Intel740 Graphics Accelerator

Datasheet

February 1998

Order Number: 290618-001

Intel740 Features

- **Hyper Pipelined Architecture**
  - 2X AGP Support
  - Parallel Data Processing (PDP)
  - Precise Pixel Interpolation (PPI)
  - Direct Memory Execution (DME)
  - Sustained 3D Performance
    - –1.1 Meg-Triangles/Second Peak
    - – 425-500K Triangles/Second Full Featured (66 Mega-Pixel Peak)
    - – 45-55 Meg-Pixels/Second Full Featured
  - Full Sideband Accelerated Graphics Port (AGP) Initiator Support
  - Optimized For 440LX Intel AGPsets
- **3D Graphics Visual Enhancements**
  - Flat & Gouraud Shading
  - Mip-Mapping With Bilinear Filtering (11 LODs)
  - Color Alpha Blending For Transparency
  - Real Time Texture Paging and Video Texturing
  - Fogging & Atmospheric Effects
  - Specular Lighting
  - Edge Antialiasing
  - Stippling Or “Screen Door” Transparency
  - Backface Culling
  - Z Buffering
- **3D Graphics Texturing Enhancements**
  - Per Pixel Perspective Correct Texture Mapping
  - Texture Sizes From 1x1 To 1024x1024 pixels (Rect/Sq)
  - Texture Formats
    - – Palletized 1,2,4, or 8-bit
    - – RGBA 1555,565,4444
    - – Compressed 4:2:2, 555, 1544
  - Texture Color Keying
  - Texture Chroma Keying
  - Integrated Hardware Palette
- **Local Memory Controller**
  - 2 to 8 MB SGRAM/SDRAM
  - High-Speed 64-Bit 100 MHz Interface
- **Display/BIOS/I^2C**
  - Integrated 24-bit 220 MHz RAMDAC
  - Pixel Palette Pipeline Frequency: 203 MHz
  - Display Resolution: 320x200 Up To 1600x1200 Pixels
  - Gamma Corrected Video
  - Multi-Monitor Capability
  - Read/Write Access To 256KB Flash
  - ACPI Compliant
  - Display Data Channel: DDC 2B Compliant
- **2D Graphics**
  - Up To 1600x1200 In 8 Bit Color Format at 75 MHz
  - Up to 1024x768 in 8/16/24 Bit Color Format @ 85 MHz Refresh Rate
  - Hardware Accelerated Functions
    - – Programmable 64x64 3 Color Transparent Cursor
    - – 3 Operand Raster BitBLTs
    - – StretchBLT
    - – Color Expansion
  - Hardware Overlay Engine
  - Hardware Double Buffering
- **Video**
  - Software DVD MPEGII Ready
  - Hardware DVD MPEGII Capable
  - Video Capture Support (16 Bit Or 8 Bit)
  - Full Bi-directional VMI CCIR601 Video Port
  - Programmable Video Output Characteristics For VGA and SVGA
  - Full Motion Video
  - Video Conferencing Support
  - Intercast & VBI Support
  - NTSC And PAL TV Out Support
- **OS & Driver Support**
  - DirectX* 5.0
  - OpenGL*
    Full 2X AGP Support
  - Windows* 95
  - Windows* 95 AGP VxD
  - Windows NT* 4.0
  - Windows* 98
  - Windows NT* 5.0
- **468 Pin STBGA (Super Thermal Ball Grid Array) Package**

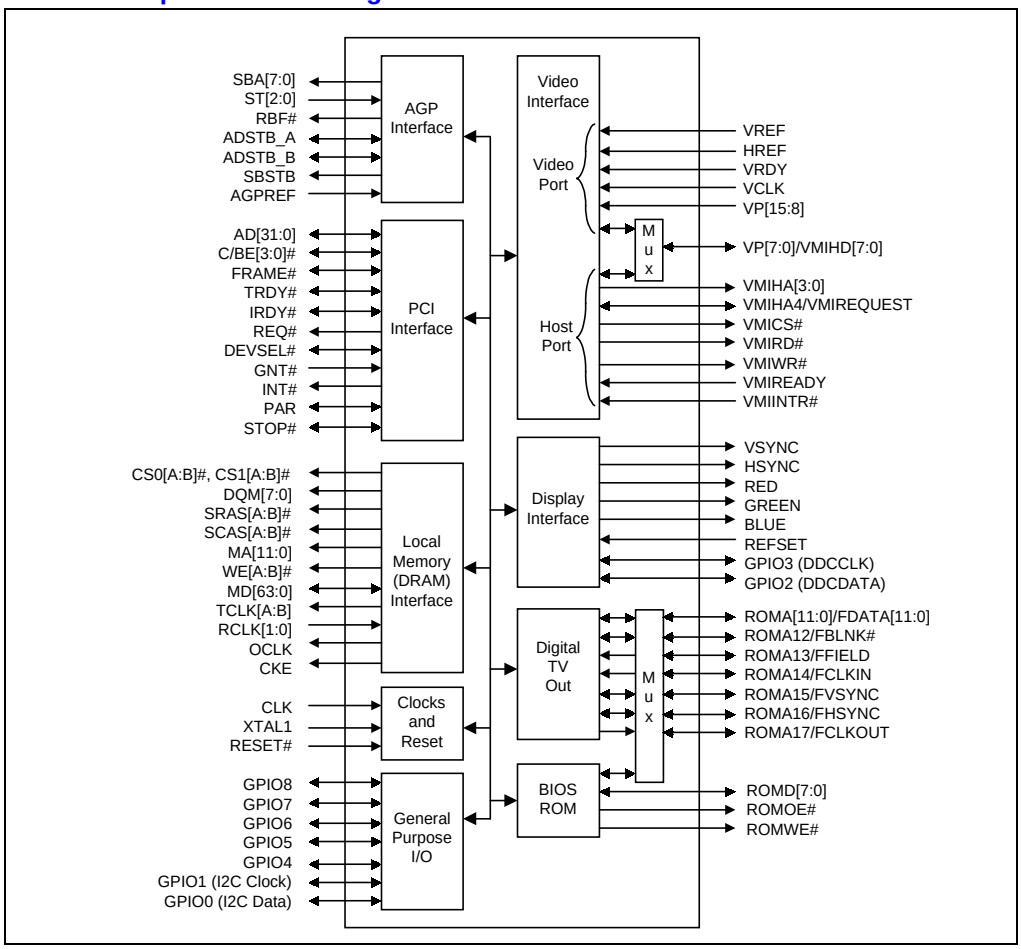
The Intel740 may contain design defects or errors known as errata which may cause the products to deviate from published specifications. Current characterized errata are available on request.

The Intel740 is a highly integrated graphics accelerator designed for the Accelerated Graphics Port (AGP). Its architecture consists of dedicated multi-media engines executing in parallel to deliver high performance 3D, 2D and video capabilities. The 3D and 2D engines are managed by a 3D/2D pipeline preprocessor allowing them a sustained flow of graphics data. The 3D engine has been architected as a deep pipeline, where performance is maximized by parallel data paths and allowing each pipeline stage to simultaneously operate on different primitives or portions of the same primitive. The Intel740 supports perspective-correct texture mapping, bilinear MIP-Mapping, Gouraud shading, alpha-blending, stippling, antialiasing, fogging and Z Buffering. A rich set of 3D instructions permit these features to be independently enabled or disabled. Textures can be located in AGP memory to free up local memory for other uses (e.g., back and Z Buffers, bitmaps, etc.).

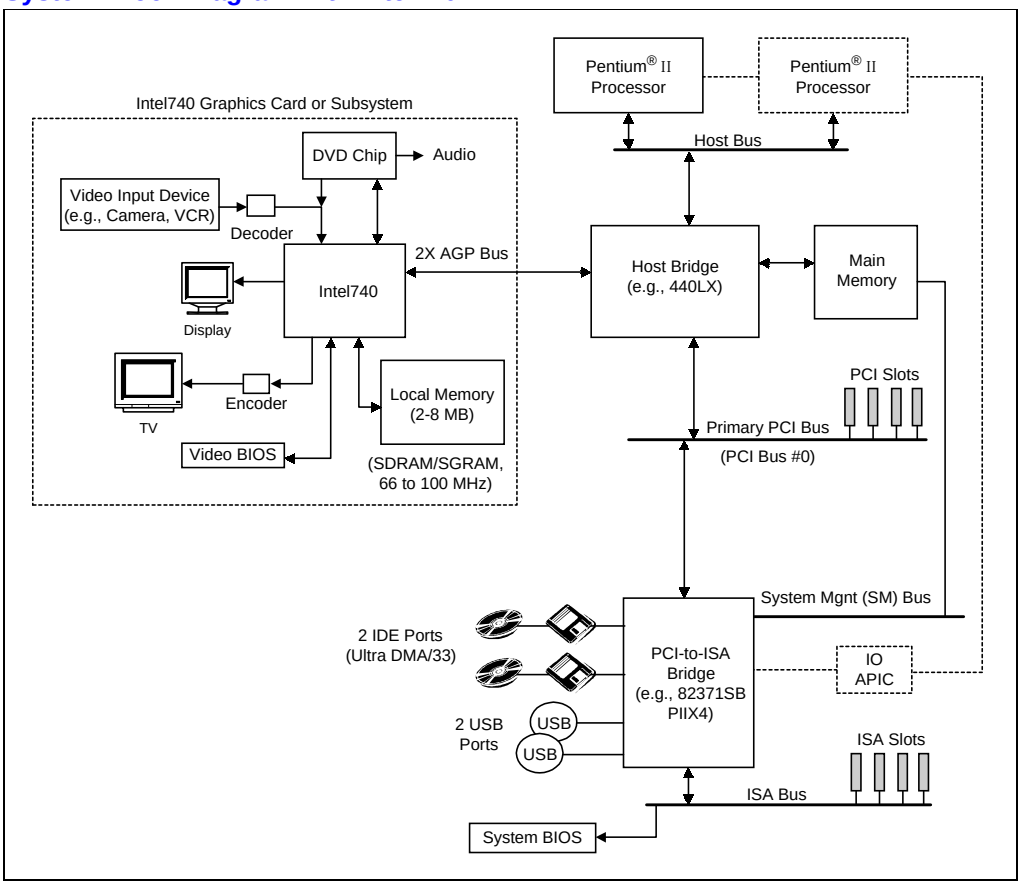
The Intel740 2D capabilities includes BLT and STRBLT engines, a hardware cursor, and an extensive set of 2D registers and instructions. The high performance 64-bit BitBLT engine provides hardware acceleration for many common Windows* operations.

The Intel740 also includes dedicated video engines for support of video conferencing and other video applications. The Intel740's Video Module Interface (VMI) has both a Host Port and Video Port that provide the interface to the DVD chip. A full bi-directional VMI Host Port is supported for low-cost hardware MPEGII DVD decode. The Video Port interface also provides video capture for video input devices. The Video interface includes support for full motion video, MPEGII/DVD, Intercast and VBI and NTSC and PAL TV out.

Intel740 Simplified Block Diagram



System Block Diagram with Intel740



Contents

Figures

Tables

Architectural Overview 1

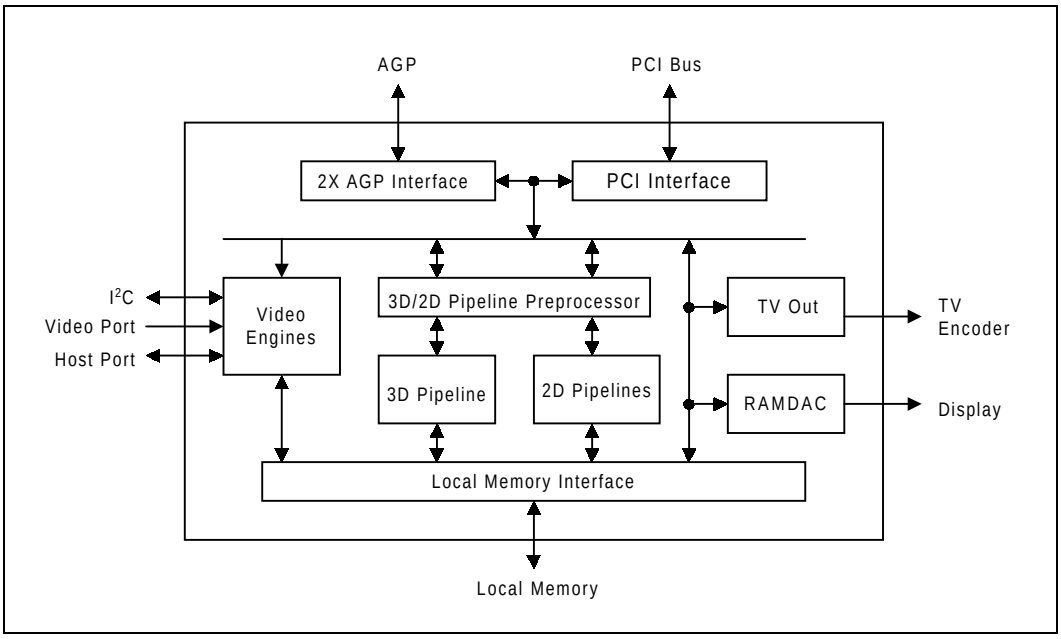
The Intel740 is a highly integrated graphics accelerator designed for the Accelerated Graphics Port (AGP). Its architecture consists of dedicated multi-media engines executing in parallel to deliver high performance 3D, 2D and video capabilities. The 3D and 2D engines are managed by the 3D/2D pipeline preprocessor allowing them a sustained flow of graphics data. The Intel740 also includes dedicated video engines for support of video conferencing and other video applications. Each of these functional units is described in detail in the *Functional Description* Chapter.

1.1 3D Engine

The Intel740 is capable of delivering a high rate of sustained 3D graphics performance with full 3D feature set functionality. This constant high level of performance is delivered through the Intel740's hyper-pipelined 3D architecture and the incorporation of specific graphics architectural enhancements. With the use of Direct Memory Execution (DME), the Intel740 fully utilizes the bandwidth of AGP and memory, benefiting the heavy data demands of 3D. DME is a technique that allows the Intel740 to store and execute textures in system memory instead of local graphics memory. This provides high levels of performance and unlimited texture sizes.

Architectural enhancements within the 3D pipeline ensure that the Intel740 uses this data in the most efficient way possible. Parallel Data Processing (PDP) allows several commands to be executed at the same time in the graphics pipeline. This translates into consistent high-performance regardless of the number of features enabled in a scene. Precise-Pixel Interpolation (PPI) contributes to the hyper-pipelined 3D quality with the Intel740's unique texture engine that delivers precise accuracy in interpolation operations of pixel values and color values. This detailed pixel processing maintains a high level of image quality in every scene.

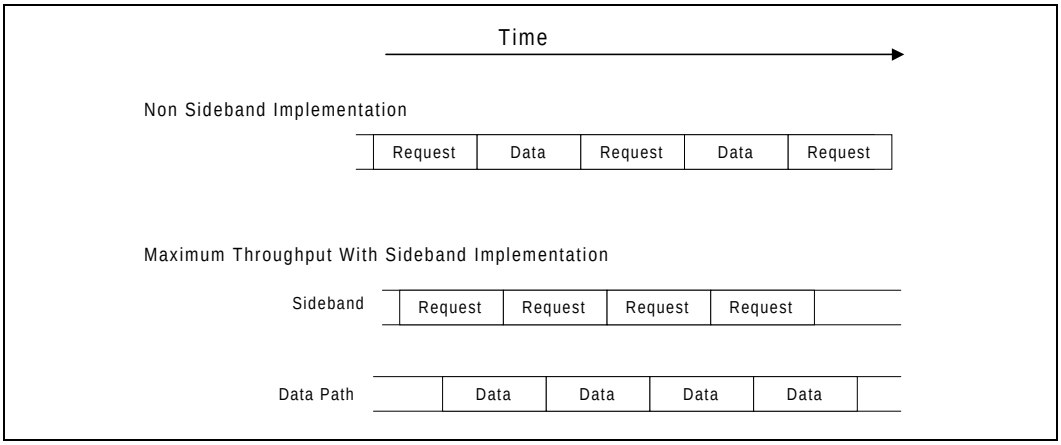
Figure 1-1. The Intel740 Architectural Interfaces



The DME architecture means that full 2X AGP implementation is integrated into the Intel740 with sideband operations supporting Type 1, Type 2, and Type 3 sideband cycles. This allows 533 MB/s peak data transfers. Type 3 support permits textures to be located anywhere in the 32-bit system memory address space. Deep buffering allows the Intel740 to receive data at this high rate and handle any latencies associated with AGP transactions.

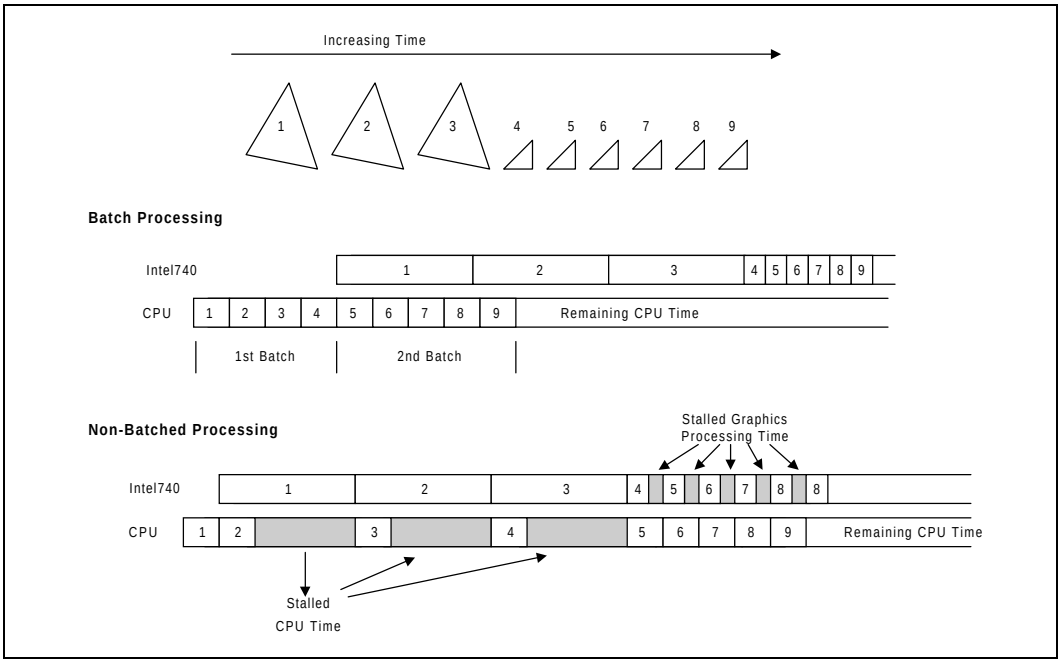
Sideband addressing gives the Intel740 the ability to issue multiple requests without having to wait for data to be returned. This allows the Intel740 to achieve the highest possible sustained data transfer rates across 2X AGP and makes DME possible.

Figure 1-2. The Intel740 Implementation of Sideband Addressing



To provide the highest level of system concurrency and performance the Intel740 is optimized for a batch processing mode of triangle delivery. Batch processing frees up the CPU for intelligent 3D gaming and more complex geometry processing. This batch processing allows the CPU to place a “batch” of triangles in memory and begin on another batch of triangles without the need to perform handshaking with the Intel740.

Figure 1-3. Batch Processing on the Intel740—A Conceptual View



The DME capabilities of the Intel740 maximize the amount of memory available for rendering (Figure 1-4). The Intel740 is capable of executing directly from AGP memory. This “direct execution” avoids the “thrashing” of local memory associated with an architecture that must load local memory from AGP or system memory. As such, textures can be executed directly from AGP memory allowing performance to be sustained even when the texture footprint increases.

As Figure 1-5 indicates, the Intel740 is capable of rendering from local memory while textures are being executed from AGP memory through parallel arbitration. This arbitration allows a combined memory peak bandwidth of 1.3 GB/s. The capability to support two open pages in local memory coupled with an additional memory channel in AGP memory supports the 3D rendering model of color (front/back buffers), z, and textures. The Intel740 also supports 2D rendering through the use of three raster operands (pattern, source and destination).

Figure 1-4. The Intel740’s Ability to Execute Textures Directly From AGP Memory

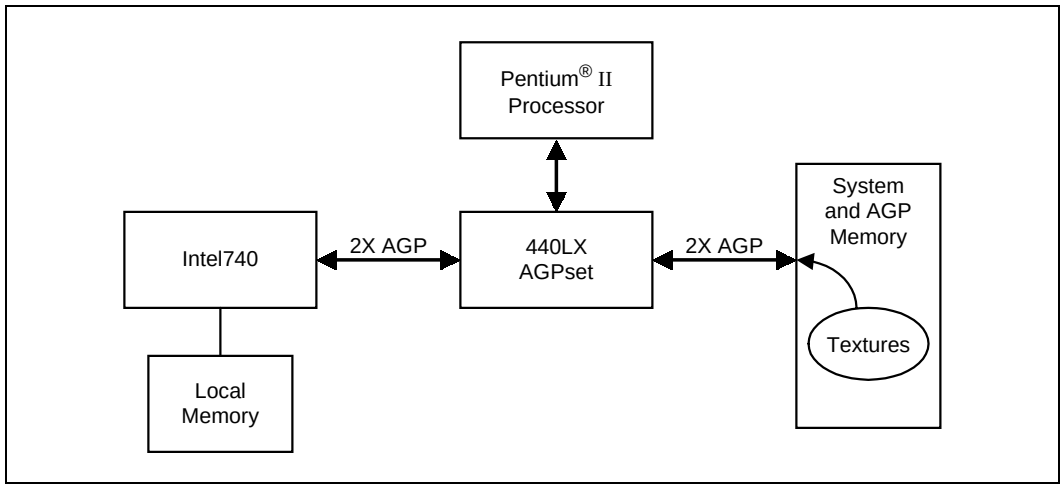
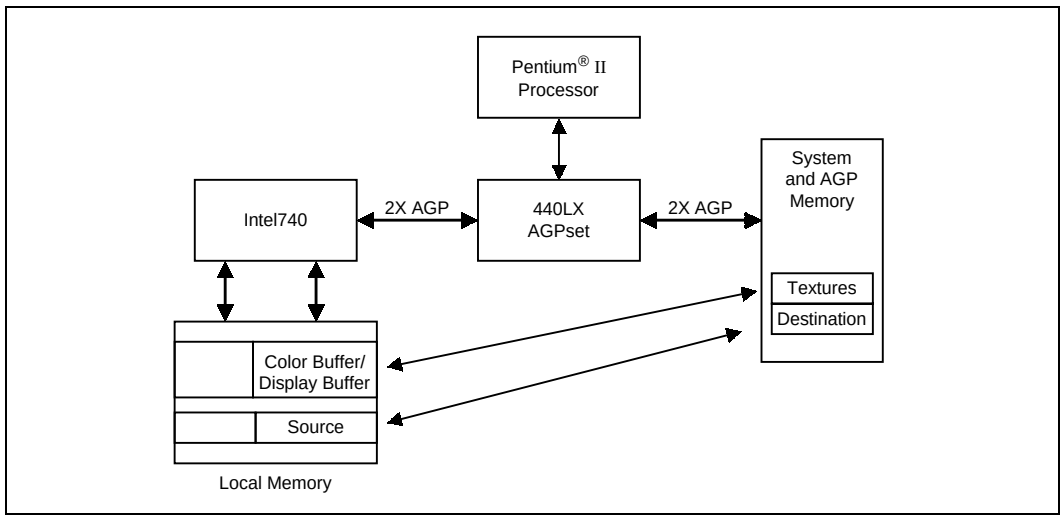


Figure 1-5. The Intel740 Functioning as Two Memory Controllers



Included in the Intel740's architecture are dedicated 3D pipeline enhancements. These enhancements are designed to manage the way in which 3D data is requested from memory and then used within the compute engine. While parallelism is employed among each of the Intel740's engines, the 3D pipeline calculates 3D data in a highly parallel fashion. With this architecture, the 3D rasterizer is able to compute four fully textured, shaded, fogged and Z Buffered pixels per clock. The 3D pipeline requests data from memory so that memory locality is maximized, regardless of triangle size or orientation. This results in fewer page misses, higher cache efficiency, and a highly sustained 3D graphics output independent of the complexity of the 3D scene being rendered. By combining memory efficiencies and processing data efficiencies, the Intel740 is capable of a high rate of sustained 3D performance.

1.2 2D Engine

The Intel740's 64-bit BitBLT engine provides hardware acceleration for many common Windows operations. There are two primary BitBLT functions: Fixed BitBLT (BLT) and Stretch BitBLT (STRBLT). The term BitBLT refers to block transfers of pixel data between memory locations. Use of the BLT engine accelerates the Graphical User Interface (GUI) of Microsoft* Windows. Hardware is included for all 256 Raster Operations (ROPs) defined by Microsoft*, including transparent BitBLT. The BLT engines can be used for various functions including:

- Moving rectangular blocks of data between memory locations
- Pixel format conversion
- Data Alignment
- Performing logical operations

The Intel740 uses instructions to invoke BLT and STRBLT operations, permitting software to set up instruction buffers and use batch processing as described in the *3D/2D Instruction Processing (Pipeline Preprocessor)* Section. Like the 3D engine described in the previous section, 2D batch processing takes full advantage of 2X AGP and frees up the CPU. Deep buffering allows the Intel740 to receive data at a high rate and handle any latencies associated with AGP transactions.

1.3 Video Module Interface (VMI)

The Intel740 VMI consists of a Video Port and a Host Port. The Host Port provides an enhanced VMI 1.4 Mode B Port. The enhancements allow burst modes of operation. The Intel740 Video Port is used to receive decompressed video data from a DVD chip or video data from a Video Decoder chip. A CCIR601 digital interface is supported as the primary capture standard. Using both the host and video ports, DVD, TV, Intercast, and video capture can be achieved. Use of the Intel740 overlay capability allows images from the capture engine to be displayed while being captured.

1.4 Digital TV Out

The Intel740 TVout port provides a digital output interface to either a television or monitor. The interface has a 12-bit data bus and connects to a television via a standard TV encoder chip (e.g., a Brooktree* BT869). Output to the encoder is in digital 24-bit RGB format. The following resolutions are supported:

- 320x200
- 320x240
- 640x400
- 640x480
- 720x480
- 800x600

1.5 Display

The display function contains a RAM-based Digital-to-Analog Converter (RAMDAC) that transforms the digital data from the graphics and video subsystems to analog data for the monitor. The following display modes are supported.

Table 1-1. Display Modes Supported

Resolution	Bits Per Pixel (frequency: Hz)		
	8-bit Indexed	16-bit	24-bit
320x200	60,72,75,85	60,72,75,85	60,72,75,85
320x240	60,72,75,85	60,72,75,85	60,72,75,85
512x384	60,72,75,85	60,72,75,85	60,72,75,85
640x350	85	85	85
640x480	60,72,75,85	60,72,75,85	60,72,75,85
800x600	56,60,72,75,85	56,60,72,75,85	56,60,72,75,85
1024x768	60,70,75,85	60,70,75,85	60,70,75,85
1280x1024	60,72,75,85	60,72,75	—
1600x1200	60,75	—	—

Signal Description 2

This section provides a detailed description of each signal for the Intel740. The signals are arranged in functional groups according to their associated interface.

The "#" symbol at the end of a signal name indicates that the active, or asserted, state occurs when the signal is at a low voltage level. When "#" is not present after the signal name the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of "active-low" and "active-high" signals. The term **assert**, or **assertion** indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term **negate**, or **negation** indicates that a signal is inactive.

The signal description also includes the type of buffer used for the particular signal:

I Input pin

O Output pin

OD Open Drain Output pin. This pin requires a pull-up to the VCC of the processor core

I/O Bi-directional input/output pin

The signal description also includes the type of buffer used for the particular signal:

AGP AGP/PCI-66/133MHz 3.3V Signaling Environment DC and AC Specifications.

LVTTL Low Voltage TTL compatible signals. These are also 3.3V inputs and outputs.

Table 2-1. Basic PCI/AGP Signals (Sheet 1 of 2)

Name	Type	Description
AD[31:0]	I/O, AGP	**PCI Address/Data:** When the Intel740 acts as a target on the PCI Bus, the AD[31:0] signals are inputs for address reads/writes and write data, and outputs for read data. AD[31:0] contain the address during the first clock of FRAME# assertion and data on subsequent clocks. Note that AD16 is also used as the IDSEL function. **AGP Address/Data:** The Intel740 acts as an AGP master. Data is driven or received by the Intel740 after appropriate commands are enqueued to the host-to-PCI bridge (via the side band address bus).
DEVSEL#	I/O, AGP	**PCI Device Select:** Remains as DEVSEL# when a PCI transaction is initiated (FRAME# is asserted). DEVSEL#, when asserted, indicates that the Intel740 as a PCI target device has decoded its address as the target of the current access. When the Intel740 is a PCI master, DEVSEL# is asserted by the system memory controller when it has decoded its address as the target of the current access. **AGP:** Not Used on AGP
FRAME#	I/O, AGP	**PCI Frame:** The Intel740 latches the C/BE[3:0]# and the AD[31:0] signals on the first clock edge on that it samples FRAME# asserted. FRAME# is negated by the initiator and remains negated by its own on-board pull up resistor. **AGP:** Not Used on AGP

Table 2-1. Basic PCI/AGP Signals (Sheet 2 of 2)

Name	Type	Description
IRDY#	I/O, AGP	**PCI Initiator Ready:** The assertion of IRDY# indicates the current PCI Bus initiator's ability to complete the current data phase of the transaction. **AGP Initiator Ready:** IRDY# assertion for reads indicates the AGP master can accept the next 32 bytes of read data. The master is not allowed to insert wait states during the first 32 bytes of a read transaction. However, it can insert wait states after each 32 byte block is transferred. IRDY# assertion for writes indicates that the master is ready to transfer all write data for the current transaction. Once IRDY# is asserted for writes the master is not allowed to insert wait states, (There is no FRAME# -- IRDY# relationship.)
C/BE[3:0]#	I/O, AGP	**PCI Command/Byte Enable:** PCI Bus Command and Byte Enable signals are multiplexed on the same pins. During the address phase of a transaction, C/BE[3:0]# define the bus command. During the data phase C/BE[3:0]# are used as byte enables. The byte enables determine which byte lanes carry meaningful data. **AGP Byte Enable:** Provides valid byte information during AGP write transactions and are not used during the return of AGP read data.
TRDY#	I/O, AGP	**PCI Target Ready:** The assertion of TRDY# indicates the target's ability to complete the current data phase of the transaction. **AGP Target Ready:** TRDY# assertion indicates that the AGP target is ready to provide read data for the entire transaction (when the transfer size is less than or equal to 32 bytes) or is ready to transfer the initial or subsequent block (32 bytes) of data, when the transfer size is greater than 32 bytes. The target is allowed to insert wait states after each block is transferred on both read and write transactions.
PAR	I/O, AGP	**Parity:** PAR is “even” and is calculated on 36 bits (AD[31:0] and C/BE[3:0]#). Even parity means a count of the number of “1s” within the 36 bits plus PAR should be even. PAR is always calculated on 36 bits regardless of the valid byte enables. PAR drives PAR for read data phase and is only guaranteed to be valid for 1 PCI clock after the data phase. PAR is driven and tri-stated identically to the AD[31:0] lines except for this extra 1 clock delay. **AGP:** This signal is not used for AGP cycles.
STOP#	I/O, AGP	**PCI Stop:** This signal remains as STOP# when a PCI transaction is initiated (FRAME# is asserted). **AGP:** This signal is not used for AGP cycles.
REQ#	O, AGP	**PCI Request:** This signal Indicates to the arbiter that the Intel740 wants to use the PCI bus AGP Request. **AGP Request:** Sideband signals are used.
GNT#	I, AGP	**PCI Grant:** Indicates to the Intel740 that it has been granted access to the bus. **AGP Grant:** ST[2:0] data is only valid when GNT# is asserted. When GNT# is negated, these signals have no meaning and must be ignored.
INT#	O, AGP	**PCI and AGP INT:** INT# is used to request an interrupt.

NOTE: All PCI signals behave according to the PCI 2.1 specification.

Table 2-2. AGP Specific Signals

Name	Type	Description
AGP Addressing (Sideband Signals)		
SBA[7:0]	O, AGP	**Side Band Address:** The SBA[7:0] bus is used to pass address and command to the target from the master. The Intel740 does not use the PIPE# protocol; thus, this is the only way of enqueueing requests to the AGP target.
AGP Flow Control and Status Signals		
RBF#	O, AGP	**Read Buffer Full:** This signal indicates if the Intel740 is ready to accept previously requested low priority read data or not. When asserted, the arbiter is not allowed to return low priority read data to the Intel740.
ST[2:0]	I, AGP	**Status Bus:** ST[2:0] provide information from the AGP arbiter to a master on what it may do. ST[2:0] have meaning to the Intel740 when its GNT# is asserted. When GNT# is negated, these signals have no meaning and must be ignored. 000 Indicates that previously requested low priority read data is being returned to the master. 001 Indicates that previously requested high priority read data is being returned to the master. 010 Indicates that the master is to provide low priority write data for a previous enqueued write command. 011 Indicates that the master is to provide high priority write data for a previous enqueued write command. This is not used by the Intel740 and will not be responded to. 100 Reserved 101 Reserved 110 Reserved 111 Indicates that the master has been given permission to start a bus transaction. In particular, this encoding is recognized by the Intel740 PCI as a grant to be a PCI master. The Intel740 does not use this functionality to post its status word to the host memory via PCI.
AGP Clocking Signals		
ADSTB_A	I/O, AGP	**AD Bus Strobe:** This signal provides timing for double clocked data on the AD[15:0] bus.
ADSTB_B	I/O, AGP	**AD Bus Strobe:** This signal is a copy of ADSTB_A that provides timing for double clocked data on the AD[31:16] bus.
SBSTB	O, AGP	**Side Band Strobe.** This signal provides timing for SBA[7:0].
AGPREF	Ref	**Analog Reference:** This is the analog reference voltage for the AGP port. This pin should be connected to a reference voltage source of 0.4 VDDQ which is used to determine the trip point of the AGP buffers.

Table 2-3. DRAM (Local Memory) Interface Signals

Name	Type	Description
CS0[A:B]#/ CS1[A:B]#	O, LVTTL	**Chip Select (SDRAM):** For the memory row configured with SDRAM, these signals select the particular SDRAM components during the active state. CSxA# and CSxB# are multiple copies of the same signal.
DQM[7:0]	O, LVTTL	**Input/Output Data Mask (SDRAM):** These signals control the memory array and act as synchronized output enables during read cycles and as a byte enables during write cycles. For write cycles, byte masking functions are performed during the same clock when write data is driven (i.e., 0 clock latency).
SRASA#/ SRASB#	O, LVTTL	**SDRAM Row Address Strobe (SDRAM):** SRASA# and SRASB# are multiple copies of the same logical signal (for loading purposes) used to generate SDRAM commands encoded on the SRASx/SCASx/WEx signals. When SRAS# is sampled active at the rising edge of the SDRAM clock, the row address is latched into the SDRAMs. These signals drive the SDRAM array directly without any external buffers.
SCASA#/ SCASB#	O, LVTTL	**SDRAM Column Address Strobe (SDRAM):** SCASA# and SCASB# are multiple copies of the same logical signal (for loading purposes) used to generate SDRAM commands encoded on SRAS#/SCAS#/WE# signals. When SCAS# is sampled active at the rising edge of the SDRAM clock, the column address is latched into the SDRAMs. These signals drive the SDRAM array directly without any external buffers.
MA[11:0]	O, LVTTL	**Memory Address(SDRAM):** MA[11:0] are used to provide the multiplexed row and column addresses to DRAM.
WEA#/ WEB#	O, LVTTL	**Write Enable Signal (SDRAM):** WEx# is asserted during writes to DRAM. WEA# and WEB# are multiple copies of the same signal.
MD[63:0]	I/O, LVTTL	**Memory Data (SDRAM):** These signals provide a 64-bit data path to the DRAM array. Note that these signals are internally connected to 20 KΩ (nominal) pull-down resistors to prevent the memory data bus from floating when no DRAM is active.
TCLK[A:B]	O, LVTTL	**Transmit Clock for SDRAM:** There are 2 copies of TCLK.
RCLK0	I, LVTTL	**Receive Clock Low:** RCLK0 is a copy of OCLK appropriately phase adjusted to match the SDRAM/board timings. RCLK0 is used to latch the lower 32 bits of data from SDRAM.
RCLK1	I, LVTTL	**Receive Clock High:** RCLK1 is a copy of OCLK appropriately phase adjusted to match the SDRAM/board timings. RCLK1 is used to latch the upper 32 bits of data from SDRAM.
OCLK	O, LVTTL	**Output Clock:** OCLK is a copy of TCLK and is used to derive the RCLKxs.
CKE	O, LVTTL	**Clock Enable:** This signal is used for SDRAM powerdown control. All SDRAM banks must be precharged before CKE is negated.

Table 2-4. BIOS ROM Signals

Pin Name	Type	Description
ROMA[17:0]/ FDATA[11:0}, FBLNK# FFIELD FCLKIN, FVSYNC#, FHSYNC#, FCLKOUT	I/O	**BIOS ROM Address (Outputs):** The Intel740 can access up to a 256 KB of dedicated BIOS ROM through ROMA[17:0]. ROMA[15:0] are externally strapped to encode the subsystem vendor ID (see *Strapping Option* section for details). Note that these signals are internally connected to pull-down resistors (20 KΩ nominal value). The ROMA[17:0] pins are multiplexed with the Digital TV Out signals.
ROMD[7:0]	I/O	**BIOS ROM Data:** The Intel740 is capable of reading or writing data to the BIOS with these pins.
ROMOE#	O	**BIOS ROM Output Enable:** This signal selects a read of BIOS ROM.
ROMWE#	O	**BIOS ROM Write Enable:** This signal selects a write of BIOS ROM.

Table 2-5. Video Interface Signals (Sheet 1 of 2)

Pin Name	Type	Description
VMI Video Port		
VREF	I, LVTTL	**Vertical Reference Input:** Vertical reference input for video capture.
HREF	I, LVTTL	**Horizontal Reference Input:** Horizontal reference input for video capture.
VRDY	I, LVTTL	**Video System Ready:** This input signal indicates that the video system has placed valid data on the VP[15:0] bus.
VCLK	I, LVTTL	**VAFC Video Input Clock:** VCLK is the continuous master clock driven from the video system.
VP[15:8]	I, LVTTL	**Video Capture Data Port (Upper Byte):** VP[15:8] are the video data inputs to the Intel740. They are used when the video capture port is in 8-bit mode. For 16-bit mode, VP[15:0] are used. VP[15:0] also provide strapping options for the subsystem identification which the Intel740 recognizes during a hard reset (see Strapping Options section for details). Note that these signals are internally connected to 20 KΩ (nominal) pull-down resistors. This signal is tri-stated during a hard reset.
VP[7:0]/ VMIHD[7:0]	I/O, LVTTL	**Video Capture Data Port (Lower Byte):** VP[7:0] are the video data inputs to the Intel740. This pins are shared with the bi-directional VMI host port data signals (VMIHD[7:0]). VP[15:0] also provide strapping options for the subsystem identification which the Intel740 recognizes during a hard reset (see Strapping Options section for details). Note that these signals are internally connected to 20 KΩ (nominal) pull-down resistors. This signal is tri-stated during a hard reset.
VMI Host Port		
VP[7:0]/ VMIHD[7:0]	I, LVTTL	**VMI Host Data:** VP[15:8] are the video data input/output signals to the Intel740. These pins are shared with the VMI video capture port input signals (VP[7:0]). VP[15:0] also provide strapping options for the subsystem identification which the Intel740 recognizes during a hard reset (see Strapping Options section for details). Note that these signals are internally connected to 20 KΩ (nominal) pull-down resistors.

Table 2-5. Video Interface Signals (Sheet 2 of 2)

Pin Name	Type	Description
VMIHA[3:0] VMIHA4/ VMIREQUEST	O, LVTTL I/O, LVTTL	**VMI Host Address:** The VMIHA[4:0] output signals provide the host address. **VMI Request:** VMIREQUEST is a data request to the Intel740 for DVD DMA data.
VMICS#	O, LVTTL	**VMI Chip Select:** The Intel740 asserts this signal for read/write operations to the DVD chip.
VMIRD#	O, LVTTL	**VMI Read:** This signal indicates a VMI read cycle.
VMIWR#	O, LVTTL	**VMI Write:** This signal indicates a VMI write cycle.
VMIREADY	I, LVTTL	**VMI Ready:** VMIREADY indicates to the Intel740 that the DVD device is ready for data. This signal is used to insert wait states.
VMIINTR#	I, LVTTL	**VMI Interrupt Request:** This signal provides a VMI interrupt request from the DVD chip to the Intel740.

Table 2-6. Display Interface Signals

Pin Name	Type	Description
VSYNC	O	**CRT Vertical Synchronization:** This signal is used as the vertical sync (polarity is programmable) or "Vsync Interval".
HSYNC	O	**CRT Horizontal Synchronization:** This signal is used as the horizontal sync (polarity is programmable).
RED	O, Analog	**Red (Analog Video Output):** This signal is a CRT Analog video output from the internal color palette DAC. The DAC is designed for a 37.5Ω equivalent load on each pin (e.g., 75Ω resistor on the board, in parallel with the 75Ω CRT load).
GREEN	O, Analog	**Green (Analog Video Output):** This signal is a CRT analog video output from the internal color palette DAC. The DAC is designed for a 37.5Ω equivalent load on each pin (e.g., 75Ω resistor on the board, in parallel with the 75Ω CRT load).
BLUE	O, Analog	**Blue (Analog Video Output):** This signal is a CRT Analog video output from the internal color palette DAC. The DAC is designed for a 37.5Ω equivalent load on each pin (e.g., 75Ω resistor on the board, in parallel with the 75Ω CRT load).
REFSET	I, NA	**Resistor Set:** Set point resistor for the internal color palette DAC. A 560 Ω 1% resistor is required between REFSET and VSSA.
GPIO3 **(DDCCLK)**	I/O	**CRT Monitor DDC Interface clock.** This GPIO signal is configurable as a CRT Monitor DDC Interface clock. (Also referred to as VESA* “Display Data Channel”; also referred to as the “Monitor Plug-n-Play” interface.) For DDC1, DCLK and DDAT provide a unidirectional channel for Extended Display ID. For DDC2, DCLK and DDAT can be used to establish a bi-directional channel based on the I^2C protocol. The host can request Extended Display ID or Video Display Interface information over the DDC2 channel.
GPIO2 **(DDCDATA)**	I/O	**CRT Monitor DDC Interface data.** This GPIO signal is configurable as CRT Monitor DDC Interface Data.

Table 2-7. Digital TV Out Signals

Pin Name	Type	Description
ROMA[11:0]/ **FDATA[11:0]**	I/O, LVTTL	**TV Data Out:** Data to be output to the TV encoder.
ROMA14/ **FCLKIN**	I, LVTTL	**TV Pixel Clock Input:** Pixel Clock from TV encoder.
ROMA16/ **FHSYNC**	I/O, LVTTL	**Horizontal Sync Input:** HSYNC signal for TV out interface. The high/low active state is programmable.
ROMA15/ **FVSYNC**	I/O, LVTTL	**Vertical Sync Input:** VSYNC signal for TV out interface. The high/low active state is programmable.
ROMA13/ **FFIELD**	I, LVTTL	**Field Control Input:** This signal indicates the field polarity. The polarity is programmable.
ROMA12/ **FBLNK#**	I/O, LVTTL	**Composite Blanking Control Output:** Indicates active pixels when FBLNK# is negated. FBLNK# is sampled on FCLKOUT edges.
ROMA17/ **FCLKOUT**	O, LVTTL	**TV Pixel Clock Output:** Pixel clock sent with data to ensure proper latching of data.

NOTE: The Digital TV Out signals are multiplexed with the BIOS ROM signals.

Table 2-8. General Purpose Input/Output Signals

Pin Name	Type	Description
GPIO0 (I^2C Data)	I/O, LVTTL	**General Purpose Input/Output 0:** This signal is a standard GPIO signal. However, for Intel board and software driver designs, GPIO0 is used for I^2C data. This signal is tri-stated during a hard reset.
GPIO1 (I^2C Clock)	I/O, LVTTL	**General Purpose Input/Output 1:** This signal is a standard GPIO signal. However, for Intel board and software driver designs, GPIO1 is used for the I^2C clock. This signal is tri-stated during a hard reset.
GPIO[3:2]	I/0, LVTTL	**General Purpose Input/Output [3:0]:** These are standard configurable GPIO signals. However, for Intel's board design and software driver design these signals have dedicated uses. See the Display Interface Signal description (Table 2-6) for the dedicated uses of GPIO[3:2].
GPIO4	I/0, LVTTL	**General Purpose Input/Output 4:** This configurable GPIO signal has added functionality. It has added interrupt functionality such that a high to low transition can trigger an interrupt event. For example, it can be used for thermal fault sensing, or other events that do not occur during powerdown modes. This signal is tri-stated during a hard reset.
GPIO5	I/0, LVTTL	**General Purpose Input/Output 5:** GPIO6 is a standard configurable GPIO signal. A suggested use is for Fan Control. This signal is tri-stated during a hard reset.
GPIO6	I/0, LVTTL	**General Purpose Input/Output 6:** GPIO6 is a standard configurable GPIO signal. A suggested use is to control the video capture port mux from VMI devices. This signal is tri-stated during a hard reset.
GPIO7	I/0, LVTTL	**General Purpose Input/Output 7:** GPIO7 is a standard configurable GPIO signal (no added functionality). This signal is tri-stated during a hard reset.
GPIO8	I/0, LVTTL	**General Purpose Input/Output 8:** This configurable GPIO signal has added functionality. GPIO8 defaults to tracking the D3 state for Sleep Mode control for Video BIOS. This signal is low during a hard reset.

Table 2-9. Clock and Reset Signals

Pin Name	Type	Description
CLK	I, AGP	**Clock:** Provides timing for AGP and PCI control signals, and is the input clock for most of the Intel740, except those parts that are controlled from VREFCLK.
XTAL1	I	**Reference Oscillator Clock:** This signal is an input for 66.667 MHz.
RESET#	I, AGP	**Reset:** PCI Bus Reset forces the Intel740 to a known state.

Table 2-10. Supply, Ground, and Test Signals

Pin Name	Type	Description
GNDIN	I	**DAC Ground Input:** Provides a ground path for current not needed by DACs. This signal should be connected to VSS ground plane.
VCCA VSSA	I	**Analog Power/Ground:** These signals are the power supply (VCCA) and ground (VSSA) for the DAC and Phase Lock Loop. They provide power supply isolation for the DAC and clock synthesizers. VCCA should be connected to 3.3 volts.
VCCP	I	**Digital Power (Peripheral):** VCCP is the power supply for the peripheral signals, except for the AGP signals. VCCP should be connected to 3.3 volts.
VDDQ	I	**Digital Power Supply (AGP Bus):** VDDQ should be connected to 3.3 volts.
VCCDP	I	**Digital Power Supply (PLL):** This signal is the digital power supply for the on-chip PLL and should be connected to 3.3 volts.
VCCAP	I	**Analog Power Supply (PLL):** This signal is the analog power supply for the on-chip PLL and should be connected to 3.3 volts.
VCC	I	**Digital Power Supply (Core):** VCC should be connected to 2.7 volts.
VSS	I	**Digital Power Supply Ground:** This is the ground for the digital circuitry.
TEST	I	**Test Enable:** This input, when asserted, enables the NAND Tree test mode.

2.1 Strapping Options

The Intel740 provides strapping options for the Subsystem Vendor ID and the Subsystem ID. These values can be read via the SVID register (PCI offset 2Ch) and SID register (PCI offset 2Eh), respectively.

- The Subsystem Vendor ID is set by externally strapping ROMA[15:0]. Bits [15:0] of the ID value correspond to ROMA[15:0], respectively.
- The Subsystem ID is set by externally strapping VIP[15:0]. Bits [15:0] of the ID value correspond to VIP[15:0], respectively.

The ID is latched as the Intel740 comes out of hard reset. Note that these signals are internally connected to pull-down resistors (20 KΩ nominal value). This reduces the number of external straps necessary to change the corresponding register bit default value.

Register Description 3

The Intel740 contains an extensive set of registers and instructions for controlling graphics operations. Graphics drivers provide the software interface at this architectural level. The register/instruction interface is transparent at the Application Programmers Interface (API) level and thus, beyond the scope of this document. General aspects of the Intel740 register/instruction interface include:

- **PCI Configuration Registers.** Standard PCI 2.1 compliant interface. The PCI Configuration registers are located in PCI Configuration space. These registers are described in the *PCI Configuration Registers* Section.
- **System Configuration.** These registers (XRxx registers) provide Intel740 extensions that have been added to the sub-addressing scheme defined by the VGA standard. While this register set is beyond the scope of this document, the DRAM control registers are included (*DRAM Configuration Registers* Section) to show the Intel740 local memory interface control options. These registers are located in both CPU memory space and CPU I/O space (not PCI Configuration space). For a detailed description of the Local Memory Interface, refer to the *Local Memory Interface* Section.
- **2D Registers (VGA and Extended VGA Registers).** These registers are a combination of registers defined by IBM* when the Video Graphics Array (VGA) was first introduced, and others that Intel has added to support graphics modes that have color depths, resolutions, and hardware acceleration features that go beyond the original VGA standard.
- **2D Instructions.** This set of instructions provides an efficient method for controlling BLT and Stretch BLT operations. These instructions obsolete the need for writing to BLT/STRBLT registers.
- **3D Instructions.** The Intel740 has a rich set of 3D instructions. Software uses these instructions to control various aspects of the setup and rendering process. For example, software can issue a set of instruction packets to set the appropriate processing states within the Intel740 3D engine and then send render instruction packets containing the vertex data to be processed according to those states.
- **Video Registers.** An set of video registers has been added to the sub-addressing extensions of the VGA standard. These registers provide a variety of control features for the video engine (e.g., video capture, overlay, etc.)
- **DVD Registers.** A set of DVD registers are used to control the Video Module Interface (VMI) including various features for controlling DVD chips.
- **Address Mapping.** Except for the PCI Configuration registers, the Intel740 register sets are mapped at standard I/O locations. In addition, the register sets and instruction ports are memory mapped, allowing fast register access and providing multiple monitor support. Thus, PIO accesses can be either via I/O space or memory space. The base address for the memory map range is programmed via the MMADR register located in PCI Configuration space.

To provide the user with a more complete understanding of the PCI interface and the Local Memory Interface the PCI Configuration registers and DRAM Control registers are described in this section. The following nomenclature is used for access attributes:

RO	**Read Only**. If a register is read only, writes to this register have no effect.
R/W	**Read/Write**. A register with this attribute can be read and written.
R/WC	**Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of 1 clears (sets to 0) the corresponding bit and a write of 0 has no effect.

3.1 PCI Configuration Registers

The PCI registers provide AGP control/status and power management control/status. Software allocates system memory space by programming base address registers in this PCI configuration space. The Intel740 contains PCI configuration registers that set the base memory address for the Local Memory, Video BIOS ROM, and the memory map for the Intel740 graphics registers/ instructions.

Some of the registers described in this section contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back.

In addition to reserved bits within a register, the configuration space contains reserved locations. Software should not write to reserved configuration locations in the device-specific region (above address offset 3Fh).

Configuration Space registers are accessed through configuration cycles on the PCI bus by the Host bridge (using configuration mechanism #1). The selection of the Intel740 as a target of a PCI configuration cycle is performed by asserting the AD16 signal (IDSEL function) when AD[1:0] are “00” (Type 0 configuration transaction). Neither double data strobing nor side-band protocol are supported for configuration cycles. For more details on performing PCI configuration cycles refer to the *PCI Local Bus Specification, Revision 2.1*.

Figure 3-1 shows an example PCI bus hierarchy when using a host bridge that contains both a Host-to-PCI and a PCI-to-PCI bridge (e.g., the 440LX). In the PCI bus hierarchy, the primary PCI bus is the highest level bus in the hierarchy and is PCI bus #0. The PCI-to-PCI bridge function provides access to the AGP/PCI bus that connects to the Intel740. This bus is below the primary bus in the PCI bus hierarchy and is represented as PCI Bus #1.

Figure 3-1. PCI Bus Hierarchy (Using 82443LX as Host Bridge)

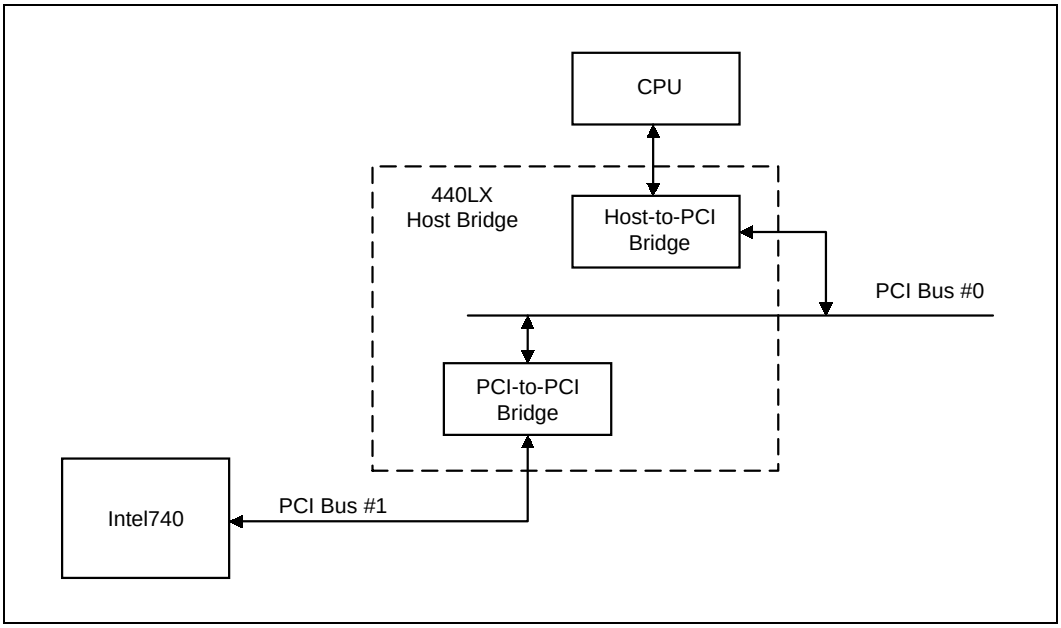


Table 3-1. PCI Configuration Space

Address Offset	Symbol	Register Name	Access	Default
PCI Specific Registers				
00-01h	VID	Vendor Identification	RO	8086h
02-03h	DID	Device Identification	RO	7800h
04-05h	PCICMD	PCI Command	R/W	0000h
06-07h	PCISTS	PCI Status	RO	02B0h
08h	RID	Revision Identification	RO	See stepping information
09–0Bh	CLASSC	PCI Class Code	RO	030000h
0Ch	CLS	Cache Line Size	RO	00h
0Dh	MLT	Master Latency Timer	RO	00h
0Eh	HDR	Header Type	RO	00h
0Fh	BIST	BIST Register	RO	00h
10–13h	LMADR	Local Memory Range Address	R/W	00000008h
14–17h	MMADR	Memory Mapped Range Address	R/W	00000000h
18–1Fh	—	Reserved	—	—
2C–2Dh	SVID	Subsystem Vendor Identification	RO	ssssh
2E–2Fh	SID	Subsystem Identification	RO	ssssh
30–33h	ROMBA	Video ROM Base Address	R/W	00000000h
34h	CAPPTR	Capabilities Pointer	RO	D0h
35–3Bh	—	Reserved	—	—
3Ch	ILR	Interrupt Line	R/W	00h
3Dh	IPR	Interrupt Pin	RO	01h
3Eh	MINGNT	Minimum Grant	RO	00h
3Fh	MAXLAT	Maximum Latency	RO	00h
Intel740 Specific Registers				
40–EFh	—	Reserved	—	—
D0–D3h	AGPCAP	AGP Capabilities	RO	00100002h
D4–D7h	AGPSTS	AGP Status	RO	1F000203h
D8–DBh	AGPCMD	AGP Master Command	R/W	00000000h
DC–DDh	PM_CAPID	Power Management Capabilities Identification	RO	0001h
DE–DFh	PM_CAP	Power Management Capabilities	RO	0201h
E0–E1h	PM_CS	Power Management Control/Status	R/W	0000h
E2–FFh	—	Reserved	—	—

3.1.1 VID—Vendor Identification Register

Address Offset: 00h–01h
Default Value: 8086h
Attribute: Read Only

The VID Register contains the vendor identification number. This 16-bit register, combined with the Device Identification register, uniquely identify any PCI device. Writes to this register have no effect.

Bit	Description
15:0	**Vendor Identification Number.** This is a 16-bit value assigned to Intel.

3.1.2 DID—Device Identification Register

Address Offset: 02h–03h
Default Value: 7800h
Attribute: Read Only

This 16-bit register, combined with the Vendor Identification register, uniquely identifies any PCI device. Writes to this register have no effect.

Bit	Description
15:0	**Device Identification Number.** This is a 16 bit value assigned to the Intel740.

3.1.3 PCICMD—PCI Command Register

Address Offset: 04h–05h
Default: 0000h
Access: Read Only, Read/Write

This 16-bit register provides basic control over the Intel740's ability to respond to PCI cycles. The PCICMD Register in the Intel740 disables the Intel740 PCI compliant master accesses to main memory.

Bit	Descriptions
15:10	Reserved.
9	**Fast Back-to-Back (FB2B)—RO.** (Not Implemented). Hardwired to 0.
8	**SERR# Enable (SERRE) —RO.** (Not Implemented). Hardwired to 0.
7	**Address/Data Stepping—RO.** (Not Implemented). Hardwired to 0.
6	**Parity Error Enable (PERRE) —RO.** (Not Implemented). Hardwired to 0. Since the Intel740 belongs to the category of devices that does not corrupt programs or data in system memory or hard drives, the Intel740 ignores any parity error that it detects and continues with normal operation.
5	**Video Palette Snooping (VPS) —RO.** Hardwired to 0 to disable snooping.
4	**Memory Write and Invalidate Enable (MWIE) —RO.** Hardwired to 0. The Intel740 does not support memory write and invalidate commands.
3	**Special Cycle Enable (SCE) —RO.** This bit is hardwired to 0. The Intel740 ignores Special cycles.
2	**Bus Master Enable (BME) —RO.** Hardwired to 1 to enable the Intel740 to function as a PCI compliant master.
1	**Memory Access Enable (MAE) —R/W.** Controls the Intel740's response to memory space accesses. 0 = Disable (default). 1 = Enable.
0	**I/O Access Enable (IOAE) —R/W.** This bit controls the Intel740's response to I/O space accesses. 0 = Disable (default). 1 = Enable.

3.1.4 PCISTS—PCI Status Register

Address Offset: 06h–07h
Default Value: 02B0h
Access: Read Only, Read/Write Clear

PCISTS is a 16-bit status register that reports the occurrence of a PCI compliant master abort and PCI compliant target abort. PCISTS also indicates the DEVSEL# timing that has been set by the Intel740 hardware.

Bit	Descriptions
15	**Detected Parity Error (DPE) —RO.** 0 = Since the Intel740 does not detect parity, this bit is hardwired to 0.
14	**Signaled System Error (SSE) —R/WC.** 1 = Set when the Intel740 asserts SERR#. 0 = Cleared by writing a 1.
13	**Received Master Abort Status (RMAS) —R/WC.** 1 = Set by the Intel740 as a master when its transaction is terminated by a target abort from the Host Bridge. 0 = Cleared by writing a 1.
12	**Received Target Abort Status (RTAS) —R/WC.** 1 = This bit is set by the Intel740 as a bus master when its transaction is terminated by a target abort from the Host Bridge. 0 = Cleared by writing a 1.
11	**Signaled Target Abort Status (STAS).** 0 = The Intel740 does not use target abort semantics. Hardwired to 0.
10:9	**DEVSEL# Timing (DEVT) —RO.** This 2-bit field indicates the timing of the DEVSEL# signal when the Intel740 responds as a target. 01 = Medium decode device.
8	**Data Parity Detected (DPD) —RO.** 0 = Since Parity Error Response is hardwired to disabled (and the Intel740 does not do any parity detection), this bit is hardwired to 0.
7	**Fast Back-to-Back (FB2B)—RO.** 1 = Intel740 accepts fast back-to-back when the transactions are not to the same agent. Hardwired to 1.
6	**User Defined Format (UDF) —RO.** 0 = Hardwired to 0.
5	**66 MHz PCI Capable (66C) —RO.** 1 = 66 MHz PCI capable. Hardwired to 1.
4	**CAP LIST—RO.** 1 = This bit is set to 1 to indicate that the register at 34h provides an offset into the function's PCI Configuration Space containing a pointer to the location of the first item in the list.
3:0	Reserved.

3.1.5 RID—Revision Identification Register

Address Offset: 08h
Default Value: See latest stepping information
Access: Read Only

This register contains the revision number of the Intel740. These bits are read only and writes to this register have no effect.

Bit	Description
7:0	**Revision Identification Number.** This is an 8-bit value that indicates the revision identification number for the Intel740. The four lsbs are for process differentiation and the four msbs indicate stepping.

3.1.6 CC—Class Code Register

Address Offset: 09h–0Bh
Default Value: 030000h
Access: Read Only

This register contains the device programming interface information related to the Sub-Class Code and Base Class Code definition for the Intel740. This register also contains the Base Class Code and the function sub-class in relation to the Base Class Code.

Bit	Description
23:16	**Base Class Code (BASEC).** 03 = Display controller.
15:8	**Sub-Class Code (SCC).** 00h = VGA compatible.
7:0	**Programming Interface (PI).** 00h = Hardwired as a Display controller.

3.1.7 CLS—Cache Line Size Register

Address Offset: 0Ch
Default Value: 00h
Access: Read only

The Intel740 does not supports this register as a PCI slave.

Bit	Description
7:0	**Cache Line Size (CLS).** This field is hardwired to 0s. The Intel740 as a PCI compliant master does not use the Memory Write and Invalidate command and, in general, does not perform operations based on cache line size.

3.1.8 MLT—Master Latency Timer Register

Address Offset: 0Dh
Default Value: 00h
Access: Read Only

The Intel740 does not support the programmability of the master latency timer because it does not perform bursts.

Bit	Description
7:0	**Master Latency Timer Count Value.** Hardwired to 0s.

3.1.9 HDR—Header Type Register

Address Offset: 0Eh
Default Value: 00h
Access: Read Only

This register contains the Header Type of the Intel740.

Bit	Description
7:0	**Header Type (HTYPE).** This is an 8-bit value that indicates the Header Type for the Intel740. This code has the value 00h, indicating a basic (i.e., single function) configuration space format.

3.1.10 BIST—BIST Register

Address Offset: 0Fh
Default Value: 00h
Access: Read Only

This register is used for control and status of Built In Self Test (BIST).

Bit	Descriptions
7	**BIST Supported.** BIST is not supported. This bit is hardwired to 0.
6:0	Reserved.

3.1.11 LMADR—Local Memory Range Address Register

Address Offset: 10–13h
Default Value: 00000008h
Access: Read/Write, Read Only

This register requests allocation for the Intel740 local memory. The allocation is for 16 MB and the base address is defined by bits [31:24].

Bit	Descriptions
31:24	**Memory Base Address—R/W.** Set by the OS, these bits correspond to address signals [31:24].
23:4	**Address Mask—RO.** 0s = 16 MB address range (Hardwired to 0s)
3	**Prefetchable Memory—RO.** 1 = Enable (Hardwired to 1 to enable prefetching.)
2:1	**Memory Type—RO.** 0 = 32-Bit Address (Hardwired to 0)
0	**Memory/IO Space—RO.** 0 = Memory Space (Hardwired to 0)

3.1.12 MMADR—Memory Mapped Range Address Register

Address Offset: 14–17h
Default Value: 00000000h
Access: Read/Write, Read Only

This register requests allocation for the Intel740 registers and instruction ports. The allocation is for 512 KB and the base address is defined by bits [31:19].

Bit	Descriptions
31:19	**Memory Base Address—R/W.** Set by the OS, these bits correspond to address signals [31:19].
18:4	**Address Mask—RO.** 0s = 512 KB address range (Hardwired to 0s)
3	**Prefetchable Memory—RO.** 0 = Disable (Hardwired to 0 to prevent prefetching.
2:1	**Memory Type—RO.** 00 = 32-Bit Address (Hardwired to 0s to indicate a 32-bit address).
0	**Memory / IO Space—RO.** 0 = Memory Space (Hardwired to 0).

3.1.13 SVID—Subsystem Vendor Identification Register

Address Offset: 2C–2Dh
Default Value: ssssh (s=strapping option)
Access: Read Only

This value in this register represents the strapping options on ROMA[15:0].

Bit	Descriptions
15:0	**Subsystem Vendor ID.** Bits [15:0] correspond to ROMA[15:0], respectively. The bit value is a 1 when the corresponding signal is strapped high and a 0 when the corresponding signal is strapped low (or left unconnected).

3.1.14 SID—Subsystem Identification Register

Address Offset: 2E–2Fh
Default Value: ssssh (s=strapping option)
Access: Read Only

This value in this register represents the strapping options on VIP15:0].

Bit	Descriptions
15:0	**Subsystem Identification.** Bits [15:0] correspond to VIP[15:0], respectively. The bit value is a 1 when the corresponding signal is strapped high and a 0 when the corresponding signal is strapped low (or left unconnected).

3.1.15 ROMADR—Video BIOS ROM Base Address Register

Address Offset: 30–33h
Default Value: 00000000h
Access: Read/Write, Read Only

The ROMADR register requests allocation for the Intel740 BIOS ROM. The allocation is for 256 KB. This register is only used for add-in type cards with an onboard ROM. Other PCI methods are used for ROM images integrated with the System BIOS.

Bit	Descriptions
31:18	**ROM Base Address—R/W.** These bits are set by the OS and correspond to CPU memory address signals [31:18].
17:11	**Address Mask—RO.** 0s = 256 KB address range (Hardwired to 0s).
10:1	Reserved. Hardwired to 0s.
0	**ROM BIOS Enable—R/W.** 1 = ROM is accessible in CPU address space. 0 = ROM not accessible.

3.1.16 CAPPOINT—Capabilities Pointer Register

Address Offset: 34h
Default Value: D0h
Access: Read Only

Bit	Descriptions
7:0	**Capabilities Pointer Value.** This field contains an offset into the function's PCI Configuration Space for the first item in the New Capabilities Linked List, which is at address D0h.

3.1.17 INTRLINE—Interrupt Line Register

Address Offset: 3Ch
Default Value: 00h
Access: Read/Write

Bit	Descriptions
7:0	**Interrupt Connection.** Used to communicate interrupt line routing information. POST software writes the routing information into this register as it initializes and configures the system. The value in this register indicates which input of the system interrupt controller that the device's interrupt pin is connected.

3.1.18 INTRPIN—Interrupt Pin Register

Address Offset: 3Dh
Default Value: 01h
Access: Read Only

Bit	Descriptions
7:0	**Interrupt Pin.** As a single function device, the Intel740 specifies INTA# as its interrupt pin. 01h = INTA#.

3.1.19 MINGNT—Minimum Grant Register

Address Offset: 3Eh
Default Value: 00h
Access: Read Only

Bit	Descriptions
7:0	**Minimum Grant Value.** 00h = The Intel740 does not burst as a PCI compliant master.

3.1.20 MAXLAT—Maximum Latency Register

Address Offset: 3Fh
Default Value: 00h
Access: Read Only

Bit	Descriptions
7:0	**Maximum Latency Value.** 00h = The Intel740 has no specific requirements for how often it needs to access the PCI bus.

3.1.21 AGPCAP—AGP Capabilities Register

Address Offset: D0h–D3h
Default Value: 00100002h
Access: Read Only

Bits	Description
31:24	Reserved. Read as 0s.
23:20	**Major Revision Number**. 0001 = Major revision number of AGP interface specification.
19:16	**Minor Revision Number**. 0000 = Minor revision number of AGP interface specification.
15:8	**Next Capabilities List Item (NEXT_PTR).** Contains the pointer to next item in capabilities list. This field points to the ACPI registers at DCh.
7:0	**Capabilities ID (CAP_ID).** 02h = This field identifies the linked list item as containing the AGP registers.

3.1.22 AGPST—AGP Status Register

Address Offset: D4h–D7h
Default: 1F000203h
Type: Read Only

Bits	Description
31:24	**Requests (RQ).** RQ=1Fh (the maximum number of AGP command requests the Intel740 can manage). 00h = depth of 1 . . FFh = depth of 256.
23:10	Reserved. Read as 0s.
9	**Sideband Addressing (SBA).** SBA=1 to indicate that the Intel740 supports sideband addressing.
8:2	Reserved. Read as 0s.
1:0	**RATE.** This field indicates all data transfer rates supported by this device. 00 = Reserved (not allowed value) 01 = 1X 10 = 2X 11 = Reserved (not allowed value) Note: Data transfer applies to AD and SBA buses.

3.1.23 AGPCMC—AGP Master Command Register

Address Offset: D8h–DBh
Default: 00000000h
Access: Read/Write

Bits	Description
31:24	**Request Queue Depth (RQ_DEPTH).** The RQ_DEPTH field must be programmed with 1 less than the maximum number of pipelined operations the master is allowed to enqueue in the target. This value must be equal to or less than the value set in the AGP compliant target. Bits [31:30] are hardwired to 0s. 00h = depth of 1 ... ... 3Fh = depth of 64.
23:10	Reserved. Read as 0s.
9	**Sideband Addressing Enable (SBA_ENABLE).** 1 = Enable. 0 = Disable.
8	**AGP Enable.** 1 = Enable AGP traffic 0 = No AGP traffic.
7:3	Reserved. Read as 0s.
1:0	**Data Rate.** This field selects the data transfer rate (AD and SBA Buses). One (and only one) bit in this field must be set to indicate the desired data transfer rate. The same bit must be set for both the Intel740 and the AGP slave. 00 = Reserved (not allowed value) 01 = 1X 10 = 2X 11 = Reserved (not allowed value)

3.1.24 PM_CAPID—Power Management Capabilities ID Register

Address Offset: DCh–DDh
Default Value: 0001h
Access: Read Only

Bits	Description
15:8	**NEXT_PTR.** This contains a pointer to next item in capabilities list. 00h = Indicates that this is the final capability in the list.
7:0	**CAP_ID.** SIG defines this ID as 01h for power management.

3.1.25 PM_CAP—Power Management Capabilities Register

Address Offset: DEh–DFh
Default Value: 0201h
Access: Read Only

Bits	Description
15:11	**PME Support.** This field indicates the power states in which the Intel740 may assert PME#. Hardwired to 0 to indicate that the Intel740 does not assert the PME# signal.
10	**D2.** Indicates support for D2 power management state. 0 = Not supported. Hardwired to 0.
9	**D1.** Indicates support for D1 power management state. 1 = Supported. Hardwired to 1.
8:6	Reserved. Read as 0s.
5	**Device Specific Initialization (DSI).** 1 = Hardwired to 1 to indicate that special initialization of the Intel740 is required before generic class device driver is to use it.
4	**Auxiliary Power Source.** Hardwired to 0.
3	**PME Clock.** 0 = Hardwired to 0 to indicate Intel740 does not support PME# generation.
2:0	**Version.** 001 = Hardwired to 001b to indicate there are 4 bytes of power management registers implemented.

3.1.26 PM_CS—Power Management Control/Status Register

Address Offset: E0h–E1h
Default Value: 0000h
Access: Read/Write

Bits	Description
15	**PME_Status—R/WC.** 1 = This bit is 0 to indicate that Intel740 does not support PME# generation from D3 (cold).
14:13	Data Scale (Reserved)—RO. The Intel740 does not support data register. This bit always returns 0 when read, write operations have no effect.
12:9	Data_Select (Reserved)—RO. The Intel740 does not support data register. This bit always returns 0 when read, write operations have no effect.
8	**PME_En—R/W.** 0 = This bit is 0 to indicate that PME# assertion from D3 (cold) is disabled.
7:2	Reserved. Always returns 0 when read, write operations have no effect.
1:0	**PowerState—R/W.** This field indicates the current power state of Intel740 and can be used to set the Intel740 into a new power state. If software attempts to write an unsupported state to this field, write operation must complete normally on the bus, but the data is discarded and no state change occurs. 00 = D0 01 = D1 10 = D2 11 = D3

3.2 DRAM Configuration Registers

The DRAM Configuration registers are part of the System Configuration register set (XRxx registers) and are provided here to show the programming options for the DRAM interface. The complete System Configuration register set is beyond the scope of this document. Note that he DRAM Configuration registers are not part of the PCI Configuration space register set described in the previous section of this document.

The DRAM Configuration registers are accessed through a sub-addressing scheme by writing the index of the desired register into the Configurations Extensions Index Register at I/O address 3D6h (or memory mapped address offset 3D6h), and then accessing the desired register through the Configurations Extension Data Register located at I/O address 3D7h (or memory mapped address offset 3D7h). Note that these registers can be accessed via I/O addresses or memory addresses where the memory address is an offset from the base address programmed into the MMADR register (PCI offset 14h).

3.2.1 XRX—Configurations Extension Index Register

IO/Mem Address: 3D6h
Default: 00h
Attributes: Read/Write

Bit	Description
7:0	Extension Register Sequence Index. This field contains the sequence index value used to access extension data registers.

3.2.2 XR50—DRAM Row Type (DRT) Register

IO/Mem Address: 3D7h (index=50h)
Default: 00h
Attributes: Read/Write

This 8-bit register identifies the width of the DRAM interface and whether the DRAM rows are populated and should be programmed by BIOS.

Bit	Description
7	Reserved.
6	**Memory Data DRAM Interface Width (MDW).** 1 = Reserved 0 = 64-bit (default).
5:0	**DRAM Row Type (DRT).** Each pair of bits in this register corresponds to the DRAM row identified by the corresponding DRB register. **DRT Bits** **Corresponding DRB Register** DRT[2:0] DRB[0], row 0 (default=0) DRT[5:3] DRB[1], row 1 (default=0) **DRT Pair** **Corresponding DRB Register** 000 Populated with SDRAM 001 Reserved 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Empty Row

3.2.3 XR51—DRAM Control (DRAMC) Low Register

IO/Mem Address: 3D7h (index=51h)
Default: 1Ch
Attributes: Read/Write

Bit	Description
7:6	Reserved.
5	**RAS-to-CAS Override (RCO).** 0 = Determined by CL bit (default). 1 = 2 Local memory clock periods. RAS#-to-CAS# delay (t_{RCD}) (i.e., row activate command to read/write command).
4	**CAS# Latency (CL).** In units of local memory clock periods. **Bit** **CL** **RAS#-to-CAS# delay (t_{RCD})** 0 2 2 1 3 3 (default)
3	**RAS# Riming (RT).** This bit controls RAS# active to precharge, and refresh to RAS# active delay (in local memory clocks). **Bit** **RAS# act. To precharge (t_{RAS})** **Refresh to RAS# act. (t_{RC})** 0 5 8 1 7 10 (default)
2	**RAS# Precharge Timing (RPT).** This bit controls RAS# precharge (in local memory clocks). 0 = 3 clocks. 1 = 4 clocks (default).
1:0	Reserved.

3.2.4 XR52—DRAM Control (DRAMC) High Register

IO/Mem Address: 3D7h (index=52h)
Default: 10h
Attributes: Read/Write

Bit	Description
7:5	Reserved.
4	**Cacheline Fetch Enable.** Software for SDRAM initialization must make sure to disable cacheline fetch during the initialization sequence. 0 = Disable (i.e., QWord instead). 1 = Enable.
3	Reserved.
2:0	**Special Mode Select (SMS).** These bits select special SDRAM modes used for testing and initialization. The NOP command must be programmed first before any other command can be issued. 000 = **Normal SDRAM mode.** (Normal, default). 001 = **NOP Command Enable (NCE).** This state forces cycles to DRAM to generate SDRAM NOP commands. 010 = **All Banks Precharge Command Enable (ABPCE).** This state forces cycles to DRAM to generate an all banks precharge command. 011 = **Mode Register Command Enable (MRCE).** This state forces all cycles to DRAM to be converted into MRS commands. The command is driven on the MA[11:0] lines. MA[2:0] correspond to the burst length, MA3 corresponds to the wrap type, and MA[6:4] correspond to the latency mode. MA[11:7] are driven to 00000 by the Intel740. The BIOS must select an appropriate host address for each row of memory such that the right commands are generated on the MA[6:0] lines, taking into account the mapping of host addresses to local memory addresses. 100 = **CBR Cycle Enable (CBRCE).** This state forces cycles to DRAM to generate SDRAM CBR refresh cycles. 101 = **Reserved.** 11X = **Reserved.**

3.2.5 XR53—DRAM Extended Control (DRAMEC) Register

IO/Mem Address: 3D7h (index=53h)
Default: 01h
Attributes: Read/Write

This 8-bit register provides additional controls for local memory DRAM operating modes and features.

Bit	Description
7:2	Reserved
1:0	**DRAM Refresh Rate (DRR).** The DRAM refresh rate is adjusted according to the frequency selected by this field. Note that refresh is also disabled via this field, and that disabling refresh results in the eventual loss of DRAM data; although refresh can be briefly disabled without data loss. Bits [2:0] must be set to a valid refresh rate as soon as possible once DRAM testing is completed. 00 = Refresh Disabled 01 = 60 MHz (default) 1X = Reserved (10 is Fast refresh test mode)

3.2.6 XR54—DRAM Timing (DRAMT) Register

IO/Mem Address: 3D7h (index=54h)
Default: 78h
Attributes: Read/Write

This 8-bit register controls main memory DRAM timings.

Bit	Description
7	Reserved.
6:5	**DRAM Read Command Timing (DRCT).** The DRAM read command timings are controlled by the DRCT field. Slower rates may be required in certain system designs to support loose layouts or slower memories. Most system designs will be able to use one of the faster command mode timings. 00 = Reserved 01 = Reserved 10 = Reserved 11 = 1 (default)
4:3	**DRAM Write Command Timing (DWCT).** This field controls the DRAM write command timings. Slower rates may be required in certain system designs to support loose layouts or slower memories. Most system designs will be able to use one of the faster command mode timings. 00 = Reserved 01 = Reserved 10 = Reserved 11 = 1 (default)
2:0	Reserved.

3.2.7 XR55—DRAM Row Boundary 0 (DRB0) Register

IO/Mem Address: 3D7h (index=55h)
Default: 08h
Attributes: Read/Write

Intel740 supports up to 2 rows of DRAM. Each row is 64 bits wide. The DRAM Row Boundary Registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the boundary addresses in 1 Mbyte granularity.

DRB0=Total amount of memory in row 0 (in 1 Mbyte quantities)

Bit	Description
7:4	Reserved.
3:0	**DRAM Row Boundary (DRB).** This 4-bit value is compared against A[23:20] to determine the upper address limit of the corresponding DRAM row. (DRB minus previous DRB=row size)

3.2.8 XR56—DRAM Row Boundary 1 (DRB1) Register

IO/Mem Address: 3D7h (index=56h)
Default: 08h
Attributes: Read/Write

Intel740 supports up to 2 rows of DRAM. Each row is 64 bits wide. The DRAM Row Boundary Registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the boundary addresses in 1 Mbyte granularity.

DRB1=Total amount of memory in row 0 + row 1 (in 1 Mbyte quantities)

Bit	Description
7:4	Reserved.
3:0	**DRAM Row Boundary (DRB).** This 4-bit value is compared against A[23:20] to determine the upper address limit of the corresponding DRAM row. (DRB minus previous DRB=row size)

Functional Description 4

This chapter describes the main functional blocks of the Intel740 including an overview of the register/instruction interface, 3D and 2D capabilities, video capabilities, display, and the various hardware interfaces.

4.1 3D/2D Instruction Processing (Pipeline Preprocessor)

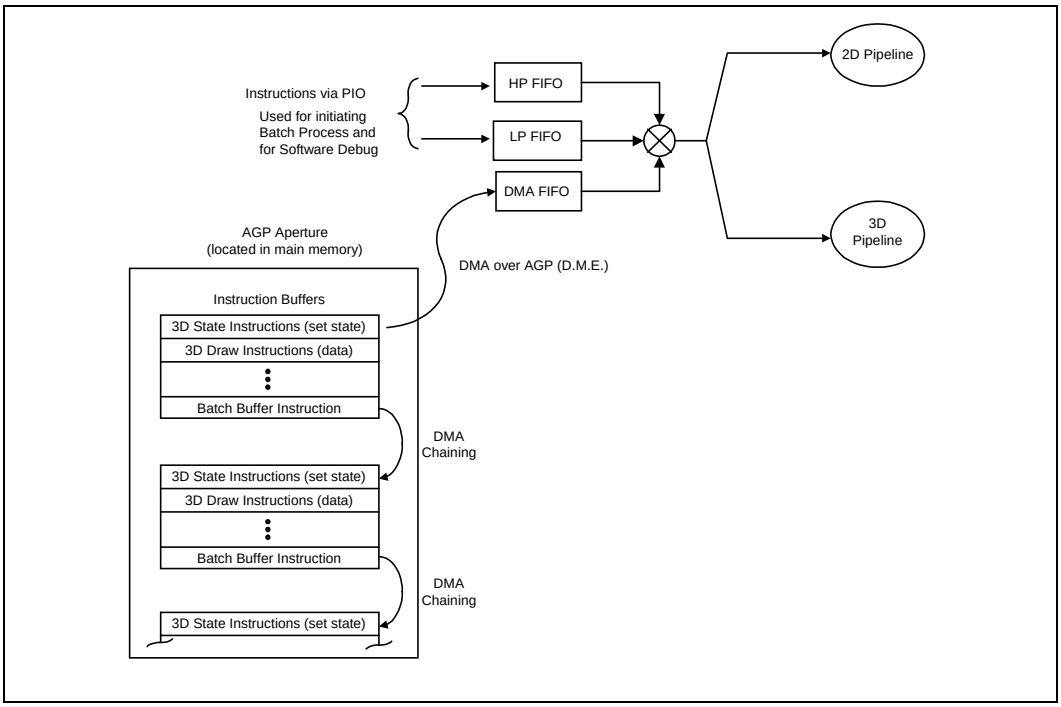
The 3D and 2D drawing engines are controlled by Intel740 instructions. For 3D operations, instructions set 3D pipeline states and control the processing functions.

For 2D operations, instructions provide an efficient method for invoking BLT and Stretch BLT operations. This obsoletes the need for writing to BLT/STRBLT registers.

Internal FIFOs provide a steady stream of instructions to the drawing engines (Figure 4-1). The 2D and 3D engines execute instructions from one of the three instruction interface FIFOs — High Priority FIFO (HPF), Low Priority FIFO (LPF), and DMA FIFO. The High Priority and Low Priority FIFOs are used for software debug and for initiating the Batch Process. The DMA FIFO Is used during batch processing.

Batch Processing. Software sets up instruction packets in AGP memory buffers and then instructs the Intel740 to process the buffers. The Intel740 fetches instructions (using its DME capability) from AGP memory into its DMA FIFO. Buffers can be chained together permitting a series of instruction buffers to be executed without CPU involvement. Batch processing is the primary method for issuing 2D and 3D instruction streams to the Intel740. This method provides significant performance advantages over writing directly to FIFOs including: 1) Reduced software overhead, 2) Efficient DMA instruction fetches from AGP memory, and 3) Software can more efficiently set up instruction packets in buffers in AGP memory (faster writes) than writing to FIFOs.

Figure 4-1. 3D/2D Pipeline Preprocessor



Note: Instruction buffers can contain either 2D or 3D instructions. This example only shows 3D instructions.

4.2 3D Engine

The Intel740 3D engine has been architected as a deep pipeline, where performance is maximized by allowing each stage of the pipeline to simultaneously operate on different primitives or portions of the same primitive. Figure 4-2 is a conceptual representation of this pipeline highlighting the parallel data paths. The Intel740 supports perspective-correct texture mapping, bilinear MIP-Mapping, Gouraud shading, alpha-blending, stippling, antialiasing, fogging and Z Buffering. These features can be independently enabled or disabled via a set of 3D instructions. Textures can be located in AGP memory to free up local memory for other uses (e.g., back and Depth Buffers, bitmaps, etc.).

The main blocks of the pipeline are the Setup Engine, Scan Converter, Texture Pipeline, and Color Calculator block. A typical programming sequence would be to send instructions to set the state of the pipeline followed by rendering instructions containing 3D primitive vertex data.

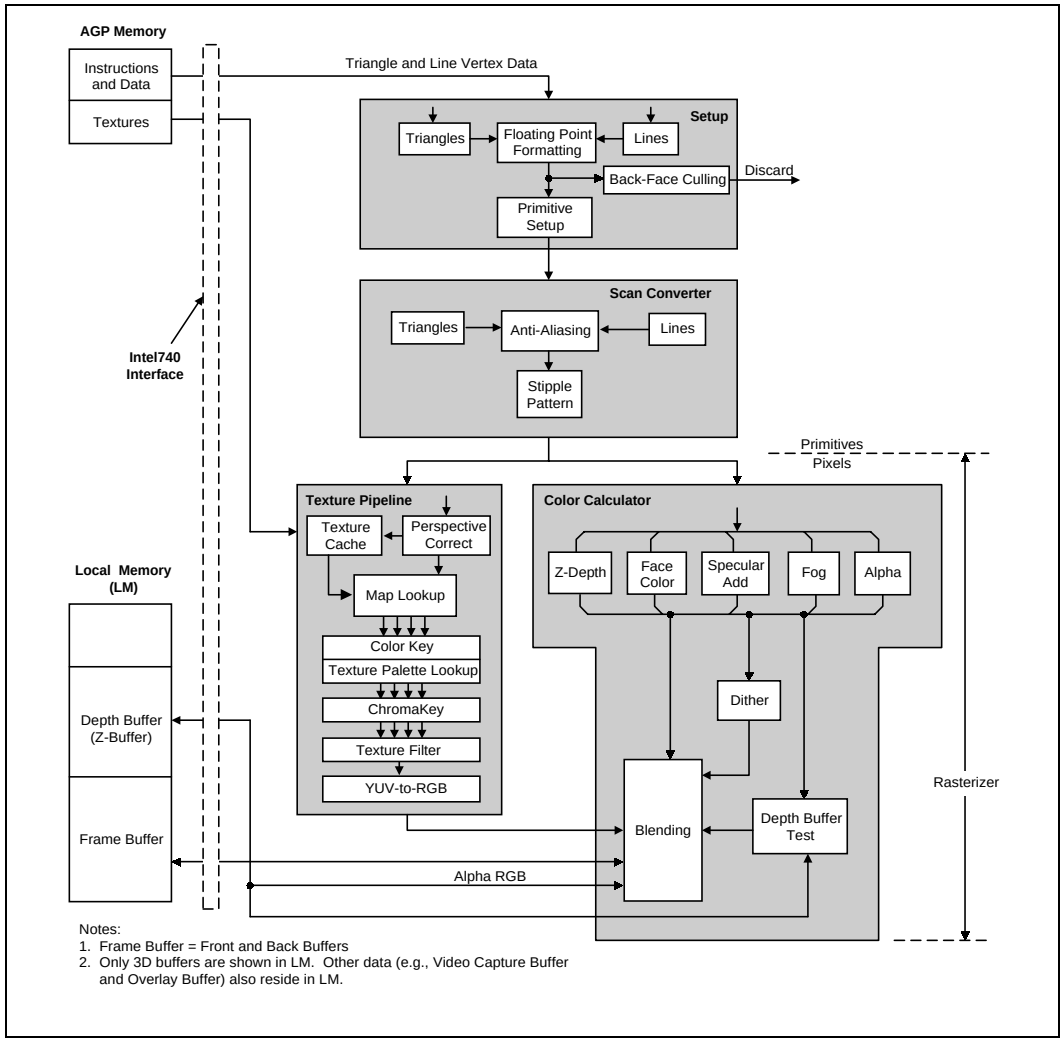
4.2.1 Buffers

The Intel740 2D, 3D, and video capabilities provide control over a variety of graphics buffers. To aid the rendering process, the Intel740 Frame Buffer contains two hardware buffers—the Front Buffer (display buffer) and the Back Buffer (render buffer). While the Back Buffer may actually coincide with (or be part of) the visible display surface, a separate (screen or window-sized) Back Buffer is used to permit double-buffered drawing. That is, the image being drawn is not visible

until the scene is complete and the Back Buffer made visible (via an instruction) or copied to the Front Buffer (via a 2D BLT operation). By rendering to one and displaying from the other, the possibility of image tearing is removed.

The Intel740 3D pipeline operates on the Back Buffer and the Z Buffer. The pixels' 16-bit (or 15-bit) RGB colors are stored in the Back Buffer. The Z Buffer can be used to store 16-bit depth values or 5-bit "destination alpha" values (for antialiasing, etc.). The Intel740 3D Instruction set provides a variety of controls for the buffers (e.g., initializing, flip, clear, etc.).

Figure 4-2. The Intel740's 3D Pipeline (Parallel Data Paths)



4.2.2 Setup

The setup stage of the pipeline takes the input data associated with each vertex of the line or triangle primitive and computes the various parameters required for scan conversion. This stage formats data such that four pixels per clock are output from the rasterizing engine. In formatting this data, the Intel740 maintains sub-pixel accuracy. Data is dynamically formatted for each rendered polygon and output to the proper processing unit. As part of setup, the Intel740 removes polygons from further processing, if they are not facing the user's viewpoint (referred to as "Back Face Culling").

4.2.3 Scan Conversion

The Intel740 scan conversion stage decomposes line and triangle primitives into spans of pixels, generating the X,Y coordinates and possibly-interpolated attributes (color, depth, fog, etc.) associated with each pixel. This data is then passed to the subsequent pipeline stages. The span of pixels throughput is independent of triangle size and shape.

The Intel740 rasterization engine is designed to work on all triangles without degrading performance based on size or orientation. The 3D features that can be invoked at this stage permit an efficient flow of pixels to subsequent stages of the pipeline (e.g., antialiasing, pattern stipple, monochrome).

4.2.4 Rasterizing

4.2.4.1 Texturing

The Intel740 texture processor allows an image, pattern, or video to be placed on the surface of a 3D polygon. Textures can be located in AGP memory. Being able to use textures directly from AGP memory means that large complex textures can easily be handled without the limitations imposed by the traditional approach of only using local graphics memory.

Traditional storage of textures in local memory is useful if all textures fit within the available memory; otherwise, each new texture must be fetched from system memory and stored in the local memory before use. When switches occur between textures such that the same texture ends up being swapped in and out of local memory because of triangle ordering, “thrashing” of local memory results. The Intel740 texture processor avoids “thrashing” local memory by executing directly from AGP memory.

The texture processor receives the texture coordinate information from the setup engine and the texture blend information from the scan converter. The texture processor performs texture color-/chroma-key matching, texture filtering (up to bilinear interpolation), and YUV to RGB conversions.

The Intel740 supports up to 11 Levels-of-Detail (LODs) ranging in size from 1024x1024 to 1x1 texels. (A texel is defined as a texture map pixel.) Textures need not be square. Included in the texture processor is a small cache which provides efficient mip-mapping.

The Intel740 has options for four different types of mip-mapping:

- **Nearest.** Texel with coordinates nearest to the desired pixel are used. (This is used if only one LOD is present.)
- **Linear.** A weighted average of a 2x2 area of texels surrounding the desired pixel are used. (This is used if only one LOD is present.)
- **Mip Nearest.** This is used if many LODs are present. The appropriate LOD is chosen and the texel with coordinates nearest to the desired pixel are used.
- **Mip Linear.** This is used if many LODs are present. The appropriate LOD is chosen and a weighted average of a 2x2 area of texels surrounding the desired pixel are used. This is also referred to a bi-linear mip-mapping.

The Intel740 can store each of theses mip-maps in any of the following formats:

- Palettized
- 1555 ARGB
- 0565 ARGB
- 4444 ARGB
- 422 YUV
- 0555YUV
- 1544 YUV

Many texture mapping modes are supported. Perspective correct mapping is always performed. As the map is fitted across the polygon, the map can be tiled, mirrored in either the U or V directions, or mapped up to the end of the texture and no longer placed on the object (this is known as clamp mode). The way a texture is combined with other object attributes is also definable. The complete listing of attributes is summarized in Table 4-1.

Table 4-1. Texture Mapping Modes

Texture Combining Option	Color Equation	Alpha Equation
Decal	RGB Co = Ct RGBA Co = Ct	Ao = Af Ao = Af
Modulate	RGB Co = Cf*Ct RGBA Co = Cf*Ct	Ao = Af Ao = At
Decal Alpha	RGB Co =Ct RGBA Co = (1-At)*Cf+At*Ct	Ao = Af Ao = Af
Modulate Alpha	RGB Co = Cf*Ct RGBA Co = Cf*Ct	Ao = Af Ao = Af*At
Decal Mask	RGB Co = Ct RGBA If (Am) Co = Ct Else Co=Cf	Ao = Af Ao = Af
Modulate Mask	RGB Co = Cf*Ct RGBA If (Am) Co = Cf*Ct Else Co=Cf	Ao = Af Ao = Af
Decal Alpha Texture Alpha	RGB Co =Ct RGBA Co = (1-At)*Cf+At*Ct	Ao = Af Ao = At
Decal Intrinsic Alpha	RGB Co = (1-Af)*Cf+Af*Ct RGBA Co = (1-Af)*Cf+Af*Ct	Ao = Af Ao = Af
Decal Intrinsic Alpha Texture Alpha	RGB Co = (1-Af)*Cf+Af*Ct RGBA Co = (1-Af)*Cf+Af*Ct	Ao = Af Ao = At
Decal Mask Feature	RGB Co = Ct RGBA If (Am) Co = Ct Else Kill Pixel	Ao = Af Ao = Af
Modulate Mask Feature	RGB Co = Cf*Ct RGBA If (Am) Co = Cf*Ct Else Kill Pixel	Ao = Af Ao = Af

NOTES:
1. Cf = intrinsic (flat or Gouraud Interpolated) color of feature
2. Af = intrinsic (flat or Gouraud Interpolated) alpha of feature
3. Ct = color form texture data
4. At = alpha from texture data
5. Am = lsb of nearest neighbor alpha from texture data
6. Co = color output of texture blend function
7. Ao = alpha output of texture blend function

Texture ColorKey and ChromaKey

ColorKey and ChromaKey describe two methods of removing a specific color or range of colors from a texture map before it is applied to an object. For “nearest” texture filter modes, removing a color simply makes those portions of the object transparent (the previous contents of the back buffer show through). For “linear” texture filtering modes, the texture filter is modified if only the non-nearest neighbor texels match the key (range).

ColorKeying occurs with paletted textures, and removes colors according to an index (before the palette is accessed). When a color palette is used with indices to indicate a color in the palette, the indices can be compared against a state variable “ColorKey Index Value” and if a match occurs and ColorKey is enabled, then this value’s contribution is removed from the resulting pixel color. The Intel740 defines index matching as ColorKey.

ChromaKeying can be performed for both paletted and non-paletted textures, and removes texels that fall within a specified color range. The ChromaKey mode refers to testing the RGB or YUV components to see if they fall between high and low state variable values. If the color of a texel contribution is in this range and ChromaKey is enabled, then this contribution is removed from the resulting pixel color.

4.2.4.2 Color Calculator

The Color Calculator includes a series of optional pipeline stages that compute the final pixel RGBA color by combining it with other color values (e.g., the fog color, specular color, filtered/converted texture color, coverage value, and/or the “destination” pixel color obtained from the back buffer). The (pre-alpha-blended) alpha and depth values are then tested to determine whether the back color and/or Depth Buffer will be updated with new values.

The Intel740 incorporates two types of shading: flat and Gouraud. Flat shading takes a specified attribute from the first passed vertex and uses this attribute as is throughout the entire polygon. Gouraud shading takes three attributes as defined by the vertices and interpolates over the entire polygon. Four attribute coefficients can be calculated:

- **Alpha** - This attribute defines the transparency or opacity of the polygon.
- **Fog** - This attribute adds the effect of density to the atmosphere. As an object goes further away from the viewer, it appears to become more “cloudy” or “foggy” than closer objects.
- **Specular** - This attribute adds the effect of a “hot spot” on an object which corresponds to the shininess of its material property. The specular highlight can be varied by the amount specified for each red, green, and blue component.
- **Color** - This attribute defines the base color of the object and is specified in red, green, and blue.

Each of these pixel color attributes is calculated in parallel with the texture processor’s calculating of the pixel texture attributes. As each of the results is calculated in both the texture processor and color calculator, the values can be dithered to 5 or 6 bits. The final role of the color calculator is to combine the results from the texture processor and pixel attributes, and to perform alpha blending, fogging, and antialiasing. Each pixel’s fog factor is used to calculate the amount of fogging performed on the pixel.

Alpha blending involves combining previously written pixel data with current data to create some level of transparency defined by the earlier calculated alpha component. The available functions for alpha blending are:

- Zero
- Source Color
- Source Alpha
- Destination Alpha
- Destination Color
- Source Alpha Saturation
- Both Inverse Source Alpha
- One
- Inverse Source Color
- Inverse Source Alpha
- Inverse Destination Alpha
- Inverse Destination Color
- Both Source Alpha

These blend factors are calculated for the source and destination pixels and the results are multiplied by the source and destination components. To reduce rendering artifacts source alpha blending requires polygons to be sorted from back to front. Destination alpha blending requires polygons to be sorted from front to back and the alpha value to be in the Z Buffer. The Intel740 allows pixel alpha values to be computed and compared against stored alpha values for a test to be made when writing to the Frame Buffer. The tests are:

- Never
- Equal
- Greater
- Gequal
- Less
- Lequal
- Notequal
- Always

The Intel740 relieves the processor from having to pre-sort triangles through the use of a Z Buffer. This Z Buffer is 16-bits. The same tests made with alpha values (see above bullets) can also be performed with z values.

4.3 The 2D Operation

The Intel740 contains BLT and STRBLT engines, a hardware cursor, and an extensive set of 2D registers and instructions.

Intel740 VGA Registers and Enhancements

The 2D registers are a combination of registers defined by IBM* when the Video Graphics Array (VGA) was first introduced, and others that Intel has added to support graphics modes that have color depths, resolutions, and hardware acceleration features that go beyond the original VGA standard. Intel740 improves upon VGA by providing additional features that are used through numerous additional registers.

The Intel740 also supports up to 8 MB of Frame Buffer memory. As an improvement on the VGA standard Frame Buffer port-hole, the Intel740 also maps the entire Frame Buffer into part of a single contiguous memory space at a programmable location, providing what is called “linear” access to the Frame Buffer. The size of this memory is 8 MB, and the base address is set via a PCI configuration register.

4.4 Fixed Blitter (BLT) and Stretch Blitter (STRBLT) Engines

The Intel740's 64-bit BitBLT engine provides hardware acceleration for many common Windows* operations. There are two primary BitBLT functions: Fixed BitBLT (BLT) and Stretch BitBLT (STRBLT). The term BitBLT refers to block transfers of pixel data between memory locations. The word "fixed" is used to differentiate from the Stretch BLT engine. The BLT engines can be used for the following:

- Move rectangular blocks of data between memory locations
- Pixel format conversion
- Data Alignment
- Perform logical operations

The Intel740 has instructions to invoke BLT and STRBLT operations, permitting software to set up instruction buffers and use batch processing as described in the *3D/2D Instruction Processing (Pipeline Preprocessor)* Section. Note that these instructions replace the need to do PIO directly to BLT and STRBLT registers.

4.4.1 Fixed BLT Engine

The rectangular block of data does not change as it is transferred between memory locations. The allowable memory transfers are between: AGP memory and local memory, local memory and AGP memory, AGP memory and AGP memory, and local memory and local memory. Data to be transferred can consist of regions of memory, patterns, or solid color fills. A pattern will always be 8 x 8 pixels wide and may be 1, 8, or 16 bits per pixel.

The Intel740 has the ability to expand monochrome data into a color depth of 8, 16, or 24 bits. BLTs can be either opaque or transparent. Opaque transfers, move the data specified to the destination. Transparent transfers, compare destination color to source color and write according to the mode of transparency selected.

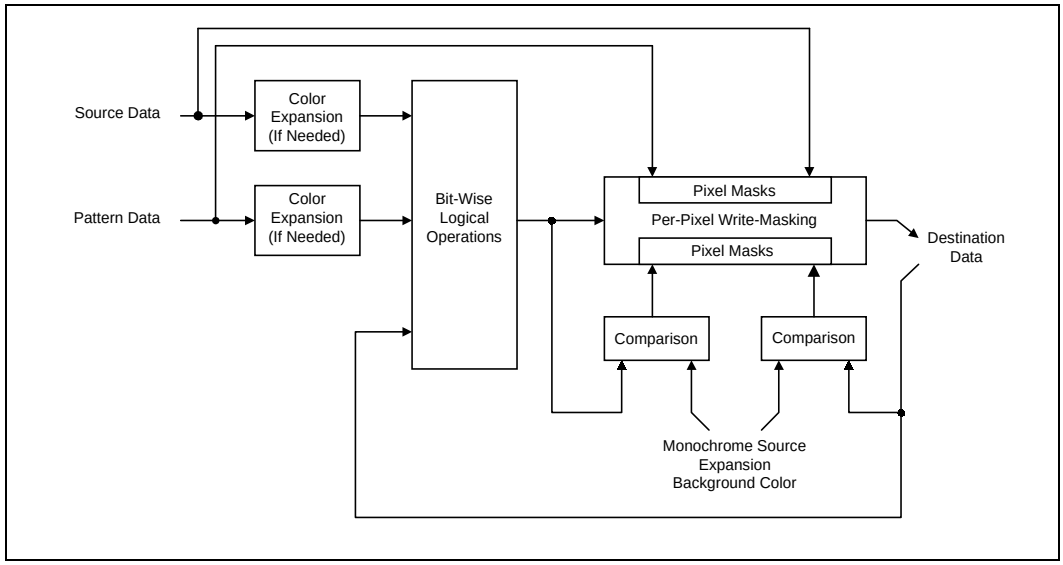
Data is horizontally and vertically aligned at the destination. If the destination for the BLT overlaps with the source memory location, the Intel740 can specify which area in memory to begin the BLT transfer. Use of this BLT engine accelerates the Graphical User Interface (GUI) interface of Microsoft* Windows. Hardware is included for all 256 Raster Operations (Source, Pattern, and Destination) defined by Microsoft*, including transparent BitBLT.

While the BitBLT engine is often used simply to copy a block of graphics data from the source to the destination, it also has the ability to perform more complex functions. The BitBLT engine is capable of receiving three different blocks of graphics data as input as shown in Figure 4-3 (source data, destination data, and pattern data). The source data may exist either in the Frame Buffer or it may be provided by the host CPU from some other source such as AGP memory. The pattern data always represents an 8x8 block of pixels that must be located in the Frame Buffer, usually within the off-screen portion. The data already residing at the destination may also be used as an input, but this data must also be located in the Frame Buffer.

The BitBLT engine may use any combination of these three different blocks of graphics data as operands, in both bit-wise logical operations to generate the actual data to be written to the destination, and in per-pixel write-masking to control the writing of data to the destination. It is intended that the BitBLT engine will perform these bit-wise and per-pixel operations on color graphics data that is at the same color depth that the rest of the graphics system has been set. However, if either the source or pattern data is monochrome, the BitBLT engine has the ability to

put either block of graphics data through a process called “color expansion” that converts monochrome graphics data to color. Since the destination is often a location in the on-screen portion of the Frame Buffer, it is assumed that any data already at the destination will be of the appropriate color depth.

Figure 4-3. BLT Engine Block Diagram and Data Paths



4.4.2 Stretch BitBLT Engine

Stretch BitBLT can stretch source data in the X and Y directions to a destination larger or smaller than the source. Stretch BitBLT functionality expands a region of memory into a larger or smaller region using replication and interpolation.

Stretch BitBLT also provides format conversion and data alignment. Through a hardware DDA algorithm, expansion and shrinking can occur in both the horizontal and vertical directions in both integer and non-integer values.

4.4.3 Hardware Cursor

The Intel740 allows up to 16 cursor patterns to be stored in 4KB. Two sets of registers, contain the x and y position of cursor relative to the upper left hand corner of the display. The following six cursor modes are provided:

- 32x32 2 bpp AND/XOR 2-plane mode
- 128x128 1 bpp 2-color mode
- 128x128 1 bpp 1-color and transparency mode
- 64x64 2 bpp 3-color and transparency mode
- 64x64 2 bpp AND/XOR 2-plane mode
- 64x64 2 bpp 4-color mode

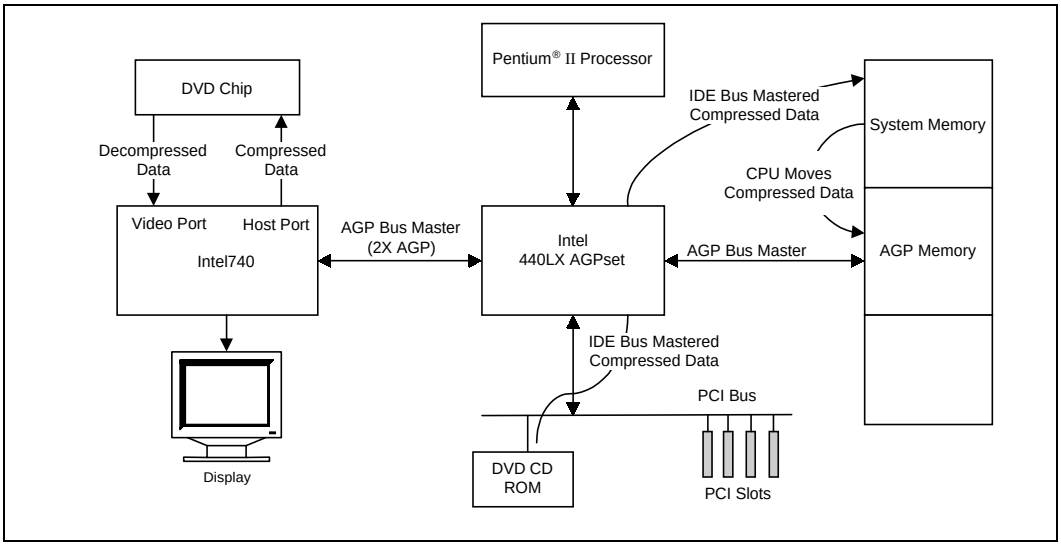
4.5 Video Module Interface (VMI)

The Intel740 VMI consists of a Video Port and a Host Port. The Host Port provides an enhanced VMI 1.4 Mode B Port. The enhancements allow burst modes of operation. The Intel740 Video Port is used to receive decompressed video data from a DVD chip or video data from a Video Decoder chip. A CCIR601 digital interface is supported as the primary capture standard. Using both the host and video ports, DVD, TV, Intercast, and video capture can be achieved. Use of the Intel740 overlay capability allows images from the capture engine to be displayed while being captured.

Figure 4-4 illustrates the flow of data through a system doing DVD playback. Typically, a DVD drive will be attached to EIDE where the DVD compressed data is bus mastered into main memory. The data flow steps are:

1. IDE bus master DVD data to main memory
2. CPU moves compressed data to AGP memory
3. Intel740 uses AGP bus master to move compressed data to the DVD decoder chip (via the Host Port).
4. Decompressed data for the display is then sent back to the Intel740 through the video port.

Figure 4-4. Data Flow For DVD Playback



Note: The Host port is also a bi-directional port for DVD chip register read/write access.

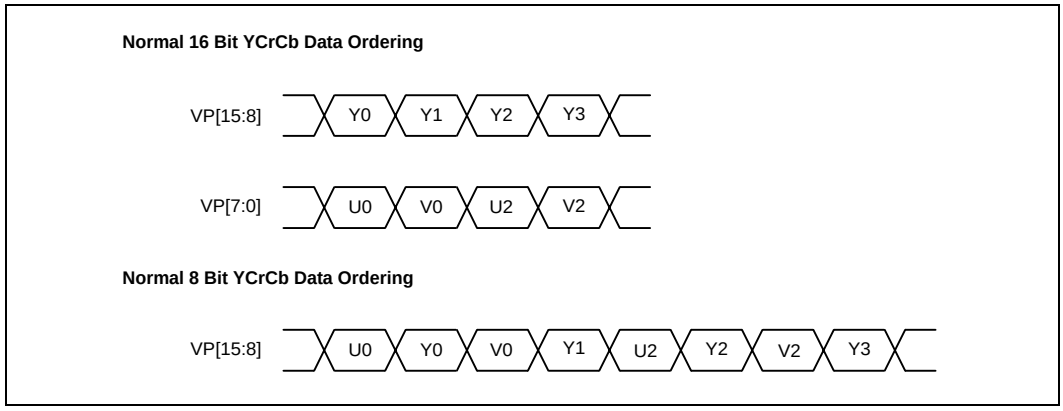
4.5.1 Video Port

The Intel740 Video Port receives decompressed video data from a DVD chip or video data from a Video Decoder chip. A CCIR601 digital interface is supported as the primary capture standard.

4.5.1.1 Capture Formats

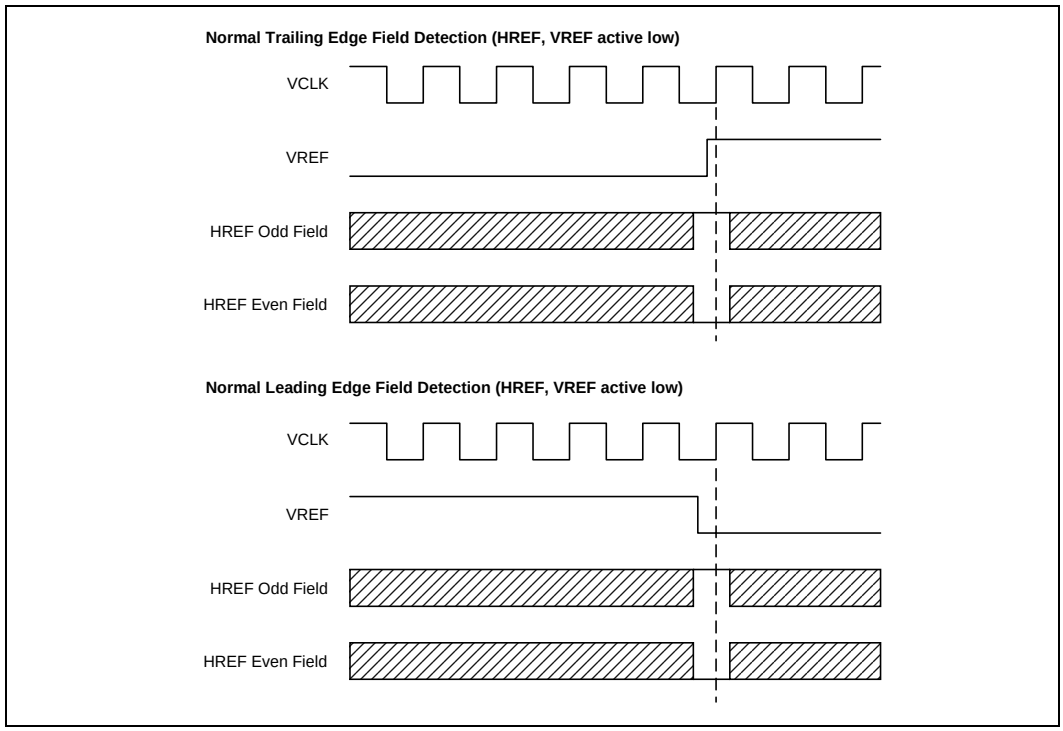
The video capture port can accept digital data in YUV 4:2:2, RGB-15, or RGB-16 format. The data port can be configured for either 16 or 8 bits. Byte ordering is programmable and UV order in YUV format is also programmable. For 16-bit mode, luminance data is received on VP[15:8] (default) or VP[7:0]. For 8-bit mode, time multiplexed luminance/chrominance data is received on VP[15:8]. For RGB input data, red is in the most significant byte position followed by green and blue.

Figure 4-5. Normal 8 and 16 Bit YCrCb Data Ordering



The capture port supports both interlaced and non-interlaced data formats. For interlaced data, fields are detected via the relationship of the VREF and HREF signals. Edge detection and polarity of the VREF and HREF signals are programmable. The Intel740 can be programmed to examine the state of HREF on either the leading or trailing edge of VREF. If HREF is active, then the field is even; if it is inactive, then the field is odd. Interlaced video data can be stored in memory either in interleaved frame format or progressive field format.

Figure 4-6. Capture Port Field Detection



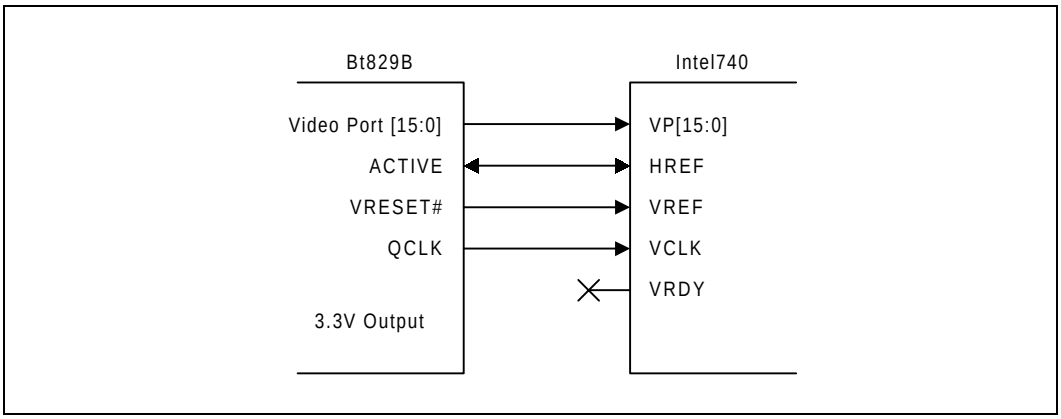
4.5.1.2 Video Port Capabilities

The video port can capture line widths up to a maximum of 1024 pixels. This capture size fully supports DVD line widths. Video Data may be arbitrarily cropped and down scaled in both the horizontal and vertical directions. Horizontal down scaling is achieved through either pixel dropping or applying a 2-tap filter. Vertical down scaling is achieved through line dropping. Video can be mirrored in the horizontal and/or vertical directions.

4.5.1.3 Video Port Example Hook Up

Figure 4-7 illustrates an example of how the video port pins can be connected. For a detailed description, refer to *The Intel740 Design Guide*.

Figure 4-7. Video Port Connection Example



4.5.2 Host Port

The host port provides the host addressees, compressed data and register programming read/write data and address cycles to the DVD chip. Three different cycle types are used: the host write cycle, host read cycle, and burst write cycle.

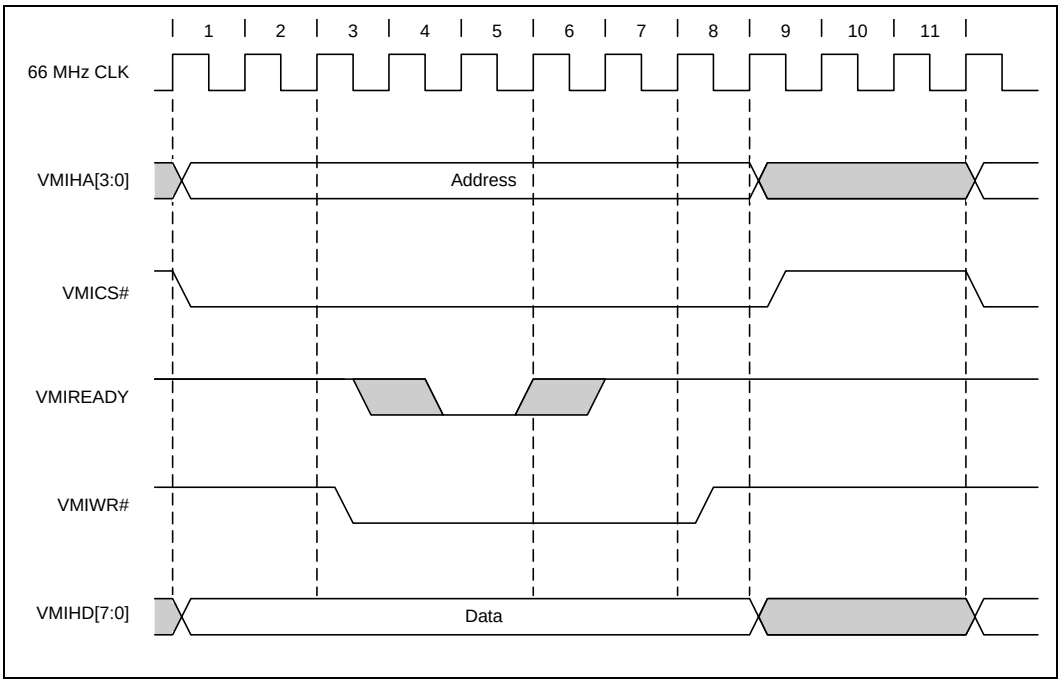
VMI Version 1.4 Mode B protocol is extended. The first extension is the addition of a fifth address bit and the second is the addition of a VMIREQUEST signal used for burst mode writes to the DVD chip. These functions share the same pin.

4.5.2.1 Host Write Cycle

The host write cycle is used to program the internal register set of the DVD chip for device initialization, mode changes and other housekeeping. The VMIHA[3:0] address, along with the VMICS# pin assertion, is used by the DVD chip to access the corresponding register. The Intel740 uses the VMIHD[7:0] data bus to write data to the DVD chip. Figure 4-8 shows the timing when a 66 MHz clock is used. Note that the VMIREADY signal is asynchronous to the clock, may have a short asserted time, and may be in an unknown state when VMICS# is asserted. VMIREADY can be tested at the rising edge of clock 7 and if low, wait states should be inserted until VMIREADY is asserted.

The host write cycle is driven by the CPU writing to the location range[x-y] in the Intel740 memory mapped I/O space. The Intel740 is capable of handling DVD devices that have access times as long as 10 ms.

Figure 4-8. VMI Host Write Cycle



Cycle Detail For VMI Host Write

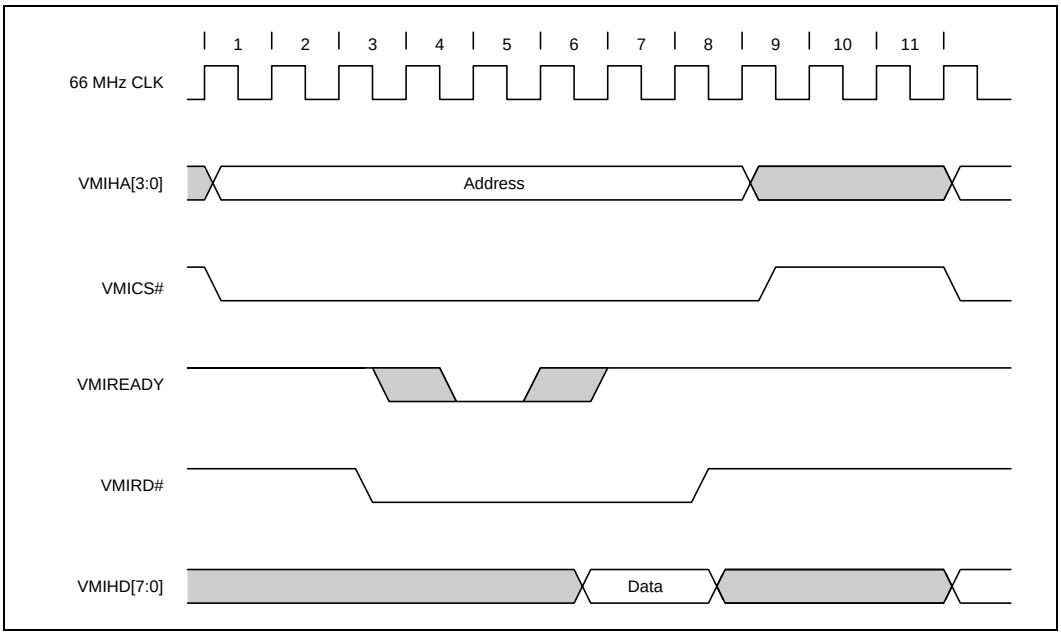
1. CPU posts a write to a DVD Device register; burst writes are put on hold
2. Wait for VMI Bus idle
3. Assert VMIHA[3:0], VMIHD[7:0], VMICS#
4. Wait 2 clocks, then assert VMIWR#
5. Wait 5 clocks, then start testing “Synchronized VMIREADY”; if low, insert wait states; if high, negate VMIWR# (“Synchronized VMIREADY” is delayed 2 clocks from VMIREADY)
6. On the next clock negate VMIHA[3:0] and VMICS#; tri-state VMIHD[7:0]
7. Bus in VMICS# high state for 2 clocks
8. Bus in idle state

4.5.2.2 Host Read Cycle

The host read cycle is used to access the registers of the DVD device for FIFO status and various mode operations. The Intel740 reads the data from the DVD device’s internal registers over the VMIHD[7:0] data bus. Figure 4-9 shows the timing when a 66 MHz clock is used. The VMIREADY signal is asynchronous to the clock, may have a short asserted time, and may be in an unknown state when VMICS# is asserted. VMIREADY can be tested at the rising edge of clock 7 and if it is low, wait states should be inserted until VMIREADY is asserted. Until the rising edge of VMIREADY, wait states are inserted. VMIREADY must be sampled after clock 7.

The host read cycle is driven by the CPU reading from the location range[x-y] in the Intel740 memory mapped I/O space. For DVD devices that have access times as long as 10 ms, the read will likely cause a delayed transaction cycle. Back-to-back reads by the CPU will need to be handled.

Figure 4-9. VMI Host Read Cycle



Cycle Detail For VMI Host Read

1. CPU starts a read to a DVD device register; burst writes are put on hold
2. Wait for VMI Bus idle
3. Assert VMIHA[3:0], VMICS#
4. Wait 2 clocks, then assert VMIRD#
5. Five clocks later start testing “Synchronized VMIREADY”; if low, insert wait states; if high, latch VMIHD[7:0] and negate VMIRD# [data is returned to CPU] (“Synchronized VMIREADY” is delayed 2 clocks from VMIREADY
6. On the next clock negate VMIHA[3:0], VMICS#
7. Bus in VMICS# High state for 2 clocks
8. Bus in idle state

4.5.2.3 Burst Write cycle

To support movement of compressed data from the DVD drive to the DVD device, burst writes have been added to the VMI port. This is a programmable option in the Intel740 that is used to set the interface for VMIREQUEST.

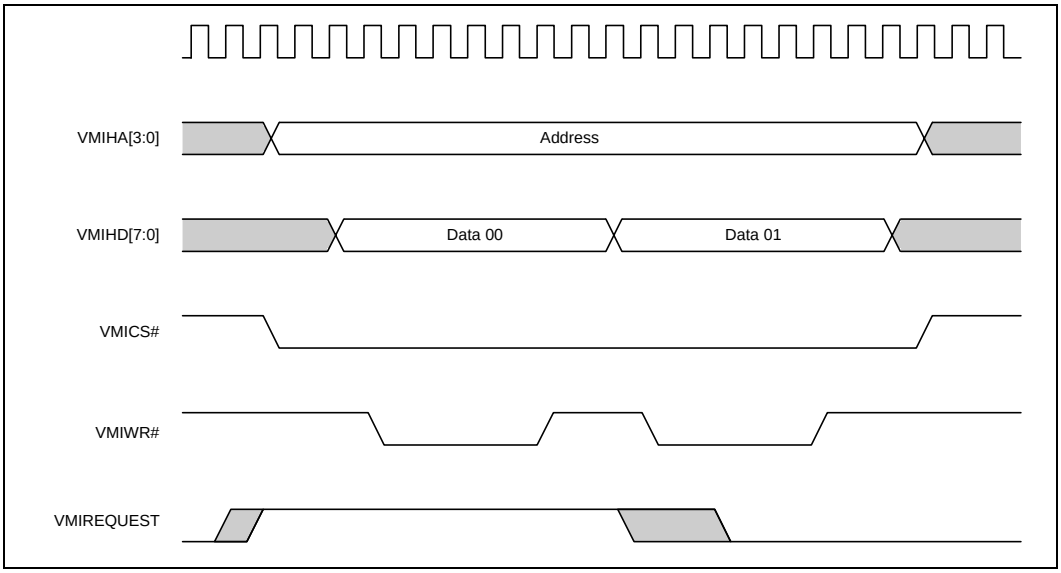
This cycle is only used by the DVD DMA Engine to transfer compressed data to the DVD Device. Figure 4-10 shows the timing when a 66 MHz clock is used. The burst write sends data to one of the DVD device’s 16 registers. The register written is actually a FIFO that accepts data as long as VMIREQUEST is asserted. When a burst write is started, it can be held for the following:

- CPU is accessing a VMI device register
- DVD DMA Hold bit is set
- VMI device VMIREQUEST is negated.

When all of these conditions are removed, the transfer continues. During the burst write, these conditions are tested once per byte transferred on the clock that VMIWR# goes high. If any of these conditions have occurred, the current byte transfer cycle will complete and the next byte write cycle will not be started. Figure 4-10 shows the timing for the burst write cycle.

Note: The DVD register read/write cycles (host access cycle) can take place during the burst write cycle. The current byte write will complete before the host access cycle can access a register. When the host access cycle is completed, the burst write will resume transferring data.

Figure 4-10. VMI Burst Write Cycle



Cycle Detail for VMI Burst Write

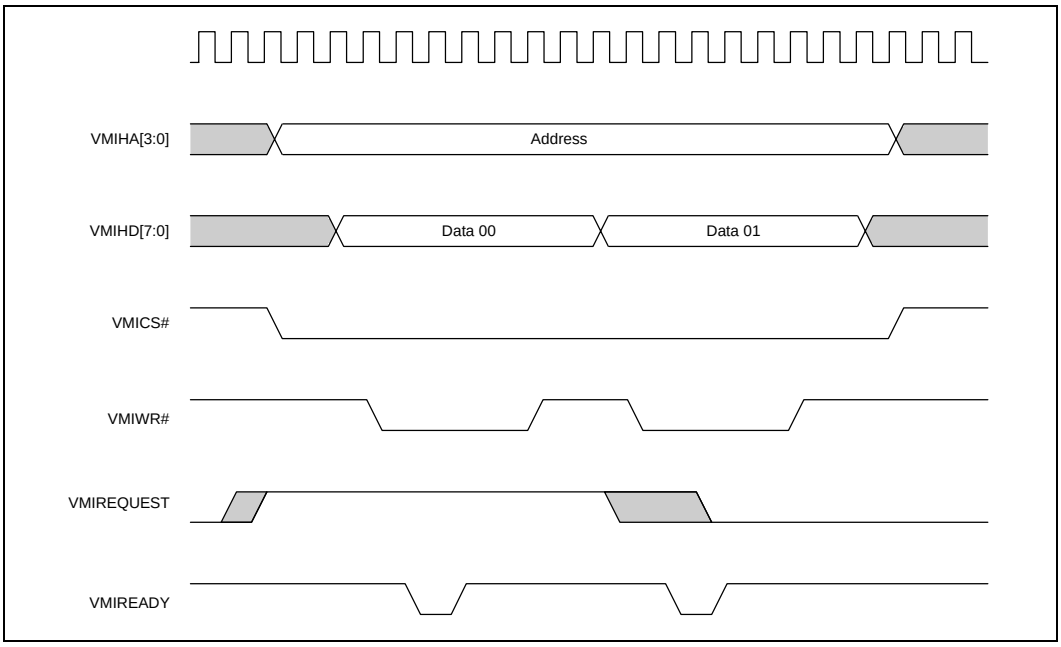
Note: Synchronized VMIREADY and VMIREQUEST are delayed by 2 clocks from the non-synchronized signal.

1. DVD DMA engine loads data into the DVD Port FIFO
2. Wait for Bus Idle
3. Test for ((Synchronized VMIREQUEST == HIGH) AND (DVD DMA Hold bit == FALSE) AND (Pending VMI Read/Write == False)); if FALSE, go to (3); if TRUE, continue
4. Assert VMIHA[3:0], VMICS# (VMIHD[7:0] can be asserted on this clock)
5. Wait 2 clocks; if VMIHD[7:0] was not asserted earlier, it has to be asserted on this clock
6. On next clock assert VMIWR#
7. Wait 5 clocks, negate VMIWR# and test for ((Synchronized VMIREQUEST == HIGH) AND (DVD DMA Hold bit == FALSE) AND (Pending VMI Read/Write == False)); if TRUE, set Next Data Flag
8. Wait 2 clocks; if Next Data Flag is set, put next data byte on VMIHD[7:0] and go to (5); else continue
9. On next clock negate VMIHA[3:0], VMICS#, tri-state VMIHD[7:0]
10. Bus in VMICS# high state for 2 clocks
11. Bus in idle state

4.5.2.4 Burst Write Cycle (Option 1)

To support multiple DVD vendors, an additional option has been added to the burst write cycle. The burst write behaves as outlined in the previous section unless this programmable option is enabled in the Intel740. The only change to the defined burst write behavior is that on the clock that VMIWR# goes high, VMIREADY is checked. If VMIREADY is high, VMIWR# goes high. If VMIREADY is low, the state machine waits for it to go high before setting VMIWR# high and continuing.

Figure 4-11. VMI Burst Write Cycle (Option 1)



Cycle Detail For VMI Burst Write

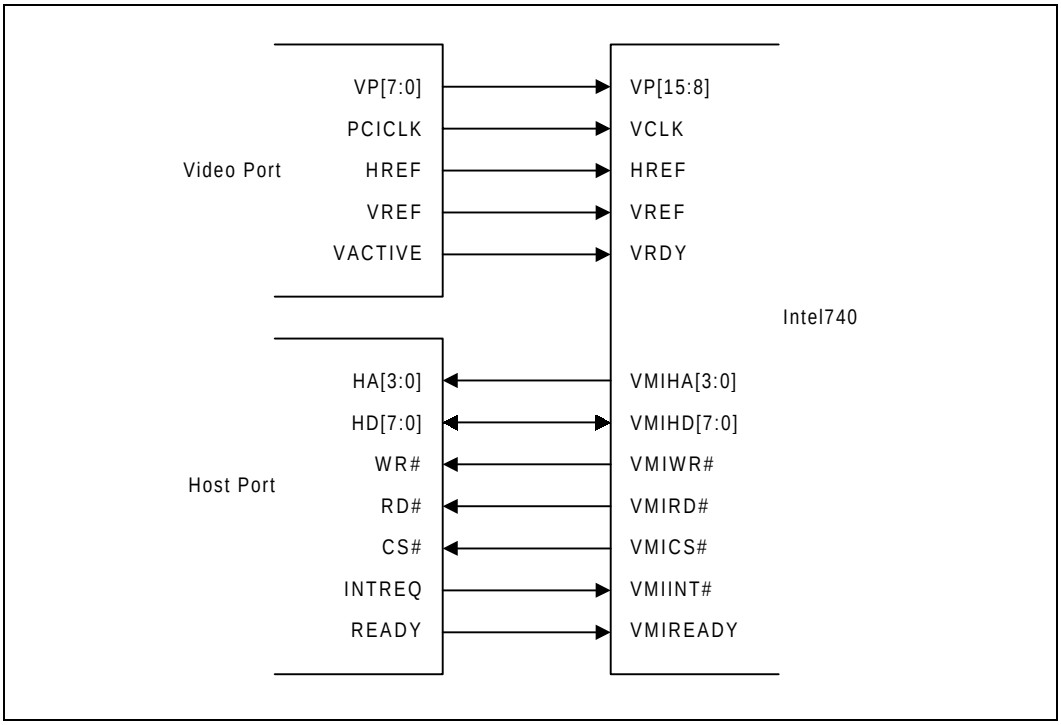
Note: Synchronized VMIREADY and VMIREQUEST are delay by 2 clocks from the non-synchronized signal.

1. DVD DMA engine loads data into the DVD Port FIFO
2. Wait for Bus Idle
3. Test for ((Synchronized VMIREQUEST == HIGH) AND (DVD DMA Hold bit == FALSE) AND (Pending VMI Read/Write == False)); if FALSE, wait; if TRUE, continue
4. Assert VMIHA[3:0], VMICS# (VMIHD[7:0] can be asserted on this clock)
5. Wait 2 clocks; if VMIHD[7:0] was not asserted earlier, it has to be asserted on this clock
6. On next clock assert VMIWR#
7. Five clocks later start testing Synchronized VMIREADY; if low, insert wait states; if high, negate VMIWR# and test for ((Synchronized VMIREQUEST == HIGH) AND (DVD DMA Hold bit == FALSE) AND (Pending VMI Read/Write == False)); if TRUE, set Next Data Flag
8. Wait 2 clocks; if Next Data Flag is set, put next data byte on VMIHD[7:0] and go to 5; else continue
9. On next clock negate VMIHA[3:0] and VMICS#; tri-state VMIHD[7:0]
10. Bus in VMICS# high state for 2 clocks
11. Bus in idle state

4.5.2.5 Host Port Example Hook Up

Figure 4-12 shows a basic connection to a VMI header containing both the video and host ports. This allows the Intel740 to operate in bi-directional mode. Note that only connections for the video are shown. For a complete design example, refer to *The Intel740 Design Guide*.

Figure 4-12. VMI Connection Example For Bi-directional Mode



4.5.3 Overlay Engine

The overlay engine provides a method of merging video capture data with the graphics data on the screen. Supported data formats include YUV 4:2:2, RGB15, and RGB16. The source data can be mirrored horizontally or vertically or both. Overlay data comes from a buffer located in local memory or AGP memory. Data can be double buffered using a pair of 32-bit buffer pointer registers. Data can either be transferred into the overlay buffer from the host or the video capture logic. Buffer swaps can be done by the host and internally synchronized with the display VSYNC. Buffer swaps also automatically happens based on the completed capture frame from the video capture engine and the display VSYNC.

The Intel740 overlay can accept line widths up to 720 pixels. Each image can be enlarged using a 6-bit expansion value filtered in both the horizontal and vertical directions. The horizontal filter is a 3-Tap FIR type and the vertical filter uses either line replication, smoothing at line boundaries, or continuous running average.

4.5.3.1 Field-Based Content

When interlaced video data is stored in progressive field format, the field-based method (also referred to as the “bob” method) of displaying video data is used to show each field individually (similar to television), using an overlay. Vertical scaling is required to stretch the image to the original aspect ratio on a progressive PC monitor. Each field is half the normal height. Thus, for example, it can be vertically zoomed by 2X using an overlay stretch to restore the correct aspect ratio.

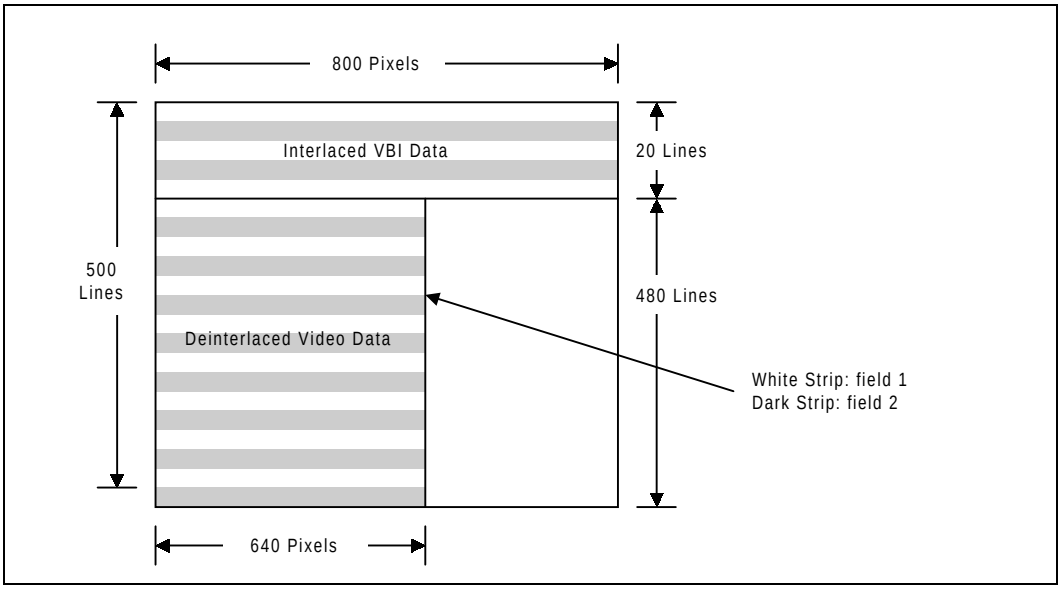
Due to the spatial interlacing of the top and bottom fields, proper vertical position adjustment is required to align the two fields. To prevent the image from jittering up and down the initial phase of the overlay vertical scalar is set to one for the top field. This accommodates the one source line offset between the two fields. This method produces 60 fps (NTSC) field display on progressive monitors and retains all temporal information.

The bob field -based display method can be automatic when showing video from the capture engine. The Intel740 is capable of using field information from incoming video to automatically adjust the overlay vertical position (auto-bob method).

4.5.4 VBI and Intercast Data

When special data is mixed with video data, as it is for VBI or Intercast, the Intel740’s scalars should not be used. It is important to use a capture chip which can send scaled video data with raw VBI data. The Intel740 will accept the VBI data and video data into the same capture buffer where software can separate the two forms of data. Figure 4-13 illustrates this mechanism.

Figure 4-13. Capture Buffer Sharing VBI and Video Data



4.6 TV Out Interface

The Intel740 is addressable as a slave. As a slave, the Intel740 accepts a clock driven by an external TV encoder chip on the FCLKIN pin. The encoder can vary the frequency of the clock to the Intel740 up to a maximum of 40 MHz. The frequency of this clock is dependent on the output resolution required. Data is driven to the encoder along with a clock and blanking signal. The Intel740 indicates that valid data is being output by negating the FBLNK# signal. The encoder should sample FBLNK# with every edge of FCLKOUT. Data should be sampled at the first negation of FBLNK# and then on every rising and falling edge of FCLKOUT while FBLNK# is negated. The encoder can expect a continuous flow of data from the Intel740 as data will not be throttled.

The Intel740 can output to a monitor as well as the encoder chip. The following resolutions are supported by the Intel740 when outputting to the TV encoder:

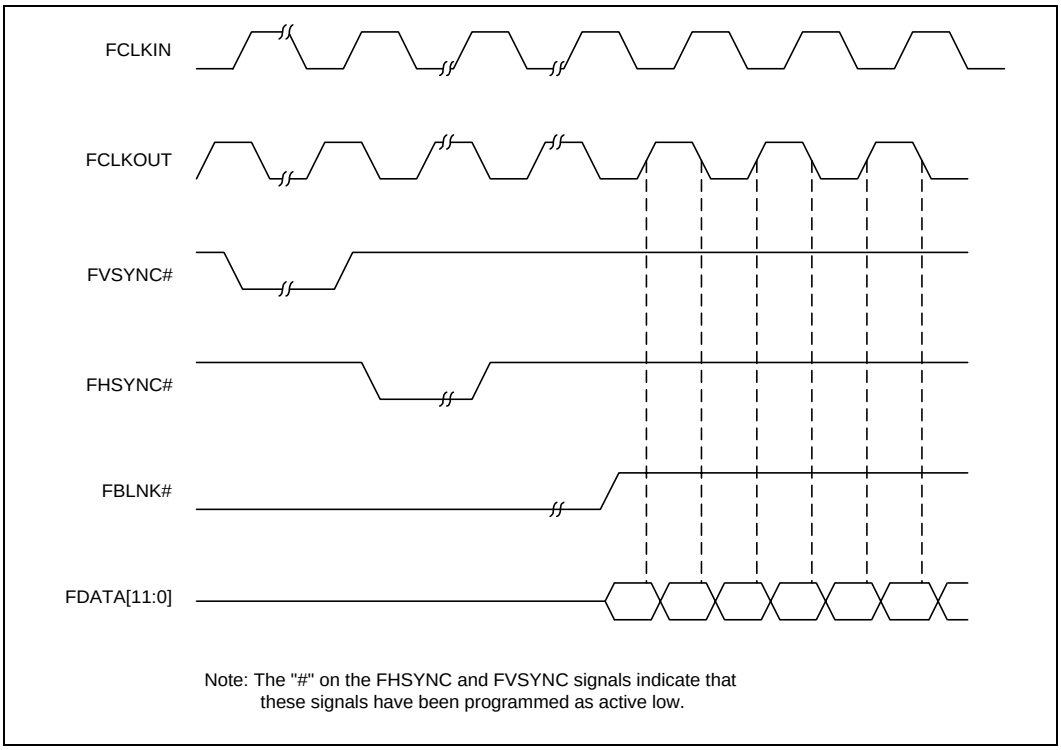
- 320x200
- 320x240
- 640x400
- 640x480
- 720x480
- 800x600

Output to the encoder is in digital 24-bit RGB format. Using a 12-bit bus, the output shown in Table 4-2 is supported:

Table 4-2. TV Out RGB Output Format

Pin	Rising Edge of CLK (24-Bit RGB)	Falling Edge of CLK (24-Bit RGB)
P11	G4	R7
P10	G3	R6
P9	G2	R5
P8	B7	R4
P7	B6	R3
P6	B5	G7
P5	B4	G6
P4	B3	G5
P3	G0	R2
P2	B2	R1
P1	B1	R0
P0	B0	G1

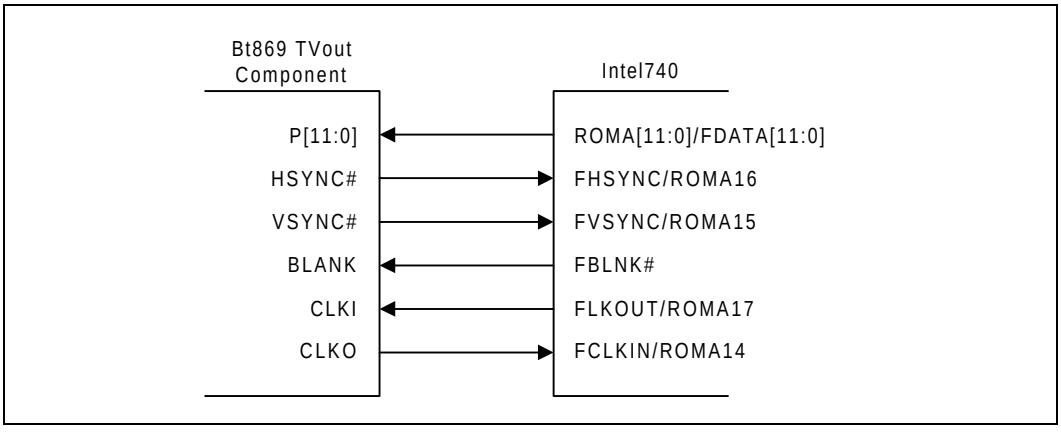
Figure 4-14. Intel740 Slave Mode



4.6.1 TV Out Connection Example

Figure 4-15 shows the connections between the Bt869 TV out component and the Intel740's TV Out pins.

Figure 4-15. TVout Connection Example



4.7 AGP/PCI Interface

The Intel740 provides an AGP/PCI interface to a host bridge. The host bridge must have AGP capabilities (e.g., the 82433LX).

2X AGP Interface

The Accelerated Graphics Port (AGP) brings new levels of performance and realism to next generation 3D graphics accelerators. The principal benefit comes from the graphics accelerator having high speed access to surface textures and other graphics surfaces in main system memory. Special performance-oriented AGP features allow much faster read/write access to these surfaces than has been possible in the past.

For Intel740 accesses to the graphics aperture (located in system memory), the AGP interface to the host bridge is used. The interface is AGP 1.0 compliant. Bus operations are permitted in both 1X and 2X mode. Full 2X AGP implementation is integrated into the Intel740 with sideband operations supporting Type 1, Type 2, and Type 3 sideband cycles. Type 3 support permits textures to be located anywhere in the 32-bit system memory address space.

Combined with side-band addressing, the Intel740 is capable of achieving the highest AGP performance possible. The side-band addressing allows the Intel740 to issue requests without having to wait for data to be written or returned from the host. Internal buffering within the Intel740 accounts for any latency over AGP. This buffering allows the various internal pipelines to proceed at full processing speed without having to wait for data.

Using 2X AGP, the various memory surfaces can be stored and executed directly from AGP memory (DME). Being executed directly from AGP memory allows an Intel740 system to store large textures efficiently for very realistic 3D rendering. When executing directly from AGP memory, the Intel740 orders its accesses to minimize page breaks and maximize memory efficiency.

PCI Interface

For memory accesses outside the graphics aperture, the PCI interface to the host bridge is used (PCI 2.1 compliant). Thus, accesses to the Intel740 registers or CPU accesses to the Intel740's local memory use the interface as a standard PCI bus. In addition, I/O accesses to the Intel740 PCI configuration space or I/O accesses to the Intel740 registers use the bus as a standard PCI Bus.

The Intel740 is capable of being a PCI 2.1 slave and limited PCI bus master. As both AGP and PCI are implemented, intermingled AGP and PCI transactions are possible.

4.8 Local Memory Interface

The Intel740 integrates a Local Memory (LM) controller that supports a 64-bit wide memory data interface running at 66 MHz or 100 MHz and memory sizes up to 8 MB. Synchronous DRAM (SDRAM) and Synchronous Graphics RAM (SGRAM) are supported.

The Local Memory interface signals are described in the *Signal Description* Chapter. The Local Memory interface is configured by a set of programmable registers. These registers are described in the *Register Description* Chapter.

4.8.1 DRAM Types

A page size of 2 KB is supported and up to 8 MB may be accessed across 2 banks. Two technologies in three densities are supported. Table 4-3 shows the various configurations of supported memory.

Table 4-3. Supported Memory Types

SDRAM Technology	SDRAM Density	SDRAM Width	DRAM Addressing	Address Size		DRAM Size	
				Row	Column	Min	Max
8Mbit	256K	32	Asymmetric	10	8	2MB	4MB
16Mbit	512K	32	Asymmetric	11	8	4MB	8MB
	1M	16	Asymmetric	12	8	8MB	8MB

Figure 4-16. 2/4 MB Local Memory Connection (64-bit Data Path)

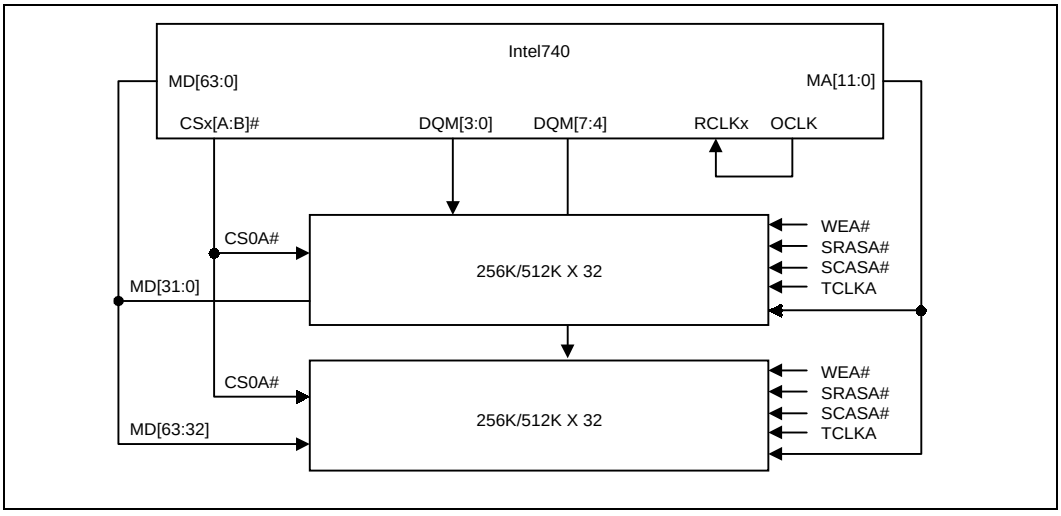


Figure 4-17. 4/8 MB Local Memory Connection (64-bit data path)

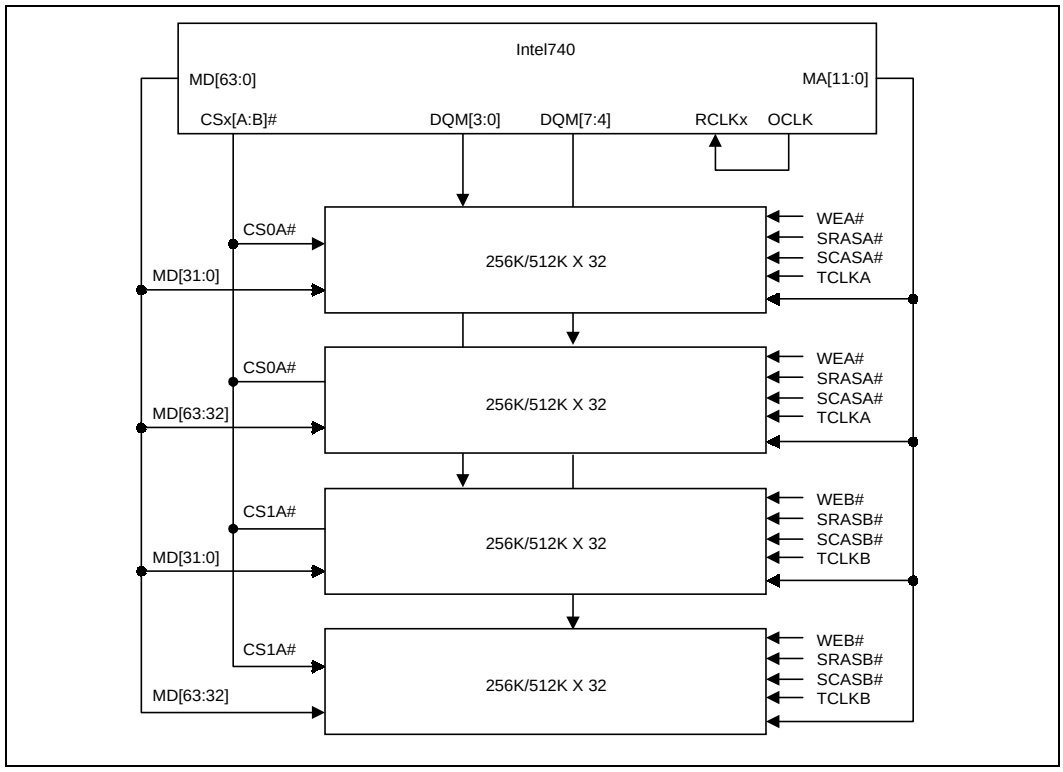


Figure 4-18. 8 MB Local Memory Connection (64-bit data path)

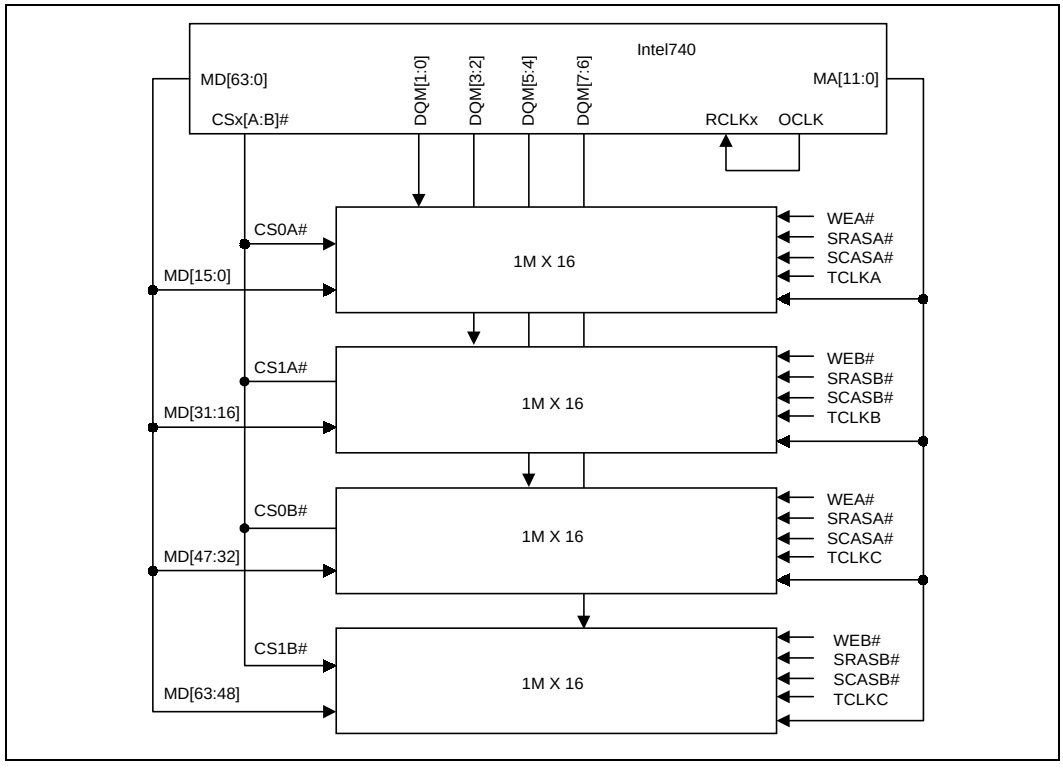


Table 4-4 shows the programming of the DRAM Row Boundary registers (XR55 and XR56) for different configurations.

Table 4-4. DRB Bit Programming

CS0#	CS1#	DRB0	DRB1	Total Memory
2MB	0	02h	02h	2MB
0MB	2MB	00h	02h	2MB
2MB	2MB	02h	04h	4MB
4MB	0MB	04h	04h	4MB
0MB	4MB	00h	04h	4MB
4MB	4MB	04h	08h	8MB
8MB	0MB	08h	08h	8MB
0MB	8MB	00h	08h	8MB

4.8.2 Address Translation

Table 4-5 shows the DRAM Address Translation for the Intel740.

Table 4-5. The Intel740 Address Translation

Memory Address MA[11:0]	11	10	9	8	7	6	5	4	3	2	1	0
Row Address	A11	A22\ A21	A21\ A11	A20	A19	A18	A17	A16	A15	A14	A13	A12
Column Address	X	X	X	X	A10	A9	A8	A7	A6	A5	A4	A3

4.8.3 SDRAM Sizing

To determine the amount of memory and speed of memory on the local memory interface, BIOS performs a series of tests. These tests cycle through the supported memory speeds, the 2 supported banks and determine how much memory is present. Typically, if a read is performed and incorrect data is received, then the BIOS determines that the memory addressed does not exist or has a slower timing than what is being used.

4.8.4 Refresh

The Intel740 supports CAS-before-RAS auto refresh.

4.9 Display

The display function contains a RAM-based Digital-to-Analog Converter (RAMDAC) that transforms the digital data from the graphics and video subsystems to analog data for the monitor. The Intel740's integrated 220 MHz RAMDAC provides resolution support up to 1600 x 1200. Circuitry is incorporated to limit the switching noise generated by the DACs. Three 8-bit DACs provide the R, G, & B signals to the monitor. Sync signals are properly delayed to match any delays from the D-to-A conversion. Associated with each DAC is a 256 pallet of colors. The RAMDAC can be operated in either direct or indexed color mode. In Direct color mode, pixel depths of 15, 16, or 24 bit can be realized. Non-interlaced mode is supported. Gamma correction can be applied to the display output.

Table 4-6. Dispaly Modes Supported

Resolution	Bits Per Pixel (frequency inHz)		
	8-bit Indexed	16-bit	24-bit
320x200	60,72,75,85	60,72,75,85	60,72,75,85
320x240	60,72,75,85	60,72,75,85	60,72,75,85
512x384	60,72,75,85	60,72,75,85	60,72,75,85
640x350	85	85	85
640x480	60,72,75,85	60,72,75,85	60,72,75,85
800x600	56,60,72,75,85	56,60,72,75,85	56,60,72,75,85
1024x768	60,70,75,85	60,70,75,85	60,70,75,85
1280x1024	60,72,75,85	60,72,75	—
1600x1200	60,75	—	—

4.10 DDC (Display Data Channel)

DDC is a standard defined by VESA. Its purpose is to allow communication between the host system and display. Both configuration and control information can be exchanged allowing plug-and-play systems to be realized. Support for DDC 2B is implemented.

4.11 I^2C Bus

The I^2C Bus is incorporated on the Intel740 as a 3.3 V Bus. The I^2C bus allows a serial communications to exist between many devices. Use of this bus allows the Intel740 to program TV encoders and video decoders for various configurations.

4.12 BIOS

The 18 address lines of the BIOS interface (ROMA[17:0]) allow support for up to 256K x 8 BIOS. Hardware support is incorporated that allows the BIOS to be in-circuit programmable. Thus, a flash BIOS can be used. BIOS is shadowed in system memory. Thus, the BIOS address lines are shared with the TV out interface and subsystem vendor ID strapping hardware. A BIOS with up to 250ns access time can be used.

Electrical, Pinout, and Package Information 5

This chapter contains Intel740 absolute maximum ratings, thermal characteristics, and DC characteristics. Pinout and package information are also provided.

5.1 Electrical Characteristics

5.1.1 Absolute Maximum Ratings

Case Temperature under Bias	0℃ to +108℃
Storage Temperature	-55℃ to +150℃
Voltage on 3.3V tolerant Pins with Respect to Ground	-0.3 to VCC + 0.3 V
Voltage on AGP tolerant Pins with Respect to Ground	-0.3 to VCC_{AGP} + 0.3V
3.3V Supply Voltage with Respect to Vss (VCC)	-0.3 to + 4.3 V

Warning: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operating beyond the "Operating Conditions" is not recommended and extended exposure beyond "Operating Conditions" may affect reliability.

5.1.2 Thermal Characteristics

The Intel740 is designed for operation at case temperatures between 0°C and 108°C. The thermal resistance of the package is given in Table 5-1.

Table 5-1. Intel740 Package Thermal Resistance

Parameter	Air Flow Meters/Second (Linear Feet per Minute)	
	0 (0)	1.0 (196.9)
$Theta_{jc}$ (°C/Watt)	7.5	
$Theta_{ja}$ (°C/Watt)	13.0	11.0

5.1.3 Power Characteristics

Table 5-2. Power Dissipation Characteristics

Symbol	Parameter	Min	Max	Unit	Notes
P_{740}	Thermal Power Dissipation For the Intel740		TBD	W	
I_{LEAK}	5.0V to 3.3V Power Supply Leakage Current		1	uA	
I_{CC-740}	Power Supply Current		TBD	mA	

5.1.4 D.C. Characteristics

Table 5-3. D.C. Characteristics

Symbol	Parameter	Min	Max	Unit	Notes
VIL1	LVTTL input low voltage	-0.3	0.8	V	
VIH1	LVTTL input high voltage	2.0	Vcc+0.3	V	
VIL2	AGP input low voltage	-0.5	0.3Vcc	V	
VIH2	AGP input high voltage	Vcc+0.5	0.5Vcc	V	
VOL1	LVTTL output low voltage		0.4	V	
VOH1	LVTTL output high voltage	2.4		V	
VOL2	AGP output low voltage		0.1Vcc	V	
VOH2	AGP output high voltage	0.9Vcc		V	
IOL1	LVTTL output low current		3	mA	
IOH1	LVTTL output high current	-2		mA	
IOL2	AGP output low current		1.5	mA	
IOH2	AGP output high current	-2		mA	
IIL1	Input leakage current		+ 10	uA	
IIH1	Input leakage current		- 10	uA	
IIL2	AGP input leakage current		±10	uA	
IIH2	AGP Input high leakage current		70	uA	Vin = 2.7V
VAGPREF	AGP Reference Voltage		0.4V_{DDQ}	V	

5.2 Pinout Information

Table 5-4. Intel740 Pinout (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	VSS	VSS	VSS	VSS	VSS	ROMD1	ROMA0	ROMA2	ROMA4	ROMA6	ROMA12	ROMA16	GPIO8
B	VSS	VSS	VSS	VSS	VSS	ROMD2	ROMD0	ROMA1	ROMA3	ROMA5	ROMA7	ROMA15	GPIO7
C	VSS	VSS	VSS	VSS	VSS	ROMD3	ROMD5	ROMD7	ROMOE#	ROMA9	ROMA13	ROMA17	ROMWE#
D	VSS	VSS	VSS	VSS	VSS	VCCP	ROMD4	ROMD6	ROMA10	ROMA11	ROMA8	ROMA14	VCCP
E	VSS	VSS	VSS	VSS	VSS	VSS	VCC	VCCP	VCC	VCCP	VCC	VCCP	VCC
F	VCCDP	CLK	VSSA	VCCAP	VSS	VSS	VSS						
G	ST0	REQ#	RESET#	INT#	VDDQ	VSS							
H	SBA0	RBF#	ST2	ST1	GNT#								
J	SBSTB	SBA2	SBA3	SBA1	VDDQ								
K	AD31	SBA6	SBA4	SBA7	SBA5								
L	AD27	AD29	AD28	AD30	VDDQ						VSS	VSS	VSS
M	ADSTB_B	AD25	C/BE3#	AD24	AD26						VSS	VSS	VSS
N	AD21	AD23	AD20	AD22	VDDQ						VSS	VSS	VSS
P	AD19	AD17	AD16	AD18	VDDQ						VSS	VSS	VSS
R	C/BE2#	AGPREF	FRAME#	TRDY#	STOP#						VSS	VSS	VSS
T	IRDY#	DEVSEL#	PAR	AD15	VDDQ						VSS	VSS	VSS
U	C/BE1#	AD14	AD12	AD13	AD11								
V	AD10	AD8	AD9	C/BE0#	VDDQ								
W	ADSTB_A	AD7	AD5	AD6	AD4								
Y	AD3	AD1	AD2	AD0	VDDQ	VSS							
AA	VCCDP	VSSA	VSSA	VCCAP	VSS	VSS	VSS						
AB	VSS	VSS	VSS	VSS	VSS	VSS	VCC	VCCP	VCC	VCCP	VCC	VCCP	VCC
AC	VSS	VSS	VSS	VSS	VSS	VCCP	MD2	MD6	MD10	MD14	RCLK0	MD16	VCCP
AD	VSS	VSS	VSS	VSS	VSS	MD0	MD4	MD8	MD12	DQM0	DQM2	MD18	MD20
AE	VSS	VSS	VSS	VSS	VSS	MD1	MD7	NC	MD11	MD15	DQM3	MD19	MD23
AF	VSS	VSS	VSS	VSS	VSS	MD3	MD5	MD9	MD13	DQM1	MD17	MD21	MD25

NOTES:
1. The ROMAx signals are multiplexed with the Digital TV Out signals. See the following Alphabetical Pin List table for details.
2. The VP[7:0] are multiplexed with the VMI Host Data signals (VMIHD[7:0]). See the following Alphabetical Pin List table for details.
3. VMIHA4 is multiplexed with VMIREQUEST.
4. GPIO[3:0] have the following specified uses in Intel supplied board and software designs: GPIO3=DDCCLK, GPIO2=DDCDATA, GPIO1=I^2C Clock, and GPIO0=I^2C Data.

Table 5-5. Intel740 Pinout (Top View)

14	15	16	17	18	19	20	21	22	23	24	25	26	
GPIO6	VMIHA4	VMIHA0	VMIRDY	VP1	VP5	VP9	VP13	VSS	VSS	VSS	VSS	VSS	A
GPIO5	VMIHA3	VMICS#	VMIINTR#	VP3	VP7	VP11	VP15	VSS	VSS	VSS	VSS	VSS	B
GPIO4	VMIHA2	VMIWR#	VP0	VP4	VP8	VP12	GNDIN	VSS	VSS	VSS	VSS	VSS	C
VCCP	VMIHA1	VMIRD#	VP2	VP6	VP10	VP14	REFSET	VSS	VSS	VSS	VSS	VSS	D
VCC	VCCP	VCC	VCCP	VCC	VCCP	VCC	VSS	VSS	VSS	VSS	VSS	VSS	E
						VSS	VSS	VSS	VSSA	RED	VCCA	GREEN	F
							VSS	VCCP	VCCA	BLUE	VCCP	VCCA	G
								VCC	XTAL1	VSSA	NC	NC	H
								VCCP	NC	VRDY	VSYNC	HSYNC	J
								HREF	GPIO0	GPIO1	GPIO2	GPIO3	K
VSS	VSS	VSS						VCCP	VREF	VCLK	TEST	MD63	L
VSS	VSS	VSS						MD62	MD60	MD58	MD61	MD59	M
VSS	VSS	VSS						VCCP	MD56	MD54	NC	MD57	N
VSS	VSS	VSS						VCCP	MD50	MD52	MD53	MD55	P
VSS	VSS	VSS						DQM4	DQM6	MD48	MD49	MD51	R
VSS	VSS	VSS						VCCP	MD46	RCLK1	DQM5	DQM7	T
								MD40	MD42	MD44	MD45	MD47	U
								VCCP	MD38	NC	MD41	MD43	V
								MD32	MD34	MD36	MD37	MD39	W
							VSS	VCCP	CS0B#	CS1B#	MD33	MD35	Y
						VSS	VSS	VSS	CKE	NC	CS0A#	CS1A#	AA
VCC	VCCP	VCC	VCCP	VCC	VCCP	VCC	VSS	VSS	VSS	VSS	VSS	VSS	AB
VCCP	MD26	MD30	MA2	MA6	MA10	SCASB#	VCCP	VSS	VSS	VSS	VSS	VSS	AC
MD22	MD24	MD28	MA0	MA4	MA8	SCASA#	OCLK	VSS	VSS	VSS	VSS	VSS	AD
NC	MD31	MA3	MA7	MA11	WEB#	SRASB#	TCLKB	VSS	VSS	VSS	VSS	VSS	AE
MD27	MD29	MA1	MA5	MA9	WEA#	SRASA#	TCLKA	VSS	VSS	VSS	VSS	VSS	AF

NOTES:

1. The ROMAx signals are multiplexed with the Digital TV Out signals. See the following Alphabetical Pin List table for details.
2. The VP[7:0] are multiplexed with the VMI Host Data signals (VMIHD[7:0]). See the following Alphabetical Pin List table for details.
3. VMIHA4 is multiplexed with VMIREQUEST.
4. GPIO[3:0] have the following specified uses in Intel supplied board and software designs: GPIO3=DDCCLK, GPIO2=DDCDATA, GPIO1=I^2C Clock, and GPIO0=I^2C Data.

Table 5-6. Alphabetical Pin List (Sheet 1 of 3)

Name	Ball #	Type	Name	Ball #	Type	Name	Ball #	Type
AD0	Y4	I/O	CS0A#	AA25	O	MD0	AD6	I/O
AD1	Y2	I/O	CS0B#	Y23	O	MD1	AE6	I/O
AD2	Y3	I/O	CS1A#	AA26	O	MD2	AC7	I/O
AD3	Y1	I/O	CS1B#	Y24	O	MD3	AF6	I/O
AD4	W5	I/O	DEVSEL#	T2	I/O	MD4	AD7	I/O
AD5	W3	I/O	DQM0	AD10	O	MD5	AF7	I/O
AD6	W4	I/O	DQM1	AF10	O	MD6	AC8	I/O
AD7	W2	I/O	DQM2	AD11	O	MD7	AE7	I/O
AD8	V2	I/O	DQM3	AE11	O	MD8	AD8	I/O
AD9	V3	I/O	DQM4	R22	O	MD9	AF8	I/O
AD10	V1	I/O	DQM5	T25	O	MD10	AC9	I/O
AD11	U5	I/O	DQM6	R23	O	MD11	AE9	I/O
AD12	U3	I/O	DQM7	T26	O	MD12	AD9	I/O
AD13	U4	I/O	FRAME#	R3	I/O	MD13	AF9	I/O
AD14	U2	I/O	GNDIN	C21	I	MD14	AC10	I/O
AD15	T4	I/O	GNT#	H5	I	MD15	AE10	I/O
AD16	P3	I/O	GPIO0 (I^2C Data)	K23	I/O	MD16	AC12	I/O
AD17	P2	I/O	GPIO1 (I^2C Clock)	K24	I/O	MD17	AF11	I/O
AD18	P4	I/O	GPIO2 (DDCDATA)	K25	I/O	MD18	AD12	I/O
AD19	P1	I/O	GPIO3 (DDCCLK)	K26	I/O	MD19	AE12	I/O
AD20	N3	I/O	GPIO4	C14	I/O	MD20	AD13	I/O
AD21	N1	I/O	GPIO5	B14	I/O	MD21	AF12	I/O
AD22	N4	I/O	GPIO6	A14	I/O	MD22	AD14	I/O
AD23	N2	I/O	GPIO7	B13	I/O	MD23	AE13	I/O
AD24	M4	I/O	GPIO8	A13	I/O	MD24	AD15	I/O
AD25	M2	I/O	GREEN	F26	O	MD25	AF13	I/O
AD26	M5	I/O	HREF	K22	I	MD26	AC15	I/O
AD27	L1	I/O	HSYNC	J26	O	MD27	AF14	I/O
AD28	L3	I/O	INT#	G4	O	MD28	AD16	I/O
AD29	L2	I/O	IRDY#	T1	I/O	MD29	AF15	I/O
AD30	L4	I/O	MA0	AD17	O	MD30	AC16	I/O
AD31	K1	I/O	MA1	AF16	O	MD31	AE15	I/O
ADSTB_A	W1	I/O	MA2	AC17	O	MD32	W22	I/O
ADSTB_B	M1	I/O	MA3	AE16	O	MD33	Y25	I/O
AGPREF	R2	I	MA4	AD18	O	MD34	W23	I/O
BLUE	G24	O	MA5	AF17	O	MD35	Y26	I/O
C/BE0#	V4	I/O	MA6	AC18	O	MD36	W24	I/O
C/BE1#	U1	I/O	MA7	AE17	O	MD37	W25	I/O
C/BE2#	R1	I/O	MA8	AD19	O	MD38	V23	I/O
C/BE3#	M3	I/O	MA9	AF18	O	MD39	W26	I/O
CKE	AA23	O	MA10	AC19	O	MD40	U22	I/O
CLK	F2	I	MA11	AE18	O	MD41	V25	I/O

Table 5-6. Alphabetical Pin List (Sheet 2 of 3)

Name	Ball #	Type	Name	Ball #	Type	Name	Ball #	Type
MD42	U23	I/O	RESET#	G3	I	SBA1	J4	O
MD43	V26	I/O	ROMA0/ FDATA0	A7	I/O	SBA2	J2	O
MD44	U24	I/O	ROMA1/ FDATA1	B8	I/O	SBA3	J3	O
MD45	U25	I/O	ROMA2/ FDATA2	A8	I/O	SBA4	K3	O
MD46	T23	I/O	ROMA3/ FDATA3	B9	I/O	SBA5	K5	O
MD47	U26	I/O	ROMA4/ FDATA4	A9	I/O	SBA6	K2	O
MD48	R24	I/O	ROMA5/ FDATA5	B10	I/O	SBA7	K4	O
MD49	R25	I/O	ROMA6/ FDATA6	A10	I/O	SBSTB	J1	O
MD50	P23	I/O	ROMA7/ FDATA7	B11	I/O	SCASA#	AD20	O
MD51	R26	I/O	ROMA8/ FDATA8	D11	I/O	SCASB#	AC20	O
MD52	P24	I/O	ROMA9/ FDATA9	C10	I/O	SRASA#	AF20	O
MD53	P25	I/O	ROMA10/ FDATA10	D9	I/O	SRASB#	AE20	O
MD54	N24	I/O	ROMA11/ FDATA11	D10	I/O	ST0	G1	I
MD55	P26	I/O	ROMA12/ FBLINK#	A11	I/O	ST1	H4	I
MD56	N23	I/O	ROMA13/ FFIELD	C11	I/O	ST2	H3	I
MD57	N26	I/O	ROMA14/ FCLKIN	D12	I/O	STOP#	R5	I/O
MD58	M24	I/O	ROMA15 VSYNC	B12	I/O	TCLKA	AF21	O
MD59	M26	I/O	ROMA16 FHSYNC	A12	I/O	TCLKB	AE21	O
MD60	M23	I/O	ROMA17 FCLKOUT	C12	I/O	TEST	L25	I
MD61	M25	I/O	ROMD0	B7	I/O	TRDY#	R4	I/O
MD62	M22	I/O	ROMD1	A6	I/O	VCLK	L24	I
MD63	L26	I/O	ROMD2	B6	I/O	VMICS#	B16	O
OCLK	AD21	O	ROMD3	C6	I/O	VMIHA0	A16	O
PAR	T3	I/O	ROMD4	D7	I/O	VMIHA1	D15	O
RBF#	H2	O	ROMD5	C7	I/O	VMIHA2	C15	O
RCLK0	AC11	I	ROMD6	D8	I/O	VMIHA3	B15	O
RCLK1	T24	I	ROMD7	C8	I/O	VMIHA4/ VMIREQUEST	A15	O
RED	F24	O	ROMOE#	C9	O	VMIINTR#	B17	I
REFSET	D21	I	ROMWE#	C13	O	VMIRD#	D16	O
REQ#	G2	O	SBA0	H1	O	VMIRDY	A17	I

Table 5-6. Alphabetical Pin List (Sheet 3 of 3)

Name	Ball #	Type	Name	Ball #	Type	Name	Ball #	Type
VMIWR#	C16	O	VP7/ VMIHD7	B19	I/O	VP15	B21	I/O
VP0/ VMIHD0	C17	I/O	VP8	C19	I/O	VRDY	J24	I
VP1/ VMIHD1	A18	I/O	VP9	A20	I/O	VREF	L23	I
VP2/ VMIHD2	D17	I/O	VP10	D19	I/O	VSYNC	J25	O
VP3/ VMIHD3	B18	I/O	VP11	B20	I/O	WEA#	AF19	O
VP4/ VMIHD4	C18	I/O	VP12	C20	I/O	WEB#	AE19	O
VP5/ VMIHD5	A19	I/O	VP13	A21	I/O	XTAL1	H23	I
VP6/ VMIHD6	D18	I/O	VP14	D20	I/O			

Table 5-7. Pinout For Power, Ground, and No Connects

Name	Pin Number
VCC	AB7, AB9, AB11, AB13, AB14, AB16, AB18, AB20, E7, E9, E11, E13, E14, E16, E18, E20, H22
VCCA	F25, G23, G26
VCCAP	F4, AA4
VCCDP	F1, AA1
VCCP	AB8, AB10, AB12, AB15, AB17, AB19, AC6, AC13, AC14, AC21, D6, D13, D14, E8, E10, E12, E15, E17, E19, G25, G22, J22, L22, N22, P22, T22, V22, Y22
VDDQ	G5, J5, L5, N5, P5, T5, V5, Y5
VSSA	AA2, AA3, F3, F23, H24
VSS	A1, A2, A3, A4, A5, A22, A23, A24, A25, A26, B1, B2, B3, B4, B5, B22, B23, B24, B25, B26, C1, C2, C3, C4, C5, C22, C23, C24, C25, C26, D1, D2, D3, D4, D5, D22, D23, D24, D25, D26, E1, E2, E3, E4, E5, E6, E21, E22, E23, E24, E25, E26, F5, F6, F7, F20, F21, F22, G6, G21, L11, L12, L13, L14, L15, L16, M11, M12, M13, M14, M15, M16, N11, N12, N13, N14, N15, N16, P11, P12, P13, P14, P15, P16, R11, R12, R13, R14, R15, R16, T11, T12, T13, T14, T15, T16, Y6, Y21, AA5, AA6, AA7, AA20, AA21, AA22, AB1, AB2, AB3, AB4, AB5, AB6, AB21, AB22, AB23, AB24, AB25, AB26, AC1, AC2, AC3, AC4, AC5, AC22, AC23, AC24, AC25, AC26, AD1, AD2, AD3, AD4, AD5, AD22, AD23, AD24, AD25, AD26, AE1, AE2, AE3, AE4, AE5, AE22, AE23, AE24, AE25, AE26, AF1, AF2, AF3, AF4, AF5, AF22, AF23, AF24, AF25, AF26
NC	AA24, AE8, AE14, H25, H26, J23, N25, V24

5.3 Package Information

This specification outlines the mechanical dimensions for the Intel740. The package is a 468 super thermal ball grid array (STBGA).

Figure 5-1. The Intel740 Package Dimensions (468 STBGA) —Top and Side Views

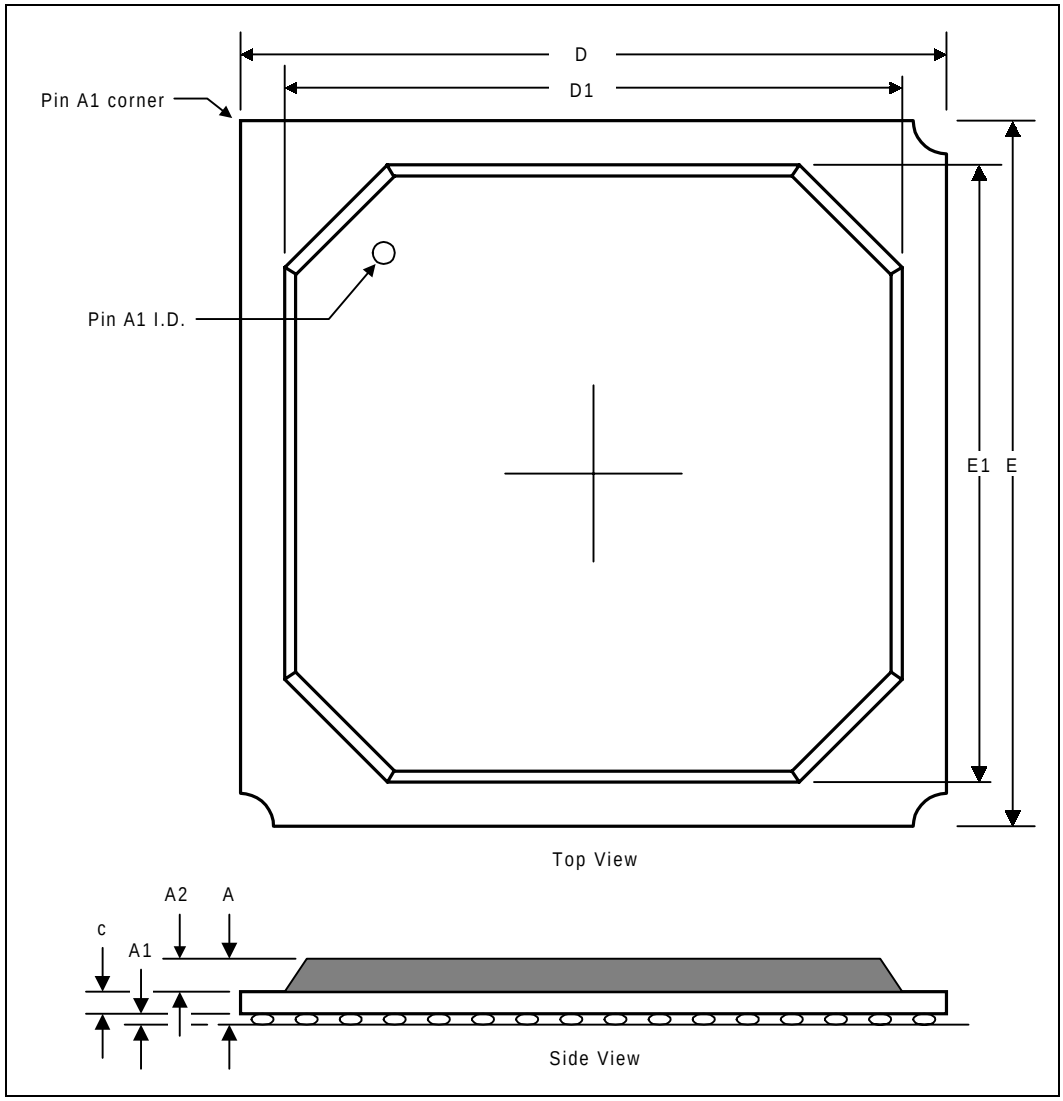


Figure 5-2. The Intel740 Package Dimensions (468 STBGA)

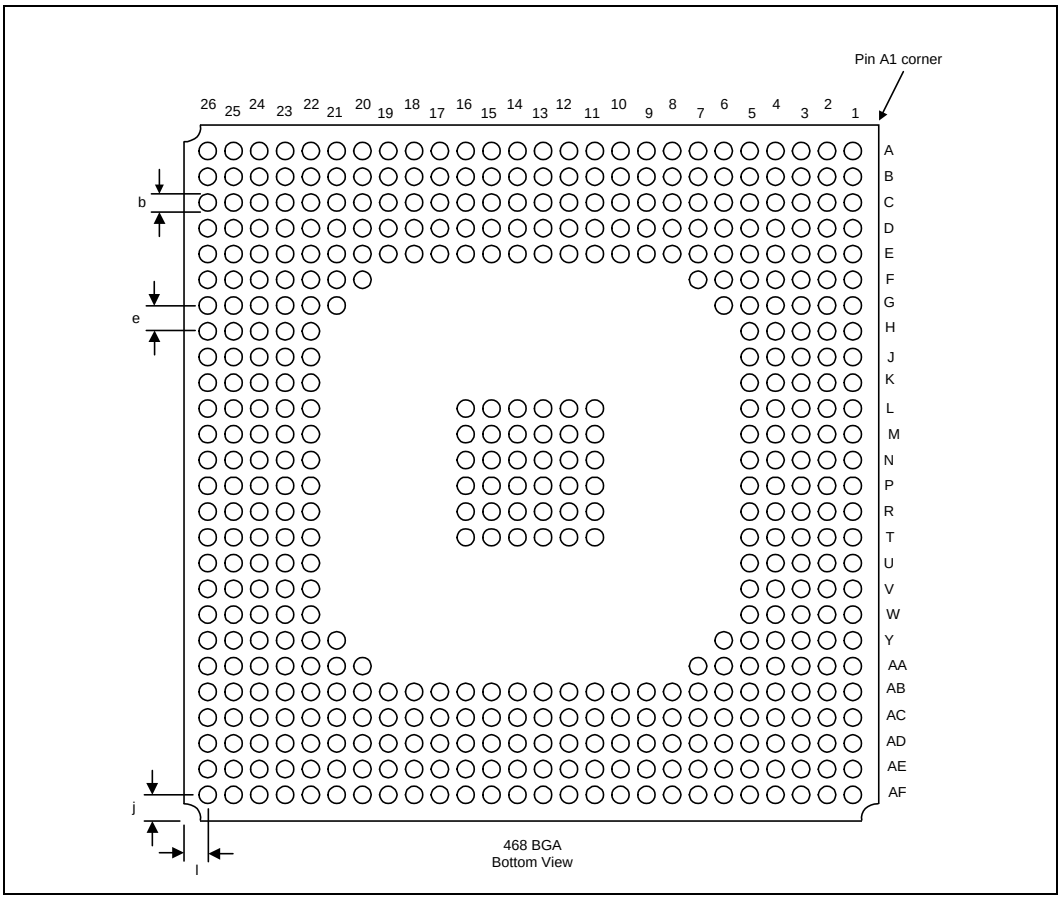


Table 5-8. Intel740 Package Dimensions (468 STBGA)

Symbol	e=1.27 mm (solder ball pitch)			Note
	Min	Nominal	Max	
A	2.17	2.38	2.59	
A1	0.50	0.60	0.70	
A2	1.12	1.17	1.22	
D	34.80	35.00	35.20	
D1	29.75	30.00	30.25	
E	34.80	35.00	35.20	
E1	29.75	30.00	30.25	
I	1.63 REF.			
J	1.63 REF.			
M	26 x 26 Matrix			
N	4.92			
b	0.60	0.75	0.90	
c	0.55	0.61	0.67	

Testability 6

The Intel740 test modes described in this chapter are for Automated Test Equipment (ATE) board level testing. To enable a test mode, the ROMA16 pin is sampled just after Reset. If its value is a 1, then TEST and (SRASB#, SCASB#, WEB#) are used to latch and decode the Test Modes. For a particular test mode, the values shown in Table 6-1 are presented on the indicated signal pins.

Table 6-1. Intel740 Test Mode Select

ROMA16	TEST	CS1[B:A]#	SRASB#	SCASB#	WEB#	Test Mode Enabled
1	1	00	0	1	0	Tri-State All Outputs
1	1	00	0	0	1	NAND Chain Test
0	X	XX	X	X	X	Disable All Test Modes

The NAND chain test allows the ATE to test the signal pin connectivity. There are four NAND chains that are enabled simultaneously.

Table 6-2. NAND Chain Outputs

Pins for NAND Chain	Purpose
DDRSTR3#	NAND Chain 0 Output
GPIO0	NAND Chain 1 Output
GPIO1	NAND Chain 2 Output
DDREF	NAND Chain 3 Output

The following signals are excluded from the NAND Chain:

- Analog outputs (Red, Green, Blue)
- RESET#
- TEST
- ROMWE
- ROMOE#
- DDRSTR[2:0]#

The NAND chain ordering is given in Table 6-3.

Table 6-3. NAND Chain Ordering

NAND Chain 0	NAND Chain 1	NAND Chain 2	NAND Chain 3
INT#	MD0	CKE	VP15
REQ#	MD1	CS0A#	VP13
ST0	MD3	CS0B#	VP14
GNT#	MD2	CS1A#	VP12
ST1	MD4	CS1B#	VP11
ST2	MD7	MD33	VP9
RBF#	MD5	MD35	VP10
SBA0	MD6	MD32	VP8
SBA1	MD8	MD34	VP7
SBA3	MD9	MD36	VP5
SBA2	MD10	MD37	VP6
SBSTB	MD12	MD39	VP4
SBA5	MD11	MD38	VP3
SBA7	MD13	MD41	VP1
SBA4	MD14	MD43	VP2
SBA6	DQM0	MD40	VP0
AD31	MD15	MD42	VMIINT#
AD30	DQM1	MD44	VMIRDY
AD28	RCLK0	MD45	VMIRD#
AD29	DQM2	MD47	VMIWR#
AD27	DQM3	MD46	VMICS#
AD26	MD17	RCLK1	VMIHA0
AD24	MD16	DQM5	VMIHA1
C/BE3#	MD18	DQM7	VMIHA2
AD25	MD19	DQM4	VMIHA3
ADSTB_B	MD21	DQM6	VMIHA4
AD22	MD20	MD48	GPIO4
AD20	MD23	MD49	GPIO5
AD23	MD25	MD51	GPIO6
AD21	MD27	MD50	GPIO8
AD19	MD22	MD52	GPIO7
AD17	MD29	MD53	ROMA16
AD16	MD31	MD55	ROMA15
AD18	MD24	MD57	ROMA17
C/BE2#	MD26	MD54	ROMA14
FRAME#	MA1	MD56	ROMA12
TRDY#	MA3	MD59	ROMA7
STOP#	MD28	MD61	ROMA13

Table 6-3. NAND Chain Ordering

IRDY#	MD30	MD58	ROMA8
DEVSEL#	MA5	MD60	ROMA6
PAR	MA7	MD62	ROMA5
AD15	MA0	MD63	ROMA9
C/BE1#	MA2	VCLK	ROMA11
AD14	MA9	VREF	ROMA4
AD12	MA11	GPIO3 (DDCCLK)	ROMA3
AD13	MA4	GPIO2 (DDCDATA)	ROMA10
AD11	MA6	HREF	ROMA2
AD10	MWEA#	HSYNC	ROMA1
AD8	MWEB#	VSYNC	ROMD7
AD9	MA8	VRDY	ROMD6
C/BE0#	MA10		ROMA0
ADSTRB_A	SRASA#		ROMD0
AD7	SRASB#		ROMD5
AD5	SCASA#		ROMD4
AD6	SCASB#		ROMD1
AD4	TCLKA		ROMD2
AD3	TCLKB		ROMD3
AD1	OCLK		
AD2			
AD0			