

# Intel's i750 (R) Video Processor -- The Programmable Solution

Ryan Manepally  
Dave Sprague

Intel Princeton Operation  
313, Enterprise Drive,  
Plainsboro NJ 08536.

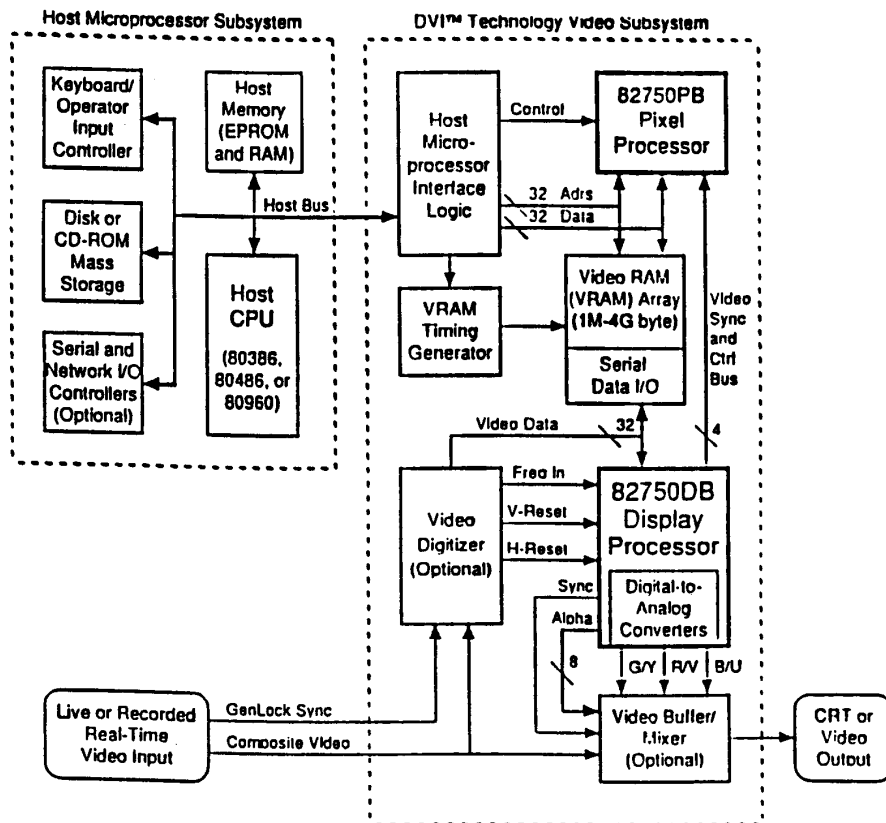
**Abstract:** On November 5 1990, Intel announced the first(\*) multimedia components for personal computers, workstations and standalone players based on DVI (TM) technology. The fully programmable i750 video processor integrates full screen full motion digital video, high quality audio, high speed graphics, text and still images in digital multimedia applications, by combining the speed of hardware and the versatility of software. The i750 video processor consists of two components: The 82750PB, pixel processor and the 82750DB, display processor. The devices are compatible with existing DVI motion video algorithms, and emerging international standards such as still image compression developed by the Joint Photographic Experts Group (JPEG).

This paper gives an overview of the two devices and their operation.

**Keywords:** Pixel Processor, Display Processor, JPEG, DVI Technology, Multimedia

**Introduction:** The i750 video processor system overview:

The figure below shows the basic architecture of a video subsystem incorporating DVI Technology. An image decompression example is used to describe how the DVI subsystem functions.



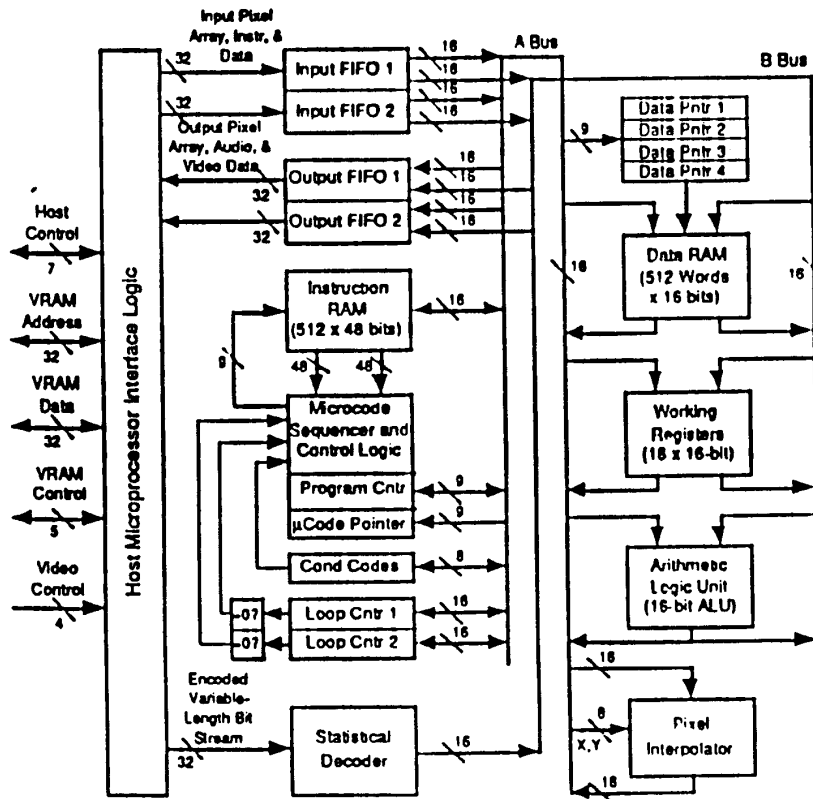
The relation of the 82750PB and 82750DB to the host is clearly shown. During a typical decompression, the host processor first copies compressed data coming from either a hard disk or a CD ROM, into one area of VRAM and tells the pixel processor where the data is located. The pixel processor retrieves the compressed data, decompresses it, performs other transformations as needed, and builds a decompressed image elsewhere in the same VRAM. Each pixel in the decompressed image is described by a luminance (Y) and two chrominance parameters (U and V). The Y, U and V images are stored as planar bitmaps in memory, which means that they are stored as three separate bitmaps in memory. This results in optimum read/write efficiency between the pixel processor and VRAM, and the fastest possible inner loop instruction execution time.

The display processor retrieves the partially expanded image data from VRAM, performs post-processing operations such as two dimensional UV interpolation, YUV to RGB conversion, digital to analog conversion, to complete image expansion and generates synchronization and control signals as required. The pixel processor continues decompressing subsequent images so that the playback can continue uninterrupted, resulting in motion video. To facilitate this, the pixel

processor decompresses each frame on average faster than the time required to display that frame. The host processor is responsible for initializing the DVI hardware, coordinating communication with the external peripherals, supervising the high level operator interface and retrieving different data sequences from disk based on operator requirements.

During capture or recording of live video, the reverse of this process occurs. Analog signals from a live video source are digitized by a video digitizer and stored in VRAM buffer. Once compressed by the pixel processor, the compressed data is transferred by the host to mass storage devices.

**The 82750PB Pixel Processor:** The 82750PB is a fully programmable video processor designed primarily for video applications. It contains more than 300,000 transistors and is fabricated using Intel's patented 1-micron CHMOS-IV (\*) technology. It comes in a 132 pin PQFP package, and initially operates up to 25Mhz. It can compress, decompress and integrate motion video with other multimedia applications in real time. A block diagram of the pixel processor CPU with the pixel interpolator and statistical decoder is shown below.



The pixel processor CPU is capable of performing several operations during each clock cycle. Microcoded software routines are downloaded into the internal instruction RAM and later executed. Surrounding the CPU are several dedicated hardware blocks akin to a hardwired processor. The difference is that the same hardware can execute a wide range of algorithms for data compression, decompression, and special effects like zooming, scaling and the like. Since the pixel processor builds each image on a pixel by pixel basis, it must compute color parameters for each pixel of every frame quickly enough so that the pixel can be displayed in real time. This means that the 82750PB must decompress, perform other transformations such as rotate, zoom and merge, and then compute color parameters for each pixel in less time than it would take each pixel to appear on the display. The complexity of this process poses a formidable challenge calling for a very specialized architecture. In many ways the 82750PB resembles a standard high performance microprocessor, with an on chip ALU, 16 working registers, 512-48 bit words of instruction RAM, 512-16 bit words of data RAM, and a 32 bit interface to external VRAM. The other discrete functional blocks as shown in the block diagram, help the ALU achieve the high performance requirements for video processing. These are the loop counters, FIFO memory pointers, pixel interpolator and statistical decoder. The processor can perform the equivalent of six conventional microprocessor instructions per clock cycle. This means that one would need a microprocessor running at 150MHz, to equal the performance of a 25MHz 82750PB.

The ALU implements one instruction per clock cycle. Operands can be incremented, decremented, summed, subtracted, masked, cleared, complemented and so forth. Several new instructions are customized for pixel calculations. For example, addition and subtraction can optionally "saturate", thus yielding the maximum or minimum representable value for each pixel without testing for overflow or underflow. The ALU can be split into two to add or subtract signed offsets to two 8 bit pixel values at once.

The CPU uses a 48-bit instruction format, with 12 opcode and operand fields. Operands can be taken from or results written to the registers, memory pointers and counters, on chip data memory, external memory, pixel interpolator or statistical decoder. Each instruction has four operand fields. This means that, during a single clock cycle, two operands can be retrieved and two stored. Another operand field specifies the address of the

next instruction. There is also a condition code field for conditional branches that specifies which instruction executes next. Since no special instructions are needed for branching, they effectively execute in zero clock cycles.

**FIFO Memory Interfaces:** Video processing applications must handle large data arrays, far larger than can fit on a conventional microprocessor core. While data caches and scratch pad memory can increase the data handling capacity of the microprocessor, some video processing applications can exceed the capacity of even a fairly large data cache. This calls for an efficient way to communicate between the pixel processor CPU and the large VRAM array. A steady stream of compressed data can be retrieved from the VRAM, at that time a steady stream of computed pixel values can be written to the VRAM with four (two input and two output) FIFO channels. Each FIFO automatically packs and unpacks data words, buffers data, and initiates external transfers to memory. Two channels provide continuous streams of data input, useful for loading microcode routines, compressed data or uncompressed images from the external array. The other two provide a path for external data which is useful for building a final image in memory suitable for display. The FIFOs are initialized by specifying the RAM address from which the data should be read or written, size of the data (8 or 16 bits) and whether the external data is stored in ascending or descending order. Input FIFOs automatically load pairs of 32-bit words and break them up into the requested width. Output FIFOs assemble words into double 32-bit word blocks to be written into VRAM. Each time a microcode routine requests a source operand from an input FIFO the next data element is driven onto the bus. Each FIFO is double buffered so that while previously fetched data is being read, the transfer of the next data is taking place. This ensures that data will always be available before it is needed. Data elements written to the output FIFOs by the ALU or the pixel interpolator are combined to form blocks of 32 bit words and are written to the next location in memory. Output FIFOs are also double buffered, so that microcode execution can proceed while write operations are taking place. FIFOs are interlocked, so that one cannot read from an empty FIFO or write to an already full FIFO. Attempts to do so will freeze the microcode routine until the requested operation can proceed.

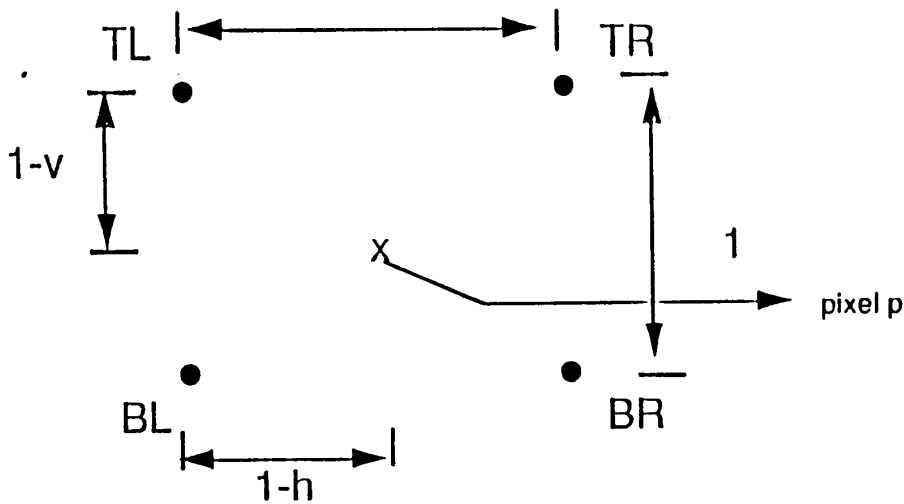
**Pixel Interpolator:** The pixel interpolator performs anti-aliasing functions. When a displayed object is the exact replica of the originally recorded image, it is said to have been faithfully reproduced. This means that no additional transformations are done while displaying the object and the decompression is relatively straightforward. If, however, additional transformations are required, such as enlarging, reducing, scrolling, rotating or intentionally distorting, then there is no longer a simple relationship between the stored pixel values and those that are displayed. Any adjustments to the image lead to artifacts which appear as poor video quality to the viewer. To remove the artifacts, there exist several software anti-aliasing algorithms that increase the display resolutions until these artifacts become imperceptible. These algorithms take up too many CPU clock cycles to perform transformations in real time. The pixel interpolator, fortunately, which is a dedicated hardware block on the 82750PB, excels in this sort of low level, well defined function.

Below is an example of how the pixel processor performs pixel interpolation. The case used here is of an enlargement transformation used on a stored image.

When a stored image is enlarged, the space between each of its stored pixels exceeds the space between those to be displayed. This means that the displayed pixel falls somewhere between two rows and columns of pixels from the original image. In principle, the color value of each displayed pixel can be derived by averaging the

color values of the four pixels surrounding it, weighted according to the physical displacement of the pixel from each of its surrounding neighbors. If  $p$  is a pixel whose value is to be derived from its neighbors, TR, TL, BR, and BL, then the weighting for  $p$  would be as shown above.

**Statistical decoder:** In video processing, as in any other type of computation, some values occur more frequently than others. In such cases, a statistical encoding technique is used to reduce storage space. The principle behind this technique is to use fewer bits to represent more frequently occurring values. A common type of statistical encoding technique is called Huffman encoding. Data is stored as a long series of variable length bit strings. Each substring encodes its own length and value, without consideration to the natural boundaries between bytes and words. During a typical decode operation a conventional computer must first find a string of data, examine the initial bit pattern to find out the length of the word, and then expand the word back to its full size, requiring many clock cycles for computation. The statistical decoder performs Huffman like computations automatically for a variety of codes and conventions. Software initially specifies the starting address of an encoded bit stream and its encoded conventions. Thereafter programs can retrieve an indefinite series of expanded substrings by designating the statistical decoder as a source operand for other operations.



If TR, TL, BR and BL denote Top Right, Top Left, Bottom Right and Bottom Left, the four pixels that surround pixel  $p$ ,  $h$  is the horizontal displacement from TR and  $v$  is the vertical displacement from BR, then the weighting ( $W$ ) of  $p$  with respect to the four surrounding pixels is given by the following equation:

$$W = TL(1-h)(1-v) + TR(h)(1-v) + BR(h)(v) + BL(1-h)(v)$$

**The 82750DB Display Processor:** As the name suggests the 82750DB display processor concerns itself with the display aspects of an image. It is a 132 pin PQFP chip also fabricated using the 1 micron CHMOS-IV process. Below is a block diagram of the display processor. It initially operates at 28MHz, to deliver upto VGA standard resolutions. The chip also handles NTSC and PAL. It retrieves decompressed images by sending a service request to the pixel processor to download display information through the serial register port of VRAM. It then reads sequential 32-bit data into the pixel processing and VU interpolator units within it. The data is processed as four 8-bit values, two 16-bit values or a single 32-bit value. Often the display data is stored in two parts, with complete luminance information for each pixel and a 4:1 subsampled set of chrominance information. In this mode, the display processor computes chroma information for the intermediate pixels. Pixel values are sent to color look up tables (CLUT), which transform the 8-bit indices into three arbitrary 8 bit values. For graphics, 256 colors can be selected from a possible 16.8 million colors at any given time. The CLUT can be bypassed to output the pixel value directly.

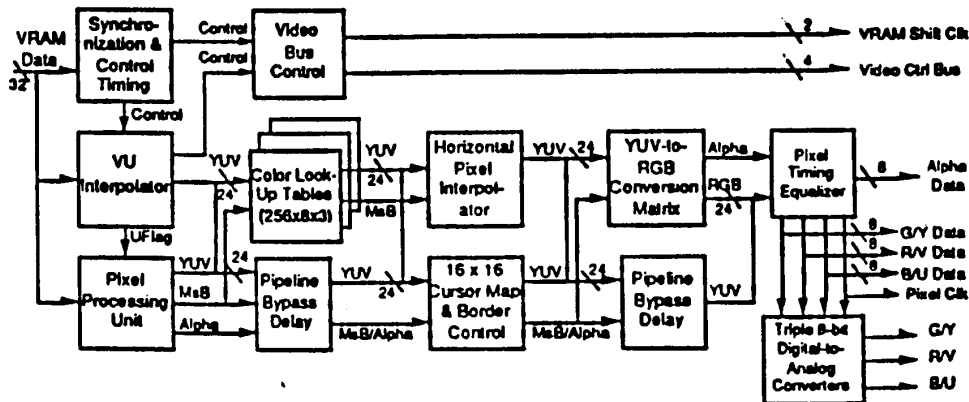
The cursor block can be programmed with a bitmap for an arbitrary 16 X 16 pattern and can be automatically superimposed over any large image, thus reducing computational overhead. There is a YVU to RGB matrix converter which can be bypassed. Also on board are triple 8 bit DAC's. These provide analog information outputs if required. Writeable configuration parameters allow the user to program operating mode, screen size, pixel resolution, etc. Each of the units within are heavily pipelined, enabling data to pass through at a uniform fast rate. Bypass paths around each unit produce the same internal pipeline delay, so enabling and disabling a function will not disrupt the

system timing.

**Synchronization and timing unit:** The display processor produces color and timing control signals for the CRT. Signals are supervised by the synchronization and timing control unit, which also generates blanking and retrace controls for the CRT. This unit uses horizontal and vertical reset inputs to genlock the 82750DB to an external video sync signal. Characteristics like CRT resolution, refresh rate, interlace mode, size and position of retrace and blanking, and number of bits per pixel are all programmable.

**Cost Comparisons:** When it comes to video subsystem costs, the i750 video processor provides a very economical solution. Below is a cost breakup (10,000 quantity) of a system using the i750 video processors.

Feature	Cost
Motion Video	]
JPEG Still Image	]
Graphics/Video Special Effects	] \$85
Display Control	]
DAC's	]
Glue Logic	\$20
Estimated Cost	\$105



**Conclusion:** The philosophy used to design the i750 video processors was to keep the architecture programmable. Programmability is a key element in the next several years since algorithms with which systems must be compatible will change frequently. Programmability is also required to cost efficiently support all elements of multimedia i.e., video special effects, fast graphics and text as well as video compression and decompression. Lastly a programmable solution allows OEM's to differentiate their products. The difference between hardwired processors and programmable solutions is similar to the difference

between a typewriter and a personal computer. Hardwired approaches may be suitable for disk drive controllers or LAN adapters, where designs are relatively well understood, but are particularly inappropriate for multimedia systems, where changes to the silicon have to be made to maintain compatibility with emerging standards. The i750 architecture in the 82750PB/DB combines the best of both worlds, combining the speed of hardware and the versatility of software, making it **the only programmable processor available today for the most demanding multimedia needs.**

**CHMOS-IV** is a a patented process of Intel Corp.  
**DVI** is a trademark of Intel Corp.  
**i750** is a registered trademark of Intel Corp.

\*An earlier generation of DVI components are available on DVI system and board level products. The 82750PB and 8275DB are the first component level DVI products.