. Software should check these bits before writing to the Command FIFO. If the number of writes exceeds the number of empty FIFO entries, the overflow bit (Bit 7) will be set.

6-4 Reserved

7 FIFO overflow. This bit indicates a Command FIFO overflow condition. A read to the register clears the status.
0: FIFO operating normally
1: FIFO has overflown, one or more writes have been dropped.

Default: 0h

Pixel Map n Base Pointer Register Offset = 17-14 R/W

Bit Description

31-0 This register specifies the start of a Pixel Map in byte address.

Pixel Map n Width Register Offset = 19, 18 R/W

Bit Description

11-0 This register specifies the width of a Pixel Map minus one in pixel units. The programmed value is one less than the actual value.

15-12 Reserved

Pixel Map n Height Register Offset = 1B, 1A R/W

Bit Description

11-0 This register specifies the height of a Pixel Map minus one in pixel units. The programmed value is one less than the actual value.

15-12 Reserved

Pixel Map n Format Register Offset = 1C R/W

Bit Description

2-0 Specifies the number of bits/pixel in the Pixel Map

Bits 1,0	Number of bits/pixel
000	1-bit
001	Reserved
010	Reserved
011	8-bits
100	16-bits
101	32-bits
110	Reserved
111	Reserved

3 Motorola/Intel format selection for the Pixel Map
0: Intel format (little endian)
1: Motorola format (big endian)
Normal software practice uses Motorola format for 1 bpp, and Intel format for 8 bpp or higher color depths.

6-4 Reserved

7 System memory indication

0: Map is in local frame buffer, no special treatment is needed.
1: Map is in system memory, master cycle or CPU assisted cycle is needed.

Source and Destination Pixel Maps can have 1-, 8-, 16-, or 32-bits/pixel. Pattern Pixel Map can only have 1-bit/pixel. Programming Pattern Pixel Map to be anything other than 1-bit/pixel will produce undefined results. Mask Map Format Register cannot be programmed at all because it is assumed to be 1-bit/pixel. However, one should still program it to be 1-bit/pixel to ensure future compatibility.

Bresenham Error Term Register **Offset = 21, 20** **R/W**

Bit Description
13-0 This register specifies the Bresenham Error Term for the Line Draw function. This value is ((2*deltaY) - deltaX). This number is 14-bit sign extended two's complement ranged from -8192 to +8191.
15-14 Reserved

Bresenham Constant K1 **Offset = 25, 24** **R/W**

Bit Description
13-0 This register specifies the Bresenham Constant K1 for the Line Draw function. This value is (2*deltaY). This number is 14-bit sign extended two's complement ranged from -8192 to +8191.
15-14 Reserved

Bresenham Constant K2 **Offset = 29, 28** **R/W**

Bit Description
13-0 This register specifies the Bresenham Constant K2 for the Draw Line function. This value is 2*(deltaY - deltaX). This number is 14-bit sign extended two's complement ranged from -8192 to +8191.
15-14 Reserved

Direction Steps Register **Offset = 2F-2C** **R/W**

This register specifies up to four Draw and Step codes and initiate a Draw and Step operation by writing to byte three of this register. Each code is one byte. A write to "2F" starts the draw/step operation. The stop code "00" is used to terminate the draw and step operation when the operation requires less than 4 bytes.

Bit Description
3-0 Number of steps for code-1 from 1-16. The programmed number is one less than the actual drawn length.
4 Move/draw operation select for code-1
0: move operation - update X-Y pointers but no pixels are drawn
1: draws pixels as normal
7-5 Direction step for code-1. Direction is as follows:

11-8 Number of steps for code-2 from 1-16. The programmed number is one les[illegible] actual drawn length.
12 Move/draw operation select for code-2
0: move operation - update X-Y pointers but no pixels are drawn
1: draws pixels as normal
15-13 Direction step for code-2
19-16 Number of steps for code-3 from 1-16. The programmed number is one less than actual drawn length.

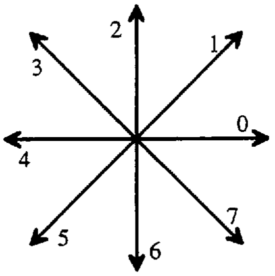


Figure 6. - Draw & Step Direction Codes

20 Move/draw operation select for code-3
0: move operation - update X-Y pointers but no pixels are drawn
1: draws pixels as normal
23-21 Direction step for code-3
27-24 Number of steps for code-4 from 1-16. The programmed number is one less than actual drawn length.
28 Move/draw operation select for code-4
0: move operation - update X-Y pointers but no pixels are drawn
1: draws pixels as normal
31-29 Direction step for code-4

Write to byte 3 (Bits 31-24), will start the Draw & Step process.

Figure A

Address	**2C**	**2D**	**2E**	**2F**
	Code-1	Code-2	Code-3	Code-4

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Direction Code			Move/Draw	Number of steps			

ROP Register **Offset = 48** **R/W**

Bit Description
3-0 These bits specified the Background ROP function to be performed between Destination and Source pixels during an operation where the Pattern pixel value is 0.
7-4 These bits specified the Foreground ROP function to be performed between Destination and Source pixels during an operation where the Pattern pixel value is 1.

Bits 7-4 (3-0)	Function	Reverse Polish
0h	zeros	0
1h	(NOT source) AND (NOT destination)	SDon
2h	(NOT source) AND destination	DSna
3h	NOT source	Sn
4h	source AND (NOT destination)	SDna
5h	NOT destination	Dn
6h	source XOR destination	SDx
7h	(NOT source) OR (NOT destination)	SDan
8h	source AND destination	SDa
9h	source XOR (NOT destination)	SDnx
Ah	destination	D
Bh	(NOT source) OR destination	DSno
Ch	source	S
Dh	source OR (NOT destination)	SDno
Eh	source OR destination	SDo
Fh	ones	1

With the combination of Foreground ROP and Background ROP under control of Pattern, the OTI-64107 can support all 256 ROP as defined by Windows.

Destination Color Compare Condition Register **Offset = 4A** **R/W**

Bit Description
2-0 These bits specify the Color Compare Condition under which Destination update is inhibited.

Bits 2-0	Destination Color Compare Condition
000	Always true (disable update)
001	Reserved
010	Dest = color compare value
011	Reserved
100	Dest <> color compare value
101	Reserved
110	Always false (enable update)
111	Reserved

7-3 Reserved
Default: 06h

Destination Color Compare Value Register **Offset = 4F-4C** **R/W**

Bit Description
31-0 Color value to be compared to Destination pixels when Color Compare is enabled. Only the corresponding number of bits-per-pixel in the Destination are required in this register. For example, if the Destination is 8 bits-per-pixel, only the eight low order bits of this register are used, and only Bit 0 is used for 1 bpp.

Pixel Bit Mask Register **Offset = 53-50** **R/W**

Bit Description
31-0 This register specifies which bits within each pixel are subject to be updated.
0 = Disable update for the particular bit
1 = Enable update for the particular bit
Only the corresponding number of bits-per-pixel in the Destination are required in this register. For example. if the Destination is 8 bits-per-pixel, only the eight low-order bits of this register are used, and o it 0 is used for 1 bpp.

Foreground Color ister **Offset = 5B-58** **R/W**

Bit Description
31-0 This regis cifies the Foreground Color to be used as the Foreground Source during operations. Onl orresponding number of bits-per-pixel in the Destination are required in this register. Fo. example, if the Destination is 8 bits-per-pixel, only the eight low order bits of this register are used, and only Bit 0 is used for 1 bpp.

Background Color Register **Offset = 5F-5C** **R/W**

Bit Description
31-0 This register specifies the Background Color to be used as the Background Source during operations. Only the corresponding number of bits-per-pixel in the Destination are required in this register. For example, if the Destination is 8 bits-per-pixel, only the eight low-order bits of this register are used, and only Bit 0 is used for 1 bpp.

Operation Dimension 1 Register **Offset = 61, 60** **R/W**

Bit Description
11-0 This register specifies the width of the rectangle to be drawn by the PxBlt function. The programmed number should be one less than the required number.
15-12 Reserved

Operation Dimension 2 Register **Offset = 63, 62** **R/W**

Bit Description
11-0 This register specifies the height of the rectangle to be drawn by the PxBlt function. The programmed number should be one less than the required number.
15-12 Reserved

The following registers 6C-7B have two stages, first stage is the register where data is written to, and the second stage is the actual counter where the data is read from. Since the counters are not loaded until a Co-Processor start is issued, it is not possible to read the registers correctly after they are written.

Mask Map Origin X Offset Register **Offset = 6D, 6C** **R/W**

Bit	Description
12-0	This register specifies the X Offset of the Mask Map origin relative to the origin of the Destination Map.
15-13	Reserved

Mask Map Origin Y Offset Register **Offset = 6F, 6E** **R/W**

Bit	Description
12-0	This register specifies the Y Offset of the Mask Map origin relative to the origin of the Destination Map.
15-13	Reserved

Source X Pointer Register **Offset = 71, 70** **R/W**

Bit	Description
11-0	This register specifies the X coordinate of the Source Map.
15-12	Reserved

Source Y Pointer Register **Offset = 73, 72** **R/W**

Bit	Description
11-0	This register specifies the Y coordinate of the Source Map.
15-12	Reserved

Pattern X Pointer Register **Offset = 75, 74** **R/W**

Bit	Description
11-0	This register specifies the X coordinate of the Pattern Map.
15-12	Reserved

Pattern Y Pointer Register **Offset = 77, 76** **R/W**

Bit	Description
11-0	This register specifies the Y coordinate of the Pattern Map.
15-12	Reserved

Destination X Pointer Register **Offset = 79, 78** **R/W**

Bit	Description
12-0	This register specifies the X coordinate of the Destination Map.
15-13	Reserved

Destination Y Pointer Register **Offset = 7B, 7A** **R/W**

Bit	Description
12-0	This register specifies the Y coordinate of the Destination Map.
15-13	Reserved

Pixel Operations Register **Offset = 7F-7C** **R/W**

Bit	Description
2-0	These bits specify the direction octant for PxBlt and Line Draw operations.

Bits 2,1,0	Direction Octant for PxBlt
00x	Start at top left corner of area, increasing right and down
01x	Start at bottom left corner of area, increasing right and up
10x	Start at top right corner of area, increasing left and down
11x	Start at bottom right corner of area, increasing left and up
Bits 2-0	**Direction Octant for Line Draw**
000	Octant 0
.	.
111	Octant 8

3	Reserved

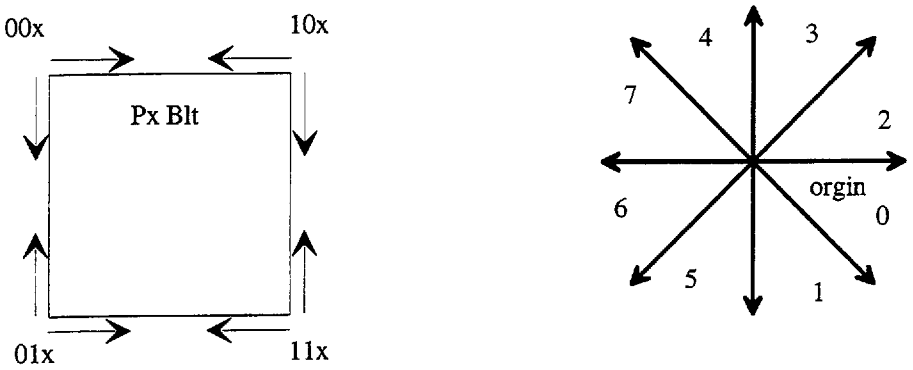
5-4 Drawing Mode Register. These bits determine the attributes of Line Draw and Draw and Step operations.

Bits 5,4	Drawing Mode
00	Draw all pixels
01	Draw first pixel null
10	Draw last pixel null
11	Draw area boundary

7-6 Mask Pixel Map Control

Bits 7,6	Mask Functions
00	Mask Map Disabled
01	Mask Map Boundary Enabled
10	Mask Map Enabled
11	Reserved

Figure B



Px Blt Direction Code

Line Draw Octant Code

11-8 Reserved

15-12 Pattern Pixel Map Control

Bits 15-12	Pattern Map Control
0000	Reserved.
0001	Use Pixel Map A for Pattern Map
0010	Use Pixel Map B for Pattern Map
0011	Use Pixel Map C for Pattern Map
01xx	Reserved
1000	Use Foreground ROP
1001	Generate Pattern from Source
1010	Reserved
.	.
1111	Reserved

19-16 Destination Pixel Map Control.

Bits 19-16	Destination Map Control
0000	Reserved
0001	Use Pixel Map A for Destination Map
0010	Use Pixel Map B for Destination Map
0011	Use Pixel Map C for Destination Map
0100	Reserved
.	.
1111	Reserved

23-20 Source Pixel Map Control

Bits 23-20	Source Map Control
0000	Reserved
0001	Use Pixel Map A for Source Map

0010	Use Pixel Map B for Source Map
0011	Use Pixel Map C for Source Map
0100	Reserved
.	.
1111	Reserved

27-24 Co-processor Function Control

Bits 27-24	Function Control
0000	Reserved
0001	Reserved
0010	Draw and Step Read
0011	Line Draw Read
0100	Draw and Step Write
0101	Line Draw Write
0110	Reserved
0111	Reserved
1000	PxBlt
1001	Inverting PxBlt
1010	Area Fill PxBlt
1011	TextBlt. This mode is designed to enhance text output. To use this mode, the monochrome font must be stored as pattern map, and the font (pattern) map base address plus map width must always be within one quad word (64-bit), and operation dimension one must be less than or equal to map width (no tiling). For example, if base address=0, then map width can be any size up to 64; if base address=38h, then map width can only be up to 8.
1100	Fast Pattern Copy
.	.
1111	Reserved

29-28 Foreground Source. These bits specify the foreground source to be combined with Destination when the Pattern pixel is a 1.

Bits 29-28	Foreground Source
00	Use Foreground Color Register
01	Reserved
10	Use Source pixel map
11	Reserved

31-30 Background Source. These bits specify the background source to be combined with Destination when the Pattern pixel is a 0.

Bits 31-30	Background Source
00	Use Background Color Register
01	Reserved
10	Use Source pixel map
11	Reserved

CHAPTER 8: VGA REGISTERS

8.1 VGA Register Summary

General Registers

Register Description	R/W	Port	Index	Bits	Block
Miscellaneous Output Register	R/W	3CC/3C2	-	7	BI
Input Status Register 0	RO	3C2	-	2	BI
Input Status Register 1	RO	3?A	-	4	BI
Feature Control Register	R/W	3CA/3?A	-	1	BI

Note: ? = B for monochrome, ? = D for color

Sequencer Registers

Register Description	R/W	Port	Index	Bits	Block
Sequencer Address	R/W	3C4	-	3	SEQ
Reset	R/W	3C5	0	2	SEQ
Clocking Mode	R/W	3C5	1	5	SEQ
Map Mask	R/W	3C5	2	4	SEQ
Character Map Select	R/W	3C5	3	6	SEQ
Memory Mode	R/W	3C5	4	3	SEQ

CRT Controller Registers

Register Description	R/W	Port	Index	Bits	Block
CRT Controller Address	R/W	3?4	-	6	CRTC
Horizontal Total	R/W	3?5	0	8	CRTC
Horizontal Display Enable End	R/W	3?5	1	8	CRTC
Start Horizontal Blanking	R/W	3?5	2	8	CRTC
End Horizontal Blanking	R/W	3?5	3	8	CRTC
Start Horizontal Retrace Pulse	R/W	3?5	4	8	CRTC
End Horizontal Retrace	R/W	3?5	5	8	CRTC
Vertical Total	R/W	3?5	6	8	CRTC
Overflow	R/W	3?5	7	8	CRTC
Preset Row Scan	R/W	3?5	8	7	CRTC
Maximum Scan Line	R/W	3?5	9	8	CRTC
Cursor Start	R/W	3?5	A	6	CRTC
Cursor End	R/W	3?5	B	7	CRTC
Start Address High	R/W	3?5	C	8	CRTC
Start Address Low	R/W	3?5	D	8	CRTC
Cursor Location High	R/W	3?5	E	8	CRTC
Cursor Location Low	R/W	3?5	F	8	CRTC
Vertical Retrace Start	R/W	3?5	10	8	CRTC
Vertical Retrace End	R/W	3?5	11	8	CRTC
Vertical Display Enable End	R/W	3?5	12	8	CRTC
Offset	R/W	3?5	13	8	CRTC
Underline Location	R/W	3?5	14	7	CRTC
Start Vertical Blank	R/W	3?5	15	8	CRTC
End Vertical Blank	R/W	3?5	16	8	CRTC
CRTC Mode Control	R/W	3?5	17	7	CRTC
Line Compare	R/W	3?5	18	8	CRTC

Note: ? = B for monochrome, ? = D for color

Graphics Controller Registers

Register Description	R/W	Port	Index	Bits	Block
Graphics Address	R/W	3CE	-	4	GRF
Set/Reset	R/W	3CF	0	4	GRF
Enable Set/Reset	R/W	3CF	1	4	GRF
Color Compare	R/W	3CF	2	4	GRF
Data Rotate	R/W	3CF	3	5	GRF
Read Map Select	R/W	3CF	4	2	GRF
Graphics Mode	R/W	3CF	5	6	GRF
Miscellaneous	R/W	3CF	6	4	GRF
Color Don't Care	R/W	3CF	7	4	GRF
Bit Mask	R/W	3CF	8	8	GRF

Attribute Controller Registers

Register Description	R/W	Port	Index	Bits	Block
Attribute Address	R/W	3C0	-	6	ATR
Palette	R/W	3C1/3C0	00-0F	6x16	ATR
Attribute Mode Control	R/W	3C1/3C0	10	7	ATR
Overscan Color	R/W	3C1/3C0	11	8	ATR
Color Plane Enable	R/W	3C1/3C0	12	6	ATR
Horizontal Pel Panning	R/W	3C1/3C0	13	4	ATR
Color Select	R/W	3C1/3C0	14	4	ATR

DAC Registers

Register Description	R/W	Port	Index	Bits	Block
PEL Address (Write Mode)	R/W	3C8	-		BI
PEL Address (Read Mode)	WO	3C7	-		BI
DAC State	RO	3C7	-		BI
PEL Data	R/W	3C9	-		BI
PEL Mask	R/W	3C6	-		BI

8.2 General Purpose Registers

3CC/3C2 Miscellaneous Output Register **R/W**

Bit Description

0 Input/Output Address Select - This bit maps the CRTC I/O addresses for monochrome or color emulation.
0 = Monochrome emulation with CRTC addresses set to 3Bx hex, Input Status 1 register set to 3BA hex.
1 = Color emulation with CRTC addresses set to 3Dx, Input Status 1 register set to 3DA hex.

1 Enable RAM
0 = Disable Display DRAM address decode from the system microprocessor
1 = Enable Display DRAM to the system microprocessor

3,2 Clock Select - These bits are the same as Bits 1-0 of Clock Select Register. These bits represent the state of signals CSEL[1] and CSEL[0], respectively.

4 Reserved

5 Page Bit For Odd/Even - Selects between two pages of memory when in the odd or even modes.
0 = Low 64 Kbyte page of memory
1 = High 64 Kbyte page of memory

6 Horizontal Sync Polarity
0 = Positive Vertical Retrace
1 = Negative Vertical Retrace

7 Vertical Sync Polarity
0 = Positive Vertical Retrace
1 = Negative Vertical Retrace

Bits 7 and 6 are used to select the vertical size of the monitor as follows:

Bits 7,6	Vertical Size
00	Reserved
01	400 lines
10	350 lines
11	480 lines

3C2 Input Status Register 0 RO

Bit Description

3-0 Reserved

4 Switch Sense Bit - Reports the status of one of the four switches selected via the clock select of the Miscellaneous Output register. This bit allows the power-on initialization to determine if a monochrome or color monitor is connected to the system.
0 = Selected sense switch is off or 0
1 = Selected sense switch is on or 1

6,5 Reserved

7 CRT Interrupt
0 = Vertical retrace interrupt is pending
1 = Vertical retrace interrupt is cleared

3?A Input Status Register 1 R

Bit Description

0 Display Enable - Monitors the status of the display. To avoid glitches on the display, some programs use this bit to restrict screen updates to de-activate display intervals. The VGA has been designed to eliminate this software requirement, so display screen updates may be made at any time.
0 = The display of video data is enabled
1 = The display is in horizontal or vertical retrace mode

2,1 Reserved

3 Vertical Retrace
0 = Video information is being displayed
1 = A vertical retrace interval is in progress

5,4 Diagnostic Usage - Reports the status of two of the eight VGA attribute controller outputs. The values set into the Video Status MUX field of the Color Plane Enable Register determine which colors are input to these two diagnostic bits.

Bits 5,4	Pixel Data
00	P2, P0
01	P5, P4
10	P3, P1
11	P7, P6

7,6 Reserved

3CA/3?A Feature Control Register R/W

Bit Description

2-0 Reserved

3 Vertical Sync Select
0 = This bit should always be set to 0 to enable normal vertical sync output to the monitor.
1 = The vertical sync output is the logical OR of vertical sync and vertical display enable.

7-4 Reserved

8.3 Sequencer Registers

3C4 Sequencer Address Register **R/W**

Bit Description

2-0 Sequencer Address Bits - A binary value pointing to the register where data is to be read from or written to.

7-3 Reserved

3C5 Reset Register **Index=0** **R/W**

Bit Description

0 Asynchronous Reset
0 = Asynchronous clear and halt the Sequencer. This may cause data loss in the DRAMs.
1 = Bits 1 and 0 must both be 1's to allow the Sequencer to operate.

1 Synchronous Reset - This bit should be set to 0 before changing Bit 0 or Bit 3 of the Clocking Mode register or Bit 2 or Bit 3 of the Miscellaneous Output register, or all bits of Register 3DF, Index D.
0 = Synchronous clear and halt the Sequencer
1 = Bits 1 and 0 must both be 1's to allow the Sequencer to operate.

7-2 Reserved

Default: 0h

3C5 Clocking Mode Register **Index=1** **R/W**

Bit Description

0 8/9 Dot Clocks - The nine dot mode is for Alphanumeric modes only. The ninth dot equals the eighth dot for ASCII codes C0 though DF hex. Also, see the Line Graphics Character Code bit in the Attribute Mode Control register section.
0 = Directs the sequencer to generate nine dot wide character clocks
1 = Generate eight dot wide character clocks

1 Reserved

2 Shift Load
0 = If Bit 4 is set to 0, also, the video serializers are reloaded every character clock.
1 = The video serializers are reloaded every other character clock. this mode is useful when 16 bits are fetched per cycle and chained together in the shift load registers.

3 Dot Clock divided by two
0 = Select the dot clock to be the same frequency as the master clock
1 = The master clock will be divided by two to generate the dot clock. This is used for 320 and 360 horizontal PEL modes.

4 Shift 4
0 = The video serializers are reloaded every character clock
1 = The serializers are loaded every fourth character clock. This is useful when 32 bits are fetched per cycle and chained together in the shift registers.

5 Screen Off - This bit is used for fast full-screen updates.
0 = Normal screen operation
1 = Turns off the video screen and assigns the maximum memory bandwidth to the system CPU

7,6 Reserved

Default: 0h

3C5 Map Mask Register **Index=2** **R/W**

Bit Description

3-0 Map Mask for planes 3-0, respectively. Bit 0 is mask memory Plane 0, Bit 1 is mask memory Plane 1, etc. For odd/even modes, maps 0 and 1, and maps 2 and 3 should have the same map mask value. When Chain 4 mode is selected, all maps should be enabled. This is a read-modify-write operation.
0 = Disable memory write to the corresponding map
1 = Enables the system to write to the corresponding map. If all four bits are set to one, the system CPU can perform a 32-bit operation with only one memory cycle.

7-4 Reserved

3C5 Character Map Select Register Index=3 R/W

Bit Description

1,0 Character Map Select B - Selects the portion of Map 2 used to generate Alpha characters with Bit 4 as the high bit when attribute Bit 3 is 0.

Bit 4,1,0	Map	Table Location	First Character Offset Address
000	0	1st 8 k of Map 2	0 K
001	1	3rd 8 k of Map 2	16 K
010	2	5th 8 k of Map 2	32 K
011	3	7th 8 k of Map 2	48 K
100	4	2nd 8 k of Map 2	8 K
101	5	4th 8 k of Map 2	24 K
110	6	6th 8 k of Map 2	40 K
111	7	8th 8 k of Map 2	56 K

3,2 Character Map Select A - Selects the portion of Map 2 used to generate Alpha characters with Bit 5 as the high bit when attribute Bit 3 is 1.

Bit 5,3,2	Map	Table Location	First Character Offset Address
000	0	1st 8 k of Map 2	0 K
001	1	3rd 8 k of Map 2	16 K
010	2	5th 8 k of Map 2	32 K
011	3	7th 8 k of Map 2	48 K
100	4	2nd 8 k of Map 2	8 K
101	5	4th 8 k of Map 2	24 K
110	6	6th 8 k of Map 2	40 K
111	7	8th 8 k of Map 2	56 K

4 Character Map Select High Bit B

5 Character Map Select high Bit A

7,6 Reserved

Default: 0h

Bit 3 of the attribute byte normally controls the ON/OFF of the foreground intensity in text modes. This bit may be redefined as a switch between character sets. For this feature to be enabled, the following must be true:

1) The setting value of Character Map Select A does not equal the value of Character Map Select B.
2) The Memory Mode register Bit 1, must be equal to 1.
3) If either of these are not true, the first 16 K of Map 2 is used.

3C5 **Memory Mode Register** **Index=4** **R/W**

Bit Description

0 Reserved

1 Extended Memory

0 = No extended memory present. Display memory is less than 64 Kbytes.

1 = Extended memory is present. Display memory is greater than 64 Kbytes. If set to one, the VGA is permitted to use the 256 Kbytes of video memory. This also enables character map selection.

2 Odd/Even

0 = Directs even CPU addresses to access Maps 0 and 2, and odd CPU addresses to access maps 1 and 3.

1 = If Bit 3 is set to 0, this bit causes system CPU addresses to sequentially address data within a bit map.

3 Chain 4

0 = If Bit 2 is set to 1, it enables the system CPU to address data sequentially within a bit map by use of the Map Mask register.

1 = Causes two low-order address bits to select the map that will be accessed as follows:

A1	AO	Map Selected
0	0	0
0	1	1
1	0	2
1	1	3

7-4 Reserved

8.4 CRT Controller Registers

CRT Controller registers resides at either 3B4/5 for Monochrome emulation modes or 3D4/5 for Color emulation modes depending on Bit 0 of the Miscellaneous Output register at address 3C2 hex.

3?4 **CRT Controller Address Register** **R/W**

Bit Description

4-0 CRT Controller Address Bits - A binary value programmed in these bits selects one of the CRT Controller registers where data is to accessed.

5 Test Bit - Must remain 0

7,6 Reserved

3?5 **Horizontal Total Register** **Index=0** **R/W**

Bit Description

7-0 Horizontal Total - This register defines the total number of characters in the horizontal scan interval including the retrace time. This value directly controls the period of the horizontal retrace output signal. Character clock inputs to the CRT controller and are counted by an internal horizontal character counter. This value is compared with the horizontal character values to provide horizontal timings. All horizontal and vertical timings are based upon the horizontal register. The value programmed is five less than the desired value.

3?5 **Horizontal Display Enable End Register** **Index=1** **R/W**

Bit Description

7-0 Horizontal Display Enable End - The total number of displayed characters minus one. This register defines the length of the horizontal display enable signal. It determines the number of displayed characters per horizontal line.

3?5 **Start Horizontal Blanking Register** **Index=2** **R/W**

Bit Description

7-0 Start Horizontal Blanking - Determines when to start the internal horizontal blanking output signal. When the internal character counter reaches this value, the horizontal blanking signal becomes active.

3?5 **End Horizontal Blanking Register** **Index=3** **R/W**

Bit Description

4-0 End Horizontal Blanking - The horizontal blanking signal width is determined as follows: Value of Start Blanking register + width of blanking signal in character clock units = 6-bit result to be programmed into the End Horizontal blanking register. Bit 5 is located in the End Horizontal Retrace register. If these six bits equal the six least significant bits of the horizontal character counter, the horizontal blanking signal becomes inactive.

6,5 Display Enable Skew Control - These two bits indicate the magnitude of display enable skew as shown below:

Bits 6,5	Skew (in character clocks)
00	0
01	1
10	2
11	3

7 Test Bit - Must be set to 1

3?5 **Start Horizontal Retrace Pulse Register** **Index=4** **R/W**

Bit Description

7-0 Start Horizontal Retrace Pulse - This register is used to center the screen horizontally and to specify the character position at which the Horizontal Retrace Pulse becomes active. The value in this register is a binary count of the character position at which the signal becomes active.

3?5 **End Horizontal Retrace Register** **Index=5** **R/W**

Bit Description

4-0 End horizontal Retrace - The value programmed here is compared to the five least-significant bits of the horizontal character counter. When they are equal, the horizontal retrace signal becomes inactive (logical 0). To calculate the width of the retrace signal use the following algorithm: Value of Start Horizontal Retrace register + width of Horizontal Retrace signal in character clock units = 5-bit result to be programmed into the End Horizontal Retrace register.

6,5 Horizontal Retrace Delay - These bits control the skew of the Horizontal Retrace signal as follows:

Bit 6,5	Skew (in character clocks)
00	0
01	1
10	2
11	3

7 End Horizontal Blanking, Bit 5 - The first four bits are located in the End Horizontal Blanking register (index 03 hex).

3?5 **Vertical Total Register** **Index=6** **R/W**

Bit Description

7-0 Vertical Total - This is the low-order eight bits of a 10-bit register that represents the number of horizontal raster scans on the CRT screen, minus two, including vertical retrace. This value determines the period of the vertical retrace signal. Bits 8 and 9 of the Vertical Total are located in the CRT Controller Overflow Register 07, hex Bit 0 and 5, respectively.

3?5 **CRT Controller Overflow Register** **Index=7** **R/W**

Bit Description

0 Vertical Total - Bit 8 of the Vertical Total register (index 06 hex)

1 Vertical Display Enable End - Bit 8 of the Vertical Display Enable End register (index 12 hex)

2 Vertical Retrace Start - Bit 8 of Vertical Retrace Start register (index 10 hex)

3 Start Vertical Blank - Bit 8 of the Start Vertical Blank register (index 15 hex)

4 Line Compare - Bit 8 of the Line Compare register (index 18 hex)

5 Vertical Total - Bit 9 of the Vertical Total register (index 06 hex)

6 Vertical Display Enable End - Bit 9 of the Vertical Display Enable End register (index 12 hex)

7 Vertical Retrace Start - Bit 9 of the Vertical Retrace Start register (index 10 hex)

3?5 Preset Row Scan Register Index=8 R/W

Bit Description

4-0 Preset Row Scan (PEL Scrolling) - These bits specify the starting row scan count after a vertical retrace. The row scan counter increments each horizontal retrace time until a maximum row scan occurs. At maximum row scan compare time, the scan is cleared (not prese

6,5 Byte Panning Control - This field controls byte panning in modes programmed in multiple shift modes, which is required for PEL-panning operations. The Horizontal PEL Panning register in the Attribute Controller provides panning of up to eight individual PEL-panning operations. In single byte shift modes, the CRT Controller start address is incremented and attribute panning is reset to the next higher PEL. In multiple shift modes, the byte pan bits are used as extensions to the attribute PEL Panning register. Together, these bits allow up to three characters to be panned.

7 Reserved

3?5 Maximum Scan Line Register Index=9 R/W

Bit Description

4-0 Maximum Scan Line - These bits specify the number of scan lines per character row. The number to be programmed is the maximum row number minus one.

5 Start Vertical Blank - Bit 9 of the Start Vertical Blank register (index 15 hex).

6 Line Compare - Bit 9 of the Line Compare register (index 18 hex).

7 200 to 400 Line Conversion -

0 = The clock to the row scan counter is equal to the horizontal scan rate. Line doubling is disabled.

1 = The clock in the row scan counter is divided by two. This allows the older 200-line modes to be displayed as 400 lines on the display. This is referred to as line doubling.

3?5 Cursor Start Register Index=A R/W

Bit Description

4-0 Cursor Start - This field specifies the row scan line of a character line where the cursor is to begin. The number programmed is one less than the starting cursor row scan. If the Cursor Start register is programmed with a value greater than the cursor End register, no cursor is generated.

5 Cursor Off

0 = Turns on the cursor

1 = Turns off the cursor

7,6 Reserved

3?5 Cursor End Register Index=B R/W

Bit Description

4-0 Cursor End - This field specifies the row scan of a character line where the cursor is to end.

6,5 Cursor Skew - These bits control the skew of the cursor signal. Cursor skew delays the cursor by the selected number of clocks.

Bits 6,5	Skew (in character clocks)
00	0
01	1
10	2
11	3

7 Reserved

3?5 Start Address High Register Index=C R/W

Bit Description

7-0 Start Address High - This register contains the eight high-order bits of the start address. The 16-bit value, from the high-order and low-order Start Address registers, is the first address after the vertical retrace on screen refresh.

3?5 Start Address Low Register Index=D R/W

Bit Description

7-0 Start Address Low - This register contains the eight low-order bits of the start address.

3?5 **Cursor Location High Register** **Index=E** **R/W**

Bit Description

7-0 Cursor High - This register contains the eight high-order bits of the cursor location.

3?5 **Cursor Location Low Register** **Index=F** **R/W**

Bit Description

7-0 Cursor Low - This register contains the eight low-order bits of the cursor location.

3?5 **Vertical Retrace Start Register** **Index=10** **R/W**

Bit Description

7-0 Vertical Retrace Start -This register contains the eight low-order bits of the Vertical Retrace Start position, programmed in horizontal scan lines. Bit 8 and 9 are in the CRTC Overflow register (index 07 hex).

3?5 **Vertical Retrace End Register** **Index=11** **R/W**

Bit Description

3-0 Vertical Retrace End -This field determines the horizontal scan count value when the vertical retrace output signal becomes inactive. This register is programmed in units of horizontal scan lines. To obtain a vertical retrace signal of width W, use the following algorithm: Value of Start Vertical Retrace register + width of vertical retrace signal in horizontal scan units = 4-bit result to be programmed into the End Horizontal Retrace register.

4 Clear Vertical Interrupt
0 = Clears the vertical retrace interrupt flip-flop
1 = No effect

5 Enable Vertical Interrupt
0 = Enables a vertical retrace interrupt (on IRQ2). This interrupt level may be shared so the Input Status register 0, Bit 7 should be checked to find out is the VGA caused the interrupt to occur.
1 = Disable the vertical retrace interrupt

6 Select five Refresh Cycles
0 = Selects three refresh cycles. The BIOS sets this bit to 0 during a mode set, a reset, or power on.
1 = Selects five refresh cycles per horizontal line. This allow the use of the VGA chip with slow sweep rate displays (15.75 kHz).

7 Protect CRT Registers 0-7
0 = Enables writing to CRTC registers 0-7
1 = Disables writing to CRTC registers 0-7. The line compare Bit 4 in Register 07 hex is not protected.

3?5 **Vertical Display Enable End Register** **Index=12** **R/W**

Bit Description

7-0 Vertical Display Enable End - This register contains the eight low-order bits of a 10-bit register that defines the Vertical Display Enable End position. Bits 8 and 9 are located in the CRT Controller Overflow register 07 hex, Bits 1 and 6, respectively.

3?5 **Offset Register** **Index=13** **R/W**

Bit Description

7-0 Offset - This register specifies the logical line width of the screen. The starting memory address for the next character row is computed by the current byte start address + (Offset register contents x N), where N = 2 for byte addressing and N = 4 for word addressing.

3?5 Underline Location Register Index=14 R/W

Bit Description

4-0 Underline Location - This field specifies the horizontal row scan of a character row on which an underline occurs. The value programmed is one less than the scan line number desired.

5 Count By 4
0 = Normal clocking
1 = The memory address counter is clocked with the character clock divided by four

6 Doubleword Mode
0 = Normal word addressing mode
1 = Memory addresses are doubleword addresses

7 Reserved

3?5 Start Vertical Blanking Register Index=15 R/W

Bit Description

7-0 Start Vertical Blanking - This register contains the eight low-order bits of a 10-bit register. Bit 8 is in the CRTC Overflow register (index 07 hex). Bit 9 is in the Maximum Scan Line register (index 09 hex).

3?5 End Vertical Blanking Register Index=16 R/W

Bit Description

7-0 End Vertical Blanking - This register specifies the horizontal scan count value when the Vertical Blank output signal becomes inactive. It is programmed in units of horizontal scan lines. To obtain the Vertical Blank signal of width W, use the following algorithm: Value of Start Vertical Blanking register minus 1 + width of Vertical Blank signal in horizontal scan units = 8-bit result to be programmed into the End Vertical Blanking register.

3?5 CRTC Mode Control Register Index=17 R/W

Bit Description

0 Compatibility Mode Support, used for CGA compatability

0 = Row scan address Bit 0 is substituted for memory address Bit 13 during active display time.

1 = Enables memory address Bit 13 to appear on the memory address output Bit 13 of the CRT controller.

1 Select Row Scan Counter, used for Hercules monochrome adapter compatability

0 = Selects row scan counter Bit 1 for CRT memory address Bit MA14

1 = Selects MA14 counter bit for CRT memory address Bit MA14

2 Horizontal Retrace Select

0 = Selects normal horizontal retrace as the clock that controls the vertical timing counter.

1 = Selects horizontal retrace divided by two as the clock that controls the vertical timing counter. Therefore, the vertical resolution is doubled to 2048 horizontal scan lines.

3 Count By Two

0 = The memory address counter is clocked with the character clock input

1 = Clocks the memory address counter with the character clock input divided by two

4 Reserved

5 Address Wrap - Selects memory address counter Bit MA13 or Bit MA15, and it appears on the MA0 output pin in the word address mode. If the VGA is not in word address mode, MA0 counter output appears on the MA0 output pin.

0 = Selects MA13. This is selected in applications where only 64 K memory is present.

1 = Selects MA15. This should be selected in odd/even mode since 256 K of video memory is installed on the system board.

6 Word Mode or Byte Mode - Bit 6 of the End Vertical Blanking register in the CRT Controller also controls the addressing. When it is set to 0, Bit 6 of this register has control. When it is set to 1, the addressing is forced to be shifted by two bits.

0 = The word mode shifts all memory address counter bits down one bit, and the most-significant bit of the counter appears on the least-significant bit of the memory address outputs.

1 = Selects the byte address mode

7 Hardware Reset

0 = Forces horizontal and vertical retrace to clear

1 = Forces horizontal and vertical retrace to be enabled

3?5 Line Compare Register Index=18 R/W

Bit Description

7-0 Line Compare - This register is the eight low-order bits of the compare target. When the vertical counter reaches this value, the internal start of the line counter is cleared. Because of this, an area of the screen is not affected by scrolling. Bit 8 is located in the Overflow register 07 hex. Bit 9 is located in the Maximum Scan Line register 09 hex.

8.5 Graphics Controller Registers

3CE Graphics Address Register **R/W**

Bit Description

3-0 Graphics Address Bits - These bits are used to point to the other registers in the graphics section.

7-4 Reserved

3CF Set/Reset Register **Index=0** **R/W**

Bit Description

3-0 This field represents the value written to all four bits of the respective memory map when the system CPU does a memory write with Write mode 0 selected and Set/Reset mode is enabled for the corresponding map.

7-4 Reserved

3CF Enable Set/Reset Register **Index=1** **R/W**

Bit Description

3-0 Enable Set/Reset - This field enables the set/reset function.

0 = If Write mode is 0 and Set/Reset is not enabled on a map, that map is written with the value of the system CPU.

1 = If Write mode is 0 and Set/Reset is enabled on a map, the respective memory is written with the value of the Set/Reset register.

7-4 Reserved

3CF Color Compare Register **Index=2** **R/W**

Bit Description

3-0 This field represents a 4-bit color value to be compared. If the system CPU sets Read mode 1 in the graphics section and does a memory read, the data returned from the memory cycle will be a 1 in each bit position where the four maps equal the Color Compare register. All the bits of the corresponding map's byte are compared with the value of the Color Compare bits. Each of the 8-bit positions of the selected byte are then compared across the four maps and a 1 is returned in each bit position where the bits of all four maps equal their respective Color Compare values.

7-4 Reserved

3CF Data Rotate Register **Index=3** **R/W**

Bit Description

2-0 Rotate Count - This field represents a binary encoded value of the number of positions to right-rotate the system CPU data bus during system CPU memory writes. This operation is done when Write mode is 0. To write non-rotated data, the system CPU must select a count of 0.

4,3 Function Select - Data written to memory can operate logically with data already in the system CPU latches. Data can be any of the choices selected by the Write Mode register except system CPU latches, which may not be modified. If rotated data is selected, the rotate applies before the logical function. The bit functions are defined as follows:

Bits 4,3	Function
00	Data unmodified
01	Data ANDed with latch data
10	Data ORed with latch data
11	Data XORed with latch data

7-5 Reserved

3CF Read Map Select Register **Index=4** **R/W**

Bit Description

1,0 Map Select - This field represents a binary encoded value of the memory map number from which the system CPU reads data. This register has no effect on the color compare read mode. In odd/even modes the value may be 00 or 01 (10 or 11) for chained maps 0,1 (2,3).

7-5 Reserved

3CF Graphics Mode Register Index=5 R/W

Bit Description

1,0 Write Mode - The logic function specified by the Function Select register is applied to data being written to memory following modes 0, 2, and 3 below. The bit functions are defined as follows:

Bits 1,0	Function
00	Each memory map is written with the system CPU data rotated by the number of counts in the Rotate register, unless Set/Reset is enabled for the map. Maps for which Set/Reset is enabled are written with 8-bits of the value contained in the Set/Reset register for that map.
01	Each memory map is written with the contents of the system CPU latches. These latches are loaded by a system CPU Read operation.
10	Memory map n (0-3) is filled with 8-bits of the value of data bit n.
11	Each map is written with 8-bits of the value contained in the Set/Reset register for that map (Enable Set/Reset register has no effect). Rotated system CPU data is ANDed with the Bit Mask register data to form an 8-bit value that performs the same function as the Bit Mask register does in Write modes 0 and 2.

5 Shift Register - A logical 1 instructs the Shift registers in the graphics section to format the serial data stream with even-numbered bits from both maps on the even-numbered maps and odd-numbered bits from both maps on the odd maps. This bit is used for modes 4 and 5.

6 256 Color Mode
0 = Allows Bit 5 to control the loading of the Shift registers.
1 = Causes the Shift register to be loaded in a manner that supports the 256-color mode.

7 Reserved

3CF Miscellaneous Register Index=6 R/W

Bit Description

0 Graphics Mode - This bit controls text mode addressing control.
0 = Selects text mode operation
1 = Selects graphics mode. When this mode is selected, the character generator address latches are disabled.

1 Odd/Even
0 = Standard VGA addressing
1 = Replace system CPU address Bit 0 with a higher-order address bit and select odd/even maps with odd/even values of the system CPU A0 bit, respectively.

3,2 Memory Map - This field controls the mapping of the regenerative buffer into the system CPU address space. The bits are defined as follows:

Bits 3,2	Function
00	Hex A0000 for 28 K bytes
01	Hex A0000 for 64 K bytes
10	Hex B0000 for 32 K bytes
11	Hex B8000 for 32 K bytes

4-7 Reserved

3CF Color Don't Care Register Index=7 R/W

Bit Description

0 Map 0
0 = Don't participate in the color compare cycle
1 = Participate in the color compare cycle

1 Map 1
0 = Don't participate in the color compare cycle
1 = Participate in the color compare cycle

2 Map 2
0 = Don't participate in the color compare cycle
1 = Participate in the color compare cycle

3 Map 3
0 = Don't participate in the color compare cycle
1 = Participate in the color compare cycle

7-4 Reserved

3CF Bit Mask Register Index=8 R/W

Bit Description

7-0 Bit Mask - Bit mask applies to write modes 0 and 2. To preserve bits using the bit mask, data must be latched internally by reading the location. When data is written to preserve the bits, the most current data latched is written in those positions. The Bit Mask applies to all maps simultaneously.

0 = Any bit set to 0 causes the corresponding bit n in each map to be immune to change, provided that the location being written was the last location read by the system CPU.

1 = Bits set to 1 allows unimpeded writes to the corresponding bits in the maps.

8.6 Attribute Controller Registers

3C0 Attribute Address Register **R/W**

Bit Description

4-0 Attribute Address Bits -These bits point to the other registers in the Attribute section. The address and data registers share the same I/O address during write operation. An internal address flip-flop controls this selection. To initialize the flip-flop, an I/O Read instruction must be sent to the Attribute Controller at address 3BA or 3DA. This clears the flip-flop. and selects the Address register. After the Address register has been loaded with a write operation to 3C0, the next I/O write to 3C0 loads the Data register. The flip-flop toggles each time an I/O write instruction is sent to the Attribute controller.

5 Palette Address Source - Bit 5 must be cleared to 0 when loading the color Palette registers. For normal operation, Bit 5 is set to 1. This enables the video memory data to access the palette registers.

7,6 Reserved

3C1/3C0 Palette Registers Hex 00 Through 0F **Index=0-F** **R/W**

Bit Description

5-0 Palette - These bits allow a dynamic mapping between the text attribute or graphic color input value and the display color on the CRT screen. A logical 1 selects the appropriate color. The palette registers should be modified only during the vertical retrace interval to avoid problems with the display image. These internal Palette register values are sent off the chip to the video DAC, where they in turn serve as addresses into the DAC internal registers.

7,6 Reserved

3C1/3C0 Attribute Mode Control Register **Index=10** **R/W**

Bit Description

0 Graphics/Alphanumeric Mode
0 = Selects alphanumeric mode
1 = Selects graphics mode

1 Monochrome Emulation
0 = Color emulation mode is set
1 = Monochrome emulation mode is set

2 Enable line Graphics Character Codes
0 = The ninth dot will be the same as the background
1 = Enables the special line graphics character codes for Monochrome emulation mode. When enabled, this bit forces the ninth dot of a line graphic character to be identical to the eighth dot of the character. The line graphics character modes for the Monochrome emulation mode are C0 hex through DF hex. For character fonts that do not use the line graphics character codes, Bit 2 should be set to 0. Otherwise, unwanted video information will be displayed on the CRT screen.

3 Enable Blink/Select Background Intensity
0 = Selects the background intensity of the attribute input, which was available on the Monochrome and CGA adapters.
1 = Enables the blink attribute in text modes and blinking graphics modes

4 Reserved

5 PEL Panning Compatibility
0 = Line compare has no effect on the output of the PEL Panning register
1 = A successful line compare in the CRT controller forces the output of the PEL Panning registers to 0 until +VSYNC occurs, at which time the output returns to its programmed value. This bit allows a selected portion of the screen to be panned.

6 PEL Width
0 = Pixel data is changed at each cycle of the dot clock, for all modes except for mode 13 hex
1 = Pixel data is changed every second cycle of the dot clock. The video pipeline is sampled so that 8-bits are available to select a color in the 256-color mode (hex 13).

7 P5,P4 Select - This bit selects the source for the P5 and P4 digital video bits that go off the chip.
0 = P5 and P4 are the outputs of the Palette registers
1 = P5 and P4 are bits 1 and 0 of the Color Select register

3C1/3C0 Overscan Color Register Index=11 R/W

Bit Description

7-0 Overscan Color- This register determines the overscan (border) color displayed on the CRT screen. This border is a band of color around the perimeter of the display area. Its width is defined by the time when display enable and blank are both inactive and is not supported in the 40-column text modes or the 320-PEL graphics modes, except for mode 13 hex.

3C1/3C0 Color Plane Enable Register Index=12 R/W

Bit Description

3-0 Enable Color Plane - Setting any of these bits to 1, enables the respective display memory color plane.

5,4 Video Status MUX - Selects two of the eight color outputs to be available on the status port. The combinations available and the color output wiring are shown below:

Color Plane Register Bits 5,4	Input Status Register 1 Bits 5,4
00	P2, P0
01	P5, P4
10	P3, P1
11	P7, P6

6,7 Reserved

3C1/3C0 Horizontal PEL Panning Register Index=13 R/W

Bit Description

3-0 Horizontal PEL Panning - This register selects the number of picture elements (PELs) to shift the video data horizontally to the left. PEL panning is available in both graphics and text modes. In monochrome emulation text modes and modes 0+, 1+, 2+, 3+ , the image can be shifted a maximum of eight PELs. Mode 13 allows a maximum shift of three PELs. All other modes, the image can be shifted a maximum of seven PELs. The sequence for shifting the image is as follows.

PEL Panning	Number of PELs Shifted to the left		
Register Value	0+,1+,2+,3+,7,7+	All Other modes	Mode 13
0	1	0	0
1	2	1	-
2	3	2	1
3	4	3	-
4	5	4	2
5	6	5	-
6	7	6	3
7	8	7	-
8	0	-	-

7-4 Reserved

3C1/3C0 Color Select Register Index=14 R/W

Bit Description

1,0 S_color 4,5 - These two bits can be used in place of bits P4 and P5 from the Attribute Palette registers to form the 8-bit digital color value sent off-chip. This feature is used to rapidly switch between sets of colors in the video DAC.

3,2 S_color 6,7 - These two bits are the two high-order bits of the 8-bit digital color value sent off-chip in all modes but the 256 color graphics. In the 256-color graphics, the 8-bit attribute stored in video memory becomes the 8-bit digital color value set off-chip to the video DAC. These bits are also used to rapidly switch between sets of colors in the video DAC.

7-4 Reserved

8.7 DAC Registers

The following registers are external registers in the video DAC, the VGA controller only generates DACRDn/ DACWRn and routes the data to the video DAC.

3C6	**PEL Mask**	**R/W**
3C7	**DAC State Register**	**RO**
3C7	**PEL Address (Read Mode)**	**WO**
3C8	**PEL Address (Write Mode)**	**R/W**
3C9	**PEL Data Register**	**R/W**

CHAPTER 9: ELECTRICAL DATA

9.1 Maximum Ratings

Ambient Operating Temperature:	0° C to + 70° C
Storage Temperature:	65° C to + 150° C
Supply Voltage to Ground Potential:	-0.5 V to + 7.0 V
Applied Input Voltage:	-0.5 V to + 7.0 V

Stresses above those listed may cause permanent damage to the device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in this data sheet is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.2 DC Specifications

TA = 0° C to 70° C, VDD = 5V +/- 5%, VSS = 0V

Symbol	Parameter	Min	Max	Unit	Condition	Notes
Voh	Output High Voltage	2.4		V	Ioh = 400 uA	
Vol	Output Low Voltage		0.4	V	Iol = 24 mA	1,2
Vol	Output Low Voltage		0.4	V	Iol = 8 mA	3
Vol	Output Low Voltage		0.4	V	Iol = 4 mA	4
Vol	Output Low Voltage		0.4	V	Iol = 2 mA	5
Vih	Input High Voltage	2	VCC+0.5	V	TTL	6
Vil	Input Low Voltage	-0.5	0.8	V	TTL	6
Vis	Schmitt Input Voltage	2.4	VCC+0.5	V	Schmitt	6
Vic	CMOS Input Voltage	3.8	VCC+0.5	V	CMOS	6
ILI	Input Leakage Current	-10	10	uA		
OLI	Output Leakage Current	-10	10	uA		
ICC	Operating Supply Current typical normal operation typical power down	 180 80		 mA mA		
CI	Input Capacitance		8	pF		
CO	Output Capacitance		8	pF		
CIO	I/O Capacitance		8	pF		

Notes: Other than ISA bus interface and Monitor interface, there is no DC requirements on the outputs. The other interfaces only have AC requirements.

1. Output Current (Iol) Capabilities:
   24mA: SD[15:0],CINTn, M16, ZEROWSn, IO16n, IOCHRDY, (all the ISA output pins).
2. Open Drain Outputs:
   24mA: IOCHRDY, CINTn, IO16n, M16n, ZEROWSn (ISA)
3. 8mA: HSYNC, VSYNC, RASLn, CASLn/WELn, MA[7:0], CASHIn/WEHIn/MA[8], RASHn/MA[9], PCLK, P[15:0], PAR, REQn, SD[31:16], LBSELn, C/BE[3:0],FRAMEn, IRDYn, TRDYn, DEVSELn, STOPn, LOCKn
4. 4mA: P[23:16], SRCK, SRD, WEAn/CASAn, WEBn/CASBn, WECn/CASCn, WEDn/CASDn, WEEn/CASEn, WEFn/CASFn, WEGn/CASGn, WEHn/CASHn,BLANKn
5. 2mA: CSEL[3]/EEPCS, CSEL[2]/EEPSK, CSEL[1]/EEPWD, CSEL[0]/EEPRD, ROMENLn, VDVALID, MDMXn, ARDn, AWRn, BD[7:0], DACWRn, DACRDn, ARS[3:0], MD[63:0]
6. Input Structures:
   TTL: All inputs
   TTL w/pull-ups: ESYNC, EPDATA, EPCLK

This page is intentionally left blank

9.3 AC Specifications

All AC Timing Diagrams are strictly to show relative timing from one signal to the next. These diagrams are not necessarily logically correct or complete. For better understanding of logical cycles for ISA, VL and PCI bus, please refer to the respective specifications.

The AC values in here are preliminary **design** values.

9.3.1 ISA Bus Timing

ISA Memory Read/Write Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
	tBCLK	*ISA Bus Clock Period*	*80*			1
ISA1	tALE	ALE Active to Inactive	40			
ISA2	tLAS	LA[23:17] Setup to Falling Edge of ALE	15			
ISA3	tLAH	LA[23:17] Hold from Falling Edge of ALE	10			
ISA4	tM16	M16n Active from Valid LA[23:17]		40	200	
ISA5	tASMC	SA[16:0] & BHEn Setup to Memory Command Active	20			
ISA6	tAHMC	SA[16:0] & BHEn Hold to Memory Command Inactive	20			
ISA7	tMCP	Memory Command Pulse Width	80			
ISA8	t0WS	ZEROWSn Delay from Command		10	200	
ISA9	tMRDY	IOCHRDY Inactive from Memory Command Active		30	200	
ISA10	tDVMR	Read Data Valid from MRDn Active		160	200	2
ISA11	tDVRDY	Read Data Valid from IOCHRDY Active		50	200	
ISA12	tDHMR	Read Data Hold from MRDn Inactive	0		200	
ISA13	tDSMW	Write Data Setup to MRWn Active	-45			
ISA14	tDHMW	Write Data Hold from MWRn Inactive	15			

Notes:

1. This parameter is for reference only. It is intended to be the recommanded maximum bus speed. The design of this controller, however, is asynchronous to the BCLK, and therefore theoretically can run at any BCLK, as long as setup, hold and propagation delay timings meet all the above specifications.
2. tDVMR is valid for memory mapped I/O only, which is standard memory read cycle (1 wait state). Normal read access to graphics memory requires wait states, and only tDVRDY is of interest.

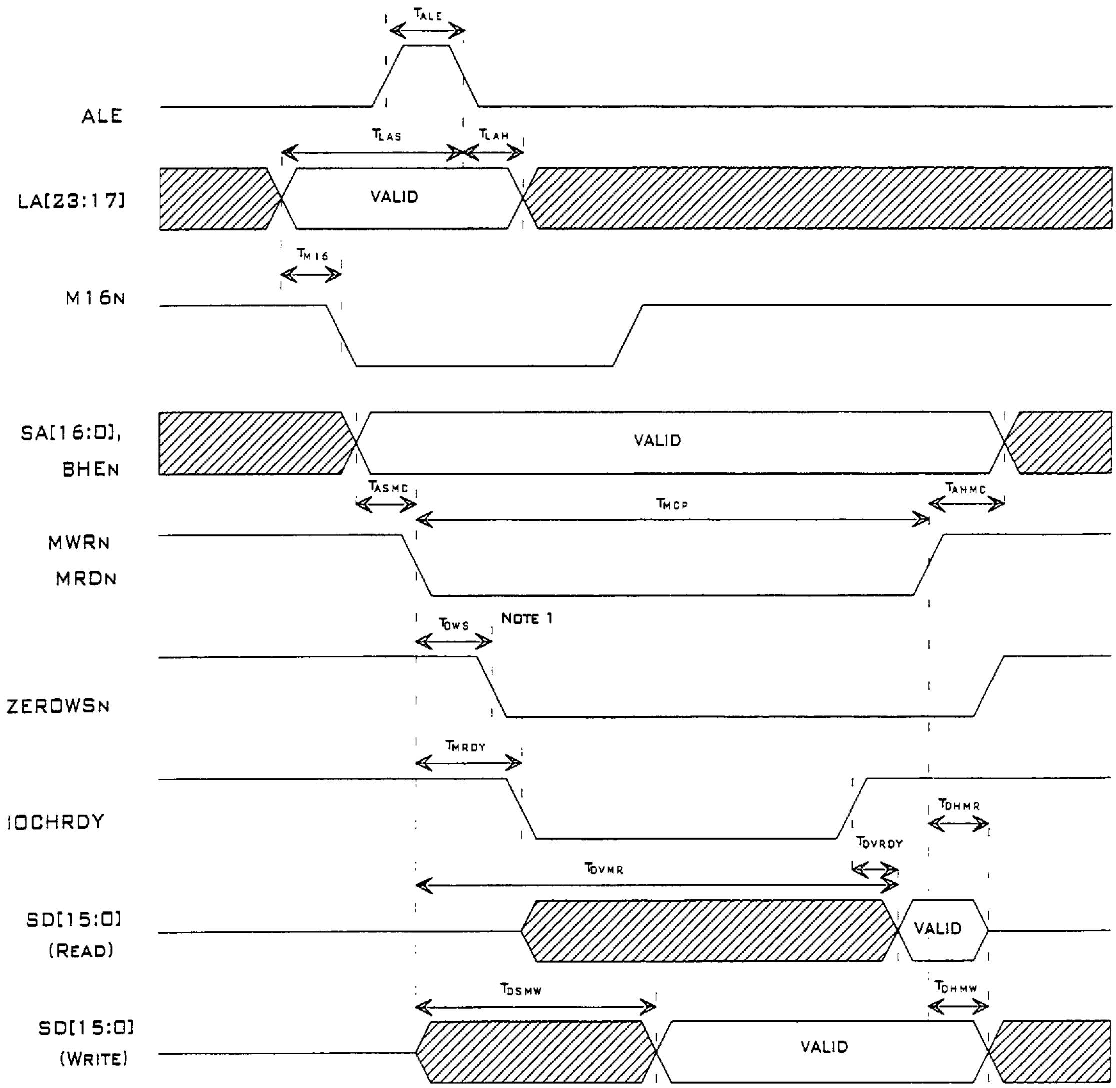


Figure 9.1 - ISA Memory Read/Write Timing

Notes: ZEROWSn and IOCHRDY are mutually exclusive, only one signal can be active at a time. The drawing merely indicates the AC timing of the signals, but not the actual logical cycle timing.

ISA ROM Read Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
ISA1	tALE	ALE Active to Inactive	40			
ISA2	tLAS	LA[23:17] Setup to Falling Edge of ALE	15			
ISA3	tLAH	LA[23:17] Hold from Falling Edge of ALE	10			
ISA4	tM16	M16n Active from Valid LA[23:17]		40	200	1
ISA5	tASMC	SA[16:0] & BHEn Setup to Memory Command Active	23			4
ISA6	tAHMC	SA[16:0] & BHEn Hold from Memory Command Inactive	20			
ISA7	tMCP	Memory Read Command Pulse Width	80			3
ISA15	tRE	ROMENLn Active from MRDn Active		40	50	1,4
	tROMA	*ROM Data Valid from SA[14:0]*		*148/448*	50	2,3,4
	tROME	*ROM Data Valid from MRDn*		*125/425*	50	2,4
ISA16	tBDSD	BD[7:0] Valid to SD[15:0] Valid		35	200	4
	t244P	*ROM Data Valid to SD[15:8] Valid through 244 Buffer*		*35*	200	1,4,5
ISA10	tDVMR	Read Data Valid from MRDn Active		160/460	200	3,4,6
ISA12	tDHMR	Read Data Hold from MRDn Inactive	0		200	

Notes:

1. These parameters are relevant only for 16-bit ROM (2 parts). For 8-bit ROM, M16n is not generated, ROMENLn is a "don't care", and no additional buffer is required.
2. This specification is for reference only. This is the required ROM access timing. Depending on tASMC, tROMA or tROME can be used to select the correct ROM speed. In general, 120ns ROM should be used for 8.33MHz. Faster buses will require faster ROMs. See note 4 for more details.
3. A/B where A is for 16-bit ROM, and B is for 8-bit ROM.
4. tDVMR is the specification that must be met. The calculation for tDVMR is not straight forward. If tROMA-tROME>tASMC, then tDVMR=tROMA-tASMC+tBDSD for 8-bit ROM, or for SD[7:0] of 16-bit ROM, and, tDVMR=tROMA-tASMC+t244 for SD[15:8] of 16-bit ROM. If tASMC>tROMA-tROME, then tDVMR=tROME+tBDSD for 8-bit ROM, or for SD[7:0] of 16-bit ROM, and, tDVMR=tROME+t244 for SD[15:8] of 16-bit ROM.
5. This specification is for reference only. The general rule of thumb is that the speed of the 244 buffer should not be more than tBDSD. See note 4 for more details.
6. tDVMR as specified here is based on 8.33MHz bus, this number is reduced for faster buses, but there is no standard for it.

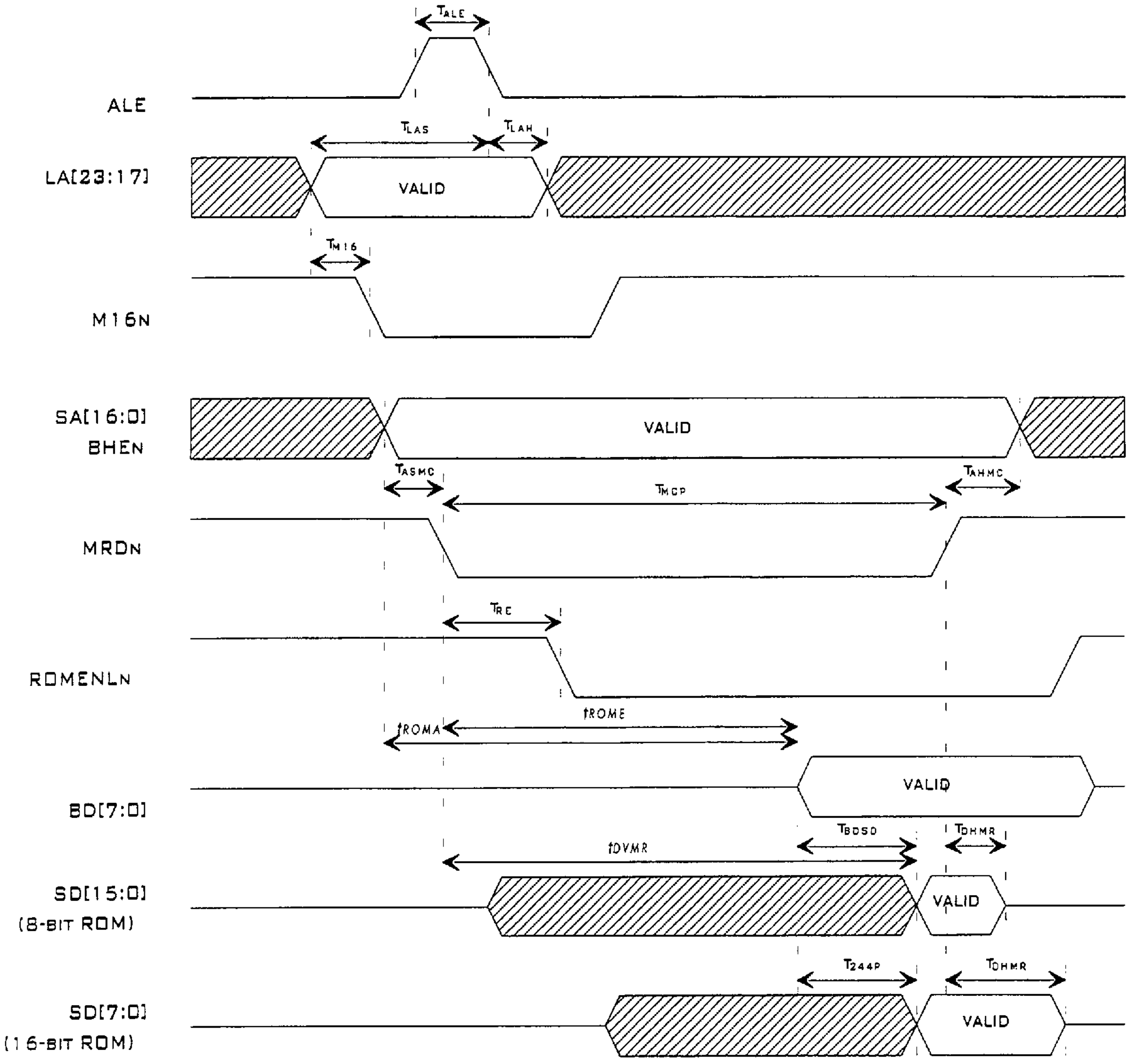


Figure 9.2 - ISA ROM Read Timing

ISA General I/O Access Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
ISA17	tASIO	SA[15:0] & BHEn Setup to I/O Command Active	25			
ISA18	tAHIO	SA[15:0] & BHEn Hold from I/O Command Inactive	25			
ISA19	tIOCP	I/O Command Pulse Width	115			
ISA20	tIO16	IO16n Active from Valid SA[15:0]		60	200	
ISA21	tDVIR	Read Data Valid from IORn Active		70	200	
ISA22	tDVRDY	Read Data Valid from IOCHRDY Active		50	200	
ISA23	tDHIR	Read Data Hold from IORn Inactive	0		200	
ISA24	tDIVW	Write Data Valid from IOWn Active	-55			
ISA25	tDHIW	Write Data Hold from IOWn Inactive	15			
ISA26	tIORDY	IOCHRDY Inactive from I/O Command Active		30	200	1

Notes:

1. IOCHRDY is normally not needed for I/O cycles. It is needed for special reset sequence and power saving modes only.

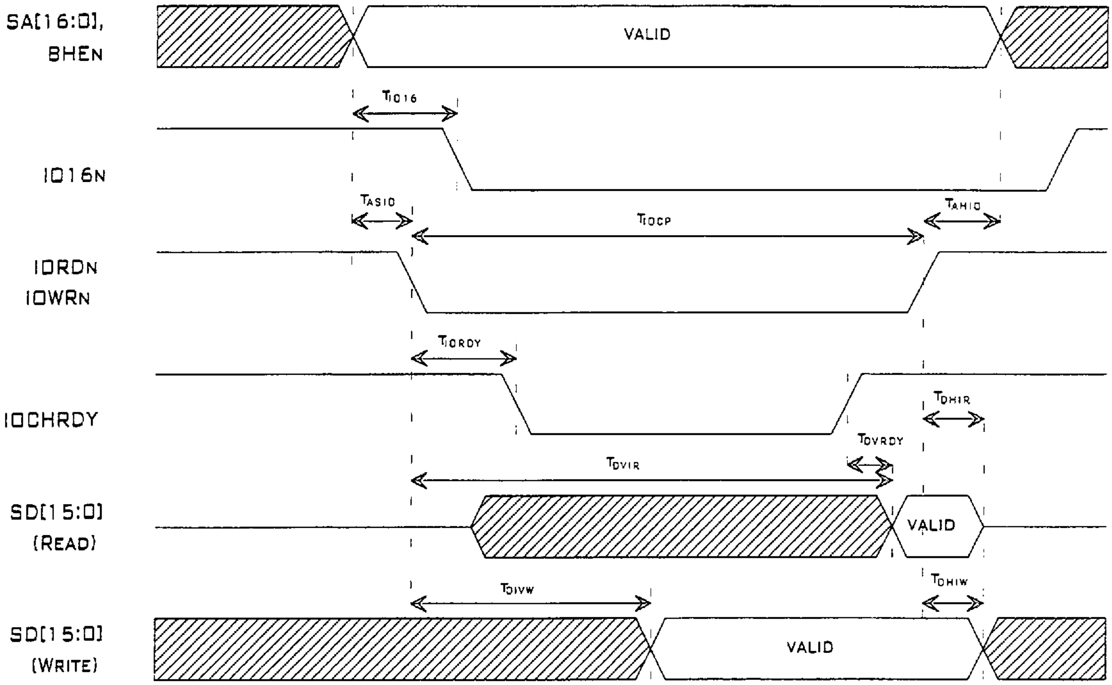


Figure 9.3 - ISA General I/O Access Timing

ISA DAC and Auxiliary I/O Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
ISA17	tASIO	SA[15:0] & BHEn Setup to I/O Command Active	25			
ISA18	tAHIO	SA[15:0] & BHEn Hold from I/O Command Active	25			
ISA19	tIOCP	I/O Command Pulse Width	115			
ISA27	tICDC	DAC Command Delay from I/O Command		25	50	
	tRLQV	*DAC Data Valid from DACRDn*		*395*	*50*	1
	tDVIR8	*SD[15:0] Valid from IORDn for DAC or Auxiliary Cycle*		*460*	*200*	2
ISA23	tDHIR	Read Data Hold from IORn Inactive	0		200	
ISA24	tDVIW	Write Data Valid from IOWn Active	-55			
ISA25	tDHIW	Write Data Hold from IOWn Inactive	15			
ISA28	tDWP	DACWRn Pulse Width	70			
ISA29	tSDBD	SD[7:0] Valid to BD[7:0] Valid		40	50	
ISA30	tBDSD	BD[7:0] Valid to SD[15:0] Valid		40	200	
ISA31	tACSn	Auxiliary Chip Select Delay from SA[15:0]		20	50	

Notes.

1. This is for reference only. This specification belongs to the DAC I/O access time.
2. This is for reference only. This is the specification for 8-bit I/O ISA bus. The real data valid time for the DAC should be calculated as follows: tDVIR=tICDC+tRLQV+tBDSD.

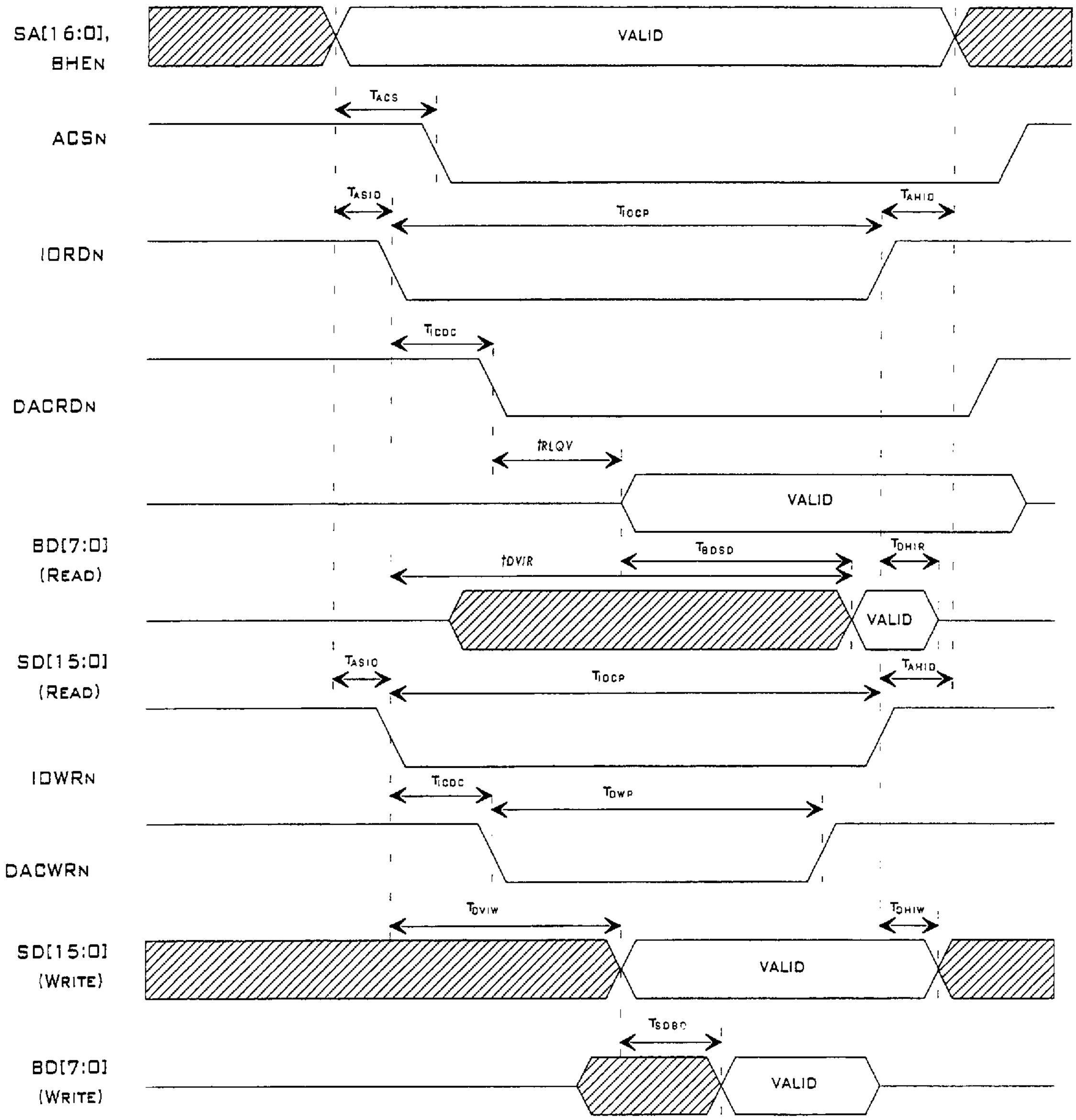


Figure 9.4- ISA DAC & Auxiliary I/O Timing

9.3.2 VL Bus Timing

VL Bus Interface Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
VL1	tPRCKP	Processor Clock Period	20			
VL2	tPRCKW	Processor Clock Pulse Width	8			
VL3	tPRCKR	Processor Clock Rise/Fall		2		
VL4	tSADS	ADS Setup Time	4			
VL5	tHADS	ADS Hold Time	2			
VL6	tSSA	Address Setup Time	4			
VL7	tHSA	Address Hold Time	2			
VL8	tSS	Status Setup Time	4			
VL9	tHS	Status Hold Time	2			
VL10	tLBSEL	LBSELn Valid from SA[31:2] & Status		20	25	
VL11	tSRDY	SRDYn Delay Time		13	75	
VL12	tSSRDYI	SRDYI Setup Time	5			
VL13	tHSRDYI	SRDYI Hold Time	2			
VL14	tVRD	Read Data Delay Time	3	14	125	
VL15	tHRD	Read Data Hold Time		2	125	
VL16	tSWD	Write Data Setup Time	4			
VL17	tHWD	Write Data Hold Time	2			

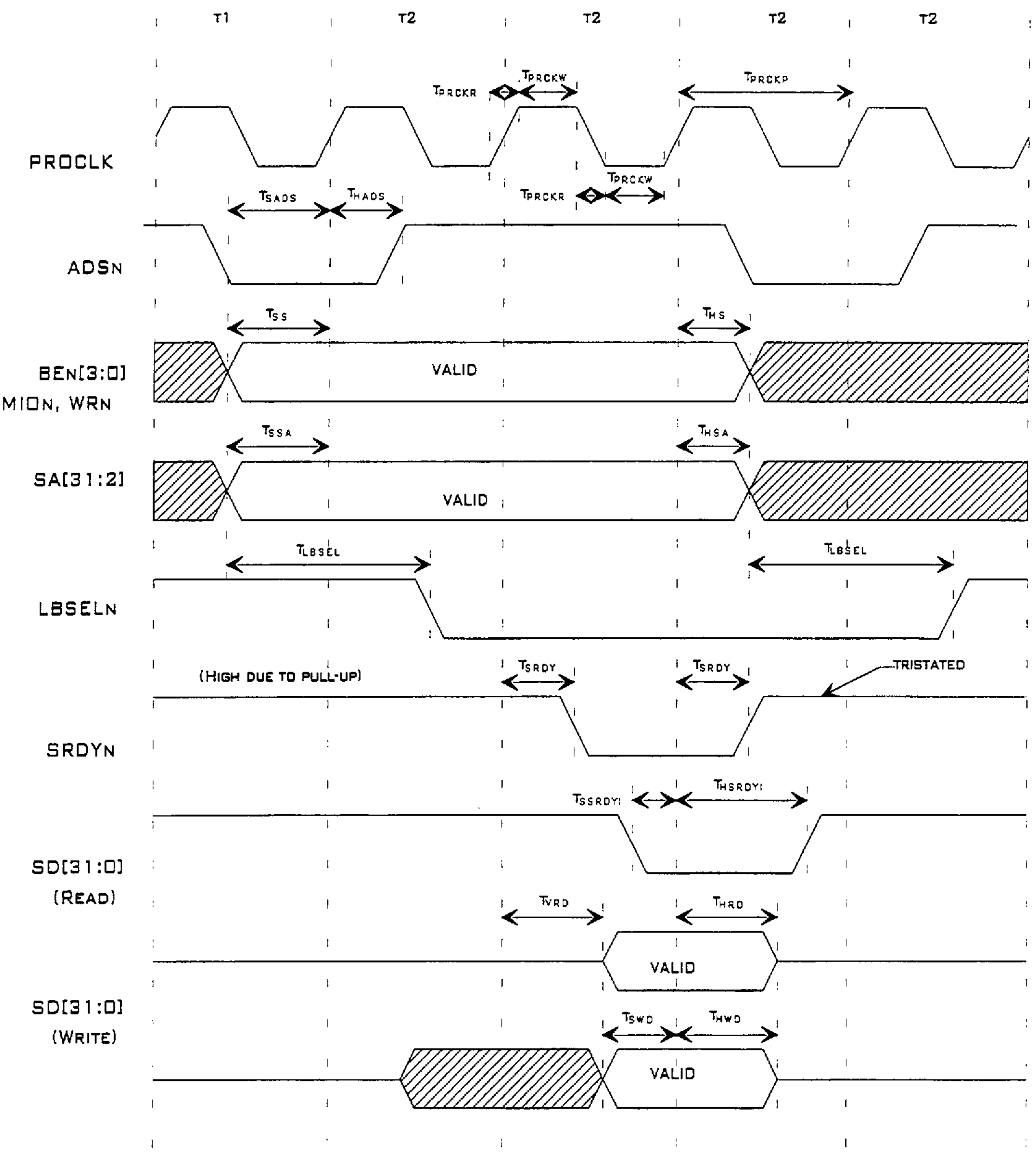


Figure 9.5 - VL Bus Interface Timing

VL ROM Read Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
VL4	tSADS	ADS Setup Time	4			
VL5	tHADS	ADS Hold Time	2			
VL6	tSSA	Address Setup Time	4			
VL7	tHSA	Address Hold Time	2			
VL8	tSS	Status Setup Time	4			
VL9	tHS	Status Hold Time	2			
VL18	tDOE	DOEn Delay from ISACMD		25	50	
ISA5	tASMC	SA[14:0] Setup to MRDn Active	23			
ISA6	tAHMC	SA[14:0] Hold from MRDn Inactive	20			
	tDVMR	*Read Data Valid from MRDn Active*		*460*	*200*	1,3
	tADSIC	*ISA Command (MRDn) Active Delay from ADSn*		*?*		1,2
	tROMA	*ROM Data Valid from SA[14:0]*		*448*	*50*	1
	tROME	*ROM Data Valid from MRDn Active*		*425*	*50*	1
	tROMO	*ROM Data Float from MRDn Inactive*	*0*		*50*	1
	t10P	*ISACMD Active from MRDn through LS10*		*35*	*50*	1
	t245P	*ROM Data Valid to SD[7:0] Valid through 245 Buffer*		*35*	*200*	1
	t245E	*DOEn Valid to SD[7:0] Valid through 245 Buffer*		*35*	*200*	1
	t245O	*SD[7:0] Tristated from DOEn Inactive through 245 Buffer*		*35*	*200*	1

Notes:

1. These parameters are for reference only.
2. This parameter is chipset dependent. A reasonable calculation for tADSMC should be as follows: tADSMC=(2*tPRCKP)+(2*tBCLK).
3. This is the required specification for 8-bit Memory on ISA bus. The actual timing can be calculated in two different ways. The worst of the 2 numbers should be used.
    1. tDVMR=tROME+t245P, or
    2. tDVMR=tROMA+t245P-tASMC.

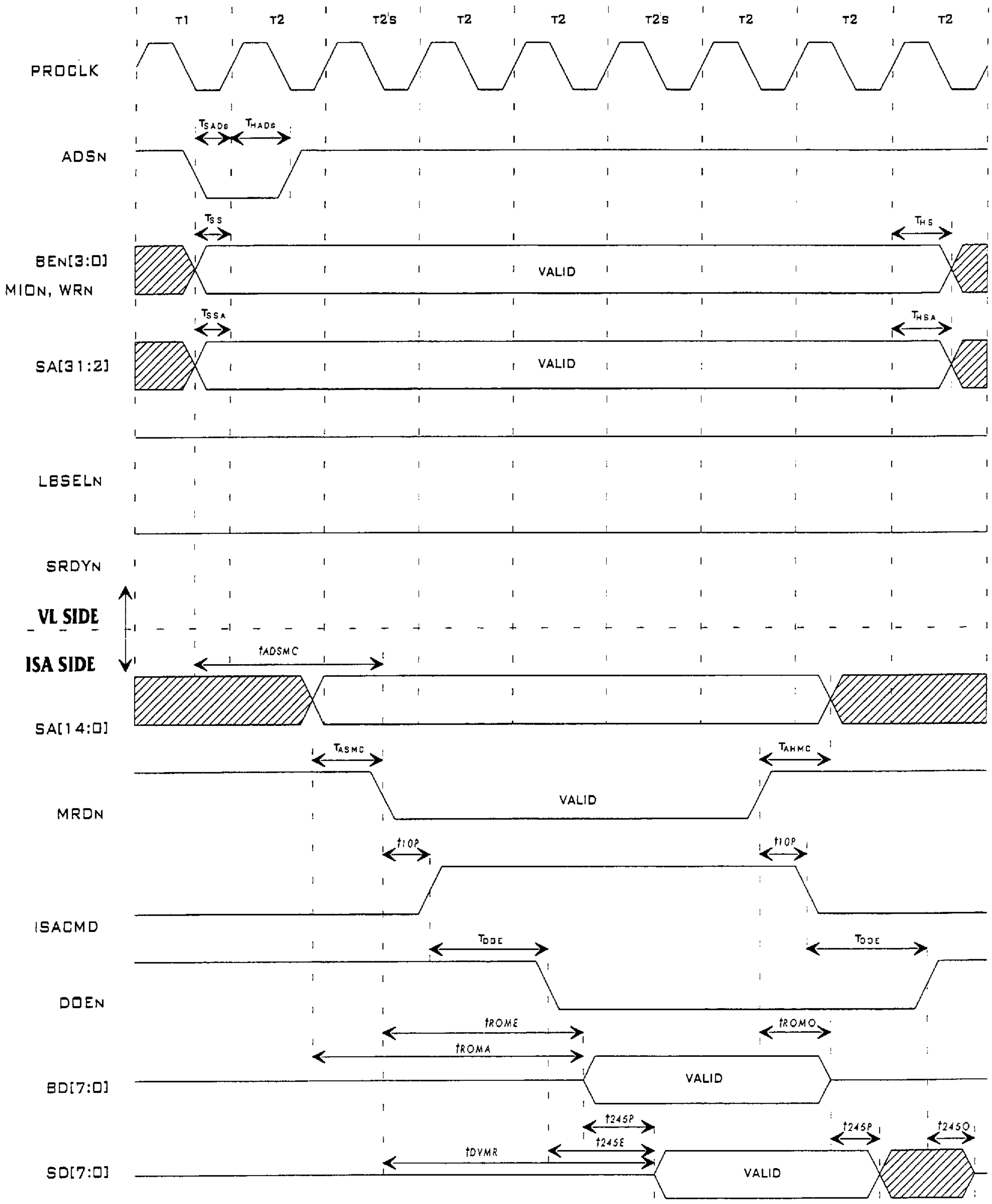


Figure 9.6 - VL ROM Read Timing

VL DAC I/O Read Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
VL4	tSADS	ADS Setup Time	4			
VL5	tHADS	ADS Hold Time	2			
VL6	tSSA	Address Setup Time	4			
VL7	tHSA	Address Hold Time	2			
VL8	tSS	Status Setup Time	4			
VL9	tHS	Status Hold Time	2			
	t10P	ISACMD Active from IORn through LS10		35	50	1
VL18	tDOE	DOEn Delay from ISACMD		25	50	
VL19	tICDC	DAC Command Delay from ISACMD		25	50	4
	tADSIC	*ISA Command (IORn) Active Delay from ADSn*		*?*		1,2
	tRLQV	*DAC Data Valid from DACRDn*		*395*	*50*	1
	tDVIR8	*SD[15:0] Valid from IORDn for DAC Cycle*		*460*	*200*	1,3
	t245P	*DAC Data Valid to SD[7:0] Valid through 245 Buffer*		*40*	*200*	1

Notes:

1. These parameters are for reference only.
2. This parameter is chipset dependent. A reasonable calculation for tADSMC should be as follows: tADSMC=(2*tPRCKP)+(2*tBCLK).
3. This is the required specification for 8-bit Memory on ISA bus. The actual timing can be calculated as follows: tDVIR8=t10P+tICDC+tRLQV+t245P.
4. Although this parameter is similar to the ISA27 parameter, it is essential to test this under VL configuration because the address decoding is from the VL bus.

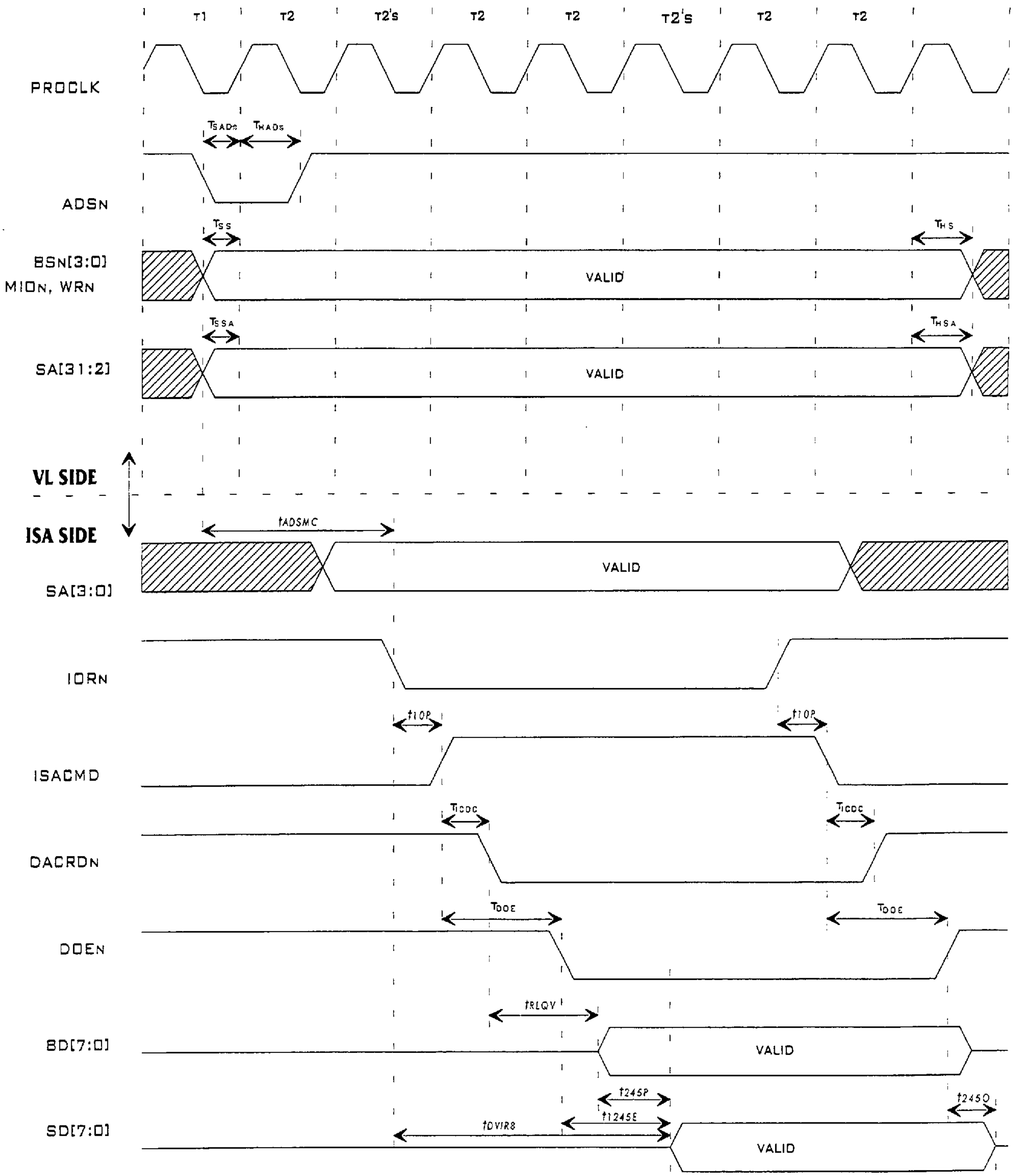


Figure 9.7 - VL DAC I/O Read Timing

VL DAC I/O Write Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
VL4	tSADS	ADS Setup Time	4			
VL5	tHADS	ADS Hold Time	2			
VL6	tSSA	Address Setup Time	4			
VL7	tHSA	Address Hold Time	2			
VL8	tSS	Status Setup Time	4			
VL9	tHS	Status Hold Time	2			
	t10P	*ISACMD Active from IOWn through LS10*		*35*	*50*	1
VL18	tDOE	DOEn Delay from ISACMD		25	50	
VL19	tICDC	DAC Command Delay from ISACMD		25	50	2
ISA24	tDVIW	Write Data Valid from IOWn Active	-55			
ISA28	tDWP	DACWRn Pulse Width	70			
	t245D	*DACWRn Valid to BD[7:0] Valid through 245 Buffer (DIR)*		*35*	*50*	1
	t245E	*DOEn Valid to BD[7:0] Valid through 245 Buffer (OEn)*		*35*	*50*	1
	t245P	*SD[7:0] Valid to BD[7:0] Valid through 245 Buffer (A-B)*		*35*	*50*	1

Notes:

1. These parameters are for reference only.
2. Although this parameter is similar to the ISA27 parameter, it is essential to test this under VL configuration because the address decoding is from the VL bus.

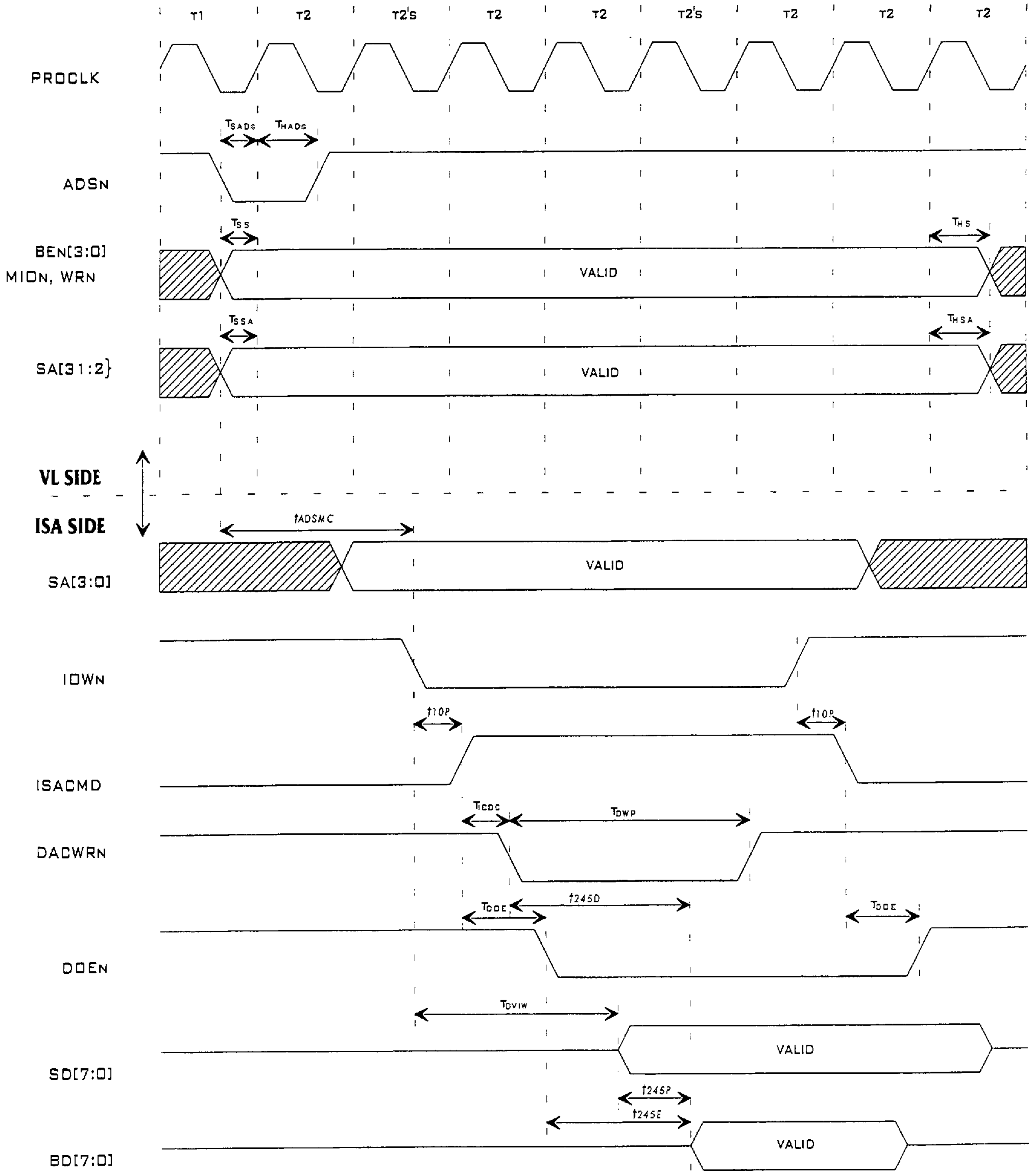


Figure 9.8 - VL DAC I/O Write Timing

9.3.3 PCI Bus Timing

PCI Bus Clock Requirement

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
PCI1	tPCKP	PCI Bus Clock Period	30			
PCI2	tPCKL	PCI Bus Clock Low Time	12			
PCI3	tPCKH	PCI Bus Clock High Time	12			

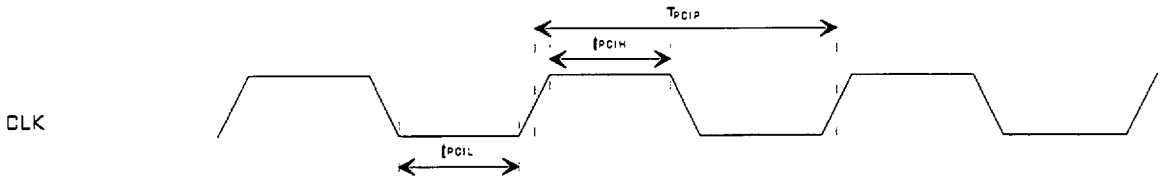


Figure 9.9 - PCI Bus Clock Requirement

PCI Bus Timing (Read Operation)

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
PCI4	tSPCI	Input Setup Time to CLK	7			
PCI5	tHPCI	Input Hold Time to CLK	0			
PCI6	tPPCI	Ouput Propagation Delay from CLK	2	11	0/50	1

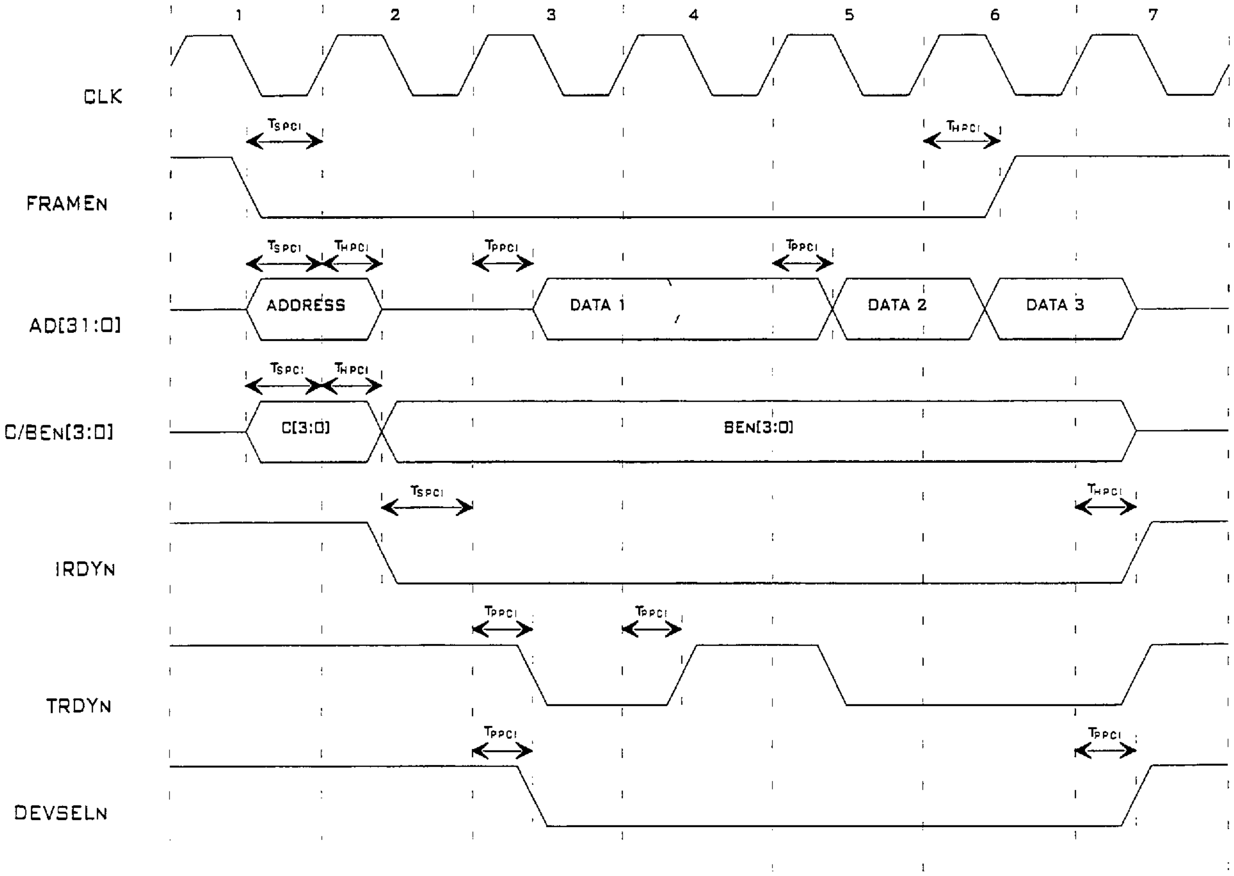


Figure 9.10 - PCI Bus Timing (Read Operation)

Notes:

1. Minimum timing is when load=0pF, and maximum timing is when load=50pF.

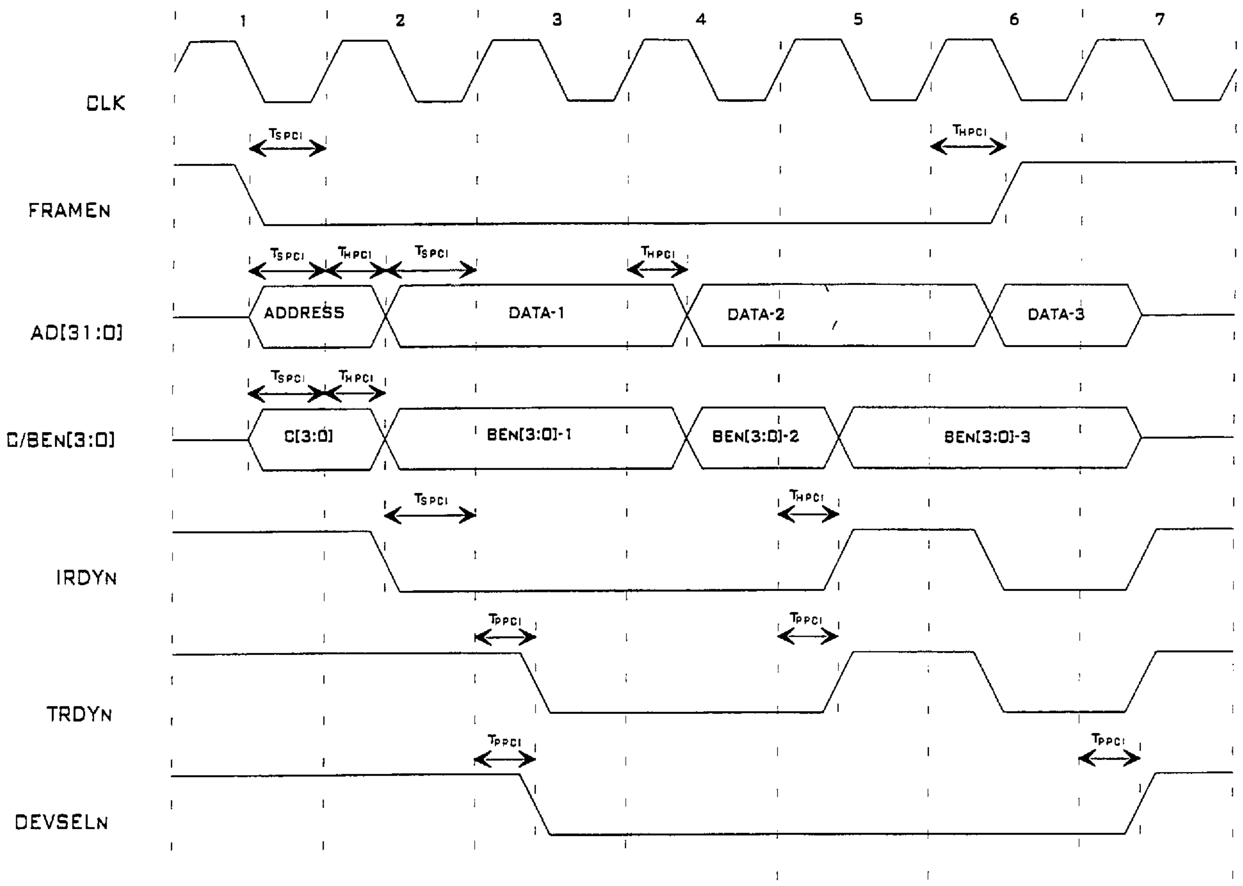


Figure 9.11 - PCI Bus Timing (Write Operation)

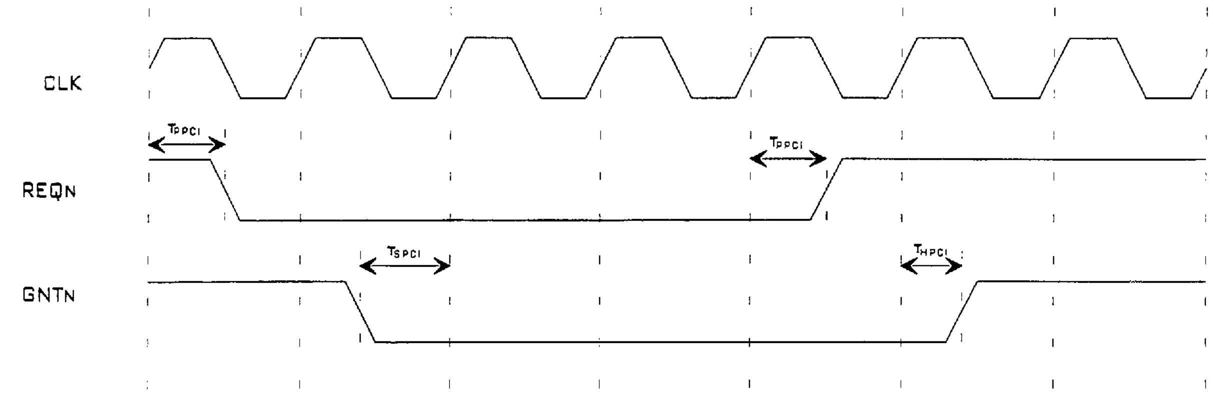


Figure 9.12 - PCI Bus Master Request Timing

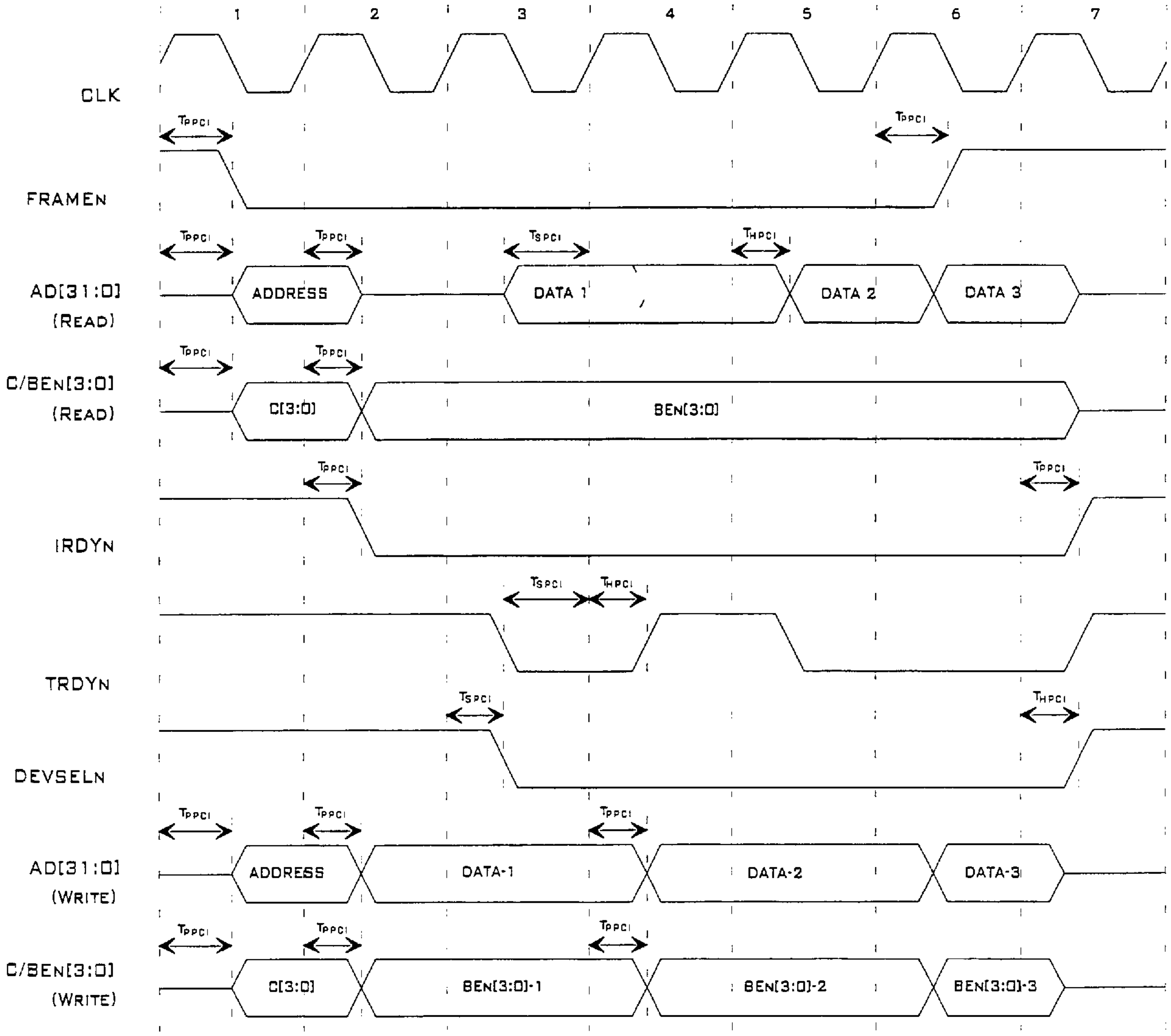


Figure 9.13 - PCI Bus Master Read/Write Timing

PCI ROM Read Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
PCI4	tSPCI	Input Setup Time to CLK	7			
PCI5	tHPCI	Input Hold Time to CLK	0			
PCI6	tPPCI	Ouput Propagation Delay from CLK	2	11	0/50	1
PCI7	tROMEN	ROMENLn Delay from CLK		25	50	
	tROMA	*ROM Data Valid from RA[14:0] (AD[30:16])*		*120*	*50*	2
	tROME	*ROM Data Valid from ROMENLn*		*60*	*50*	2
	tROMO	*ROM Data Float from ROMENLn*		*55*	*50*	2

Notes:

1. Minimum timing is when load=0pF, and maximum timing is when load=50pF.
These parameters are for reference only. They can be used to choose the appropriate EPROM speed. These requirements are based on the bus clock being 33MHz.

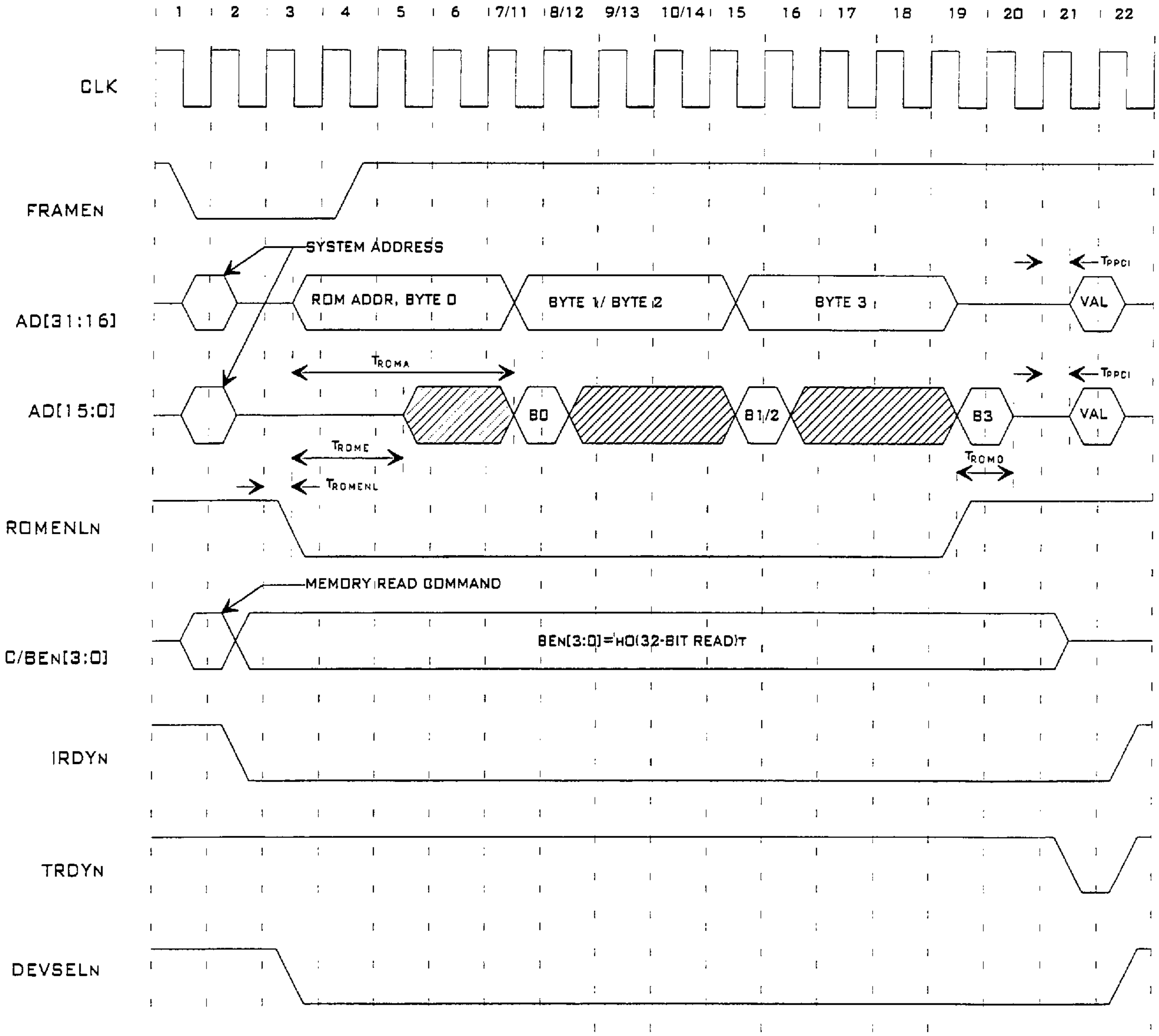


Figure 9.14 - PCI ROM Read Timing

PCI DAC & Auxiliary I/O Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
PCI4	tSPCI	Input Setup Time to CLK	7			
PCI5	tHPCI	Input Hold Time to CLK	0			
PCI6	tPPCI	Ouput Propagation Delay from CLK	2	11	0/50	1
PCI8	tARS	ARS[3:0] Delay from CLK		25	50	
PCI9	tADC	Auxiliary or DAC Command Delay from CLK		25	50	
PCI10	tBDAD	BD[7:0] Valid to AD[31:0] Valid		40	50	
PCI11	tADBD	AD[31:0] Valid to BD[7:0] Valid		40	50	
PCI12	tHADC	BD[7:0] Valid Hold from DACWRn/AWRn Inactive	tPCIP		50	
	tRLQV	*BD[7:0] Valid from ARDn or DACRDn*		*65*	*50*	2

Notes:

1. Minimum timing is when load=0pF, and maximum timing is when load=50pF.
2. This parameter is for reference only. They can be used to choose the appropriate DAC or Auxiliary I/O speed. These requirements are based on the bus clock being 33MHz.

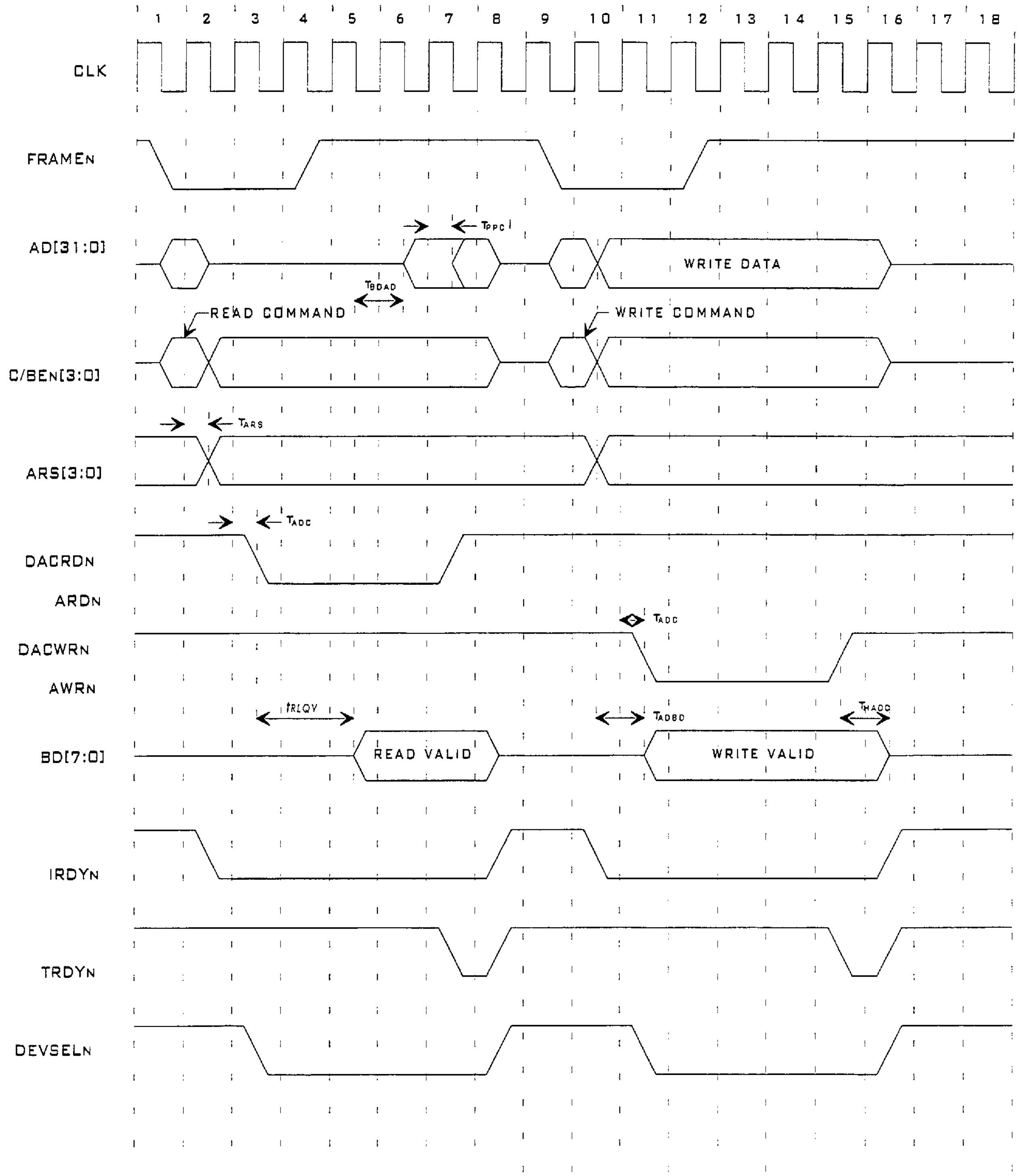


Figure 9.15 - PCI DAC & Auxiliary I/O Timing

9.3.4 - DAC Interface Timing

DAC Pixel Port Interface Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
DAC1	tVCKP	Video Clock Period	9			
DAC2	tVCKL	Video Clock Low Time	3			
DAC3	tVCKH	Video Clock High Time	3			
DAC4	tCHCH	Pixel Clock Period	9		50	1
DAC5	tCLCH	Pixel Clock Low Time	3		50	
DAC6	tCHCL	Pixel Clock High Time	3		50	
DAC7	tPVCH	P[23:0] Setup to PCLK	2		50	1
DAC8	tCHPX	P[23:0] Hold to PCLK	2		50	
DAC9	tBVCH	BLANKn Setup to PCLK	2		50	1
DAC10	tCHBX	BLANKn Hold to PCLK	2		50	

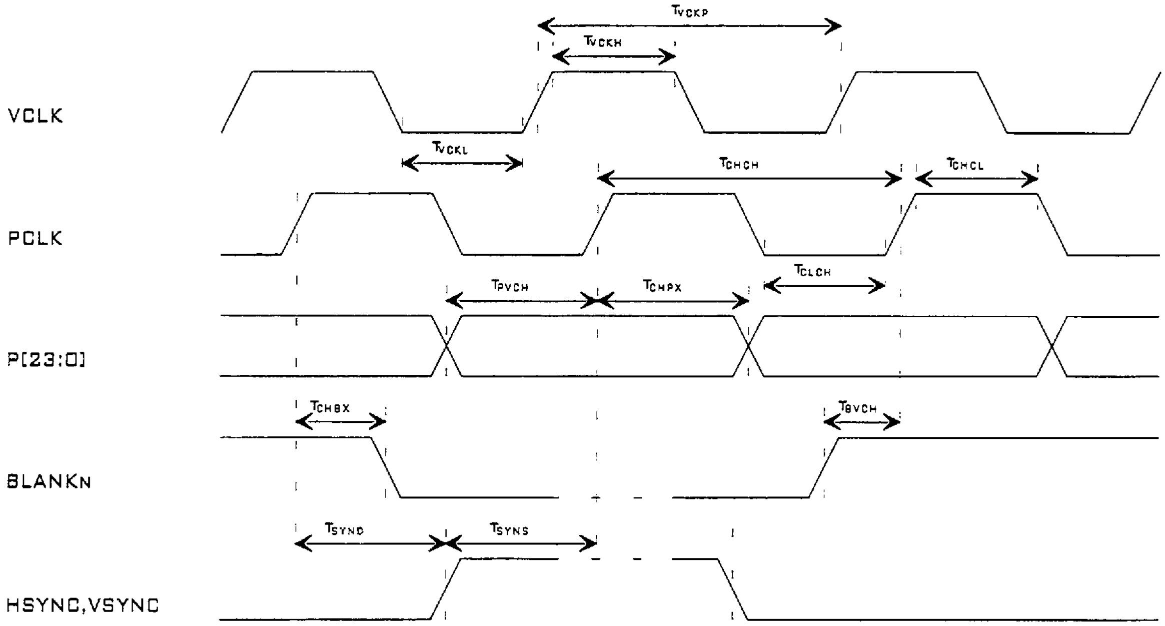


Figure 9.16 - DAC Pixel Port Interface Timing

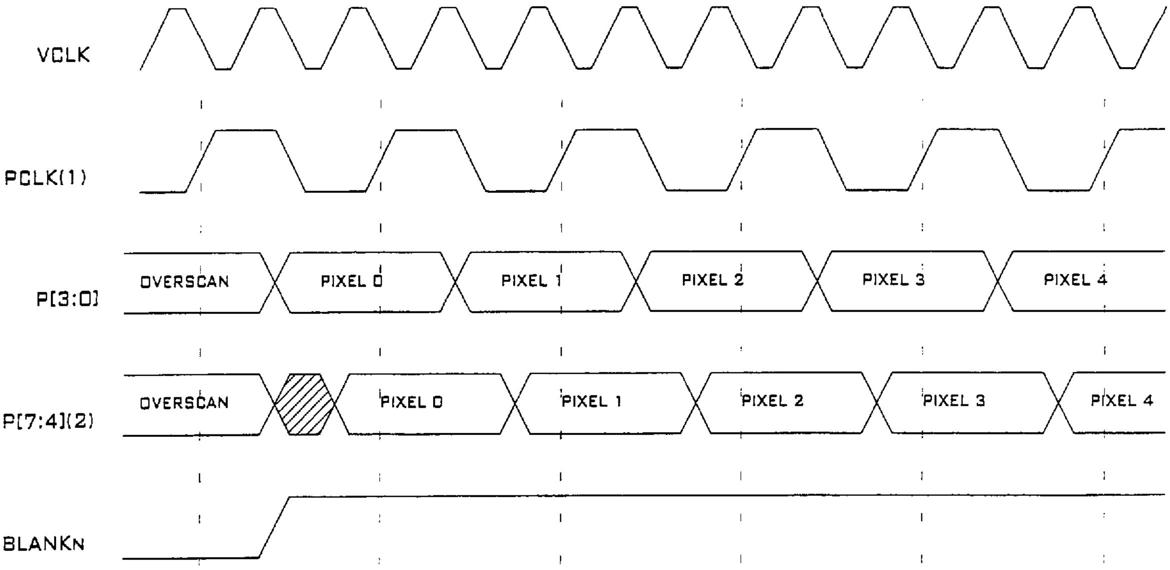


Figure 9.17 - DAC Interface - Attribute Mode 0, VGAMode 13 (8bpp, 8-bit port)

Notes:

1. PCLK frequency is half that of VCLK frequency.
2. Nibble P[7:4] is one VCLK delayed from nibble P[3:0] for the same pixel.

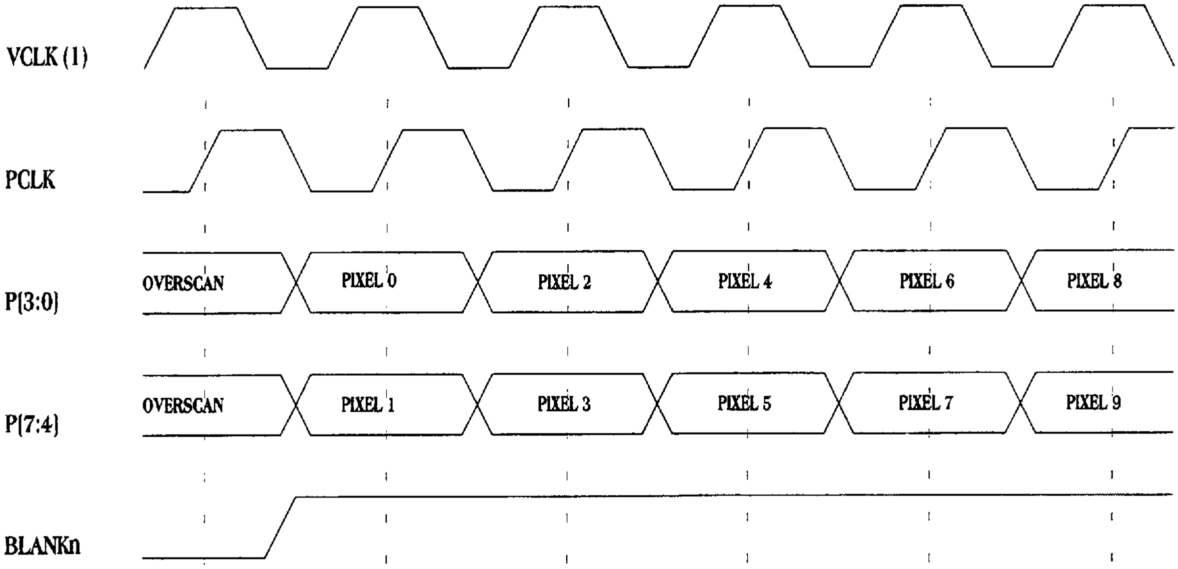


Figure 9.18 - DAC Interface - Attribute Mode 1 (4bpp, 8-bit port)

Notes:

1. The VCLK frequency is only half of the normally required frequency. For example, normal frequency for 1280x1024 70Hz is 135MHz, but it only needs to be 67.5MHz for this mode.

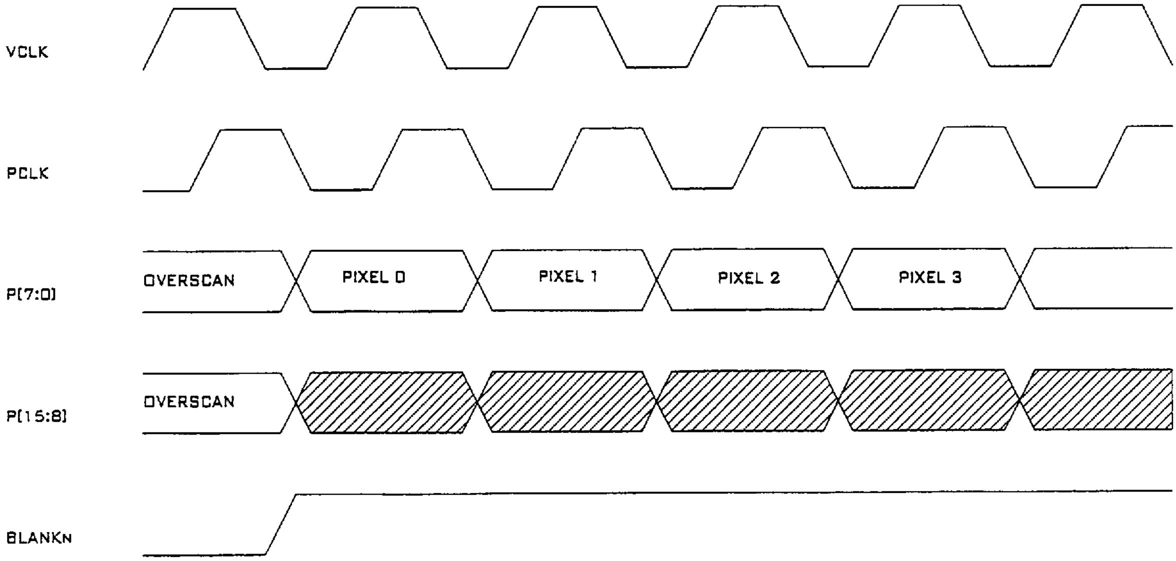


Figure 9.19 - DAC Interface - Attribute Mode 2A (8bpp, 8-bit port)

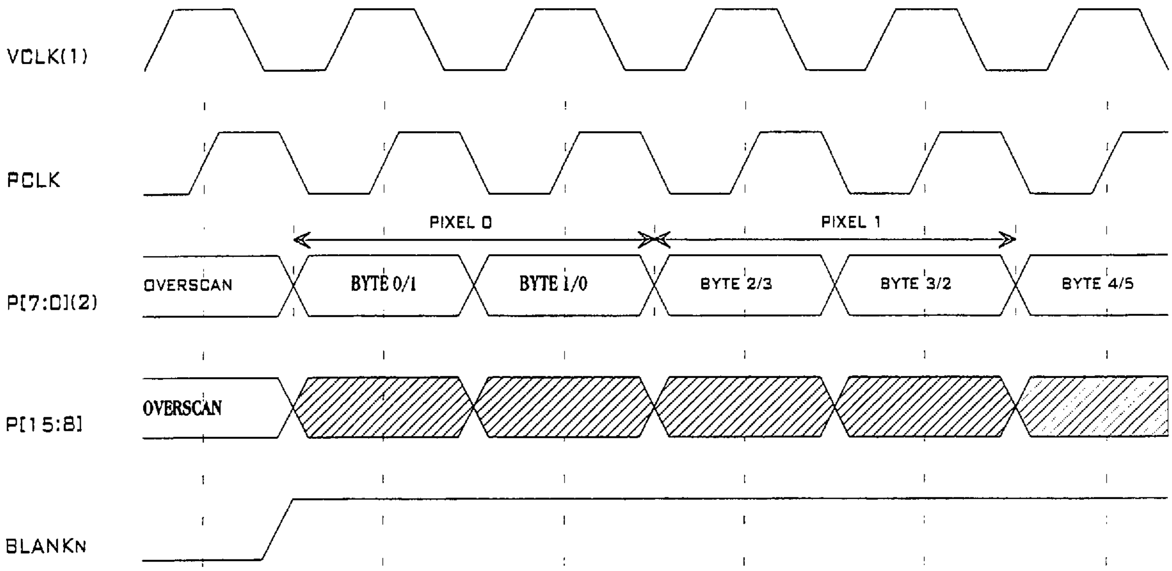


Figure 9.20 - DAC Interface - Attribute Mode 2B (15/16bpp, 8-bit port)

Notes:

1. VLCK frequency is twice that of normally required frequency. For example, the normal frequency for 640x480 60 Hz is 25MHz, but it has to be 50MHz for this mode.
2. The pixel data ordering can be byte 0, byte 1, byte 2, byte 3, ... or byte 1, byte 0, byte 3, byte 2, ..., depending on ER38b4.

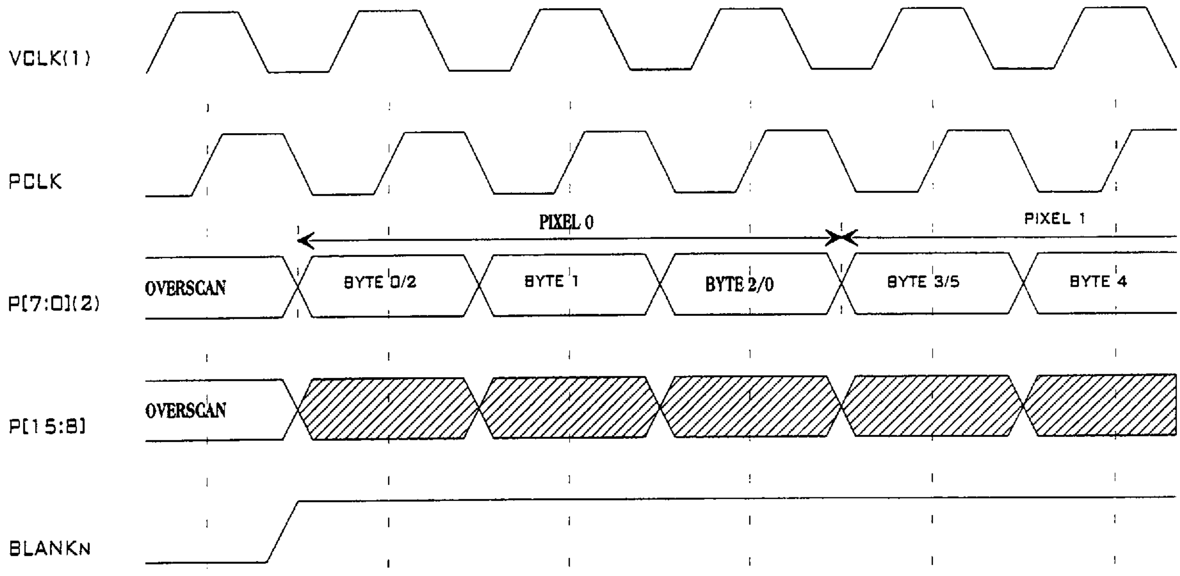


Figure 9.21 - DAC Interface - Attribute Mode 2C (24bpp, 8-bit port)

Notes:

1. VLCK frequency is three times that of normally required frequency. For example, the normal frequency for 640x480 60 Hz is 25MHz, but it has to be 75MHz for this mode.
2. The pixel data ordering can be byte 0, byte 1, byte 2, byte 3, byte 4, byte 5 ... or byte 2, byte 1, byte 0, byte 5, byte 4, byte 3, ..., depending on ER38b4.

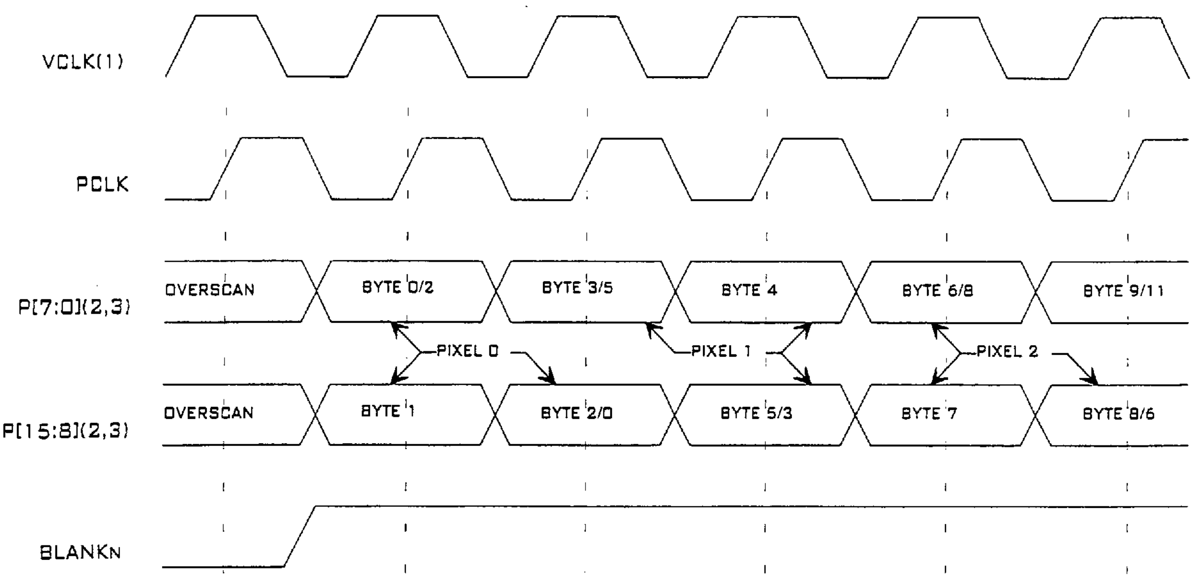


Figure 9.22 - DAC Interface - Attribute Mode 3 (24bpp, 16-bit port)

Notes:

1. VLCK frequency is 3/2 times that of normally required frequency. For example, the normal frequency for 640x480 60 Hz is 25MHz, but it has to be 37.5MHz for this mode.
2. The pixel data ordering can be byte 0, byte 1, byte 2, byte 3, byte 4, byte 5 ... or byte 2, byte 1, byte 0, byte 5, byte 4, byte 3, ..., depending on ER38b4.
3. The third byte of a pixel is always on the high byte of P bus (P[15:8]).

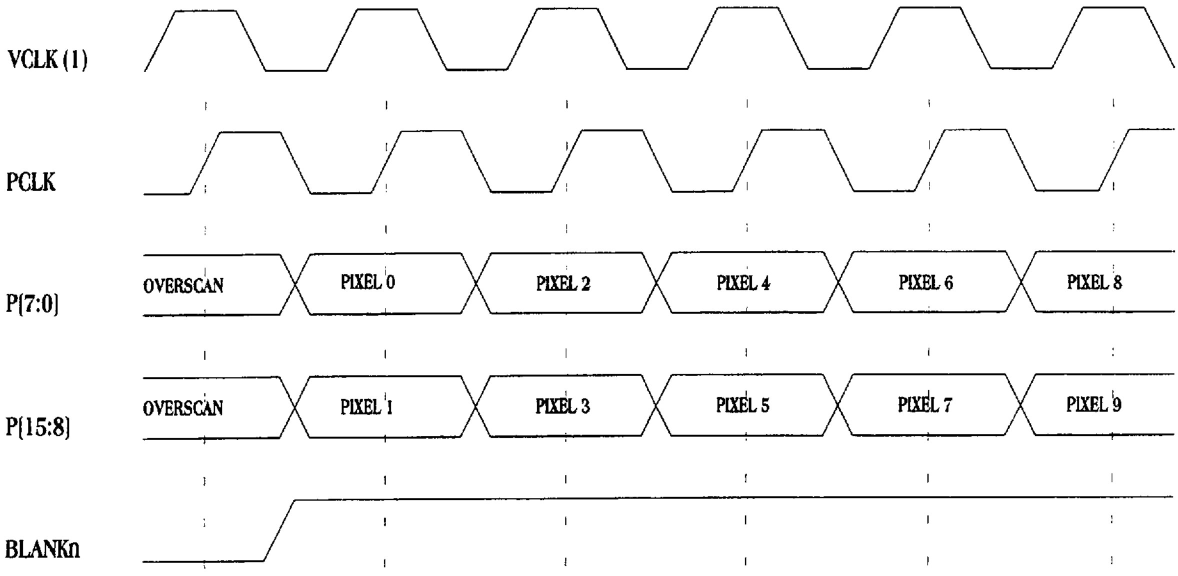


Figure 9.23 - DAC Interface - Attribute Mode 4 (8bpp, 16-bit port)

Notes:

1. The VCLK frequency is only half of the normally required frequency. For example, normal frequency for 1280x1024 70Hz is 135MHz, but it only has to be 67.5MHz for this mode.

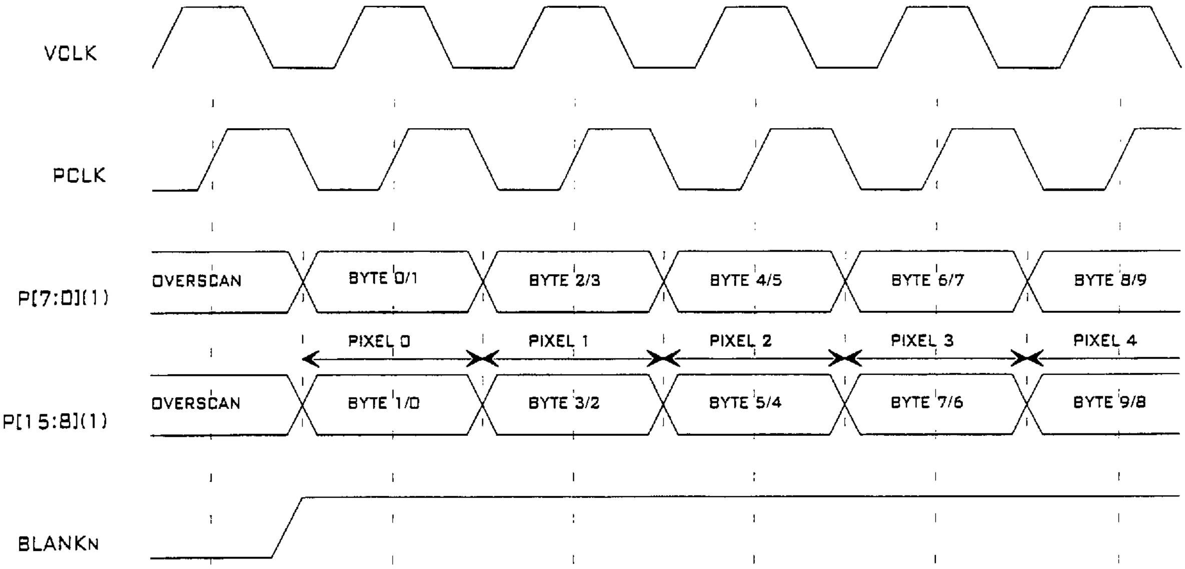


Figure 9.24 - DAC Interface - Attribute Mode 5 (15/16bpp, 16-bit port)

Notes:

1. The pixel data ordering can be byte 0, byte 1, byte 2, byte 3, ... or byte 1, byte 0, byte 3, byte 2, ..., depending on ER38b4.

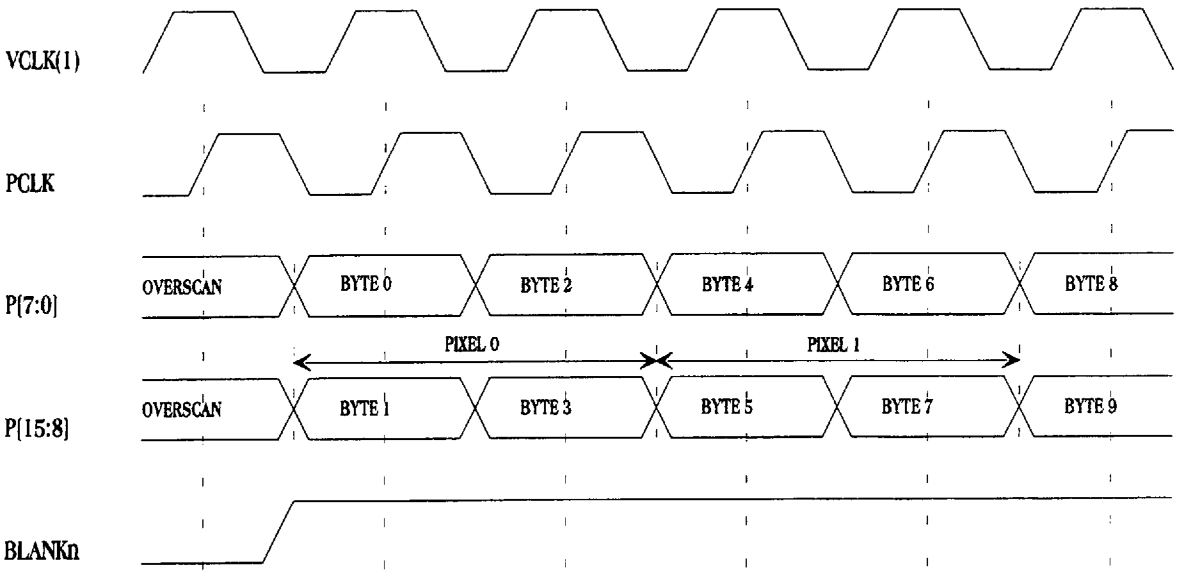


Figure 9.25 - DAC Interface - Attribute Mode 6 (32bpp, 16-bit port)

Notes:

1. VLCK frequency is twice that of normally required frequency. For example, the normal frequency for 640x480 60 Hz is 25MHz, but it has to be 50MHz for this mode.

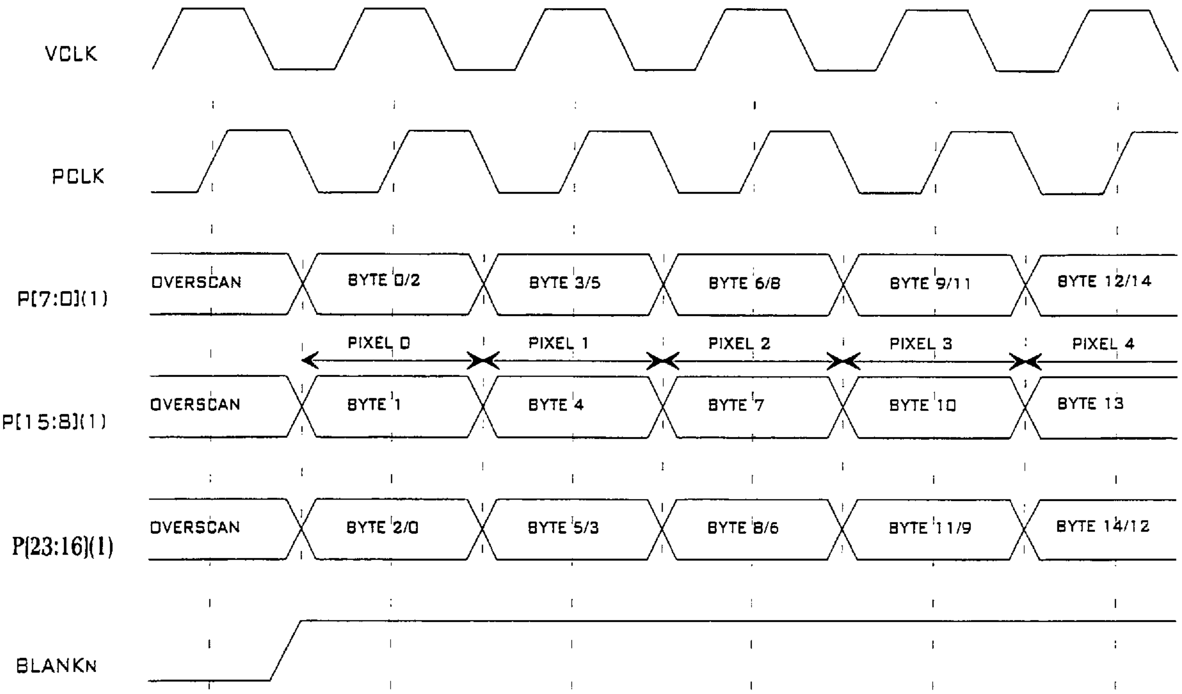


Figure 9.26 - DAC Interface - Attribute Mode 7 (24bpp, 24-bit port)

Notes:

1. The pixel data ordering can be byte 0, byte 1, byte 2, byte 3, ... or byte 2, byte 1, byte 0, byte 5, ..., depending on ER38b4.

9.3.5 - Memory Interface Timimg

Memory Clock Requirement

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pF)	Notes
M1	tMP	Memory Clock Period	15	25		1
M2	tMCKL	Memory Clock Low Time	6			
M3	tMCKH	Memory Clock High Time	6			

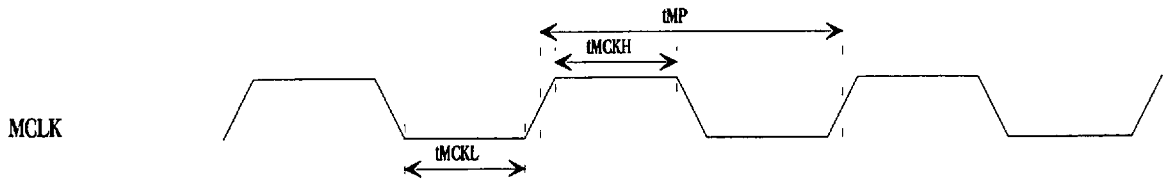


Figure 9.27 - Memory Clock Requirement

Programmable Parameters

No	Symbol	Parameter	Min (tMP)	Max (tMP)	Register	Notes
M4	tRP	RASxn Precharge	1.5	5	ER26[3,1:0]	
M9	tCP	CASxn Precharge	0.5	2	ER27[1:0]	
M8	tCAS	CASxn Pulse Width	1	2.5	ER27[2:1]	
	tRCPD	RASxn Low to CASxn Precharge Delay	1	4	ER27[5:4]	

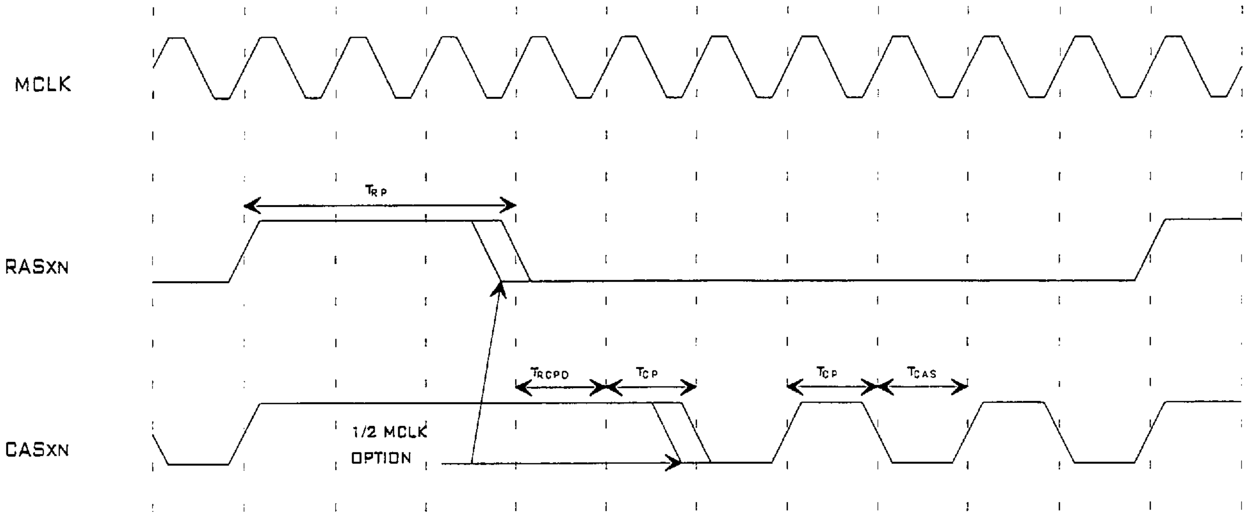


Figure 9.28 - Programmable Parameters

Memory Interface Timimg

No	Symbol	Parameter	Min (tMP)	Max (tMP)	Tol. (ns)	Load (pF)	Notes
	tRC	*Random Rd/Wr Cycle time*	*5*	*13*			1
	tPC	*Fast Page Mode Cycle Time*	*2*	*4*			1
	tRAC	*Access Time From RASn*	*3*	*8*			1
	tCAC	*Access Time From CASn*	*1*	*2.5*			1
	tAA	*Access Time From Col. Addr. (MA)*	*2*	*4*			1
	tCPA	*Access Time From CASn Precharge*	*2*	*4*			1
	tOFF	*Output Buffer Turn-off Delay ****	*1*	*2*			1
M4	tRP	RASn Precharge Time	1.5	5		100	
M5	tRAS	RASn Pulse Width	3	8		100	2
M6	tRSH	RASn Hold Referenced to CASn	1	2		100	
M7	tCSH	CASn Hold Referenced to RASn	3	8		100	
M8	tCAS	CASxn Pulse Width	1	2.5		100	
M9	tCP	CASxn Precharge Time	0.5	2		100	
M10	tRCD	RASxn to CASn Delay Time	1.5	6		100	
M11	tRAD	RASxn to Column Address Delay Time	1	4		100	
M12	tASR	Row Address Setup Time	0.5	1		100	
M13	tRAH	Row Address Hold Time	1	1.5		100	
M14	tASC	Column Address Setup Time	0.5	2		100	

No	Symbol	Parameter	Min (tMP)	Max (tMP)	Tol. (ns)	Load (pF)	Notes
M15	tCAH	Column Address Hold Time	1	2.5		100	
M16	tAR	Column Address Hold Ref. to RASxn	3	8		100	2
M17	tRAL	Column Address to RASn Lead Time	2	4		100	
M18	tWCH	WExxn Hold Ref. to CASxn	1	2		100	
M19	tWCR	WExxn Hold Ref. to RASxn	3	8		100	
M20	tWP	WExxn Pulse Width	1.5	2.5		100	
M21	tRWL	WExxn to RASn Lead Time	1.5	2.5		100	2
M22	tCWL	WExxn to CASn Lead Time	1.5	2.5		100	
M23	tWCS	WExxn Setup to CASxn	0	0.5		100	
M24	tDS	MD Setup to CASxn	1	2		100	
M25	tDH	MD Hold to CASxn	1	2		100	
M26	tDHR	MD Hold Ref. to RASxn	3	8		100	
M27	tCSR	CASn Setup to RASxn (Ref. Cycle)	0.5	1		100	3
M28	tCHR	CASn Hold to RASxn (Ref. Cycle)	2	2.5		100	3

Notes:

1. These parameters are for reference only.
2. The maximum specified is for maximum programmable for a single cycle, it is not the absolute maximum.
3. These parameters are for CAS-before-RAS refresh only.

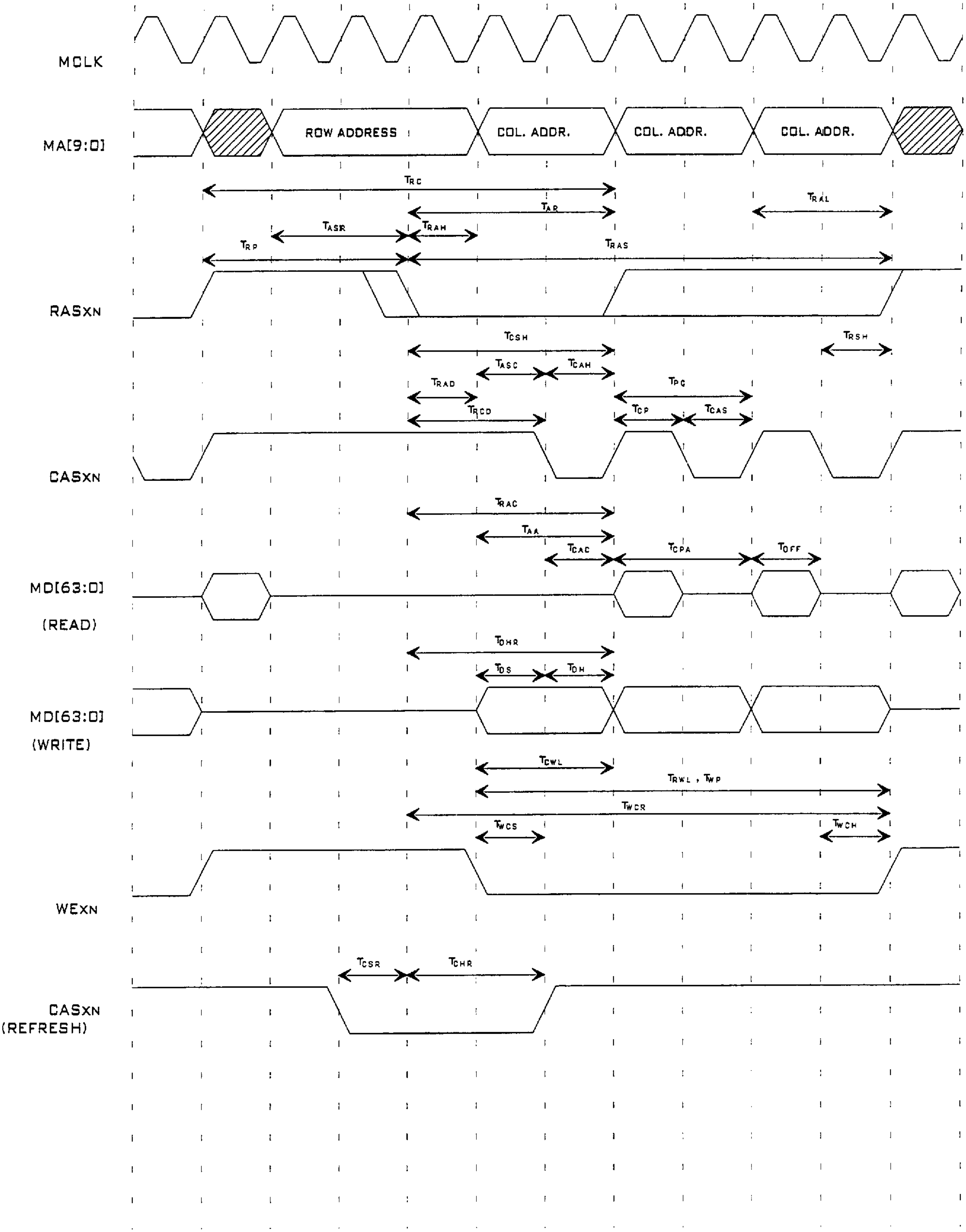


Figure 9.29 - Memory Interface Timing

Standard Memory Design for 70ns and 45ns DRAM's

Symbol	Parameter	DS1	50MHz	DS1	50Mhz	DS2	66MHz	DS2	66MHz
	Half clock option	off		on		off		on	
tMP	Memory Clock Period		20		20		15		15
tRC	Random Rd/Wr Cycle time	7	140	7	140	6	90	6	90
tPC	Fast Page Mode Cycle Time	2	40	2	40	2	30	2	30
tRAC	Access Time From RASn	4	80	4	80	3.5	52.5	3.5	52.5
tCAC	Access Time From CASn	1	20	1.5	30	1	15	1.5	22.5
tAA	Access Time From Col. Addr. (MA)	2	40	2	40	2	30	2	30
tCPA	Access Time From CASn Precharge	2	40	2	40	2	30	2	30
tOFF	Output Buffer Turn-off Delay ***	1	20	1.3	25	1.3	18.8	1.3	18.8
tRP	RASn Precharge Time	3	60	3	60	2.5	37.5	2.5	37.5
tRAS	RASn Pulse Width	4	80	4	80	3.5	52.5	3.5	52.5
tRSH	RASn Hold Referenced to CASn	1	20	1.5	30	1	15	1.5	22.5
tCSH	CASn Hold Referenced to RASn	4	80	4	80	3.5	52.5	3.5	52.5
tCAS	CASn Pulse Width	1	20	1.5	30	1	15	1.5	22.5
tRCD	RASn to CASn Delay Time	3	60	3	60	2.5	37.5	2	30
tRAD	RASn to Column Address Delay Time	1	20	1	20	1.5	22.5	1.5	22.5
tCP	CASn Precharge Time	1	20	0.5	10	1	15	0.5	7.5
tASR	Row Address Setup Time	1	20	1	20	0.5	7.5	0.5	7.5
tRAH	Row Address Hold Time	1	20	1	20	1.5	22.5	1.5	22.5
tASC	Column Address Setup Time	1	20	0.5	10	1	15	0.5	7.5
tCAH	Column Address Hold Time	1	20	1.5	30	1	15	1.5	22.5
tAR	Column Address Hold Ref. to RASn	4	80	4	80	3.5	52.5	3.5	52.5

Symbol	Parameter	DS1	50MHz	DS1	50Mhz	DS2	66MHz	DS2	66MHz
tRAL	Column Address to RASn Lead Time	2	40	2	40	2	30	2	30
tWCH	WExxn Hold Ref. to CASn	1	20	1	20	2	30	1.5	22.5
tWCR	WExxn Hold Ref. to RASn	4	80	4	80	3.5	52.5	3.5	52.5
tWP	WExxn Pulse Width	1	20	1.5	30	2	30	1.5	22.5
tRWL	WExxn to RASn Lead Time	1	20	1.5	30	1.5	22.5	1.5	22.5
tCWL	WExxn to CASn Lead Time	2	40	1.5	30	2	30	1.5	22.5
tDS	WExxn Setup to CASn	1	20	0	0	1	15	0.5	7.5
tDS	MD Setup to WExxn	1	20	0.5	10	1	15	0.5	7.5
tDH	MD Hold to WExxn	1	20	1.5	30	1	15	1.5	22.5
tDHR	MD Hold Ref. to RASn	4	80	4	80	3.5	52.5	3.5	52.5
tCSR	CASn Setup to RASn (Ref. Cycle) (3)	1	20	1	20	0.5	7.5	0.5	7.5
tCHR	CASn Hold to RASn (Ref. Cycle) (3)	2	40	2	40	2.5	37.5	2.5	37.5
tRMW	Read-Modify-Write Cycle Time	10	200	10	200	9	135	9	135
tPRMW	Page Read-Modify-Write Cycle Time	5	100	5	100	5	75	5	75

9.3.6 - Multimedia Port Timing

Multimedia Port Timing

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pf)	Notes
MM1	tMCKP	Multimedia Clock Period	30			
MM2	tMCKL	Multimedia Clock Low Time	10			
MM3	tMCKH	Multimedia Clock High Time	10			
MM4	tMMS	Multimedia Input Setup to IMCLK	4			
MM5	tMMH	Multimedia Input Hold to IMCLK	4			

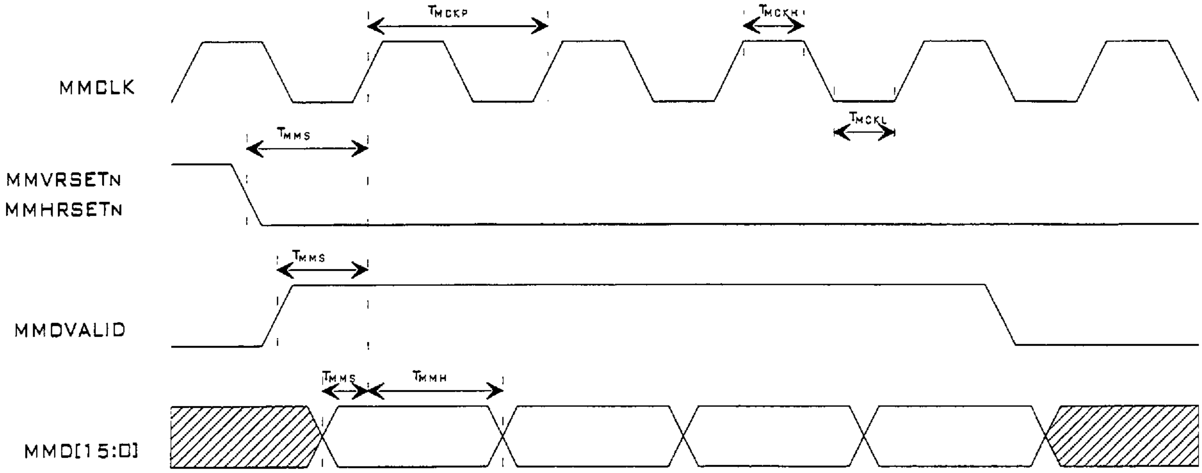


Figure 9.30 - Multimedia Port Timing

Multimedia Output Port (Video Window) Timing - MD64

No	Symbol	Parameter	Min (ns)	Max (ns)	Load (pf)	Notes
MM6	tMCKP	Video Window Clock Period	15			
MM7	tCASD	CASxn Delay from MCLK		10	50	1
MM8	tMXD	MDMXn Delay from MCLK		10	50	1
MM9	tVLD	VDVALID Delay from MCLK		10	50	
	tCAC	*MD[63:0] Valid from CASxn*		*tCAS-2*	50	
	tVDS	*Video Data Setup to MCLK*	*3*			2
	tVDH	*Video Data Hold from MCLK*	*1*			
	t374S	*MD[63:0] Setup to CASxn (Setup of 374)*		*2*	50	
	t374C	*VD[31:0] Delay from CASxn (374 clock to data delay)*		*2*	50	
	t374E	*VD[31:0] Delay from CASxn/MDMXn (374 Enable to data delay)*		*7*	50	

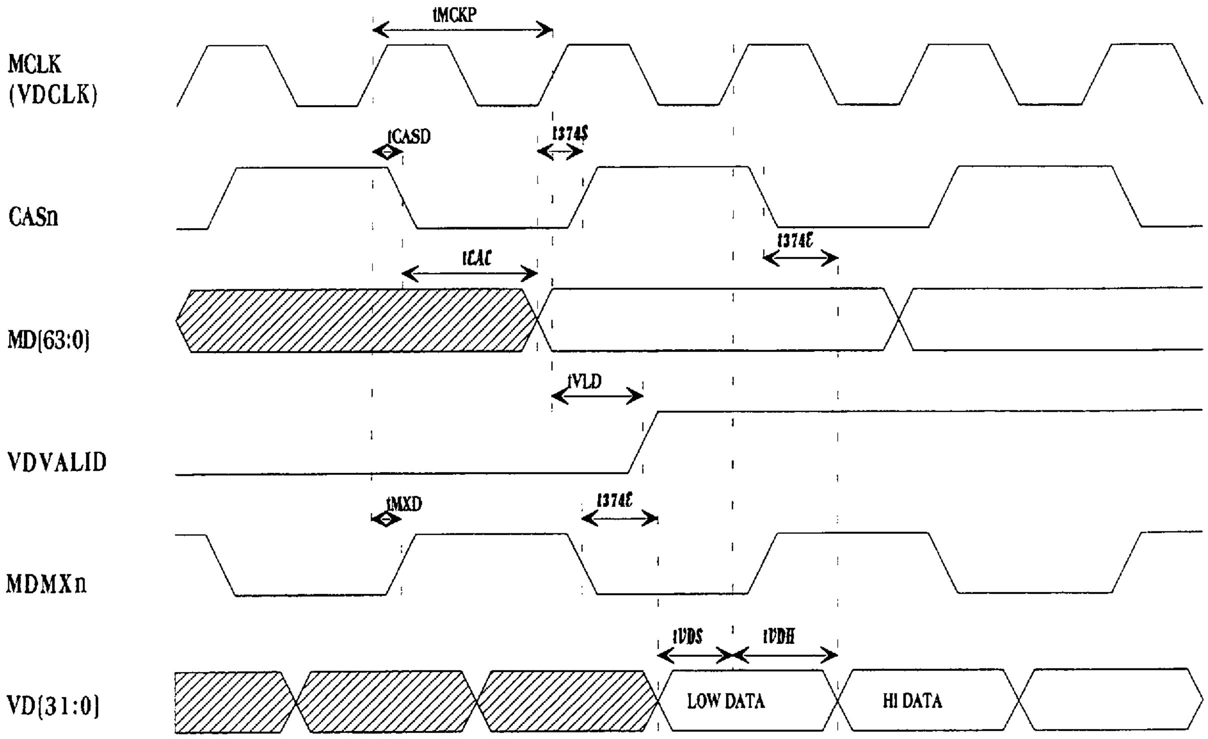


Figure 9.31 - Multimedia Output Port (Video Window) Timing - MD64

Notes:

1. The delays of these two signals should match to avoid contention on VD [31:0].
2. tVDS = tMXD + t374E or tCASD + t374C, whichever is worse. To improve tVDS, MCLK can be buffered to generate VDCLK.

CHAPTER 10: VIDEO BIOS

The video BIOS provides VGA compatible support and extended features for the OTI-64107. The following topics are covered in this section:

10.1 Overview

This section provides an overview of the capabilities of the OTI-64107 Video BIOS. The Video BIOS includes a Power-on Self Test (POST), as well as code and tables to implement the Video Modes and BIOS Functions. The POST initializes and tests the OTI-64107, Clock, DAC, and display memory during system boot. The Video Modes and BIOS Functions are described below. The primary software interface to the Video BIOS is via Interrupt 10h. The services provided by Interrupt 10h include setting the video mode, moving the cursor, reading and writing characters or pixels, scrolling up or down, and setting the color palette.

10.1.1 Video Modes

The OTI-64107 supports a large number of predefined Video Modes. A program selects a Video Mode by calling a Set Mode function in the BIOS. The BIOS supports all VGA compatible modes (see section 10.2) as well as many Extended Modes (see section 10.3). The Video Mode defines the display resolution and color depth, display timing, character font, and whether display memory is addressed as "all points addressable" graphics data or character addressable text.

10.1.2 Differences Between Text and Graphics Modes

The standard BIOS functions (described in later sections) are supported for both text modes and graphics modes. However, there are some differences.

- Attributes are defined for text modes only.
- The attribute parameter supplied to the Write Character function and the scrolling functions in text modes is redefined to be a pixel value in graphics modes.
- The Write TTY function assumes a black background by default when writing text in graphics mode.
- Characters in graphics modes can be XOR'ed to the screen.
- The cursor is not shown in graphics modes.

10.1.3 Cursor Functions

The cursor defines the screen position (text row and column) where the next output character is placed. Although the cursor is often shown as a blinking underline, its shape may be changed using the Set Cursor Type function. The BIOS normally updates the cursor position after writing a character to the screen at the current cursor position. Programs must use a cursor function to move the cursor if the updated cursor position is not appropriate.

10.1.4 Scrolling Functions

The scrolling functions move an area of the active display. There are two scrolling functions, Scroll Up and Scroll Down, which are functionally similar except for the direction of movement.

10.1.5 Character / Attribute Functions

Programs use the character/attribute function to place text at the current cursor position. There are functions available to read and write characters and attributes.

The character set is the standard PC-compatible extension of the ASCII character set.

Attributes define certain display characteristics, such as underlining, blinking, or foreground and background color. The table below shows a listing of the color mapping /attribute codes.

Table 10.1 Color Mapping / Attribute Codes

Attribute	I R G B	Monochrome	Color
00h	0 0 0 0	Black	Black
01h	0 0 0 1	Underline	Blue
02h	0 0 1 0	Video	Green
03h	0 0 1 1	Video	Cyan
04h	0 1 0 0	Video	Red
05h	0 1 0 1	Video	Magenta
06h	0 1 1 0	Video	Brown
07h	0 1 1 1	Video	White
08h	1 0 0 0	Black	Dark Gray
09h	1 0 0 1	Underline	Light Blue
0Ah	1 0 1 0	Video	Light Green
0Bh	1 0 1 1	Video	Light Cyan
0Ch	1 1 0 0	Video	Light Red
0Dh	1 1 0 1	Video	Light Magenta
0Eh	1 1 1 0	Video	Yellow
0Fh	1 1 1 1	Video	Intensified White

For read/write character functions while in the CGA graphics modes (4,5, and 6), characters are formed from a font table in the system BIOS ROM. Only the first 128 characters (characters 0-7Fh) are found there. Interrupt Vector 1Fh (memory location 0:007Ch) contains a pointer to a 1 K byte table where the font data for characters 80h-ffh are found.

For the EGA graphics modes (D- 10) and the VGA graphics modes (11h-13h), all 256 graphics characters are supplied in the Video BIOS ROM at the location specified by the pointer at Interrupt Vector 43h (location 0:010C).

For the write character functions while in graphics mode, the replication factor contained in CX on entry produces valid results only for characters contained on the same row. Continuation to succeeding lines will not operate correctly.

10.1.6 Color Palette Functions

The OTI-64107 VGA BIOS provides the ability for a programmer to define different colors to be displayed on the screen. This is performed with the Set Color Palette and Set Palette Registers functions.

10.1.7 Graphics Mode Functions

The BIOS capabilities for doing graphics are limited. They allow a program to write or retrieve the current value of a pixel at a given row and column location of a specific page. These functions are relatively slow and are insufficient for complex graphics. They are provided only as a general mechanism.

10.2 VGA Compatible Modes

Mode	Resolution	Colors	Font	Alpha Format	Video Mode	Display Mode	Buffer Start	Max Pgs. (Note 1)
0.1	320x200	16	8x8	40x25	CGA	Text	B8000	8
0*,1*	320x350	16	8x14	40x25	EGA	Text	B8000	8
0+,1+	360x400	16	9x16	40x25	VGA	Text	B8000	8
2,3	640x200	16	8x8	80x25	CGA	Text	B8000	8
2*,3*	640x350	16	8x14	80x25	EGA	Text	B8000	8
2+,3+	720x400	16	9x16	80x25	VGA	Text	B8000	8
4,5	320x200	4	8x8	40x25	CGA	Graphics	B8000	1
6	640x200	2	8x8	80x25	CGA	Graphics	B8000	1
7	720x350	4	9x14	80x25	HCG/MDA	Text	B0000	8
7+	720x400	4	9x16	80x25	VGA	Text	B0000	8
D	320x200	16	8x8	40x25	EGA	Graphics	A0000	8
E	640x200	16	8x8	80x25	EGA	Graphics	A0000	4
F	640x350	4	8x14	80x25	EGA	Graphics	A0000	2
10	640x350	16	8x14	80x25	EGA	Graphics	A0000	2
11	640x480	2	8x16	80x30	VGA	Graphics	A0000	1
12	640x480	16	8x16	80x30	VGA	Graphics	A0000	1
13	320x200	256	8x8	40x25	VGA	Graphics	A0000	1

Notes:
1. With 512K memory the maximum # of pages is doubled.

10.3 OTI-64107 Extended Modes

Mode (hex)	VESA Mode (hex)	Resolutio	Colors	Font	Alpha Format	Dot Clk (MHz) P8/P16/P24	H-freq (KHz)	V-freq (Hz)	Display Memory Required	VESA	Memory Datapath 50MHz/60MHz	P Bus Width
12		640X480	16	8X16	80X30	25.175	31.5	60	256K	-	32/32	8
12		640X480	16	8X16	80X30	31.5	37.86	72	256K	STD	32/32	8
12		640X480	16	8X16	80X30	31.5	37.5	75	256K	-	32/32	8
4E	108	80X60	16	8X8	80X60	25.175	31.5	60	256K	-	32/32	8
4F	10C	132X60	16	8X8	132X60	40	31.5	60	256K	-	32/32	8
50	109	132X25	16	8X14	132X25	40	31.5	70	256K	-	32/32	8
51	10A	132X43	16	8X8	132X43	40	31.5	70	256K	-	32/32	8
52	6A/102	800X600	16	8X16	100X37.5	36	35.156	56	256K	MFG. GL	32/32	8
52	6A/102	800X600	16	8X16	100X37.5	40	37.879	60	256K	MFG. GL	32/32	8
52	6A/102	800X600	16	8X16	100X37.5	50	48.077	72	256K	STD	32/32	8
52	6A/102	800X600	16	8X16	100X37.5	49.5	46.875	75	256K	-	32/32	8
53	101	640X480	256	8X16	80X30	25.175	31.5	60	512K	-	32/32	8
53	101	640X480	256	8X16	80X30	31.5	37.86	72	512K	STD	32/32	8
53	101	640X480	256	8X16	80X30	31.5	37.5	75	512K	-	32/32	8
54	103	800X600	256	8X16	100X37.5	36	35.156	56	512K	MFG. GL	32/32	8
54	103	800X600	256	8X16	100X37.5	40	37.879	60	512K	MFG. GL	32/32	8
54	103	800X600	256	8X16	100X37.5	50	48.077	72	512K	STD	32/32	8
54	103	800X600	256	8X16	100X37.5	49.5	46.875	75	512K	-	32/32	8
56	104	1024X768	16	8X16	128X48	44.9	35.52	87I	512K	-	32/32	8
56	104	1024X768	16	8X16	128X48	65	48.363	60	512K	MFG. GL	32/32	8
56	104	1024X768	16	8X16	128X48	75	56.69	70	512K	STD	32/32	8
56	104	1024X768	16	8X16	128X48	78.75	58.04	72	512K	-	32/32	8
56	104	1024X768	16	8X16	128X48	78.75	60.023	75	512K	STD	32/32	8
58	106	1280X102	16	8X16	160X64	80.00/40.00	48.78	87I	1M	-	32/32	8/16
58	106	1280X102	16	8X16	160X64	110.00/55.00	64.25	60	1M	-	32/32	8/16
58	106	1280X102	16	8X16	160X64	NA/65.00		70	1M	-	32/32	16
58	106	1280X102	16	8X16	160X64	NA/67.00	79.976	75	1M	STD	32/32	16

OTI-64107 Extended Modes (Cont.)

Mode (hex)	VESA Mode (hex)	Resolutio	Colors	Font	Alpha Format	Dot Clk (MHz) P8/P16/P24	H-freq (KHz)	V-freq (Hz)	Display Memory Required	VESA	Memory Datapath 50MHz/60MHz	P Bus Width
59	105	1024X768	256	8X16	128X48	44.9	35.52	87I	1M	-	32/32	8
59	105	1024X768	256	8X16	128X48	65	48.363	60	1M	MFG. GL	32/32	8
59	105	1024X768	256	8X16	128X48	75	56.69	70	1M	STD	32/32	8
59	105	1024X768	256	8X16	128X48	78.75	58.04	72	1M	-	32/32	8
59	105	1024X768	256	8X16	128X48	78.75	60.023	75	1M	STD	32/32	8
5A/5C	111/110	640X480	64K/32K	8X16	80X30	50.00/25.17	31.5	60	1M	-	32/32	8/16
5A/5C	111/110	640X480	64K/32K	8X16	80X30	63.00/31.50	37.86	72	1M	STD	32/32	8/16
5A/5C	111/110	640X480	64K/32K	8X16	80X30	63.00/31.50	37.5	75	1M	-	32/32	8/16
5B/62		640X400	32KX64K	8X16	80X25	50.00/25.17	31.5	70	512K	-	32/32	8/16
5D/60	113/114	800X600	32KX64K	8X16	100X37.5	72.00/36.00	35.156	56	1M	MFG. GL	32/32	8/16
5D/60	113/114	800X600	32K/64K	8X16	100X37.5	80.00/40.00	37.5	60	1M	MFG. GL	32/32	8/16
5D/60	113/114	800X600	32K/64K	8X16	100X37.5	100.00/50.0	48.077	72	1M	STD	64/32	8/16
5D/60	113/114	800X600	32K/64K	8X16	100X37.5	99.00/49.50	46.875	75	1M	STD	64/32	8/16
5E	107	1280X102	256	8X16	160X64	80.00/40.00	48.78	87I	2M	-	32/32	8/16
5E	107	1280X102	256	8X16	160X64	110.00/55.0	64.25	60	2M	-	64/64	8/16
5E	107	1280X102	256	8X16	160X64	NA/65.00		70	2M	-	64/64	16
5E	107	1280X102	256	8X16	160X64	NA/67.50	79.976	75	2M	STD	64/64	16
5F	112	640X480	16.8M	8X16	80X30	75.00/37/50	31.55	60	1M	-	32/32	8/16
5F	112	640X480	16.8M	8X16	80X30	94.50/47.25	37.86	72	1M	STD	64/32	8/16
5F	112	640X480	16.8M	8X16	80X30	94.50/47.25	37.5	75	1M	-	64/32	8/16
61	100	640X400	256	8X16	80X25	25.175	31.5	70	256K	-	32/32	8
63/64	116/117	1024X768	32K/64K	8X16	128X48	89.80/44.90	35.52	87I	2M	-	64/32	8/16
63/64	116/117	1024X768	32K/64K	8X16	128X48	NA/65.00	48.363	60	2M	MFG. GL	64/64	16
63/64	116/117	1024X768	32K/64K	8X16	128X48	NA/75.00	56.69	70	2M	-	64/64	16
63/64	116/117	1024X768	32K/64K	8X16	128X48	NA/78.75	58.04	72	2M	-	64/64	16
63/64	116/117	1024X768	32K/64K	8X16	128X48	NA/78.75	60.023	75	2M	STD	64/64	16

OTI-64107 Extended Modes (Cont.)

Mode (hex)	VESA Mode (hex)	Resolutio	Colors	Font	Alpha Format	Dot Clk (MHz) P8/P16/P24	H-freq (KHz)	V-freq (Hz)	Display Memory Required	VESA	Memory Datapath 50MHz/60MHz	P Bus Width
65/66	119/11A	1280X102	32K/64K	8X16	160X64	NA/80.00	48.78	87I	4M	-	64/64	16
65/66	119/11A	1280X102	32K/64K	8X16	160X64	NA/110.00		60	4M	-	NA/64	16
67	115	800X600	16.8M	8X16	100X37.5	108.00/54.00	35.156	56	2M	MFG. GL	64/32	8/16
67	115	800X600	16.8M	8X16	100X37.5	NA/60.00	37.879	60	2M	MFG. GL	64/64	16
67	115	800X600	16.8M	8X16	100X37.5	NA/75.00	48.077	72	2M	STD	N/A/64	16
67	115	800X600	16.8M	8X16	100X37.5	NA/74.25	46.875	75	2M	STD	N/A/64	16
68	118	1024X768	16.8M	8X16	8X16	NA/67.35/44.9	35.52	87I	4M	-	64/64	16/24
68	118	1024X768	16.8M	8X16	8X16	NA/97.50/65.0	48.363	60	4M	MFG. GL	N/A/64	16/24
69		640X480	4G	8X16	80X30	NA/50.00/25.1	31.5	60	2M	STD	64/32	16/24
69		640X480	4G	8X16	80X30	NA/63.00/31.5	37.86	72	2M	STD	64/64	16/24
69		640X480	4G	8X16	80X30	NA/63.00/31.5	37.5	75	2M	STD	64/64	16/24
6B		800X600	4G	8X16	100X37.5	NA/72.00/36.0	35.156	56	2M	MFG. GL	N/A/64	16/24
6B		800X600	4G	8X16	100X37.5	NA/80.00/40.0	37.879	60	2M	MFG. GL	N/A/64	16/24
6C		1024X768	4G	8X16	128X48	NA/89.80/44.9	35.52	87I	4M	-	N/A/64	16/24

Notes:

I = Interlaced mode
Std = VESA monitor timing Standard
Mfg GL = VESA monitor timing Manufacturing Guideline

10.4 VGA BIOS STANDARD FUNCTIONS

OTI-64107 VGA BIOS functions are accessed by executing interrupt 10h. The function code is placed in register AH. Additional parameters are placed in the registers as indicated in the function descriptions below.

Table 10.3 groups the BIOS functions by function. Table 10.4 orders the BIOS functions by function number.

Note that some of the following functions behave differently or expect different parameters based on whether the current display mode is text or graphics. Except where noted, these functions operate only in the VGA "compatible" Video Modes. When an Extended Video Mode has been set, it is assumed a driver is present to properly support the Operating System or Application Program under the increased resolution and/or color depth of the Extended Video Mode.

10.4.1 BIOS Functions (Sorted by function)

Name	Function
Set/Get Mode Functions	
Set Mode	0
Get Video State	F
Set Active Page	5
Cursor Functions	
Set Cursor Type	1
Set Cursor Position	2
Read Cursor Position	3
Character/Attribute Functions	
Write Character & Attribute	9
Write Character Only	A
Read Character & Attribute	8
Write TTY	E
Write String	13
Scrolling Functions	
Scroll Up	6
Scroll Down	7
Write Dot	C
Read Dot	D
Palette Functions	
Set Color Palette	B
Set Palettte Registers	10
Miscellaneous Functions	
Load Character Font Info	11
Alternate Select	12
Return DCC Info	1A
Return Functionality Info	1B
Save/Restore State	1C

10.4.2 BIOS Function Summary (Sorted by Function, Subfunction Number)

Function	Subfunction	Description
00h		Set Mode
01h		Set Cursor Type
02h		Set Cursor Position
03h		Read Cursor Position
04h		Read Light Pen Position
05h		Set Active Display Page
06h		Scroll Active Page Up
07h		Scroll Active Page Down
08h		Read Character & Attribute at Current Cursor Position
09h		Write Character & Attribute to Current Cursor Position
0Ah		Write Character Only to Current Cursor Position
0Bh		Set Color Palette
0Ch		Write Dot
0Dh		Read Dot
0Eh		Write TTY-style to Active Page
0Fh		Return Current Video State
10h		Set Palette Registers
	00h	Set Individual Palette Register
	01h	Set Overscan Register
	02h	Set All Palette Registers and Overscan Register
	03h	Toggle Intensity / Blinking Bit
	07h	Read Individual Palettte Register
	08h	Read Overscan Register
	09h	Read All Palette Registers and Overscan Register
	10h	Set Individual Color Register
	12h	Set Block of Color Registers
	13h	Select Color Page
	15h	Read Individual Color Register
	17h	Read Block of Color Registers
	1Ah	Read Current Color Page Number
	1Bh	Sum Color Values to Gray Scale
11h		Load Font Info
	00h	Load User Font
	01h	Load ROM Monochrome Font
	02h	Load ROM 8x8 Font
	04h	Load ROM 8 x 16 Font
	10h	Load User Font
	11h	Load ROM Monochrome Font
	12h	Load ROM 8x8 Font
	14h	Load ROM 8 x 16 Font
	20h	Load User Graphics Characters INT 1Fh (8x8)
	21h	Load User Graphics Characters
	22h	Load Graphics Mode ROM 8 x 14 Font
	23h	Load Graphics Mode ROM 8x8 Font
	24h	Load Graphics Mode ROM 8 x 16 Font
	30h	Return Character Font Information

Function	Subfunction	Description
12h		Alternate Select
	10h	Return Video Information
	20h	Select Alternate Print Screen Routine
	30h	Select Scan Lines for Text Modes
	31h	Default Palette Loading During Mode Set
	32h	Video Enable / Disable
	33h	Summing to Gray Scales
	34h	Cursor Emulation
	35h	Display Switch
	36h	Video Screen On / Off
13h		Write Text Sting
1Ah		Return Display Combination Code (DCC)
1Bh		Return Functionality / State Info
1Ch		Save / Restore Video State

10.4.3 Interrupt 10h Video Bios Functions

Int 10h - Function 0h - Set Video Mode

Entry: AH = 0h
AL = Bit 7 has flag indicating to preserve/clear display memory
Bits 6-0 have video mode value (refer to Table 10.1, VGA-compatible modes, or Table 10.2, extended video modes

Exit: None

The Set Mode function sets the display system to one of the predefined text or graphics modes. The display memory may be cleared or preserved depending upon the state of AL bit 7. Setting bit 7 of the AL register preserves the contents of display memory (although the appearance of the display may be altered if the new video mode interprets the display memory differently). Set Mode loads the appropriate character font into plane 2, sets the default colors in the palette, and sets the cursor position to 0, 0 for all pages. Refer to the tables at the beginning of this chapter for a list of the VGA-compatible video modes as well as the Oak extended video modes.

Int 10h - Function 1h - Set Cursor Type

Entry: AH = 01h
CH = Cursor start line (bits 4-0)
CL = Cursor end line (bits 4-0)

Exit: None

Note: Setting Bit 5 in start line (CH) causes no cursor display

This function specifies the size of the cursor in text modes. Cursor size parameters are stored in the byte at 40:60h.

Int 10h - Function 2h - Set Cursor Position

Entry: AH = 02h
DH = row
DL = column
BH = page number

Exit: None

This function updates the cursor position as indicated by the row and column coordinates in DH and DL. When the cursor position is set, all character writes and reads occur at that position (0, 0 corresponds to the upper left corner of the display).

Int 10h - Function 3h - Read Cursor Position

Entry: AH = 03h
BH = page number

Exit: DH = row
DL = column
CH = start scanline of cursor
CL = end scanline of cursor

This function reads and returns the cursor position for the specified display page.

Int 10h - Function 5h - Set Active Page

Entry: AH = 05h
AL = new page value

Exit: None

Most text display modes have multiple displayable pages, or screen images. Only one screen, called the Active Page, is displayed at a time. The other pages are accessible by the CPU, but are not visible to the user. This function sets the active page for the current display mode.

The BIOS maintains a cursor position for each page. When selecting the active page, the cursor position for that page becomes active and is displayed. Paging can be used when it is desirable to hide screen updates and for animation effects.

Int 10h - Function 6h - Scroll Active Page Up

Entry: AH = 06h
AL = Number of lines (input lines blanked at bottom of window)
(AL = 0 means blank entire window)
BH = Attribute to be used on blank line
CH,CL = Row, column of upper left corner of scroll
DH,DL = Row, column of lower right corner of scroll

Exit: None

Int 10h - Function 7h - Scroll Active Page Down

Entry: AH = 07h
AL = Number of lines (input lines blanked at top of window)
(0 means blank entire window)
BH = Attribute to be used on blank line
CH,CL = Row, column of upper left corner of scroll
DH, DL = Row, column of lower right corner of scroll

Exit: None

These two scrolling functions affect only the currently displayed active page, defining the area of the display which is moved. The calling program defines the top left corner and the bottom right corner of scroll window and the number of lines to be scrolled. The lines scrolled off the screen are lost. Note that a scroll function with "number of lines" equal to zero clears the specified window, so that all characters within that window are blanked. The attribute specified in BH is loaded into that window.

Int 10h - Function 8h - Read Character and Attribute

Entry: AH = 08h
BH = page

Exit: AL = Character read
AH = Attribute of character read (Alpha modes only) See Table 10.5

The Read Character and Attribute function reads and returns a character and the corresponding attribute from the current cursor position on the specified page. Any display page may be specified so the character may not be visible on the screen.
Note: Graphics modes must have a background color of 0 for this function to operate correctly.

Int 10h - Function 9h - Write Character and Attribute

Entry: AH = 09h
AL = Character to write
BL = Attribute of character (Alpha mode) See Table 10.5
BL = Color of character (Graphics mode)
BH = Display rage
CX = Number of times to write character

Exit: None

Note: In graphics mode, if Bit 7 of BL is 1, then the character is XOR'ed with the screen.

Int 10h - Function 0Ah - Write Character Only

Entry: AH = 0Ah
AL = Character to write
BH = Display page
BL = Foreground color (Graphics ONLY)
CX = Count of character to write

Exit: None

The Write Character Only function changes only the character data and not the corresponding attribute at the current cursor position.

Int 10h - Function 0Bh - Set Color Palette

Entry: AH = 0Bh
BH = Palettte color ID being set (0-127)
BL = Color value to be used with that color ID

Where: Color ID = 0 selects the background color (0-15)
Color ID = 1 selects the palette to be used: 0 = Green (1)/Red(2)/Brown(3)
1 = Cyan(1)/Magenta(2)/White(3)

Exit: None

In 40x25 or 80x25 text modes, the value set for palette color 0 indicates the border color (0-31, where 16-31 select the high-inten ity background set.)
This function is provided for compatibility with the CGA BIOS. For the VGA, this function is needed only for 320x200 (4-color) graphics. The VGA palette function 10h (Set Palette Registers) provides a super-set of this functions capabilities.

Int 10h - Function 0Ch - Write Dot

Entry: AH = 0Ch
BH = Page
DX = Row number
CX = Column number
AL = Color value for pixel

Exit: None

Note: If Bit 7 of AL = 1, then the color value is XOR'ed with the current contents of the dot.

This function writes a single pixel of the specified color at the specified pixel row and pixel column of the indicated display page. The color value can range from 0 to 255 depending on the display mode. In 4-color modes, the color value may be 0-3; in a 16-color mode, the color value may be 0-15; in a 256-color mode, the color value must be in the range 0-255. All VGA modes except mode 13h can also do an XOR on the current contents of the display, which is useful for display effects such as “rubber banding” and highlighting.

Int 10h - Function 0Dh - Read Dot

Entry: AH = 0Dh
BH = Page
DX = Pixel Row number
CX = Pixel Column number

Exit: Al = Color of dot read

This function reads and returns a single pixel from the specified pixel row and pixel column of the specified display page. The function return the color value of the pixel in the AL register. The color value can range from 0 to 255 depending on the number of colors displayable by the current display mode.

Int 10h - Function 0Eh - Write TTY

The Write TTY function writes a character to the screen and update the cursor position automatically. When the cursor reaches the right side of the screen it wraps to the beginning of the next line. When the cursor reaches the bottom of the screen it automatically scrolls the screen up by one line.

Entry: AH = 0Eh
AL = Character to write
BL = Foreground color in graphics mode

Exit: None

Note: The screen width and height in characters depends on the current display mode.

Certain characters are interpreted by Write TTY and result in the following actions:

- CR (carriage return, ASCII 0Dh) resets the cursor to column 0 on the same line
- LF (line feed, ASCII 0Ah) moves the cursor down one character row, leaving the column position unchanged. If the cursor reaches the bottom of the screen, the screen is scrolled up by one row.
- BS (back space, ASCII 08h) moves the cursor back by one character position
- BL (bell, ASCII 07h) cause the speaker to "beep" once, leaving the cursor position unchanged

Int 10h - Function 0Fh - Get Video State

Get Video State reads and returns information about the state of the display system.

Entry: AH = 0Fh

Exit: AL =
Bit 7 = "preserve/clear" display bit from last Set Mode
Bits 6-0 = current display mode
AH = number of character columns on screen
BH = active display page

Note: If the program uses the no blank option on the Set Mode function, this will be returned in the Get Video State function. This flag (mode byte msb) will be set if the last call to set mode was to not blank the screen.

Int 10h - Function 10h - Set Palette Registers

The Set Palette Registers function allows a program to set all 256 indexed colors available in the VGA modes.

Entry: AH = 10h

AL = 0: Set individual palette register:
BL = Palette register to be set
BH = Value to set

AL = 1: Set overscan register:
BH = Value to set

AL = 2: Set palette registers and overscan:
ES:DX = Pointer to 17 byte table: Bytes 0-15 are palette values
Byte 16 is the overscan value

AL = 3: Toggle intensify / blinking bit:
BL = 0: Enable intensify
BL = 1: Enable blinking

This redefines one of the bits in the attribute code to allow for 16 background colors. When intensify is enabled it provides 16 background colors and 16 foreground colors. When blinking is enabled it provides 8 background colors plus a blinking character.

AL = 7: Read individual palette register
BL = Palette register to be read
BH = Value Read

AL = 8: Read overscan register
BH = Value read

AL = 9: Read all palette registers and overscan
ES:DX = Pointer to 17 byte table: Bytes 0-15 are palette values
Byte 16 is the overscan value

AL = 10h:Set individual color register
BX = Color register number
CH = Green value
CL = Blue value
DH = Red value

AL = 12h:Set Block of color registers
BX = Number of first color register
CX = Number of registers to be set
ES:DX = Pointer to a table of color values.

The table should contain color values be in the sequence:<(red,green, blue), (red,green,blue).... (red,green,blue)>

Al = 13h:Select color page
BH = Paging mode or value
BL = 0 selects paging mode
BH = 0 selects four pages of 64 color registers
BH = 1 selects 16 pages of color registers
BL = 1 selects page:
BH = Number of the required page (0-3 or 0-15)

AL = 15h:Read individual color register
BX = Number of color register
CH = Green value
CL = Blue value
DH = Red value

AL = 17h:Read block of color registers
BX = Number of first color register
CX = Number of registers to be read
ES:DX = Pointer to a table to receive the color values

The table contains color values in the sequence<(red,green,blue), (red,green,blue)...(red,green, blue)>

AL = 1Ah:Read current color page number
BH = Current page
BL = Paging mode

AL = 1Bh:Sum color values to gray scale
BX = First color register to be summed
CX = Number of registers to sum

Exit: None

Int 10h - Function 11h - Load Character Font Info

Programs may load the character font sets used in the text modes. The fonts are saved in plane 2 of display memory. When in a text mode, the character font information for each character is retrieved from plane 2 and displayed at the proper position on the display.

Up to eight 256-character fonts may be stored. Two fonts may be displayed at any one time.

The character font load function should only be called after a Set Mode operation and before changing any of the characteristics of the display such as the cursor size. This is because loading a character font causes the VGA registers to be reprogrammed so that the BIOS can load the font.

Entry:	AH = 11h		
	AL = 0xh	Initiate mode set, completely resetting the video environment but maintaining display memory:	
	AL = 00h	Load User Font ES:BP = Pointer to user table CX = Count to store DX = Character offset into table BH = Number of bytes per character	BL = Block to load
	AL = 01h	Load ROM Monochrome Font:	BL = Block to load
	AL = 02h	Load ROM 8x8 Double Dot Font:	BL = Block to load
	AL = 03h	Set Block Specifier BL = Font Block Specifier	D3-D2 Attr bit-3 = 1, font 0-3 D1-D0 Attr bit-3 = 0, font 0-3

Note: When using AL = 3, a function call of AX = 1000h, BX = 0712h is recommended to set the color planes resulting in 512 characters and eight consistent colors AL = 04h Load ROM 8 x 16 character set:

BL = Target block

AL = 1xh Similar to (AL = 0x) functions except that:

- Page 0 must be active
- POINTS (bytes/character) will be recalculated
- ROWS will be recalculated from:
INT((200,350 or 400)/POINTS) - 1
- CRT_LEN will be calculated from: (ROWS + 1) * CRT_COLS * 2
- The CRTC will be reprogrammed as follows:

CR09 = POINTS - 1 (only in mode 7)	(Max Scan Line)
CR0A = POINTS - 2	(Cursor Start)
CR0B = 0	(Cursor End)
CR12 = ((ROWS +1)*POINTS)-1	(Vert Disp End)
CR14 = POINTS	(Underline Loc)

The above register calculations must be close to the original table values or undetermined results will occur. The functions in this group should only be called immediately after a mode set or undetermined results will occur.

AL = 10h	Load User Font	
	ES:BP = Pointer to user table	
	CX = Count to store	
	DX = Character offset into table	
	BH = Number of bytes per character	BL = Block to load
AL = 11h	Load ROM Monochrome Font:	BL = Block to load
AL = 12h	Load ROM 8 x 8 Double Dot Font:	BL = Block to load
AL = 14h	Load ROM 8 x 16 character set:	BL = Target block
AL = 20h	Load user graphics characters Int 1Fh (8x8)	ES:BP = Pointer to user table
AL = 21h	Load user graphics characters	ES:BP = Pointer to user table
	CX = Points (bytes per character)	
	BL = Row specifier	BL = 0 User (DL = Rows)
		BL = 1 14 (0Eh)
		BL = 2 25 (19h)
		BL = 3 43 (2Bh)
AL = 22h	Load ROM 8 x 14 Font:	BL = Row specifier
AL = 23h	Load ROM 8 x 8 Font:	BL = Row specifier
AL = 24h	Load Graphics mode ROM 8 x 16 set	
	BL = Number of rows on the screen:	
	BL = 1-14 rows	
	BL = 2-25 rows	
	BL = 3-43 rows	
AL = 30h	Get Font Information	
	BH = 0 Return Int 1Fh Pointer	
	BH = 1 Return Int 44h Pointer	
	BH = 2 Return ROM 8 x 14 Font Pointer	
	BH = 3 Return ROM 8 x 8 Font Pointer	
	BH = 4 Return ROM 8 x 8 Font Pointer (Top)	
	BH = 5 Return ROM Alternate 9x14 Pointer	
	BH = 6 Return ROM 8 x 16 Font Pointer	
	BH = 7 Return 9 x 16 Replacement Font Pointer	

Exit: CX = Points
DL = Rows
ES:BP = Pointer to table

Int 10h - Function 12h - Subfunction10h - Get Video Configuration Info

Entry:	AH = 12h	
	BL = 10h:	Subfunction number
Exit:	BH = 0:	Color mode in effect (3Dx)
	BH = 1:	Mono mode in effect (3Bx)
	BL = Memory Size: 0 = 64 K, 1 = 128 K, 2 = 193 K, 3 = 256 K	
	CH = Feature Bits	
	CL = Switch Settings	

This function retrieves configuration and mode information from the display subsystem.

Int 10h - Function 12h - Subfunction 20h - Select Alternate Print Screen

Entry:	AH = 12h	
	BL = 20h:	Subfunction number
Exit:	None	

The Select Alternate function selects an alternate Print Screen routine instead of the standard system BIOS ROM Print Screen routine. The alternate Print Screen routine prints all of the rows on the screen Many standard system board Print Screen routines print only 25 lines. This function works only in text modes, not graphics modes.

Int 10h - Function 12h - Subfunction 30h - Select Scan Lines

Entry:	AH = 12h	
	AL = Number of scan lines: 0 = 200 lines, 1 = 350 lines, 2 = 400 lines	
	BL = 30h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function sets the number of scan lines in text mode, and takes effect in the next mode set.

Int 10h - Function 12h - Subfunction 31h - Default Palette Loading

Entry:	AH = 12h	
	AL = 0	Enable palette loading
	AL = 1	Disable palette loading
	BL = 31h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function sets a bit in the BIOS Data Area to enable/disable palette loading on Mode Sets. When palette loading is disabled, neither the internal palette nor the DAC will be modified during Mode Sets or other BIOS calls.

Int 10h - Function 12h - Subfunction32h - Video Enable/Disable

Entry:	AH = 12h	
	AL = 0	Enable video
	AL = 1	Disable video
	BL = 32h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function enables/disables CPU access to the video subsystem. When the video subsystem is disabled, all I/O Port accesses and display memory accesses are disabled.

Int 10h - Function 12h - Subfunction 33h - Gray Scale Summing

Entry:	AH = 12h	
	AL = 0	Enable summing
	AL = 1	Disable summing
	BL = 33h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function enables/disables the conversion of RGB color values to gray based on a weighting of 30% red, 59% green and 11% blue.

Int 10h - Function 12 h- Subfunction 34h - Cursor Emulation

Entry:	AH = 12h	
	AL = 0	Enable emulation
	AL = 1	Disable emulation
	BL = 34h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function enables/disables BIOS cursor emulation. When cursor emulation is disabled, the cursor start/stop is set exactly by the cursor type BIOS function. When cursor emulation is enabled, the following algorithm is used:

Parameters	Cursor Type
Bit 5 = 1	No cursor
START<END =<3	Overbar cursor
START+2>=END	Underline cursor
START=>2	half-block cursor
START=<2 or END<START	Full-block cursor

Int 10h - Function 12h - Subfunction 35h - Display Switch

Entry:	AH = 12h	
	AL = 0	Initial adapter video off
	AL = 1	Initial planar video on
	AL = 2	Switch active display off
	AL = 3	Switch inactive display on
	BL = 35h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function supports using two video subsystems in the same computer. It is used when two video subsystems overlap in I/O address space.

Int 10h - Function 12h - Subfunction 36h - Video Screen On/Off

Entry:	AH = 12h	
	AL = 0	Enable video output
	AL = 1	Disable video output
	BL = 36h	Subfunction number
Exit:	AL = 12h	Indicates function is supported

This function enables/disables screen refresh. Disabling screen refresh gives the CPU full access to display memory since the CRTC does not compete for display memory access.

Int 10h - Function 13h - Write String

The Write String function writes a string of text containing one or more characters to the display. It also allows a program to write a fixed attribute for the whole screen, or a character and an attribute for each position on the screen. An option is provided to update the cursor position or to leave the cursor in the original position.

Entry: AH = 13h
ES:BP = Pointer to string to be written
CX = Character only count
DX = Cursor position to begin string
BH = Page number

AL = 0: Fixed attribute, cursor not moved
BL = attribute

AL = 1: Fixed attribute, cursor is moved
BL = attribute

AL = 2: String includes attributes, cursor not moved

AL = 3: String includes attributes, cursor is moved

Exit: None

When AL=0 or AL=1, the string contains characters only (<char>, <char>,...). When AL = 2 or AL = 3 the string contains character/attribute pairs ((<char>, <attr>), (<char>, <attr>), ...), with an attribute following each character.

The Write String function also treats the CR, LF, BS and Bell codes similar as the Write TTY function does.

Int 10h - Function 1Ah - Return Display Combination Code (DCC)

Entry: AH = 1Ah
AL = 0

Exit: AL = 1Ah Indicates function is supported

BL = Active Display Device (see table below)
BH = Alternate Display Device (see table below)

00h = No Display
01h = MDA
02h = CGA
03h = (reserved)
04h = EGA (mono)
05h = EGA (color)
06h = PGA
07h = VGA (mono)
08h = VGA (color)
09h = (reserved)
0Ah = (reserved)
0Bh = MCGA (mono)
0Ch = MCGA (color)

This function returns the display type.

Int 10h - Function 1Bh - Return Functionality/State Information

Entry: AH = 1Bh
BX = 0
ES:DI = Pointer to target buffer

Exit: AL = 1Bh indicates function is supported
ES:DI = points to the table described in the table below.

Return Functionality/State Information Table

Entry: BX = 00h

ES:DI = Buffer of size 40h bytes

(DI +00h) word - Offset to static functionality information
(DI +02h) word - Segment to static functionality information

Video States:(The following information is dynamically generated and reflects the current video state.)

(DI + 04h) byte - Video mode
(DI + 05h) byte - Columns on screen (character columns on screen)

(DI + 07h word - Length of regenerator buffer (bytes)
(DI + 09h) word - Starting address in regenerator buffer
(DI + 0Bh) word - Cursor position for eight display pages (row, column)

(DI + 1Bh) word - Cursor type setting (cursor start/end value)
(DI + 1Dh) byte - Active display page
(DI + 1Eh) word - CRT controller address (3Bx-monochrome, 3Dx-color)

(DI + 20h) byte - Current setting of 3x8 register
(DI + 21h) byte - Current setting of 3x9 register
(DI + 22h) byte - Rows on screen (character lines on screen)
(DI + 23h) word - Character height (scan lines per character)

(DI + 25h) byte - Display combination code (active)
(DI + 29h) byte - Display combination code (alternate)

(DI + 27h) word - Colors supported for current video mode
(DI + 29h) byte - Display pages supported for current video mode

(DI + 2Ah) byte - Scan lines in current video mode
=0 - 200 scan lines
=1 - 350 scan lines
=2 - 400 scan lines
=3 - 480 scan lines
=4 to 255 - reserved

(DI +2Bh) - Primary character block
=0 - Block 0
=1 - Block 1
=2 - Block 2
. . . . .
. . . . .
=255 - Block 255

This information is based on block specifier.[See (AH) = 11h, (AL) = 03h]

(DI + 2Dh) - Miscellaneous state information
Bits 7, 6 - Reserved
Bit 5 = 0 - Background intensity
= 1 - Blinking
Bit 4 = 1 - Cursor emulation active
Bit 3 = 1 - Mode set default palette loading disabled
Bit 2 = 1 - Monochrome display attached
Bit 1 = 1 - Summing active
Bit 0 = 1 - All modes on all displays active

(DI + 2Eh) byte - Reserved
(DI + 2Fh) byte - Reserved

(DI + 30h) byte - Reserved

(DI + 31h) byte - Video memory available
=0 - 64B
=1 - 128KB
=2 - 192KB
=3 - 256KB
=4 to 255 - Reserved

(DI + 32h) byte - Save pointer state information
Bits 7, 6 - Reserved
Bit 5 = 1 - DCC extension active
Bit 4 = 1 - Palette override active
Bit 3 = 1 - Graphics font override active
Bit 2 = 1 - Alpha Font override active
Bit 1 = 1 - Dynamic save area active
Bit 0 = 1 - 512 character set active

(DI + 33h) to (DI + 3Fh) 13 bytes - Reserved

Format of static functionality table:
0 = Not supported
1 = Supported

(00h) byte - Video modes
Bit 7 = Mode 07h
Bit 6 = Mode 06h
Bit 5 = Mode 05h
Bit 4 = Mode 04h
Bit 3 = Mode 03h
Bit 2 = Mode 02h
Bit 1 = Mode 01h
Bit 0 = Mode 00h

(01h) byte - Video modes
Bit 7 = Mode 0Fh
Bit 6 = Mode 0Eh
Bit 5 = Mode 0Dh
Bit 4 = Mode 0Ch
Bit 3 = Mode 0Bh
Bit 2 = Mode 0Ah
Bit 1 = Mode 09h
Bit 0 = Mode 08h

(02h) byte - Video modes
Bit 7 to 4 - Reserved
Bit 3 = Mode 13h
Bit 2 = Mode 12h
Bit 1 = Mode 11h
Bit 0 = Mode 10h

See (AH) = 00h for video mode information

(03h) to (07h) 4 bytes - Reserved

(07h) byte - Scan lines available in text modes

Bit 7 to 3 - Reserved
Bit 2 = 400 scan lines
Bit 1 = 350 scan lines
Bit 0 = 200 scan lines

See (AH) = 12h, (BL) = 30h for text mode scan line selection.

(08h) byte - Character blocks available in text modes
(09h) byte - Maximum number of active character blocks in text modes

See (AH) = 11h for character block loading interfaces.

(0Ah) byte - Miscellaneous functions
Bit 7 = Color paging [see (AH) = 10h]
Bit 6 = Color palette [see (AH) = 10h]
Bit 5 = EGA palette [see (AH) = 10h]
Bit 4 = Cursor emulation [see (AH) = 01h]
Bit 3 = Mode set default palette loading [see (AH) = 0h]
Bit 1 = Summing [see (AH) = 10h and (AH) = 12h]
Bit 0 = All modes on all displays

(0Bh) byte - Miscellaneous functions
Bit 7 to 4 Reserved
Bit 3 = DCC [see (AH) = 1Ah]
Bit 2 = Background intensity/blinking control [see (AH) = 10h]
Bit 1 = Save/restore [see (AH) = 1Ch]
Bit 0 = Light pen [see (AH) = 04h]

(0Ch to 0Dh) 2 bytes - Reserved

(0Eh) byte - Save pointer functions
Bits 7, 6 - Reserved
Bit 5 = DCC extension
Bit 4 = Palette override
Bit 3 = Graphics font override
Bit 2 = Alpha font override
Bit 1 = Dynamic save area
Bit 0 = 512-character set

(0Fh) byte - Reserved

Int 10h - Function 1Ch - Save/Restore Video State

Entry:	AH = 1Ch	
	AL = 00h	Return size of save/restore buffer
	CX -	Requested states (see supported save/restore states below)
Exit:	AL = 1Ch	Indicates function is supported
	BX	Save/restore buffer size block count [number of 64-bytes blocks for saving requested states in (CX)]
Entry:	AL = 01h	Save video state
	CX -	Requested states (see supported save/restore states below)

	(ES:BX)	Buffer pointer to save state
Exit:	AL = 1Ch	Indicates function is supported
Entry:	AL = 2h	Restore video state
	CX -	Requested states (see supported save/restore states below)
	ES:BX	Pointer to save/restore buffer
Exit:	AL = 1Ch	Indicates function is supported

Supported save/restore states:

Bit 15 to 3 - Reserved and set to 0
Bit 2 = 1 - Save/restore video DAC state and color registers
Bit 1 = 1 - Save/restore video BIOS data area
Bit 0 = 1 - Save/restore video hardware state

This function completely saves or restores the video state to or from a buffer in RAM at the address specified by the calling program. The program should first call this function with AL = 0 to determine the necessary RAM buffer size. The current video state is altered during a Save State operation. To maintain the current video state, perform a Restore State operation after saving the video state.

10.5 BIOS DATA STRUCTURES AND TABLES

10.5.1 Video Display BIOS Data Area

The Video BIOS routines maintain several variables in the BIOS Data Area at Segment 40h. The table below provides a summary of these variables' addresses, their sizes, and their contents.

Address (Segment:Offset)	Type	Description
0040:0049	Byte	Current BIOS video mode number
0040:004A	Word	Number of displayed character columns
0040:004C	Word	Size of video buffer in bytes
0040:004E	Word	Offset of start of video buffer
0040:0050	Word	Array of eight words containing the cursor position for each of eight possible video pages. The high-order byte of each word contains the character row, the low-order byte the character column.
0040:0060	Word	Starting and ending lines for alphanumeric cursor. The high-order byte contains the starting (top) line; the low-order byte contains the ending (bottom) line.
0040:0062	Byte	Currently displayed video page number
0040:0063	Word	I/O port address of CRT Controller's Address register (3B4h for monochrome, 3D4h for color).
0040:0065	Byte	Current value for Mode Control register (3B8h on MDA, 3D8h on CGA). On the VGA, the value emulates those used on the MDA and CGA.
0040:0066	Byte	Current value for the CGA Color Select register (3D9h). On the VGA, the value emulates those used on the MDA and CGA.
0040:0084	Byte	Number of displayed character rows - 1
0040:0085	Word	Height of character matrix
0040:0087	Byte	(See description next page)
0040:0088	Byte	(See description next page)
0040:0089	Byte	Miscellaneous flags (See description next page)
0040:008A	Byte	Display Combination Code table index
0040:00A8	Dword	Pointer to BIOS Save Area (See Section 10.5.2)

Mapping of INFO byte at 0040:0087 in the BIOS Data Area.

Bit	Description
7	Preserve/Clear display bit from last Mode Set (AL Register Bit 7 passed to INT 10h function 0)
6-5	Display Memory Size: 00=64K 01=128K 10=192K 11=256K
4	(Reserved)
3	1 - video susbsystem is inactive
2	(Reserved)
1	1 - video subsystem is attached to monochrome display
0	1 - alphanumeric cursor emulation is enabled

Mapping of INFO_3 byte at 0040:0088 in the BIOS Data Area. Bits 4 through 7 contain the power-on status of the feature connector. Bits 0 through 3 contain the settings of the four "configuration switches" (VGA-compatible BIOSes emulate the switch values based on the type of display attached).

Bit	Description
7	Input from feature connector on FEAT1 (bit 6 of Input Status register 0) in response to output on FC1 (bit 1 of Feature Control register)
6	Input from feature connector on FEAT0 (bit 5 of Input Status register 0) in response to output on FC1 (bit 1 of Feature Control register)
5	Input from feature connector on FEAT1 (bit 6 of Input Status register)) in response to output on FC0 (bit 0 of Feature Control register)
4	Input from feature connector on FEAT0 (bit 5 of Input Status register 0) in response to output on FC0 (bit 0 of Feature Control register)
3	Configuration switch 4 (1 -off, 0 - on)
2	Configuration switch 3 (1 -off, 0 - on)
1	Configuration switch 2 (1 -off, 0 - on)
0	Configuration switch 1 (1 -off, 0 - on)

Mapping of Flags byte at 0040:0089 in the BIOS Data Area.

Bit	Description		
7	Alphanumeric scan lines (with bit 4):		
	bit 7	bit 4	
	0	0	350-line mode
	0	1	400-line mode
	1	0	200-line mode
	1	1	(Reserved)
6	1 - display switching is enabled 0 - display switching is disabled		
5	(Reserved)		
4	(see bit 7)		
3	1 - default palette loading disabled 0 - default palette loading enabled		
2	1 - using monochrome monitor		
1	1 - gray scale summing enabled 0 - gray scale summing disabled		
0	1 - VGA active 0 - VGA not active		

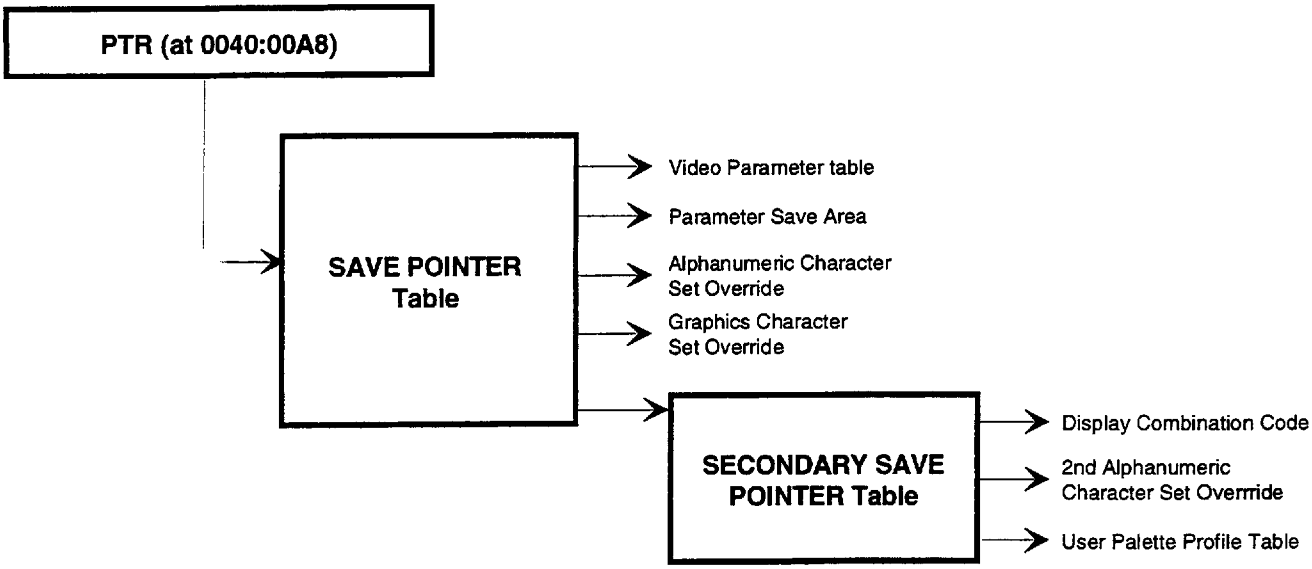
Video BIOS routines dynamically update the values in the BIOS Data Area to reflect the status of the video subsystem. Programs which directly modify the display subsystem environment without using INT 10h Video BIOS calls must update the relevant variables in the BIOS Data Area, otherwise subsequent calls to video BIOS routines will malfunction.

10.5.2 Save Areas

The Video BIOS maintains several "save areas", where video hardware and BIOS information is saved by certain BIOS routines. The video BIOS can use these save areas to supplement the BIOS Data Area. The save areas may also be used to override the video BIOS defaults for character sets, palette programming, and other configuration functions.

The video BIOS save areas are linked by a set of doubleword (segment:offset) pointers (see figure below). Use the doubleword pointer at 0040:00A8 in the BIOS Data Area to locate the save areas. This pointer contains the address of the SAVE POINTER table, which contains the addresses of as many as seven data structures, each with its own unique format and data.

The fifth address in the SAVE POINTER table is that of the SECONDARY SAVE POINTER table. This table also contains the addresses of several data structures with contents relating to the functioning of the video hardware and the BIOS.


PTR (at 0040:00A8)
SAVE POINTER Table
Video Parameter table
Parameter Save Area
Alphanumeric Character Set Override
Graphics Character Set Override
SECONDARY SAVE POINTER Table
Display Combination Code
2nd Alphanumeric Character Set Override
User Palette Profile Table

SAVE POINTER table

Offset	Type	Description
0	Dword	Address of Video Parameter table
4	Dword	Address of Parameter Save Area
8	Dword	Address of Alphanumeric Character Set Override
0Ch	Dword	Address of Graphics Charcter Set Override
10h	Dword	Address of Secondary Save Pointer table
14h	Dword	(Reserved)
18h	Dword	(Reserved)

SECONDARY SAVE POINTER table

Offset	Type	Description
0	Word	Length of Secondary Save Pointer table in bytes
2	Dword	Address of Display Combination Code table
6	Dword	Address of second Alphanumeric Character Set Override
0Ah	Dword	Address of User Palette Profile table
0Eh	Dword	(Reserved)
12h	Dword	(Reserved)
16h	Dword	(Reserved)

Aside from the SAVE POINTER and SECONDARY SAVE POINTER tables, the only data structures predefined by the Video BIOS are the Video Parameter table and the Display Combination Code table. Thus, the only initialized pointers in the SAVE POINTER table are for the Video Parameter table and the SECONDARY SAVE POINTER table. The only initialized pointer in the SECONDARY SAVE POINTER table belongs to the Display Combination Code table. All other addresses are initialized to 0.

10.5.3 Video Parameter Table

Offset	Type	Description
0	Byte	Value for CRT_COLS
1	Byte	Value for ROWS
2	Byte	Value for POINTS
3	Word	Value for CRT_LEN
5	4-byte array	Values for Sequencer registers 1-4
9	Byte	Value for Miscellaneous Output register
OAh	25-byte array	Values for CRTC registers 0-18h
23h	20-byte array	Values for Attribute Controller registers 0-13h
37h	9-byte array	Values for Graphics Controller registers 0-8

This Video Parameter Table contains configuration parameters used by the video BIOS Mode Set routines. The table contains entries for each predefined video mode.

Format of a VGA Video Parameter table entry. The VGA Video Parameter table holds 29 of these entries.

10.5.4 Parameter Save Area

Offset	Type	Description
0	16-byte array	Current contents of Graphics Controller Palette registers
10h	Byte	Current contents of Graphics Controller Overscan register
11h-0FFh	(Reserved)	

This 256-byte table contains the values of the VGA Graphics Controller palette registers (00h through 0Fh) and the Overscan register (11h). The video BIOS updates the Parameter Save Area whenever it updates the corresponding Attribute Controller registers.
Note: When a User Palette Profile overrides the default palette register values, the Parameter Save Area is updated with default values, not those in the User Palette Profile.

10.5.5 Alphanumeric Character Set Override

Offset	Type	Description
0	Byte	Length of each character definition in bytes
1	Byte	Character generator RAM bank
2	Word	Number of characters defined
4	Word	First character code in table
6	Dword	Address of character definition table
0Ah	Byte	Number of displayed character rows
0Bh	Byte array	Applicable video modes
	Byte	0FFh (end of list of video modes)

This data structure specifies an alphanumeric character set which replaces the BIOS default character set. The character set is loaded when Mode Set is called to set one of the video modes that the data structure specifies.

A second Alphanumeric Character Set Override data structure can be used to specify another 256-character set by storing its address in the SECONDARY SAVE POINTER table.

10.5.6 Graphics Character Set Override

Offset	Type	Description
0	Byte	Number of displayed character rows
1	Word	Length in bytes of each character definition
3	Dword	Address of character definition table
7	Byte array	Applicable video modes
	Byte	0FFh (end of list of video modes)

This data structure overrides the default BIOS character set selection whenever Mode Set is called to set one of the specified video modes.

10.5.7 Display Combination Code Table

Offset	Type	Description
0	Byte	Numb of entries in table
1	Byte	DCC table version number
2	Byte	Maximum display type code
3	Byte	(reserved)
4	Word array	Each pair of bytes in the array describes a valid display combination (see INT 10h function 1Ah)

The figure below lists all combinations of video subsystems that the video BIOS supports. See the description of INT 10h function 1Ah earlier in this chapter.

10.5.8 User Palett 'rofile Table

Offset	Type	Description
0	Byte	Underlining: 1 - Enable in all alphanumeric modes 0 - Enable in monochrome alphanumeric mode 1 - Disable in all alphanumeric modes
1	Byte	(Reserved)
2	Word	(Reserved)
4	Word	Number of Attribute Controller registers in table
6	Word	First Attribute Controller register number
8	Dword	Address of Attribute Controller register table
0Ch	Word	Number of video DAC Color registers in table
0Eh	Word	First video DAC Color register number
10h	Dword	Address of video DAC Color register table
14h	Byte array Byte	Applicable video modes 0FFh (End of list of video modes)

This data structure contains user-specified overrides for the default Attribute Controller Palette and Overscan register values, for the default values in the 256 video DAC color registers, and for the default value in the CRTC Underline Location register.

10.6 BIOS INTERRUPT VECTORS

05h - Print Screen (Location = 0:0014h)

The Alternate Select BIOS function can set the print screen vector so that it points to a routine that handles nonstandard rows and columns.

10h - Functions (Location = 0:0040h)

BIOS functions are accessed via this vector. Programs place a function code in AH and other calling parameters, if required, in other registers then execute an INT 10 instruction. When BIOS gains control, the appropriate code is executed to perform the function; return parameter values are left in processor registers on return to the calling program.

The functions supported by the OTI-64107 VGA BIOS allow the calling program to set the current mode, manipulate the cursor, place characters and individual pixels on the display screen, scroll the screen, load character fonts and color palette values, and read the light pen position. These functions are described in previous sections.

Functions 0h-0Fh are supported by the PC system BIOS. If a VGA board is present in the system, its BIOS takes over these functions from the system. Functions 10h-0FFh are only available to programs if the OTI-64107 VGA board is present in the system.

42h - Reserved (Location = 0:0108h)

When a VGA is installed, BIOS routines use INT42 to re-vector the standard INT 10 video pointer. (Which is the original motherboard INT 10 vector.)

43h - Graphics Character Table (Location = 0:010Ch)

BIOS routines use this vector to point to a table of dot patterns that are used when graphics characters are displayed. This table is used for the first 128 characters in video modes 4, 5, and 6. This table is also used for 256 characters in all additional graphics modes (0Dh, 0Eh, 0Fh, 10h, 11h, 12h and 13h).

1D - CRT Controller Parameter Table (Location = 0:0074h)

This is used as a pointer to the CRT controller parameters as used by the CGA. This vector is used for emulation only.

1F - Upper 128 Characters (Location = 0:007Ch)

This table is used for the upper 128 characters in modes 4, 5, and 6.

10.7 VESA SUPER VGA STANDARD

10.7.1 Introduction

This section contains the VESA, Video Electronics Standards Association, specification for a standardized interface to extended VGA video modes and functions. The specification consists of mechanisms for supporting standard extended video modes and functions that have been approved by the main VESA committee and nonstandard video modes that an individual VGA supplier may choose to add, in a uniform manner that application software can utilize without having to understand the intricate details of the particular VGA hardware.

The primary topics of this specification are definitions of extended VGA video modes and the functions necessary for application software to understand the characteristics of the video mode and manipulate the extended memory associated with the video modes.

Readers of this document should already be familiar with programming VGAs at the hardware level and Intel iAPX real mode assembly language. Readers who are unfamiliar with programming the VGA should first read one of the many VGA programming tutorials before attempting to understand these extensions to the standard VGA.

10.7.2 Goals and Objectives

The IBM VGA has become a defacto standard in the PC graphics world. A multitude of different VGA offerings exist in the marketplace, each one providing BIOS or register compatibility with the IBM VGA. More and more of these VGA compatible products implements various supersets of the VGA standards. These extensions range from higher resolutions and more colors to improved performance and even some graphics processing capabilities. Intense competition has dramatically improved the price/performance ratio, to the benefit of the end user.

However, several serious problems face a software developer who intends to take advantage of these "Super VGA"2 environments. Because there is no standard hardware implementation, the developer is faced with widely disparate Super VGA hardware architectures. Lacking a common software interface, designing applications for these environments is costly and technically difficult. Except for applications supported by OEM-specific display drivers, very few software packages can take advantage of the power and capabilities of Super VGA products.

The purpose of the VESA VGA BIOS Extension is to remedy this situation. Being a common software interface to Super VGA graphics products, the primary objective is to enable application and system software to adapt to and exploit the wide range of features available in these VGA extensions.

Specifically, the VESA BIOS Extension attempts to address the following two main issues: a) Return information about the video environment to the application and b) Assist the application in initializing and programming the hardware.

Video environment information

Today, an application has no standard mechanism to determine what Super VGA hardware it is running on. Only by knowing OEM-specific features can an application determine the presence of a particular video board. This often involves reading and testing registers located at I/O addresses unique to each OEM. By not knowing what hardware an application is running on, few, if any, of the extended features of the underlying hardware can be used.

The VESA BIOS Extension provides several functions to return information about the video environment. These functions return system level information as well as video mode specific details. Function 00h returns general system level information, including an OEM identification string. The function also returns a pointer to the supported video modes. Function 01h may be used by the application to obtain information about each supported video mode. Function 03h returns the current video mode.

Programming support

Due to the fact that different Super VGA products have different hardware implementations, application software has great difficulty in adapting to each environment. However, since each is based on the VGA hardware architecture, differences are most common in video mode initialization and memory mapping. The rest of the architecture is usually kept intact, including I/O mapped registers, video buffer location in the CPU address space, DAC location and function, etc.

The VESA BIOS Extension provides several functions to interface to the different Super VGA hardware implementations. The most important of these is Function 02h, Set Super VGA video mode. This function isolates the application from the tedious and complicated task of setting up a video mode. Function 05h provides an interface to the underlying memory mapping hardware. Function 04h enables an application to save and restore a Super VGA state without knowing anything of the specific implementation.

Compatibility

A primary design objective of the VESA BIOS Extension is to preserve maximum compatibility to the standard VGA environment. In no way should the BIOS extensions compromise compatibility or performance. Another but related concern is to minimize the changes necessary to an existing VGA BIOS. RAM, as well as ROM-based implementations of the BIOS extension should be possible.

Scope of standard

The purpose of the VESA BIOS Extension is to provide support for extended VGA environments. Thus, the underlying hardware architecture is assumed to be a VGA. Graphics software that drives a Super VGA board, will perform its graphics output in generally the same way it drives a standard VGA, i.e., writing directly to a VGA style frame buffer, manipulating graphics controller registers, directly programming the palette etc. No significant graphics processing will be done in hardware. For this reason, the VESA BIOS Extension does not provide any graphics output functions, such as Bit, line or circle drawing, etc.

An important constraint of the functionalities that can be placed into the VESA BIOS Extension, is that ROM space is severely limited in certain existing BIOS implementations.

Outside the scope of this VESA BIOS Extension is handling of different monitors and monitor timings. Such items are dealt with in other VESA fora. The purpose of the VESA BIOS Extension is to provide a standardized software interface to Super VGA graphics modes, independent of monitor and monitor timing issues.

10.7.3 Standard VGA BIOS

A primary design goal with the VESA BIOS extension is to minimize the effects on the standard VGA BIOS. Standard VGA BIOS functions should need to be modified as little as possible. This is important since ROM, as well as RAM based versions of the extension may be implemented.

However, two standard VGA BIOS functions are affected by the VESA extension. These are Function 00h (Set video mode) and Function 0Fh (Read current video state). VESA-aware applications will not set the video mode using VGA BIOS function 00h. Nor will such applications use VGA BIOS function 0Fh. VESA BIOS functions 02h (Set Super VGA mode) and 03h (Get Super VGA mode) will be used instead.

However, VESA-unaware applications (such as old Pop-Up programs and other TSRs, or the CLS command of MS-DOS), might use VGA BIOS function 0Fh to get the present video mode. Later it may call VGA BIOS function 090h to restore/reinitialize the old video mode.

To make such applications work, VESA recommends that whatever value returned by VGA BIOS function OFh (it is up to the OEM to define this number), it can be used to reinitialize the video mode through VGA BIOS function 00h. Thus, the BIOS should keep track of the last Super VGA mode in effect.
It is recommended, but not mandatory, to support output functions (such as TTY-output, scroll, set pixel, etc.) in Super VGA modes. If the BIOS extension doesn't support such output functions, bit D2 (Output functions supported) of the Mode Attributes field (returned by VESA BIOS function 01h) should be cleared.

10.7.4 Super VGA Mode Numbers

Standard VGA mode numbers are seven bits wide and presently range from 00h to 13h. OEMs have defined extended video modes the range 14h to 7Fh. Values in the range 80h to FFh cannot be used, since VGA BIOS function 00h (Set video mode) interprets Bit 7 as a flag to clear/not clear video memory.

Due to the limitations of 7-bit mode numbers, VESA video mode numbers are 15 bits wide. To initialize a Super VGA mode, its number is passed in the BX register to VESA BIOS function 02h (Set Super VGA mode).

The format of VESA mode numbers is as follows:

D0-D8=	Mode number If D8==0, this is not a VESA defined mode If D8==1,this is a VESA defined mode
D9-D14=	Reserved by VESA for future expansion (=0)
D15=	Reserved (=0)

Thus, VESA mode numbers begin at 100h. This mode numbering scheme implements standard VGA mode numbers as well as OEM-defined mode numbers as subsets of the VESA mode number. That means that regular VGA modes may be initialized through VESA BIOS function 02h (Set Super VGA mode), simply by placing the mode number in BL and clearing the upper byte (Bh).

10.7.5 CPU Video Memory Windows

A standard VGA subsystem provides 256 K bytes of memory and a corresponding mechanism to address this memory. Super VGAs and their extended modes require more than the standard 256 K bytes of memory but also require that the address space for this memory be restricted to the standard address space for compatibility reasons. CPU video memory windows provide a means of accessing this extended VGA memory within the standard CPU address space.

This chapter describes how several hardware implementations of CPU video memory windows operate, their impact on application software design, and relates them to the software model presented by the VESA VGA BIOS extensions.

The VESA CPU video memory windows functions have been designed to put the performance insensitive, nonstandard hardware functions into the BIOS while putting the performance sensitive, standard hardware functions into the application. This provides portability among VGA systems together with the performance that comes from accessing the hardware directly. In particular, the VESA BIOS is responsible for mapping video memory into the CPU address space while the application is responsible for performing the actual memory read and write operations.

This combination software and hardware interface is accomplished by informing the application of the parameters that control the hardware mechanism of mapping the video memory into the CPU address space and then letting the application control the mapping within those parameters.

Hardware design considerations

Limited to 64 K/128 K of CPU address space

The first consideration in implementing extended video memory is to give access to the memory to application software.

The standard VGA CPU address space for 16 color graphics modes is typically at segment A000h for 64 K. This gives access to the 256 K bytes of a standard VGA, i.e. 64 K per plane. Access to the extended video memory is accomplished by mapping portions of the video memory into the standard VGA CPU address space.
Every super VGA hardware implementation provides a mechanism for software to specify the offset from the start of video memory which is to be mapped to the start of the CPU address space. Providing both read and write access to the mapped memory provides a necessary level of hardware support for an application to manipulate the extended video memory.

Crossing CPU video memory window boundaries

The organization of most software algorithms which perform video operations consists of a pair of nested loops: an outer loop over rows or scan lines and an inner loop across the row or scan lines. The latter is the proverbial inner loop, which is the bottle neck to high performance software.
If a target rectangle is large enough, or poorly located, part of the required memory may be within the video memory mapped into the CPU address space and part of it may not be addressable by the CPU without changing the mapping. It is desirable that the test for remapping the video memory is located outside of the inner loop.

This is typically accomplished by selecting the mapping offset of the start of video memory to the start of the CPU address space so that at least one entire row or scan line can be processed without changing the video memory mapping. There are currently no super VGAs that allow this offset to be specified on a byte boundary and there is a wide range among super VGAs in the ability to position a desired video memory location at the start of the CPU address space.

The number of bytes between the closest two bytes in video memory that can be placed on any single CPU address is defined as the granularity of the window mapping function. Some super VGA systems allow any 4 K video memory boundary to be mapped to the start of the CPU address space, while other super VGA systems allow any 64 K video memory boundary to be mapped to the start of the CPU address space. These two example systems would have granularities of 4 K and 64 K respectively. This concept is very similar to the bytes that are accessed with a 16-bit pointer in an Intel CPU before a segment register must be changed (the granularity of the segment register or mapping, here is 16 bytes).
Note that if the granularity is equal to the length of the CPU address space, i.e., the least significant address bit of the hardware mapping function is more significant than the most significant bit of the CPU address, then the inner loop will have to contain the test for crossing the end or beginning of the CPU address space. This is because if the length of the CPU address space (which is the granularity in this case) is not evenly divisible by the length of a scan line, then the scan line at the end of the CPU address will be in two different video memory which cannot be mapped into the CPU address space simultaneously.

Operating on data from different areas

It is sometimes required or convenient to move or combine data from two different areas of video memory. One example of this is strong menus in the video memory beyond the displayed memory because there is hardware support in all VGAs for transferring 32 bits of video data with an 8 bit CPU read and write. Two separately mappable CPU video memory windows must be used if the distance between the source and destination is larger than the size of the CPU video memory window.

Combining data from two different windows

The above example of moving data from one CPU video memory window to another CPU video memory only required read access to one window and only required write access to the other window. Sometimes it is convenient to have read access to both windows and write access to one window and only required write access to the other window. Sometimes it is convenient to have read access to both windows and write access to one window. An example of this would be a raster operation where the resulting destination is the source data logically combined with the original destination data.

Different types of hardware windows

Different hardware implementations of CPU video memory windows can be supported by the VESA BIOS extension. The information necessary for an application to understand the type of hardware implementation is provided by the BIOS to the application. There are three basic types of hardware windowing implementations and they are described below.

The types of windowing schemes described below do not include differences in granularity.

Also note that is possible for a VGA to use a CPU address space of 128 K starting at segment A000h.

Single window systems

Some hardware implementations only provide a single window. This single window will be readable as well as writable. However, this causes a significant performance degradation when moving data in video memory a distance that is larger than the CPU address space.

Dual window systems

Many super VGAs provide two windows to facilitate moving data within video memory. There are two separate methods of providing two windows.

Overlapping windows

Some hardware implementations distinguish window A and window B is by looking at the CPU address within the total VGA CPU address space. When the two windows are distinguished by the CPU address within the VGA CPU address space the windows cannot share all the same address space, but they can each be both read and written.

10.7.6 Extended VGA BIOS

Several new BIOS calls have been defined to support Super VGA modes. For maximum compatibility with the standard VGA BIOS, these calls are grouped under one function number. This number is passed in the AH register to the Int 10h handler.

The designated Super VGA extended function number is 4Fh. This function number is presently unused in most, if not all, VGA BIOS implementations. A standard VGA BIOS performs no action when function call 4Fh is made. Super VGA standard VS900602 defines sub-functions 00h through 07h. Sub-function numbers 08h through 0FFh are reserved for future use.

Status information

Every function returns status information in the AX register. The format of the status word is as follows:

AL == 4Fh:	Function is supported
AL != 4Fh:	Function is not supported
AH == 00h:	Function call successful
AH == 01h:	Function call failed

Software should treat a nonzero value in the Ah register as a general failure condition. In later versions of the VESA BIOS Extension new error codes might be defined.

Function 00h - Return Super VGA information

The purpose of this function is to provide information to the calling program about the general capabilities of the Super VGA environment. The function fills an information block structure at the address specified by the caller. The information block size is 256 bytes.

Input:	AH=4Fh	Super VGA support
	AL=00h	Return Super VGA information
	ES:DI= Pointer to buffer	
Output:	AX=Status	
	All other registers are preserved	

The information block has the following structure:

```
VgaInfoBlock struc
        VESASignature   db      'VESA'          ;4 signature bytes
        VESAVersion     dw      ?               ;VESA version number
        OEMStringPtr    dd      ?               ;Pointer to OEM string
        Capabilities    db      4 dup (?)       ;capabilities of the video environment
        VideoModePtr    dd      ?               ;pointer to supported Super VGA modes
        TotalMemory     dw      ?               ;Number of 64-KB memory blocks on board
        Reserved        db      236 dup (?)     ;Remainder of VgaInfoBlock
VgaInfoBlock ends
```

The VESASignature field contains the characters 'VESA' if this is a valid block.

The VESAVersion is a binary field which specifies what level of the VESA standard the Super VGA BIOS conforms to. The higher byte specifies the major version number. The lower byte specifies the minor version number. The current VESA version number is 1.2 Applications written to use the features of a specific version of the VESA BIOS Extension, are guaranteed to work in later versions. The VESA BIOS Extension will be fully upwards compatible. The OEMStringPtr is a far pointer to a null terminated OEM-defined string. The string may used to identify the video chip, video board, memory configuration etc.., to hardware specific display drivers. There are not restrictions on the format of the string.

The Capabilities field describes what general features are supported in the video environment. The bits are defined as follows:

D0 =DAC is switchable
0 = DAC is fixed width, with six bits per primary color
1 = DAC width is switchable
D1-31 =Reserved

The VideoModePtr points to a list of supported Super VGA (VESA-defined as well as OEM-specific) mode numbers. Each mode number occupies one word (16 bits). The list of mode numbers is terminated by a -1 (0FFFFh). Please refer to chapter two for a description of VESA mode numbers. The pointer could point into either ROM or RAM, depending on the specific implementation. Either the list would be a static string stored in ROM, or the list would be generated at run-time in the information block (see above) in RAM. It is the applications responsibility to verify the current availability of any mode returned by this Function through the Return Super VGA mode information (Function 1) call. Some of the returned modes may not be available due to the video boards current memory and monitor configuration.

The TotalMemory field indicates the amount of memory installed on the VGA board. Its value represents the number of 64kb blocks of memory currently installed.

Function 01h - Return Super VGA mode information

This function returns information about a specific Super VGA video mode that was returned by Function 0. The function fills a mode information block structure at the address specified by the caller. The mode information block size is maximum 256 bytes.

Some information provided by this function is implicitedly defined by the VESA mode number. However, some Super VGA implementations might support other video modes than those defined by VESA. To provide access to these modes, this function also returns various other information about the mode.

Input:	AH = 4Fh	Super VGA support
	AL = 01h	Return Super VGA mode information
	CX =Super VGA video mode[1]	
	ES:DI = Pointer to 256-byte buffer	
Output:	AX = Status	
	All other registers are preserved	

Note: 1. The mode number must be one of those returned by Function 0

The mode information block has the following structure:

```
ModeInfoBlock   struc

; mandatory information

        ModeAtrributes          dw      ?       ; mode attributes
        WinAAttributes          db      ?       ; window A attributes
        WinBAttributes          db      ?       ; window B attributes
        WinGranularity          dw      ?       ; window granularity
        WinSize                 dw      ?       ; window size
        WinASegment             dw      ?       ; window A start segment
        WinBSegment             dw      ?       ; window B start segment
        WinFuncPtr              dd      ?       ; pointer to window function
        BytesPerScanLine        dw      ?       ; bytes per scan line

; formerly optional information (now mandatory)

        XResolution             dw      ?       ; horizontal resolution
        YResolution             dw      ?       ; vertical resolution
        XCharSize               db      ?       ; character cell width
        YCharSize               db      ?       ; character cell height
        NumberOfPlanes          db      ?       ; number of memory planes
        BitsPerPixel            db      ?       ; bits per pixel
        NumberOfBanks           db      ?       ; number of banks
        MemoryModel             db      ?       ; Memory model type
        BankSize                db      ?       ; bank size in kb
        NumberOfImagePages      db      ?       ; Number of Images
        Reserved                db      ?       ; reserved or page function

; New Direct Color fields

        RedMaskSize             db      ?       ;size of direct color red mask in bits
        RedFieldPosition        db      ?       ;bit position of lsb of red mask
        GreenMaskSize           db      ?       ;size of direct color green mask in bits
        GreenFieldPosition      db      ?       ;bit position of lsb of green mask
        BlueMaskSize            db      ?       ;size of direct color blue mask in bits
        BlueFieldPosition       db      ?       ;bit position of lsb of blue mask
        RsvdMaskSize            db      ?       ;size of direct color reserved mask in bits
        RsvdFieldPosition       db      ?       ;bit position of lsb of reserved mask
        DirectColorModeInfo     db      ?       ;Direct Color mode attributes
        Reserved                db      216 dup (?) ;remainder of ModeInforBlock
ModeInfoBlock   ends
```

The ModeAttributes field describes certain important characteristics of the video mode. Bit D0 specifies whether this mode can be initialized in the present video configuration. This bit can be used to block access to a video mode if it requires a certain monitor type, and that this monitor is presently not connected. Prior to Version 1.2 of the VESA BIOS Extension, it was not required that the BIOS return valid information for the fields after BytesPerScanline. Bit D1 was used to signify if the optional information was present. Version 1.2 of the VBE requires that all fields of the ModelnfoBlock contain valid data, except for the Direct Color Fields, which are valid only if the MemoryModel field is set to a 6 (Direct Color) or 7 (YUV). Bit D1 is now reserved, and must be set to a 1. Bit D2 indicates whether the BIOS has support for output functions like TTY output, scroll, pixel output etc. in this mode (it is recommended, but not mandatory, that the BIOS have support for all output function). If bit D2 is 1 then the BIOS must support all of the standard output function.

The field is defined as follows:

D0 = Mode supported in hardware
    0 = Mode not supported in hardware
    1 = Mode supported in hardware

D1 = 1 (Reserved)

D2 = Output functions supported by BIOS
    0 = Output functions not supported by BIOS
    1 = Output functions supported by BIOS
D3 = Monochrome/color mode (see note below)
    0 = Monochrome mode
    1 = Color mode
D4 = Mode type
    0 = Text mode
    1 = Graphics mode
D5-D15= Reserved

Note: Monochrome modes have their CRTC address at 3B4h. Color modes have their CRTC address at 3D4h. Monochrome modes have attributes in which only bit 3 (video) and bit 4 (intensity) of the attribute controller output are significant. Therefore, monochrome text modes have attributes of off, video, high intensity, blink, etc. Monochrome graphics modes are two plane graphics modes and have attributes of off, video, high intensity, and blink. Extended two color modes that have their CRTC address at 3D4h, are color modes with one bit per pixel and one plane. The standard VGA modes, 06h and 11h would be classified as color modes, while the standard VGA modes 07h and 0Fh would be classified as monochrome modes.

The BytesPerScanline field specifies how many bytes each logical scanline consists of. The logical scanline could be equal to or larger than the displayed scanline.
The WinAAttributes and WnBAttributes describe the characteristics of the CPU windowing scheme such as whether the windows exist and are read/writeable, as follows:

D0 = Window supported
    0= Window is not supported
    1= Window is supported
D1 = Window readable
    0= Window is not readable
    1= Window is readable
D2 = Window writable
    0= Window is not writeable
    1= Window is writeable
D3-D7 = Reserved

If windowing is not supported, (bit D0 = 0 for both Window A and Window B), then an application can assume that the display memory buffer resides at the standard CPU address appropriate for the MemoryModel of the mode.

WinGranularity specifies the smallest boundary, in KB, on which the window can be placed in the video memory. The value of this field is undefined if bit D0 of the appropriate WinAttributes field is not set.

WinSize specifies the size of the window in KB.

WinASegment and WinBSegment address specify the segment addresses where the windows are located in the CPU address space.

WinFuncAddr specifies the address of the CPU video memory windowing function. The windowing function can be invoked either through VESA BIOS function 05h, or by calling the function directly. A direct call will provide faster access to the hardware paging registers than using Int 10h, and is intended to be used by high performance applications. If this field is Null, then Function 05h must be used to set the memory window, if paging is supported.

The XResolution and YResolution specify the width and height of the video mode. In graphics modes, this resolution is in units of pixels. In text modes this resolution is in units of characters. Note that textmode resolutions, in units of pixels, can be obtained by multiplying XResolution and YResolution by the cell width and height, if the extended information is present.

The XCharCellSize and YCharCellSize specify the size of the character cell in pixels.

The NumberOfPlanes field specifies the number of memory planes available to software in that mode. For standard 16-color VGA graphics, this would be set to 4. For standard packed pixel modes, the field would be set to 1.

The BitsPerPixel field specifies the total number of bits that define the color of one pixel. For example, a standard VGA 4 Plane 16-color graphics mode would have a 4 in this field and a packed pixel 256-color graphics mode would specify 8 in this field. The number of bits per pixel per plane can normally by derived by dividing the BitsPerPixel field by the NumberOfPlanes field.

The MemoryModel field specifies the general type of memory organization used in this mode. The following models have been defined:

00h=	Text mode
01h=	CGA graphics
02h=	Hercules graphics
03h=	4-plane planar
04h=	Packed pixel
05h=	Non-chain 4, 256 color
06h=	Direct Color
07h=	YUV
08h-0Fh=	Reserved, to be defined by VESA
10h-0FFh	To be defined by OEM

In Version 1.1 and earlier of the VESA Super VGA BIOS Extension, OEM defined Direct Color video modes with pixel formats 1:5:5:5, 8:8:8, and 8:8:8:8 were described as a Packed Pixel model with 16, 24, and 32 bits per pixel, respectively. In Version 1.2 and later of the VESA Super VGA BIOS Extension, it is recommended that Direct Color modes use the Direct Color MemoryModel and use the MaskSize and FieldPosition fields of the ModelnfoBlock to describe the pixel format. BitsPerPixel is always defined to be the total memory size of the pixel, in bits.

NumberOfBanks. This is the number of banks in which the scan lines are grouped. The remainder from dividing the scan line number by the number of banks is the bank that contains the scan line and the quotient is the scan line number within the bank. For example, CGA graphics modes have two banks and Hercules graphics mode has four banks. For modes that don't have scanline banks (such as VGA modes 0Dh-13h), this field should be set to 1.

The BankSize field specifies the size of a bank (group of scan lines) in units of 1 KB. For CGA and Hercules graphics modes this is 8, as each bank is 8192 bytes in length. for modes that don't have scanline banks (such as VGA modes 0Dh-13h), this field should be set to 0.

The NumberOflmagePages field specifies the number of additional complete display images that will fit into the VGA's memory, at one time, in this mode. The application may load more than one image into the VGA's memory if this field is nonzero, and flip the display between.

The Reserved field has been defined to support a future VESA BIOS extension feature and will always be set to one in this version.

The RedMaskSize, GreenMaskSize, BlueMaskSize fields define the size, in bits, of the red, green, and blue components of a direct color pixel. A bit mask can be constructed from the MaskSize fields using simple shift arithmetic. For example, the MaskSize values for a Direct Color 5:6:5 mode would be 5,6,5, and 0, for the red, green, blue, and reserved fields, respectively. Note that in the YUV MemoryModel, the red field is used for V, the green field is used for Y, and the blue field is used for U. The MaskSize fields should be set to 0 in modes using a MemoryModel that does not have pixels with component fields.

The RedFieldPosition, GreenFieldPosition, BlueFieldPosition, and RsvdFieldPosition fields define the bit position within the direct color pixel or YUV pixel of the least significant bit of the respective color component. A color value can be aligned with its pixel field by shifting the value left by the FieldPosition. For example, the FieldPosition values for a Direct Color 5:6:5 mode would be 11,5, 0 and 0, for the red, green, blue, and reserved fields, respectively. Note that in the YUV MemoryModel, the red field is used for V, the green field is used for Y, and the blue field is used for U. The FieldPosition fields should be set to 0 in modes using a MemoryModel that does not have pixels with component fields.

The DirectColorModelnfo field describes important characteristics of direct color modes. Bit D0 specifies whether the color ramp of the DAC is fixed or programmable. If the color ramp is fixed, then it can not be changed. If the color ramp is programmable, it is assumed that the red, green, and blue lookup tables can be loaded using a standard VGA DAC color registers BIOS call (AX=1012h). Bit D1 specifies whether the bits in the Rsvd field of the direct color pixel can be used by the application or are reserved, and thus unusable.

D0= Color ramp is fixed/programmable
 0= Color ramp is fixed
 1= Color ramp is programmable

D1= Bits in Rsvd field are usable/reserved
 0= Bits in Rsvd field are reserved
 1= Bits in Rsvd field are usable by the application

Notes:
Version 1.1 and later VESA BIOS extensions will zero out all unused fields in the Mode Information Block, always returning exactly 256 bytes. This facilitates upward compatibility with future versions of the standard, as any newly added fields will be designed such that values of zero will indicate nominal defaults or non-implementation of optional features. (For example, a field containing a bit-mask of extended capabilities would reflect the absence of all such capabilities.) Applications that wish to be backwards compatible to Version 1.0 VESA BIOS extensions should pre-initialize the 256 byte buffer before calling Return Super VGA mode information.

Function 02h - Set Super VGA video mode

This function initializes a video mode. The BX register contains the mode to set. The format of VESA mode numbers is described in Chapter Two. If the mode cannot be set, the BIOS should leave the video environment unchanged and return a failure error code.

Input:	AH = 4Fh	Super VGA support
	AL = 02h	Set Super VGA video mode
	BX =Video mode	
	D0-D14= Video mode	
	D15= Clear memory flag	
		0= Clear video memory
		1= Don't clear video memory
Output:	AX=Status	
	All other registers are preserved	

Function 03h - Return current video mode

This function returns the current video mode in BX. The format of VESA video mode numbers is described in chapter 2 of this document.

Input:	AH = 4Fh	Super VGA support
	AL = 03h	Return current video mode
Output:	AX=Status	
	BX=Current video mode	
	All other registers are preserved	

Note: In a standard VGA BIOS, function 0Fh (Read current video state) returns the current video mode in the AL register. In D7 of AL, it also returns the status of the memory clear bit (D7 of 40:87). This bit is set if the mode was set without clearing memory. In this Super VGA function, the memory clear bit will not be returned in BX since the purpose of the function is to return the video mode only. If an application wants to obtain the memory clear bit, it should call VGA BIOS function Fh.

Function 04h - Save/Restore Super VGA video state

These functions provide a mechanism to save and restore the Super VGA video state. The functions are a superset of the three sub-functions under standard VGA BIOS function 1Ch (Save/restore video state). The complete Super VGA video state (except video memory) should be saveable/restorable by setting the requested states mask (in the CX register) to 000Fh.

Input:	AH= 4Fh	Super VGA support
	AL= 04h	Save/Restore Super VGA video state
	DL= 00h	Return save/restore state buffer size
	CX= Requested states	
	D0= Save/restore video hardware state	
	D1= Save/restore video BIOS data state	
	D2= Save/restore video DAC state	
	D3= Save/restore Super VGA state	
Output:	AX=Status	
	BX= Number of 64-byte blocks to hold the state buffer	
	All other registers are preserved	
Input:	AX= 4Fh	Super VGA support
	AL= 04h	Save/Restore Super VGA video state
	DL= 01h	Save Super VGA video state
	CX= Requested states (see above)	
	ES:BX= Pointer to buffer	
Output:	AX=Status	
	All other registers are preserved	
Input:	AH= 4Fh	Super VGA support
	AL= 04h	Save/Restore Super VGA video state
	DL= 02h	Restore Super VGA video state
	CX= Requested states (see above)	
	ES:BX= Pointer to buffer	
Output:	AX=Status	
	All other registers are preserved	

Note: Due to the goal of complete compatibility with the VGA environment, the standard VGA BIOS function 1Ch (Save/Restore VGA state) has not been extended to save the Super VGA video state. VGA BIOS compatibility requires that function 1Ch returns a specific buffer size with specific contents, in which there is no room for the Super VGA state.

Function 05h - CPU Video Memory Window Control

This function sets or gets the position of the specified window in the video memory. The function allows direct access to the hardware paging registers. To use this function properly, the software should use VESA BIOS Function 01h (Return Super VGA mode information) to determine the size, location and granularity of the windows.

Input:	AH= 4Fh	Super VGA support
	AL= 05h	Super VGA video memory window control
	BH= 00h	Select super VGA video memory window
	BL= Window number	
		0= Window A
		1= Window B
	DX= Window position in video memory (in window granularity units)	

Output:	AX=	Status
	See notes below	
Input:	AH= 4Fh	Super VGA support
	AL= 05h	Super VGA video memory window control
	BH= 01h	Return super VGA video memory window
	BL= Window number	
		0= Window A
		1= Window B
Output:	AX=Status	
	DX= Window position in video memory (in window granularity units)	
	See notes below	

Notes: This function is also directly accessible through a far call from the application. The address of the BIOS function may be obtained by using VESA BIOS Function 01h, return Super VGA mode information. A field in the ModeInfoBlock contains the address of this function. Note that this function may be different among video modes in a particular BIOS implementation so the function pointer should be obtained after each set mode.

In the far call version, no status information is returned to the application. Also, in the far call version, the AX and DX registers will be destroyed. Therefore if AX and/or DX must be preserved, the application must do so prior to making the far call.

The application must load the input arguments in BH, BL, and DX (for set window) but does not need to load either AH or AL in order to use the far call version of this function.

Function 06h - Set/Get Logical Scan Line Length

This function sets or gets the length of a logical scan line. This function allows an application to set up a logical video memory buffer that is wider than the displayed area. Function 07h then allows the application to set the starting position that is to be displayed.

Input:	AH= 4Fh	Super VGA Support
	AL = 06h	Logical Scan Line Length
	BL = 00h	Select Scan Line Length
	CX =	Desired Width in Pixels
Output:	AX =	Status
	BX =	Bytes Per Scan Line
	CX =	Actual Pixels Per Scan Line
	DX =	Maximum Number of Scan Lines
Input:	AH = 4Fh	Super VGA Support
	AL = 06h	Logical Scan Line Length
	BL = 01h	Return Scan Line Length
Output:	AX =	Status
	BX =	Bytes Per Scan Line
	CX =	Actual Pixels Per Scan Line
	DX =	Maximum Number of Scan Lines

Note: The desired width in pixels may not be achievable because of VGA hardware considerations. The next larger value will be selected that will accommodate the desired number of pixels, and the actual number of pixels will be returned in CX, BX returns a value that when added to a pointer into video memory will point to the next scan line. For example, in a mode 13h this would be 320, but in mode 12h this would be 80. DX returns the number of logical scan lines based upon the new scan line length and the total memory installed and usable in this display mode. This function is also valid in text modes. In text modes the application should find out the current character cell width through normal BIOS functions, multiply that times the desired number of character per line, and pass that value in the CX register.

Function 07h - Set/Get Display Start

This function selects the pixel to be displayed in the upper left corner of the display from the logical page. This function can be used to pan and scroll around logical screens that are larger than the displayed screen. This function can also be used to rapidly switch between two different displayed screens for double buffered animation effects.

Input:	AH = 4Fh	Supper VGA Support
	AL = 07h	Display Start Control
	BH = 00h	Reserved and must be 0
	BL = 00h	Select Display Start
	CX =	First Displayed Pixel In Scan Line
	DX =	First Displayed Scan Line
Output:	AX =	Status

Input:	AH = 4Fh	Super VGA Support
	AL = 07h	Display Start Control
	BL = 01h	Return Display Start
Output:	AX =	Status
	BH =	00h Reserved and will be 0
	CX =	First Displayed Pixel In Scan Line
	DX =	First Displayed Scan Line

Note: This function is also valid in text modes. In text modes the application should find out the current character cell width through normal BIOS functions, multiply that times the desired starting character column, and pass that value in the CX register. It should also multiply the current character cell height times the desired starting character row, and pass that value in the DX register.

Function 08h - Set/Get DAC Palette Control

This function queries and selects the operating mode of the DAC palette. Some DACs are configurable to provide 6-bits, 8-bits, or more of color definition per red, green, and blue primary color. The DAC palette width is assumed to be reset to standard VGA 6-bits per primary during a standard or VESA Set Super VGA Mode (AX=4F02h) call.

Input:	AH = 4Fh	Super VGA Support
	AL = 08h	Set/Get DAC Palette Control
	BL = 00h	Set DAC Palette Width
	BH =	Desired number of bits of color per primary (Standard VGA = 6)
Output:	AX =	Status

	BH =	Current number of bits of color per primary (Standard VGA = 6)
Input:	AH = 4Fh	Super VGA Support
	AL = 08h	Set/Get DAC Palette Control
	BL = 01h	Get DAC Palette Width
Output:	AX =	Status
	BH =	Current number of bits of color per primary (Standard VGA = 6)

An application can find out if DAC switching is available by querying bit D0 of the Capabilities field of the VgaInfoBlock structure returned by VESA Return Super VGA Information (AX=4F00h). The application can then attempt to set the DAC palette width to the desired value. If the Super VGA is not capable of selecting the requested palette width, then the next lower value that the Super VGA is capable of, will be selected. The resulting palette width is returned.

10.7.7 Application Example

The following sequence illustrates how an application would interface to the VESA BIOS Extension. The hypothetical application is VESA-aware and calls the VESA BIOS functions. However, the application is not limited to supporting just VESA-defined video modes. Thus, it will inquire what video modes are available, before setting up the video mode.

1) The application would first allocate a 256-byte buffer. This buffer will be used by the VESA BIOS to return information about the video environment. Some applications will statically allocate this buffer, others will use system calls to temporarily obtain buffer space.

2) The application would then call VESA BIOS function 00h (Return Super VGA information). If the AX register does not contain 004Fh on return from the function call, the application can determine that the VESA BIOS Extension is not present and handle such situation.

   If no error code is passed in AX, the function call was successful. The buffer has been filled by the VESA BIOS Extension with various information. The application can verify that indeed this is a valid VESA block by identifying the characters 'VESA' in the beginning of the block. The application can inspect the VESAVersion field to determine whether the VESA BIOS Extension has sufficient functionality. The application may use the OEMStringPtr to locate OEM-specific information.

   Finally, the application can obtain a list of the supported Super VGA modes, by using the VideoModePtr. This field points to a list of the video modes supported by the video environment.

3) The application would then create a new buffer and call the VESA BIOS function 01h (Return Super VGA mode information), to obtain information about the supported video modes. Using the VideoModePtr, obtained in step 2 above, the application would call this function with a new mode number until a suitable video mode is found. If no appropriate video mode is found, its up to the application to handle this situation.

   The Return Super VGA mode information function fills a buffer specified by the application with information describing the features of the video mode. The data block contains all the information an application needs to take advantage of the video mode.

   The application would examine the ModeAttributes field. To verify that the mode indeed is supported, the application would inspect bit D0. If D0 is cleared, then the mode is not supported by the hardware. This might happen if a specific mode requires a certain type of monitor, but that monitor is not present.

4) After the application has selected a video mode, the next step is to initialize the mode. However, the application might first want to save the present video mode. When the application exits, this mode would be restored. To obtain the present video mode, the VESA BIOS function 03h (Get Super VGA mode), would be used. If a non-VESA (standard VGA or OEM-specific) mode is in effect, only the lower byte in the mode number is filled. The upper byte is cleared.

5) To initialize the video mode, the application would use VESA BIOS function 02h (Set Super VGA mode). The application has from this point on full access to the VGA hardware and video memory.

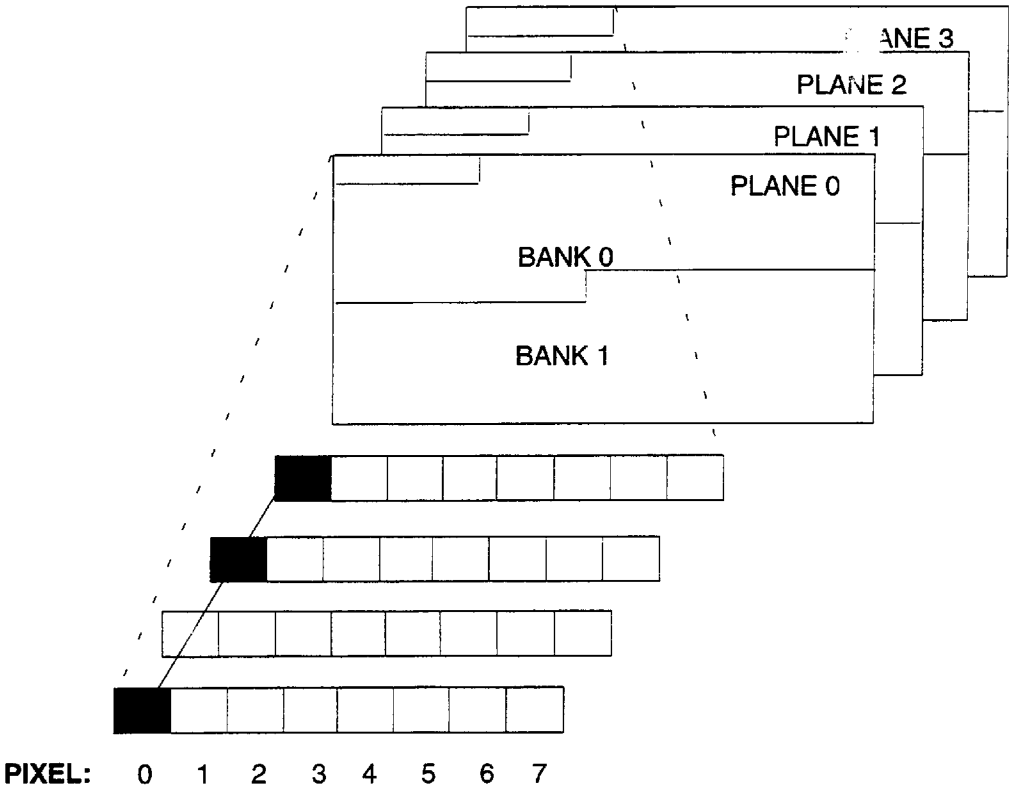
6) When the application is about to terminate, it would restore the prior video mode. The prior video obtained in step 4) above could be either a standard VGA mode, OEM-specific mode, or VESA-supported mode. It would reinitialize the video mode by calling VESA BIOS function 02h (Set Super VGA mode). The application would then exit.

10.8.7 Memory Mapping

10.8.7.1 16-Color Planar Mode Memory Organization

The Planar Video Memory denotes that a pixel is represented by bits of information dispersed across a set of planes. The 16-color Mode requires four bits per pixel, which means four display planes have the pixel represented by one bit per plane. The Display Memory is organized as bytes, and there are eight pixels per byte. The memory planes are overlaid in the CPU memory address space so that each plane occupies the same CPU address. The CPU can access any of these planes independently for read/write operations by programming the appropriate select registers.

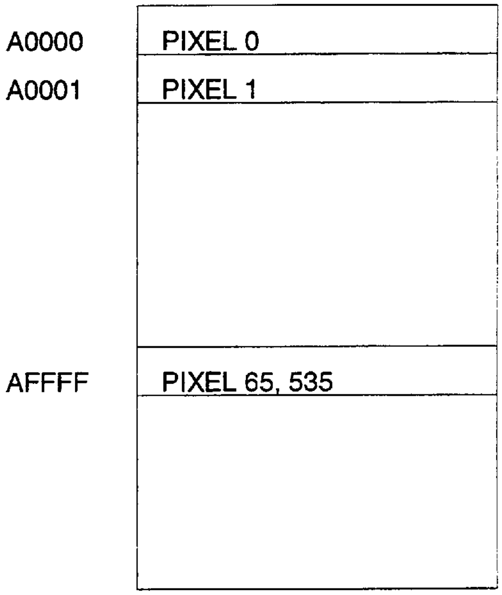
This figure shows the video memory organized in four display planes for 16-color Planar Mode. The color information for each pixel is stored in corresponding bits across four planes. The figure also depicts how Bank 0 and Bank 1 are aligned across each plane.



The standard IBM VGA supports 256K bytes of video memory. In 16-color Planar Mode, the video memory is divided into four planes with 64K bytes per plane. The 64K bytes segment of each plane is mapped normally from A0000 to AFFFF, and the CPU can easily address each of the video memory planes. The problem arises for a video mode that requires more than 64K bytes of addressable video memory per plane. For example the 1024 x 768, 16-color Planar Mode requires 98,304 bytes of Display Memory per plane, and a way to remap different segments of Display Memory into the CPU-limited address range. The Display Memory address remapping scheme is discussed in a later section.

10.8.2 256-Color Packed Pixel Modes

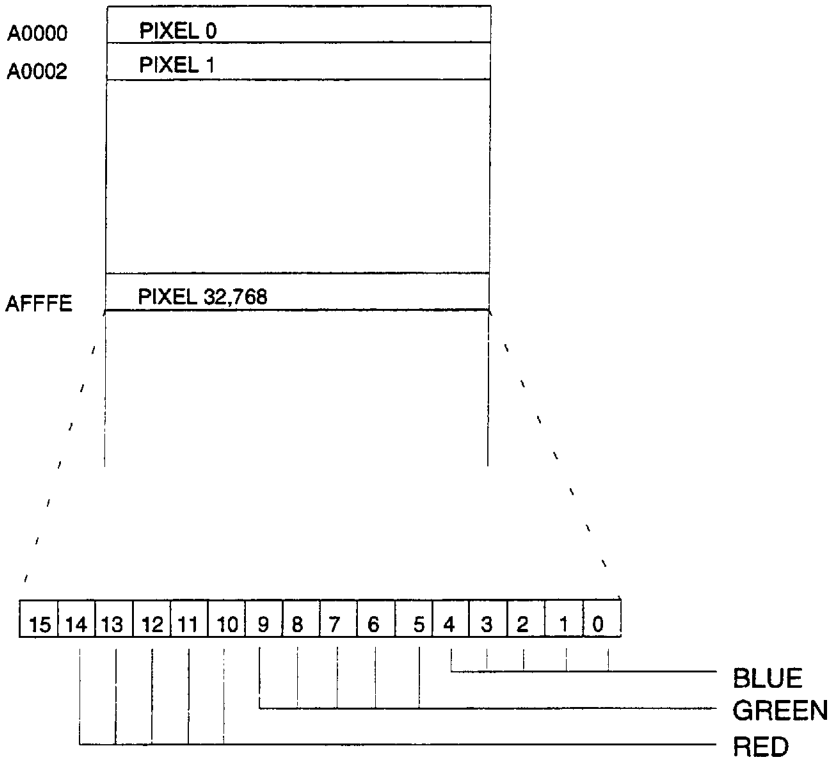
The 256-color Modes use eight bits (or one byte) per pixel; the term 'packed' means all information about each pixel is packed in one plane. The CL-GD543X memory controller handles all the address generation; to the CPU the Display Memory appears as if it were linear 64K bytes of addressable Display Memory. For high-resolution modes that require addressing beyond the 64K byte segment of video memory, the CL-GD542X has Display Memory remapping schemes to 'page-in' the Display Memory segments.



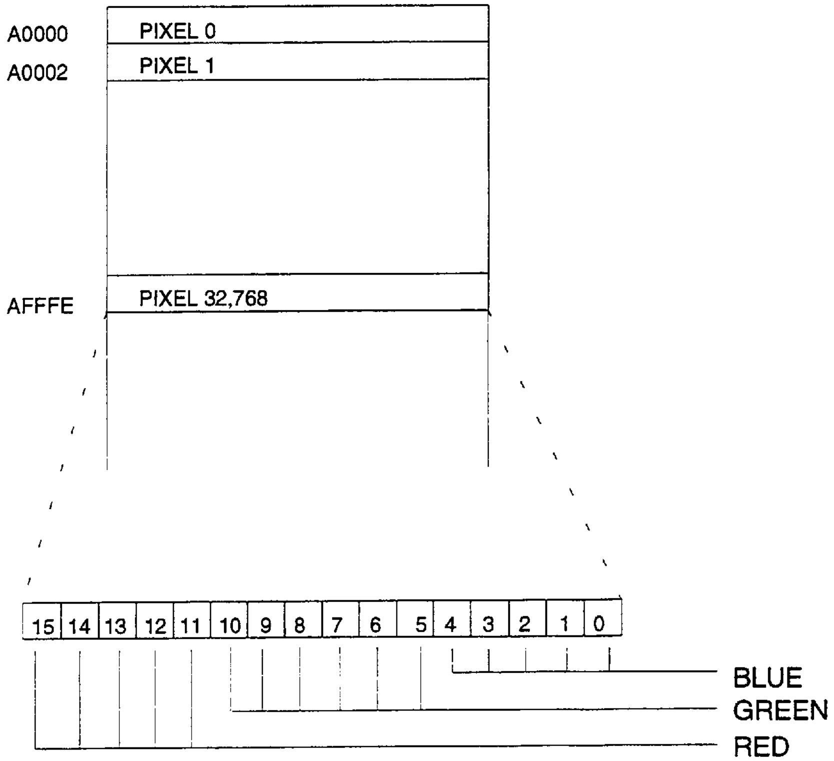
The figure shows how eight bits per pixel (Packed-pixel Mode) bytes are stored in terms of physical plane organization. The memory controller makes this organization completely transparent; therefore to the CPU the byte addressing is sequential. For example, at Segment A0000, Pixel 0 resides at offset 0, Pixel 1 at offset 1, and Pixel 65535 at offset 65535. To address Display Memory beyond 64K segment boundary, the OTI-64107 supports a Display Memory paging scheme that allows up to 1 Mbyte of Display Memory to be paged into the CPU address range.

10.8.3 DIRECT-COLOR (32,768 OR 65,536 COLORS), PACKED-PIXEL MODES

The OTI-64107 controllers support a Direct-color Mode that is capable of displaying 32, 768 or 65,536 colors simultaneously at screen resolutions of up to 800 x 600. The OTI-64107 uses its internal Palette DAC to support this TARGA- or IBM XGA-compatible Mode, where 15 or 16 bits per pixel of color information is used to display 32,768 or 65,536 colors. The VGA-standard Palette DAC Lookup Table is limited to 256 addresses. The scheme bypasses the Lookup Table, and passes 15 or 16 bits of RGB (RED, GREEN, and BLUE) color information directly to the DAC to generate the high-range colors. The OTI-64107 Direct-color Mode displays close to true-color images on standard VGA analog monitors for desktop and multimedia applications.



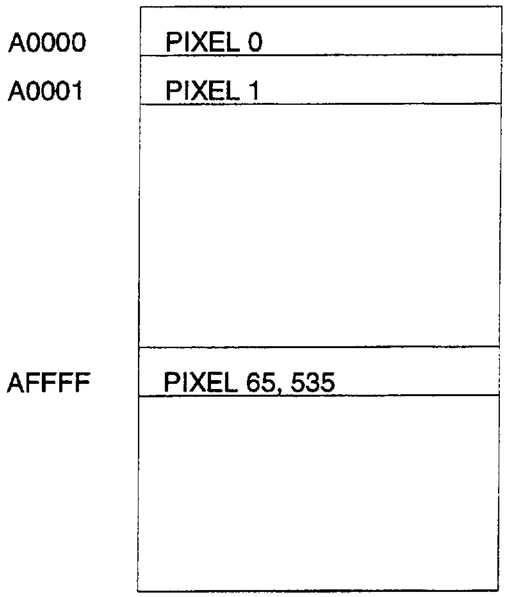
The figure shows the 15-byte (32,768 colors) Direct-color, Packed-pixel Mode organization. The RGB color information is stored in 5-5-5 format, and the MSB (Bit 15) is ignored. Two bytes per pixel are used for storing the color information. The CPU addressing is completely sequential. The 5-5-5 format represents most-significant bits of RED, GREEN, and BLUE of the 24-bit DAC output.



This figure shows the 16-bit (65,536 colors) Direct-color, Packed-pixel Mode organization. The RGB color information is stored in a 5-6-5 format comprising 5-bits of RED, 6-bits of GREEN, and 5-bits of BLUE. The color information (per pixel) is read from Display Memory, and used by the internal Palette DAC in Direct-color Mode to display 65,536 colors simultaneously.

10.8.4 TRUE COLOR 24-BIT (16,8 MILLION COLORS), PACKED-PIXEL MODES

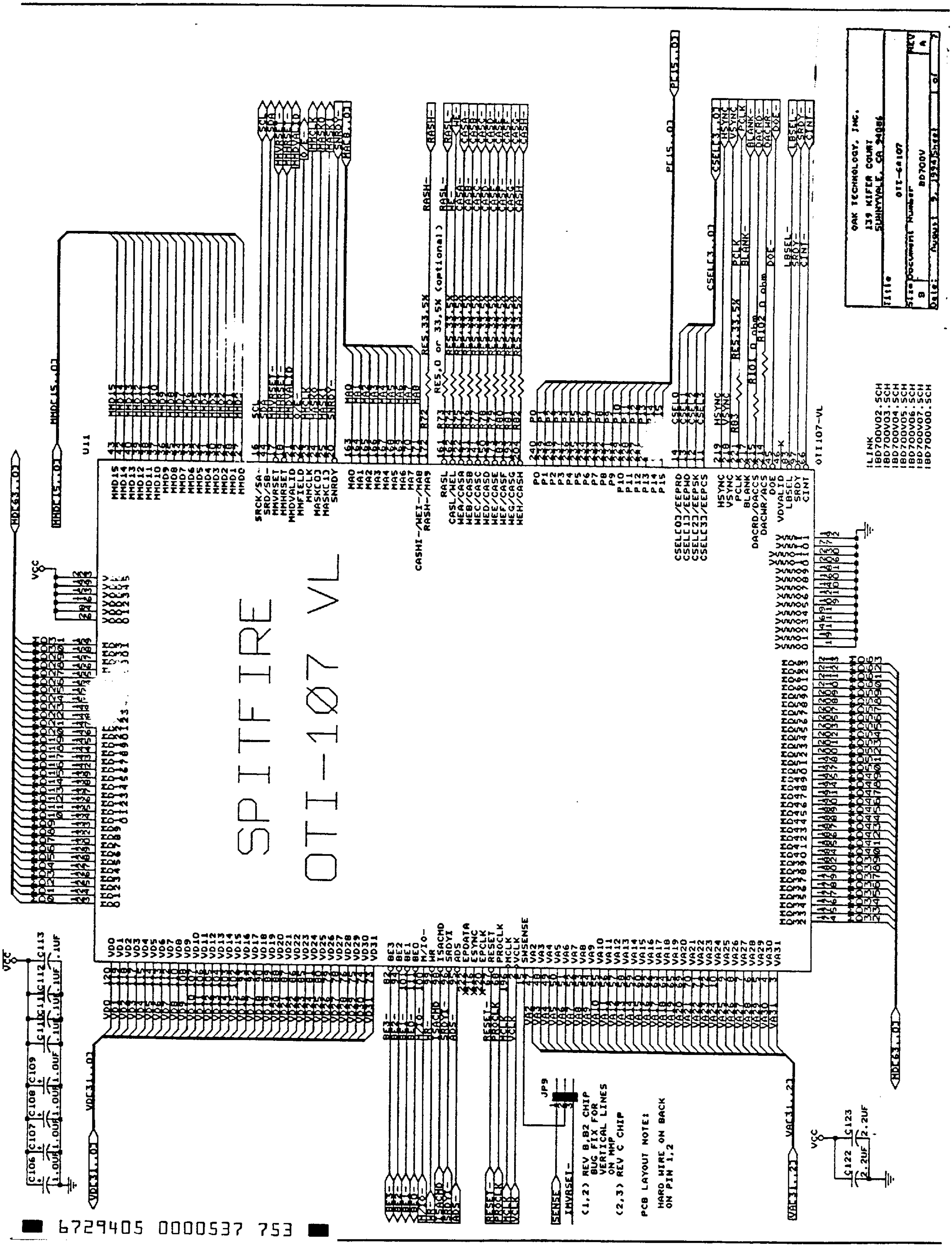
The OTI-64107 VGA controller supports a True Color Mode that is capable of displaying 16-million colors simultaneously at screen resolutions of up to 640 x 480. The OTI-64107 uses its internal Palette DAC to support this TARGA-compatible Mode, where 24 bits per pixel of color information is used to display 16-million colors. The VGA-standard Palette DAC Lookup Table is limited to 256 addresses. This scheme bypasses the Lookup Table, and passes 24 bits of RGB (RED, GREEN, and BLUE) color information directly to the DAC to generate the high-range of colors. The OTI-64107 True Color Mode offers professional-quality color images to be displayed on standard VGA analog monitors for desktop and multimedia applications.



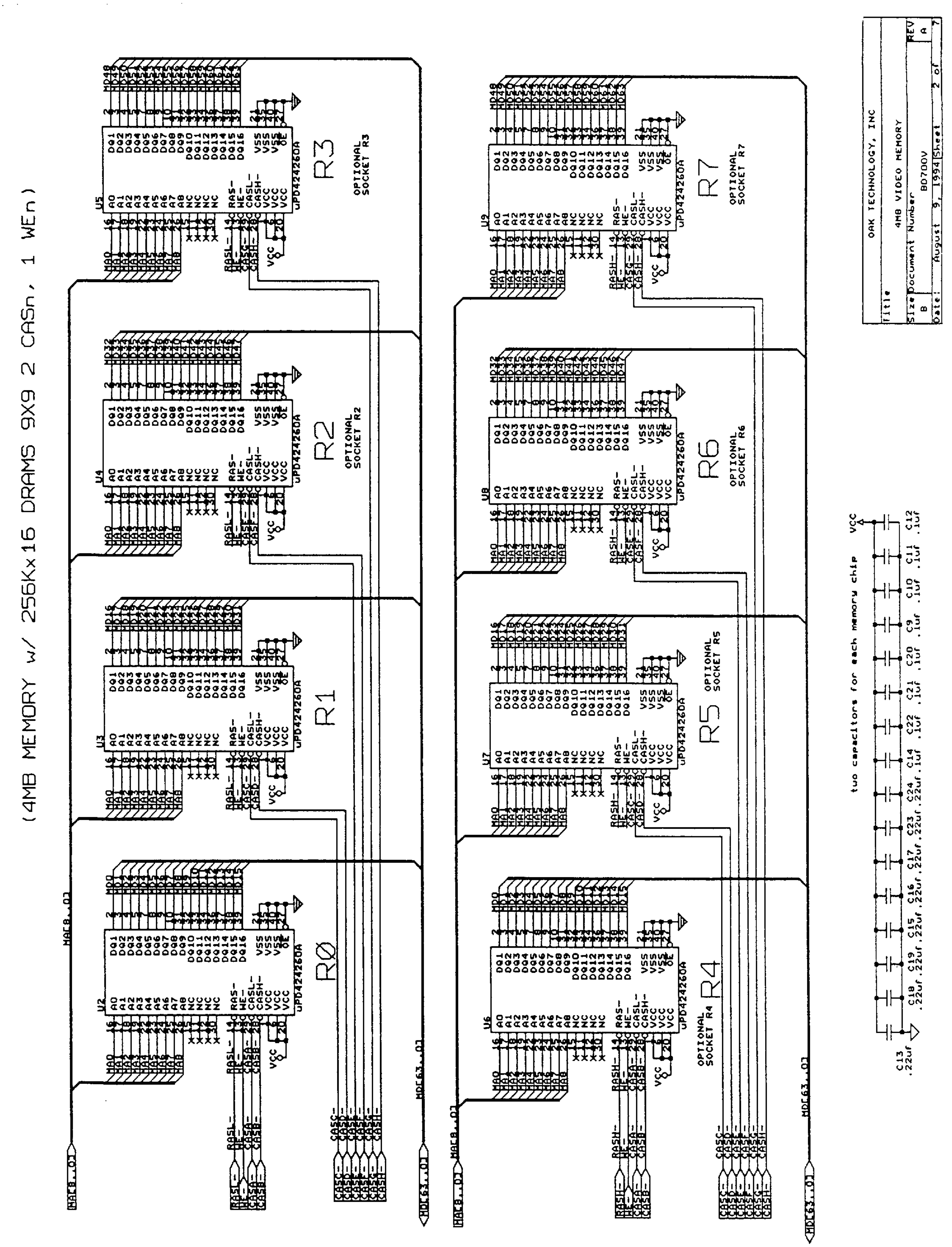
This figure shows how True Color 24-bits per pixel (RGB 8-8-8) video mode data bytes are stored in terms of physical plane organization. The memory controller makes this organization completely transparent. Hence to the CPU the byte addressing is sequential. For example, at Segment A0000, Pixel 0 RGB color information is stored in three sequential bytes: Blue color information resides at offset 0, Green color information resides at offset 1, and Red color information resides at offset 2. Pixel 1 RGB information is at offset 3. To address Display Memory beyond 64K segment boundary, the OTI-64107 supports a Display Memory paging scheme that allows up to 1 Mbyte of Display Memory to be paged into the CPU address range.

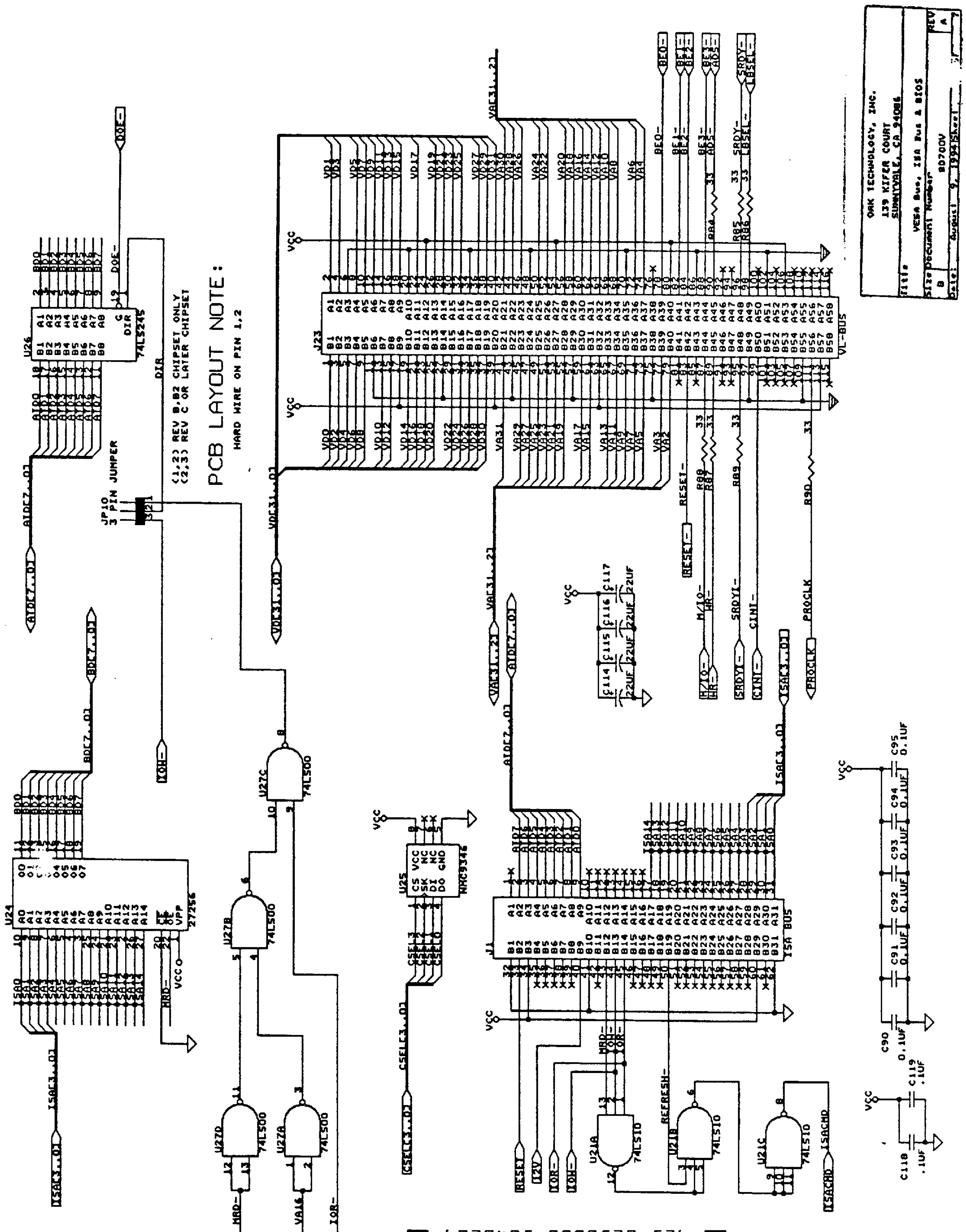
CHAPTER 11: REFERENCE DESIGN

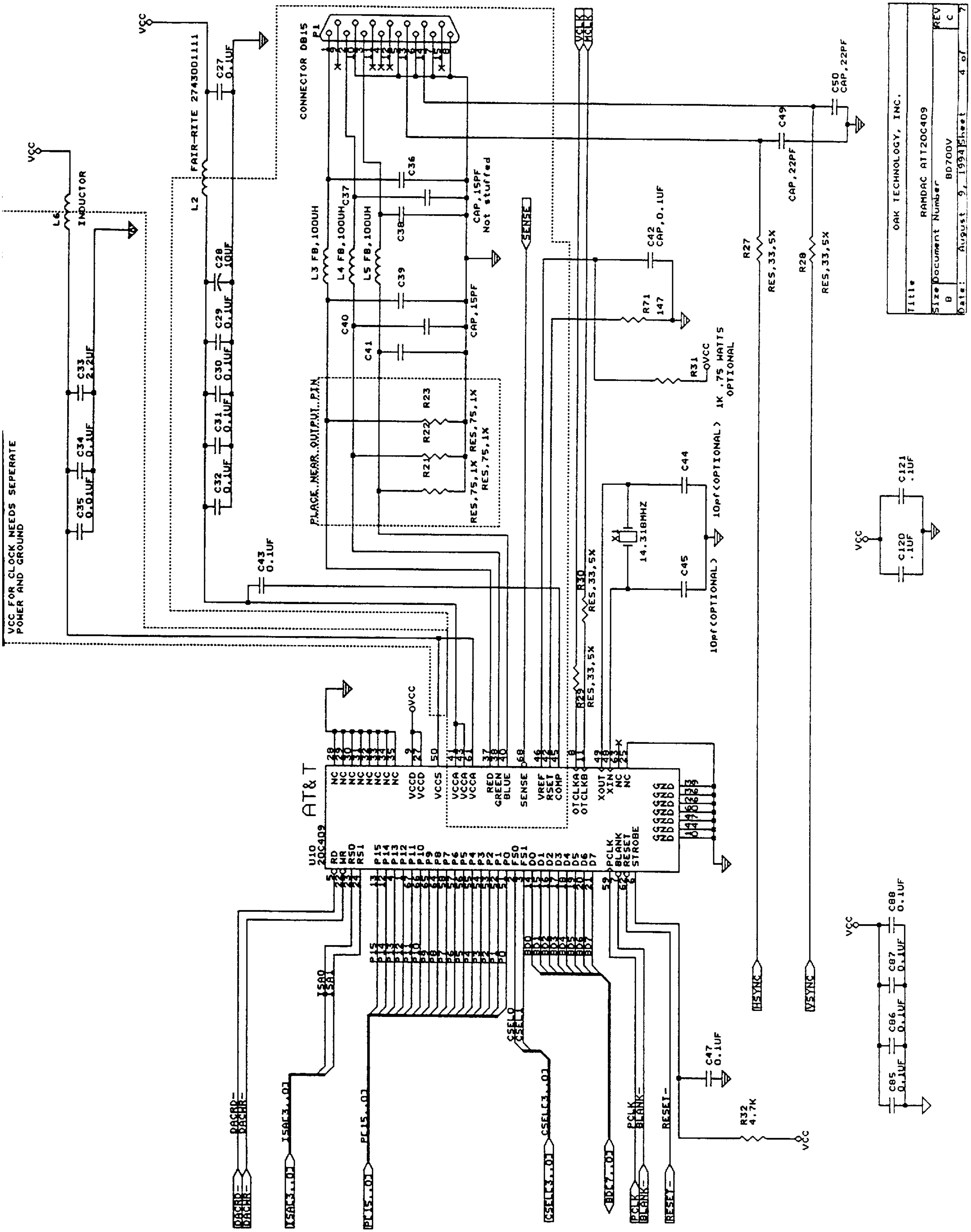
The enclosed referece designs use the Spitfire OTI-64107 in 2 Audio/Video/Graphics (AVG) boards. One design is VL based and the other is PCI based. The graphics and video use the 2Mbyte DRAM shared frame buffer of the OTI-64107. The video NTSC signal is input to the Philips SAA7110 video decoder followed by the Phillips SAA7186 color converter and scaler. The output of the SAA718 is stored in the DRAM frame buffer of the OTI-64107 through the 16-bit Multimedia port on the 64107. The VL bus design uses an AT&T20C499 *Precision*DAC (16-bit pixel port) while the PCI bus design uses the pin compatible AT&T20C499 *Precision*DAC (24-bit pixel port). Both of these reference designs are available as evalustion boards. Please contact OAK Technology for more information.

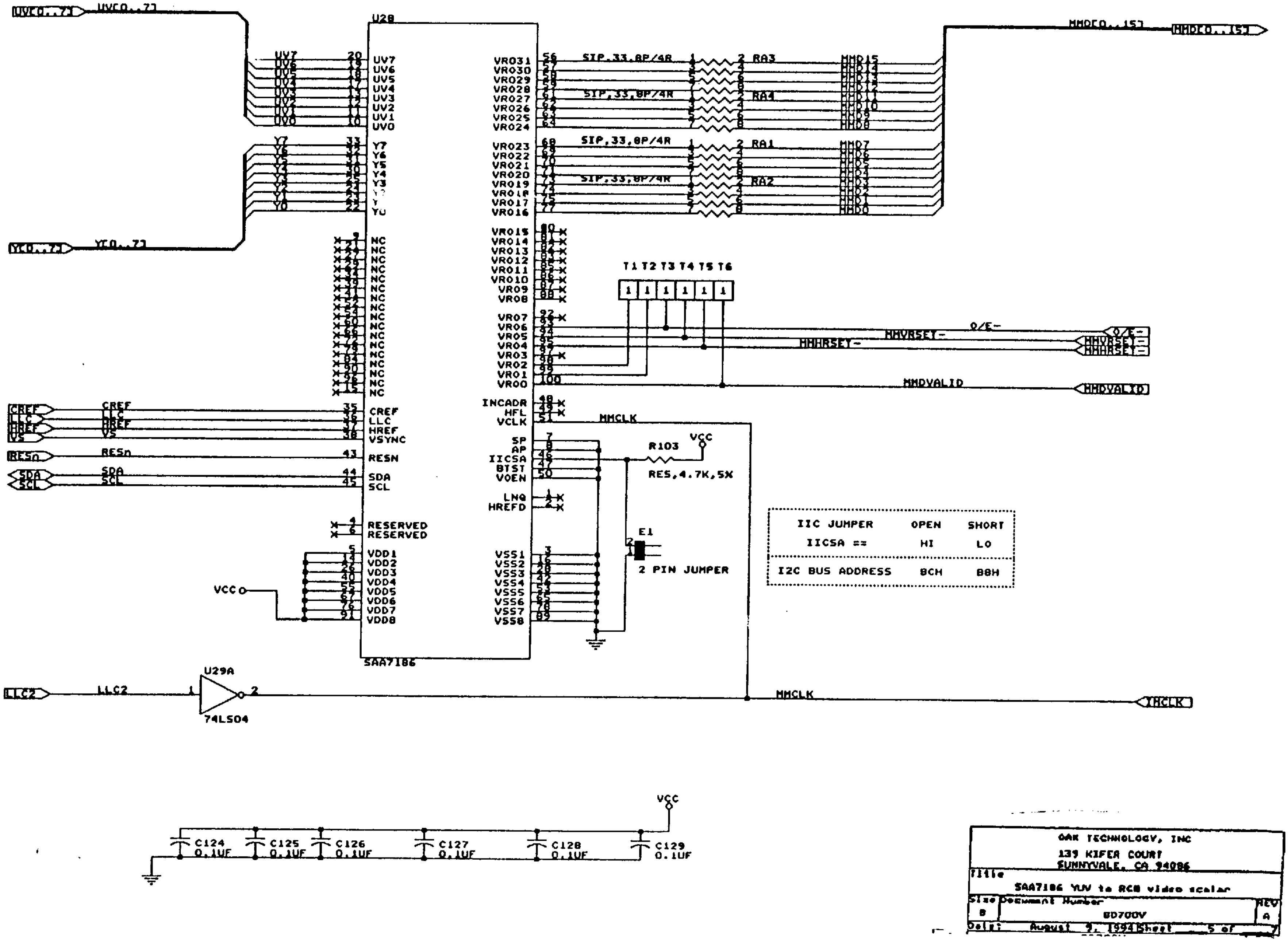


6729405 0000537 753


(4MB MEMORY w/ 256Kx16 DRAMS 9X9 2 CASn, 1 WEn)
R0
R1
R2
R3
R4
R5
R6
R7
OPTIONAL SOCKET R2
OPTIONAL SOCKET R3
OPTIONAL SOCKET R4
OPTIONAL SOCKET R5
OPTIONAL SOCKET R6
OPTIONAL SOCKET R7
uPD424260A
U2
U3
U4
U5
U6
U7
U8
U9
MA[8..0]
MD[63..0]
RASL-
RASH-
WE-
CASA-
CASB-
CASC-
CASD-
CASE-
CASF-
CASG-
CASH-
VCC
two capacitors for each memory chip
C13 .22uf
C18 .22uf
C19 .22uf
C15 .22uf
C16 .22uf
C17 .22uf
C23 .22uf
C24 .22uf
C14 .1uf
C22 .1uf
C21 .1uf
C20 .1uf
C9 .1uf
C10 .1uf
C11 .1uf
C12 .1uf
OAK TECHNOLOGY, INC
Title
4MB VIDEO MEMORY
Size
B
Document Number
B070OV
REV
A
Date:
August 9, 1994
Sheet 2 of 7


OAK TECHNOLOGY, INC.
139 KIFER COURT
SUNNYVALE, CA 94086
Title VESA Bus, ISA Bus & BIOS
Size B
Document Number 8D700V
REV A
Date: August 9, 1994
Sheet
PCB LAYOUT NOTE:
HARD WIRE ON PIN 1,2
(1,2) REV B,B2 CHIPSET ONLY
(2,3) REV C OR LATER CHIPSET
JP10 3 PIN JUMPER
U26 74LS245
U24 27256
U23 VL-BUS
J1 ISA BUS
U25 NMC9346
U27A 74LS00
U27B 74LS00
U27C 74LS00
U27D 74LS00
U21A 74LS10
U21B 74LS10
U21C 74LS10
AID[7..0]
BD[7..0]
ISA[3..0]
VD[31..0]
VA[31..2]
CSEL[3..0]
DOE-
IOW-
IOR-
MRD-
VA16
RESET
12V
RESET-
M/IO-
WR-
SRDY-
SRDYI-
CINI-
PROCLK
BE0-
BE1-
BE2-
BE3-
ADS-
LBSEL-
REFRESH-
ISACHD
VCC
C90 0.1UF
C91 0.1UF
C92 0.1UF
C93 0.1UF
C94 0.1UF
C95 0.1UF
C114 22UF
C115 22UF
C116 22UF
C117 22UF
C118 .1UF
C119 .1UF
R84 33
R85 33
R86 33
R87 33
R88 33
R89 33
R90 33


OAK TECHNOLOGY, INC.
Title RAMDAC ATT20C409
Size B
Document Number BD700V
REV C
Date: August 9, 1994
Sheet 4 of 7
U10 20C409
AT&T
CONNECTOR DB15
P1
VCC
L6 INDUCTOR
L2 FAIR-RITE 2743001111
C27 0.1UF
C28 10UF
C29 0.1UF
C30 0.1UF
C31 0.1UF
C32 0.1UF
C33 2.2UF
C34 0.1UF
C35 0.01UF
C43 0.1UF
VCC FOR CLOCK NEEDS SEPERATE POWER AND GROUND
L3 FB,100UH
L4 FB,100UH
L5 FB,100UH
C36
C37
C38
C39
C40
C41
CAP,15PF Not stuffed
CAP,15PF
PLACE NEAR OUTPUT PIN
R21
R22
R23
RES,75,1X
SENSE
C42 CAP,0.1UF
R71 147
R31 VCC
1K .75 WATTS OPTIONAL
C44
C45
10pf(OPTIONAL)
X1 14.318MHZ
R29 RES,33,5X
R30 RES,33,5X
R27 RES,33,5X
R28 RES,33,5X
C49 CAP,22PF
C50 CAP,22PF
VCLK
XCLK
HSYNC
VSYNC
C120 .1UF
C121 .1UF
C85 0.1UF
C86 0.1UF
C87 0.1UF
C88 0.1UF
C47 0.1UF
R32 4.7K
NC
VCCD
VCCS
VCCA
RED
GREEN
BLUE
SENSE
VREF
RSET
COMP
OTCLKA
OTCLKB
XOUT
XIN
RD
WR
RS0
RS1
P15 P14 P13 P12 P11 P10 P9 P8 P7 P6 P5 P4 P3 P2 P1 P0
FS0
FS1
D0 D1 D2 D3 D4 D5 D6 D7
PCLK
BLANK
RESET
STROBE
GND
DACRD-
DACWR-
ISA0
ISA1
ISA[3..0]
PC[15..0]
CSEL0
CSEL1
CSEL[3..0]
BD0 BD1 BD2 BD3 BD4 BD5 BD6 BD7
BD[7..0]
PCLK
BLANK-
RESET-



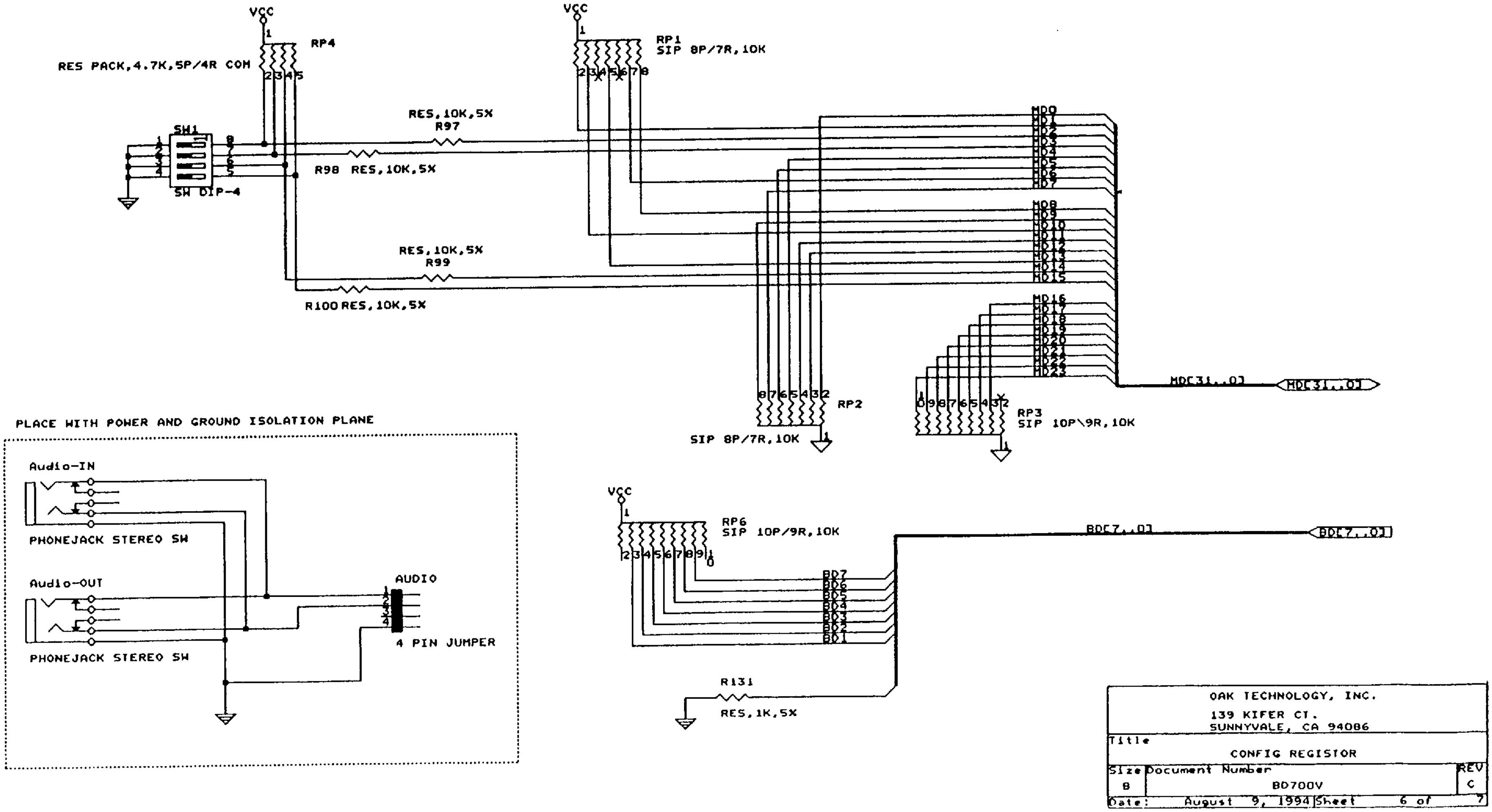
IIC JUMPER	OPEN	SHORT
IICSA ==	HI	LO
I2C BUS ADDRESS	BCH	B8H

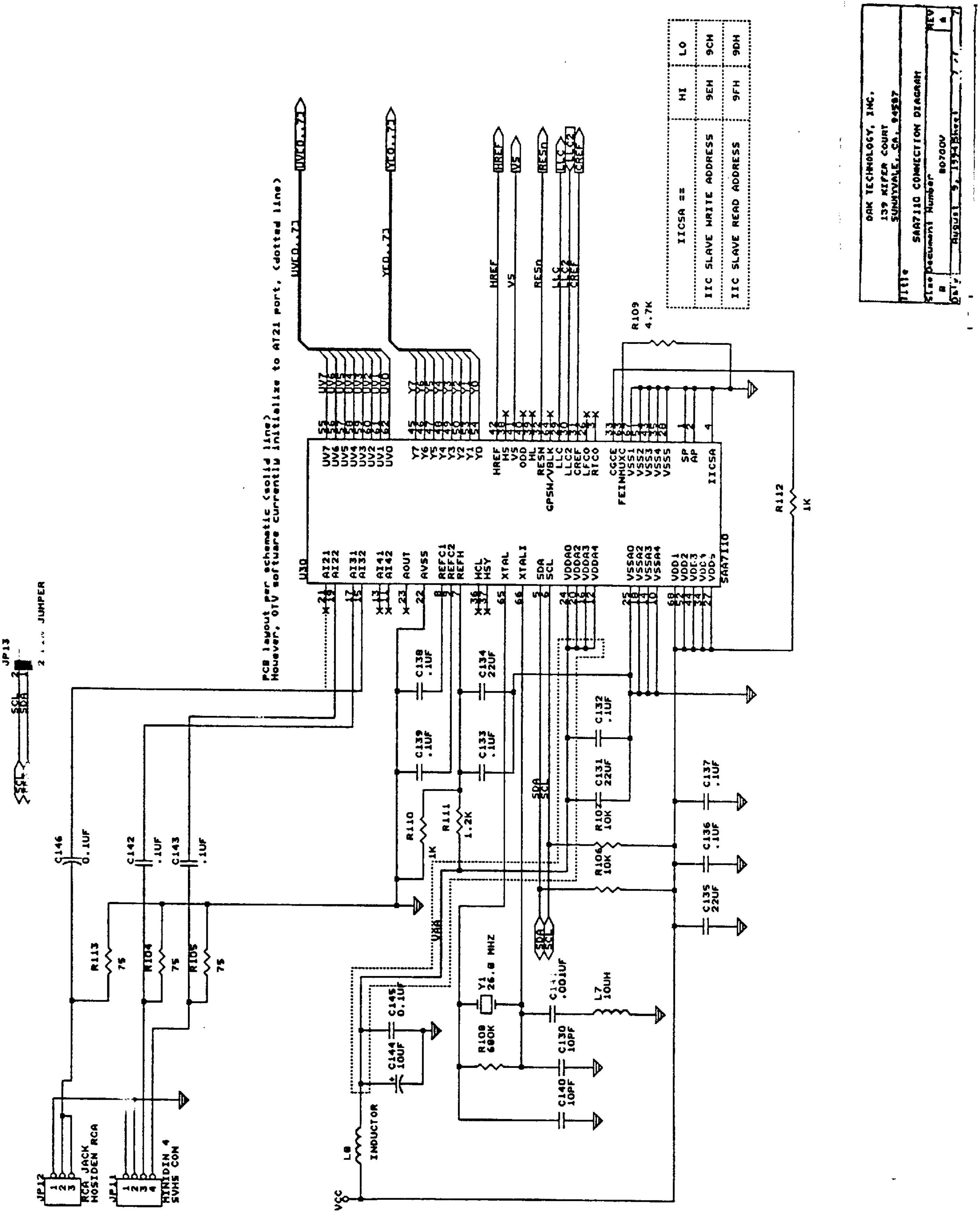
OAK TECHNOLOGY, INC
139 KIFER COURT
SUNNYVALE, CA 94086

Title		
SAA7186 YUV to RGB video scalar		
Size B	Document Number BD700V	REV A
Date: August 9, 1994	Sheet 5 of 7	

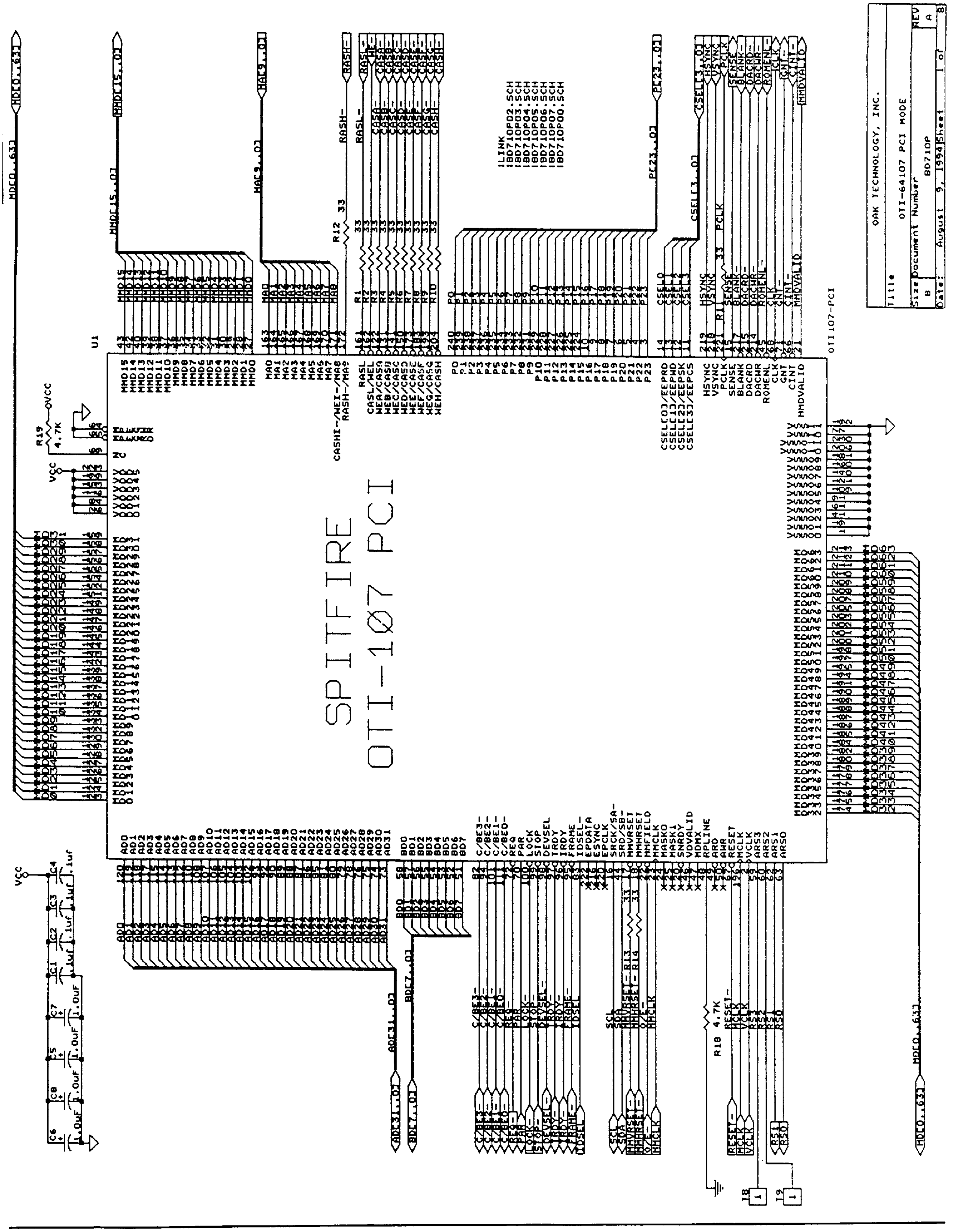
POWER ON CONFIGURATION

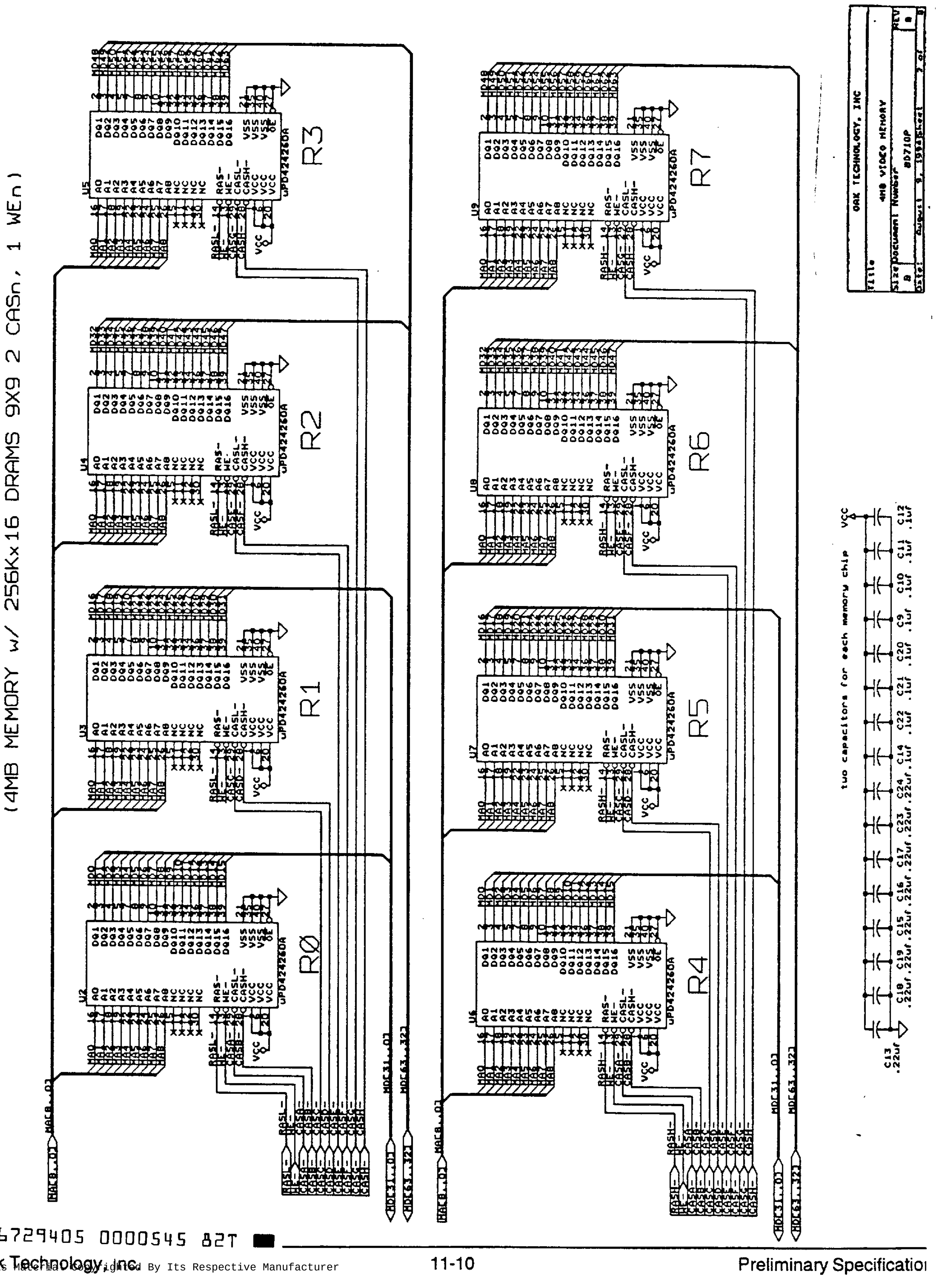
MDØ: 8-BIT BIOS (1 ROM)
MD1: ADD-ON CONFIGURATION
MD4..2: VESA LOCAL BUS TYPE
MD5..7: 256K X 16(9X9) DRAM
MD8: PIXEL BUS WIDTH (16 BITS)
MD9: NORMAL OPERATION. 1Ø7 GENERATES DACRDn AND DACWRn.
MD1Ø: ENABLE DECODING FOR ROM BIOS
MD11-12: ROM BIOS SIZE (32K BYTES)
MD13: CASHn, CASLn, WEn (9X9 TYPE)
MD14: FAST WRITE CAPABLE
MD15: VESA LOCAL BUS SPEED
MD16-23: RESERVED

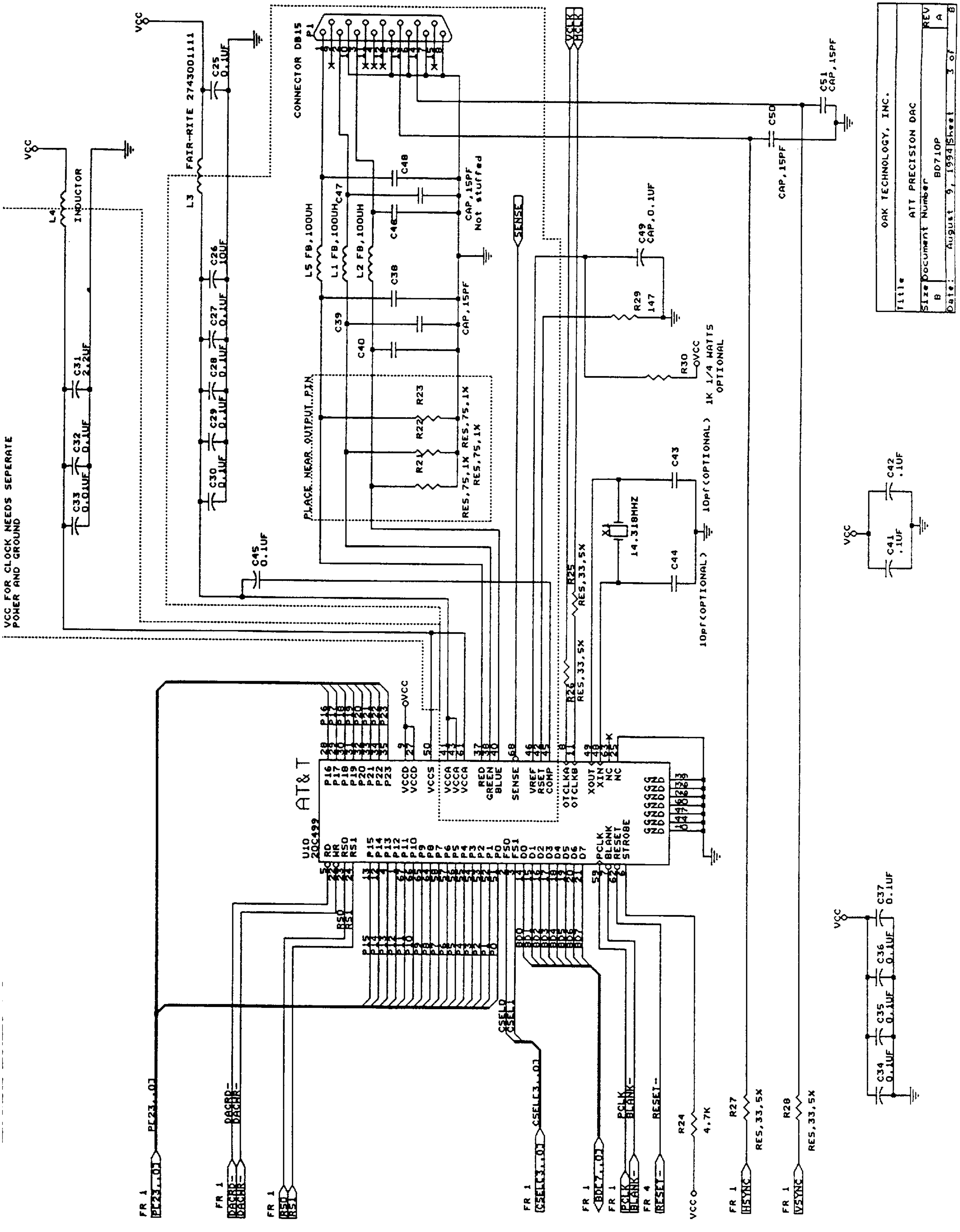


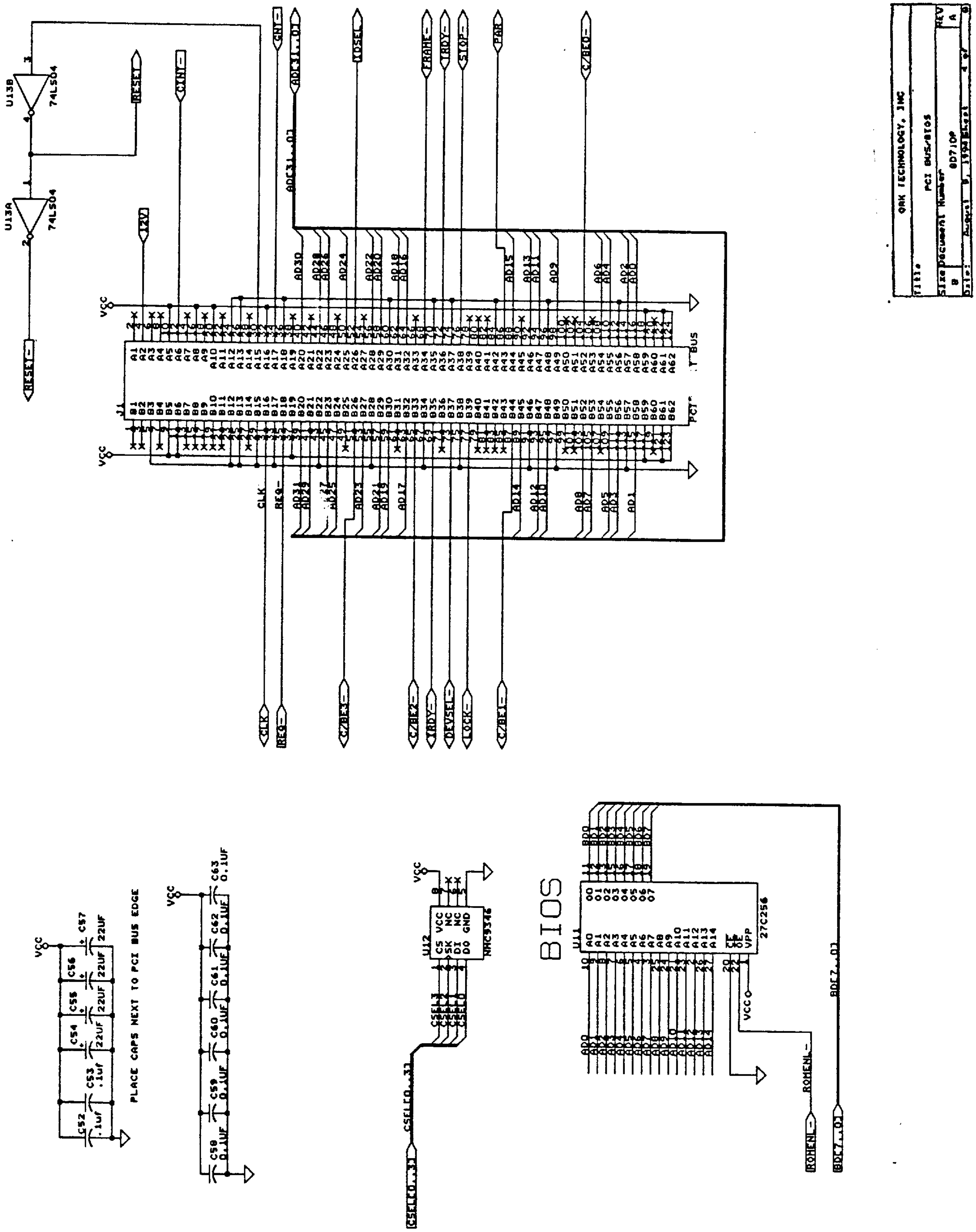


IICSA ==	HI	LO
IIC SLAVE WRITE ADDRESS	9EH	9CH
IIC SLAVE READ ADDRESS	9FH	9DH


SPITFIRE
OTI-107 PCI
U1
OTI107-PCI
OAK TECHNOLOGY, INC.
Title OTI-64107 PCI MODE
Size B
Document Number BD710P
REV A
Date: August 9, 1994
Sheet 1 of 8
!LINK
!BD710P02.SCH
!BD710P03.SCH
!BD710P04.SCH
!BD710P05.SCH
!BD710P06.SCH
!BD710P07.SCH
!BD710P00.SCH
MD[0..63]
MMD[15..0]
MA[9..0]
RASH-
RASL-
PC[23..0]
CSEL[3..0]
HSYNC
VSYNC
PCLK
SENSE
BLANK-
DACRD-
DACWR-
ROMENL-
CLK
GNT-
CINT-
MMDVALID
AD[31..0]
BD[7..0]
RESET-
MCLK
VCLK
RS1
RS0
R18 4.7K
R19 4.7K
R12 33
VCC
OVCC

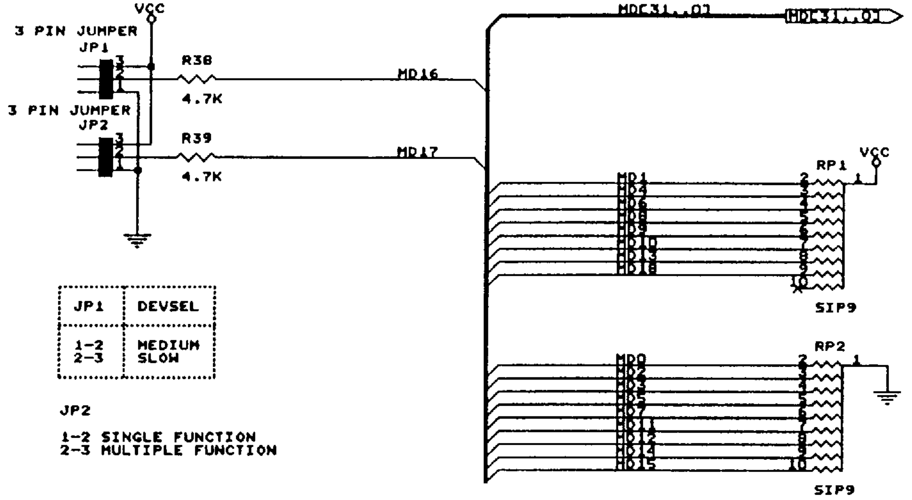

(4MB MEMORY w/ 256Kx16 DRAMS 9X9 2 CASn, 1 WEn)
R0
R1
R2
R3
R4
R5
R6
R7
U2
U3
U4
U5
U6
U7
U8
U9
uPD424260A
A0
A1
A2
A3
A4
A5
A6
A7
A8
NC
RAS-
WE-
CASL-
CASH-
VCC
VSS
OE
DQ1
DQ2
DQ3
DQ4
DQ5
DQ6
DQ7
DQ8
DQ9
DQ10
DQ11
DQ12
DQ13
DQ14
DQ15
DQ16
MA0
MA1
MA2
MA3
MA4
MA5
MA6
MA7
MA8
MA[8..0]
MD0
MD16
MD32
MD48
MD[31..0]
MD[63..32]
RASL-
RASH-
WE-
CASA-
CASB-
CASC-
CASD-
CASE-
CASF-
CASG-
CASH-
two capacitors for each memory chip
VCC
C13 .22uf
C18 .22uf
C19 .22uf
C15 .22uf
C16 .22uf
C17 .22uf
C23 .22uf
C24 .22uf
C14 .1uf
C22 .1uf
C21 .1uf
C20 .1uf
C9 .1uf
C10 .1uf
C11 .1uf
C12 .1uf
OAK TECHNOLOGY, INC
Title
4MB VIDEO MEMORY
Size
B
Document Number
8D710P
REV
B
Date:
August 9, 1994
Sheet
2 of
8


OAK TECHNOLOGY, INC.
Title ATT PRECISION DAC
Size B
Document Number BD710P
REV A
Date: August 9, 1994
Sheet 3 of 8
VCC FOR CLOCK NEEDS SEPERATE POWER AND GROUND
L4 INDUCTOR
L3 FAIR-RITE 2743001111
CONNECTOR DB15
P1
C25 0.1UF
C26 10UF
C27 0.1UF
C28 0.1UF
C29 0.1UF
C30 0.1UF
C31 2.2UF
C32 0.1UF
C33 0.01UF
C45 0.1UF
L5 FB,100UH
L1 FB,100UH
L2 FB,100UH
C46
C47
C48
CAP,15PF Not stuffed
C38
C39
C40
CAP,15PF
PLACE NEAR OUTPUT PIN
R21
R22
R23
RES,75,1X
RES,75,1X
RES,75,1X
SENSE
VCLK
HACK
C49 CAP,0.1UF
R29 147
R30 VCC
1K 1/4 WATTS OPTIONAL
C43
10pf(OPTIONAL)
X1 14.318MHZ
C44
10pf(OPTIONAL)
R25 RES,33,5X
R26 RES,33,5X
C50 CAP,15PF
C51 CAP,15PF
C41 .1UF
C42 .1UF
VCC
U10 20C492
AT&T
RD
WR
RS0
RS1
P0–P23
VCCD
VCCS
VCCA
RED
GREEN
BLUE
SENSE
VREF
RSET
COMP
OTCLKA
OTCLKB
XOUT
XIN
NC
GNDD
FS0
FS1
D0–D7
PCLK
BLANK
RESET
STROBE
CSEL0
CSEL1
FR 1 PC[23..0]
FR 1 DACRD- DACWR-
FR 1 RS0 RS1
FR 1 CSEL[3..0]
FR 1 BD[7..0]
FR 1 PCLK BLANK-
FR 4 RESET-
VCC R24 4.7K
FR 1 HSYNC R27 RES,33,5X
FR 1 VSYNC R28 RES,33,5X
C34 0.1UF
C35 0.1UF
C36 0.1UF
C37 0.1UF


OAK TECHNOLOGY, INC
Title
PCI BUS/BIOS
Size B
Document Number 8D710P
REV A
Date: August 8, 1994 Sheet 4 of 8
U13A 74LS04
U13B 74LS04
RESET
RESET-
CINT-
GNT-
12V
VCC
AD[31..0]
IDSEL
FRAME-
TRDY-
STOP-
PAR
C/BE0-
J1
PCI BUS
CLK
REQ-
C/BE3-
C/BE2-
IRDY-
DEVSEL-
LOCK-
C/BE1-
U12 NMC9346
CS SK DI DO
VCC NC NC GND
CSEL3 CSEL2 CSEL1 CSEL0
CSEL[0..3]
BIOS
U11 27C256
CE OE VPP
BD[7..0]
ROMEN-
PLACE CAPS NEXT TO PCI BUS EDGE
C52 .1uf
C53 .1uf
C54 22UF
C55 22UF
C56 22UF
C57 22UF
C58 0.1UF
C59 0.1UF
C60 0.1UF
C61 0.1UF
C62 0.1UF
C63 0.1UF

POWER ON RESET CONFIGRATION

MDØ: 8-BIT BIOS (1 ROM)
MD1: ADD-ON CONFIGURATION
MD2-4: PCI BUS
MD5-7: DRAM TYPE 256X16 9X9
MD8: PIXEL BUS WIDTH STATUS (16 BIT)
MD9: 24 BIT PIXEL BUS FOR SYNDAC PORT.
MD1Ø: ENABLE DECODING FOR ROM BIOS
MD11-12: 32K BYTES EPROM BIOS
MD13: CASxn/WExn SELECT
MD14-15: RESERVED
MD16: DEVSELn TIMING
MD17: SINGLE OR MULTIPLE FUNCTION

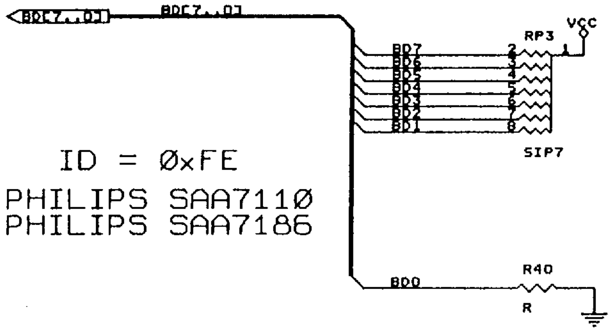


JP1	DEVSEL
1-2 2-3	MEDIUM SLOW

JP2

1-2 SINGLE FUNCTION
2-3 MULTIPLE FUNCTION

VIDEO DECODER CONFIGURATION ID

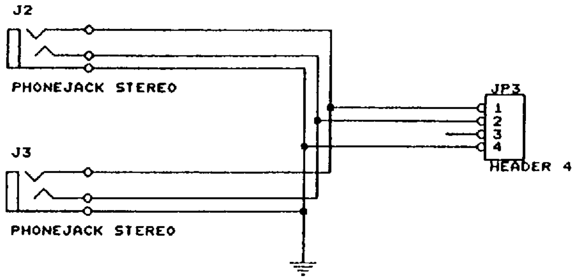


ID = ØxFE

PHILIPS SAA711Ø
PHILIPS SAA7186

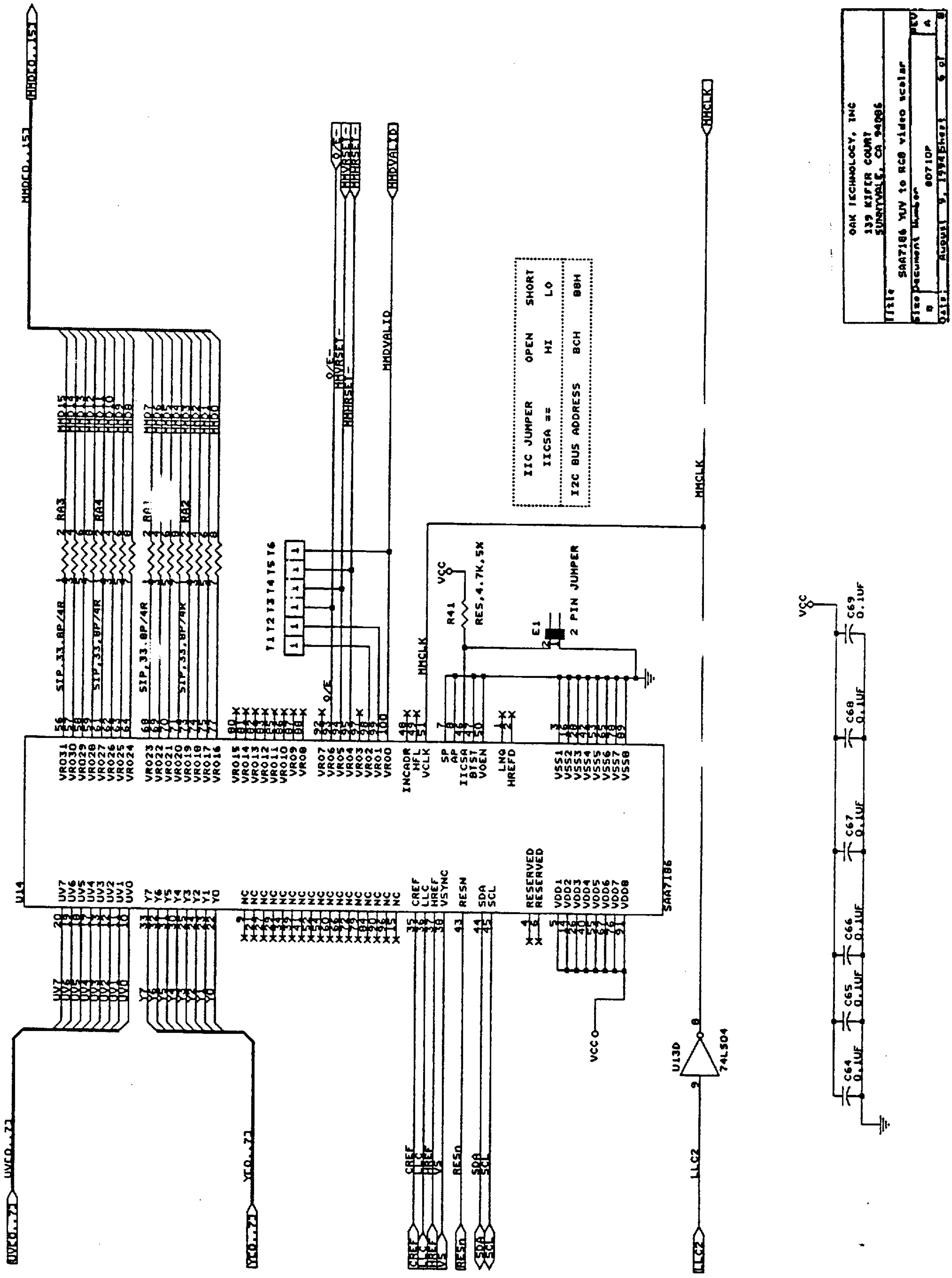
AUDIO PASS-THRU

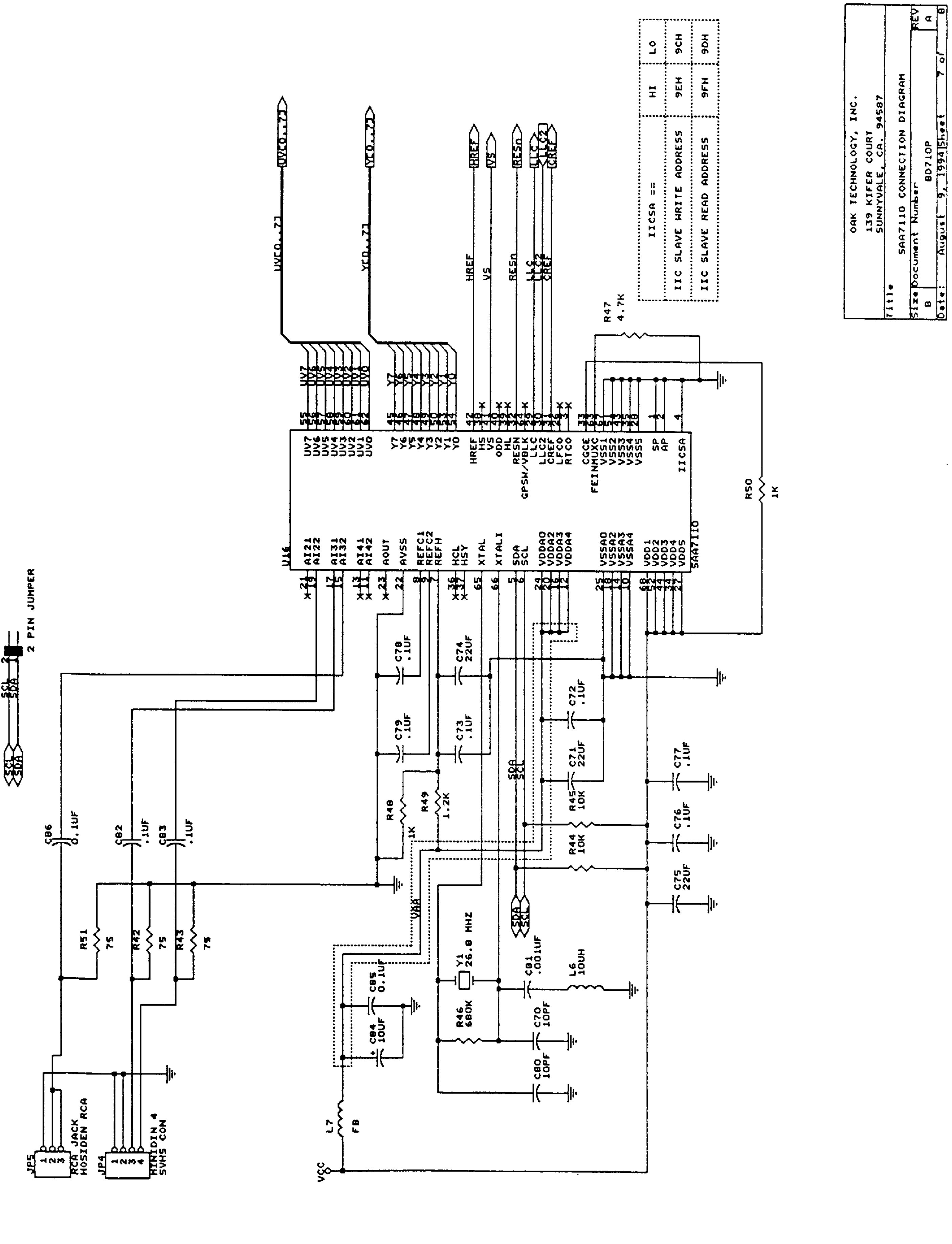
PROVIDE GROUND AND POWER CUT OUT



PROVIDE GROUND TIE TO OUTSIDE GROUND PLANE

OAK TECHNOLOGY INC.			
Title	CONFIGURATION		
Size B	Document Number BD710P		REV B
Date:	August 9, 1994	Sheet	5 of 8

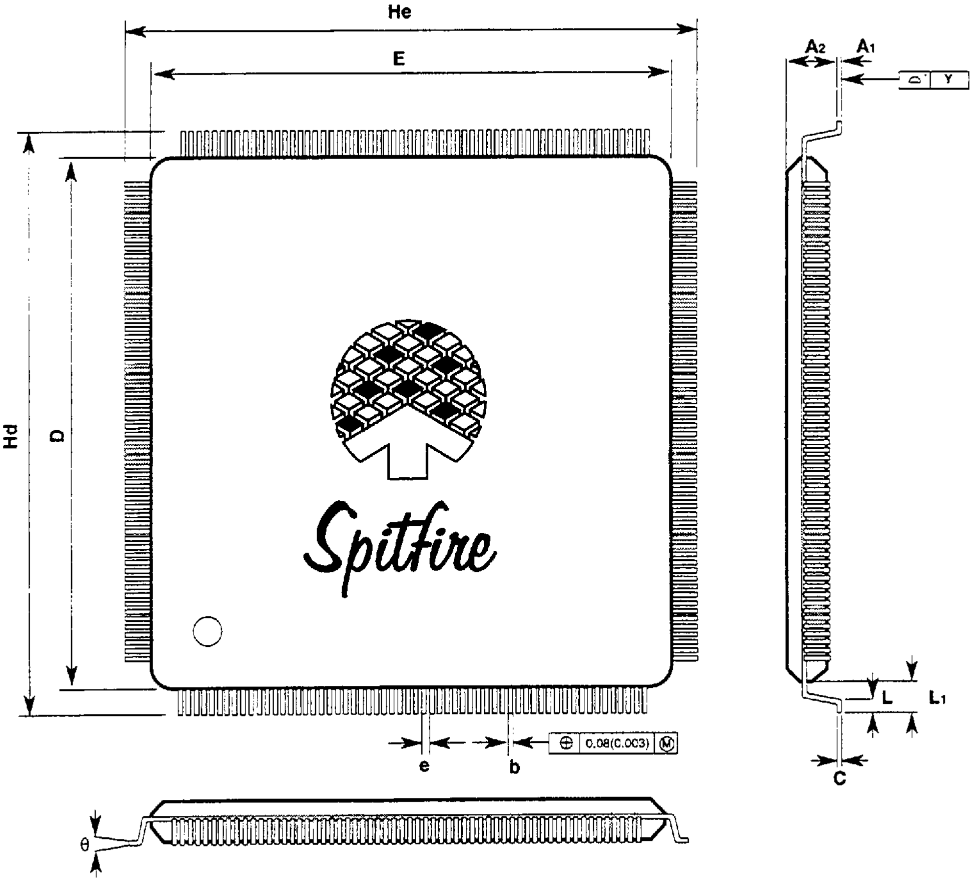

OAK TECHNOLOGY, INC
139 KIFER COURT
SUNNYVALE, CA 94086
Title SAA7186 YUV to RGB video scalar
Size B
Document Number 8071OP
REV A
Date: August 9, 1994 Sheet 6 of 8
IIC JUMPER OPEN SHORT
IICSA == HI LO
I2C BUS ADDRESS BCH B8H
U14
SAA7186
U13D
74LS04
MMDCO..15]
MMDVALID
MMCLK
O/E-
MMVRSET-
MMHRSET-
UVC0..7]
YC0..7]
CREF
LLC
HREF
VS
RESn
SDA
SCL
LLC2
VCC
R41
RES,4.7K,5X
E1
2 PIN JUMPER
RA1
RA2
RA3
RA4
SIP,33,8P/4R
T1 T2 T3 T4 T5 T6
C64 0.1UF
C65 0.1UF
C66 0.1UF
C67 0.1UF
C68 0.1UF
C69 0.1UF



IICSA ==	HI	LO
IIC SLAVE WRITE ADDRESS	9EH	9CH
IIC SLAVE READ ADDRESS	9FH	9DH

CHAPTER 12: PACKAGE DIMENSIONS

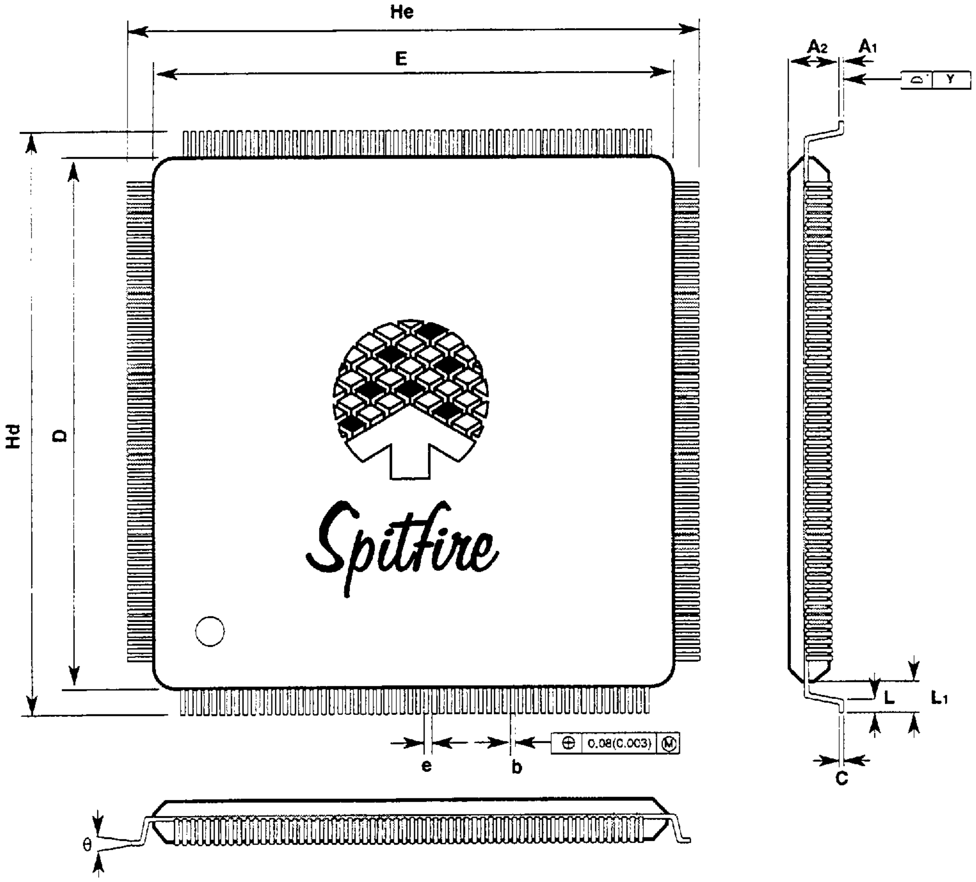
The Spitfire OTI-64107 is packaged in a 240-pin PQFP.



	MILLIMETER			INCH		
SYMBOL	Min.	Nom.	Max.	Min.	Nom.	Max.
A1	0.25	0.35	0.45	0.01	0.014	0.018
A2	3.17	3.32	3.47	0.125	0.131	0.137
b	0.1	0.2	0.3	0.004	0.008	0.012
c	0.1	0.15	0.2	0.004	0.006	0.008
D	31.9	32	32.1	1.256	1.26	1.264
E	31.9	32	32.1	1.256	1.26	1.264
e		0.5			0.02	
Hd	34.35	34.6	34.85	1.352	1.362	1.372
He	34.35	34.6	34.85	1.352	1.362	1.372
L	0.35	0.5	0.65	0.014	0.02	0.026
L1		1.3			0.051	
y			0.08			0.003
q	0°		10°	0°		10°

CHAPTER 12: PACKAGE DIMENSIONS

The Spitfire OTI-64107 is packaged in a 240-pin PQFP.



SYMBOL	MILLIMETER			INCH		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A1	0.25	0.35	0.45	0.01	0.014	0.018
A2	3.17	3.32	3.47	0.125	0.131	0.137
b	0.1	0.2	0.3	0.004	0.008	0.012
c	0.1	0.15	0.2	0.004	0.006	0.008
D	31.9	32	32.1	1.256	1.26	1.264
E	31.9	32	32.1	1.256	1.26	1.264
e		0.5			0.02	
Hd	34.35	34.6	34.85	1.352	1.362	1.372
He	34.35	34.6	34.85	1.352	1.362	1.372
L	0.35	0.5	0.65	0.014	0.02	0.026
L1		1.3			0.051	
y			0.08			0.003
q	0°		10°	0°		10°