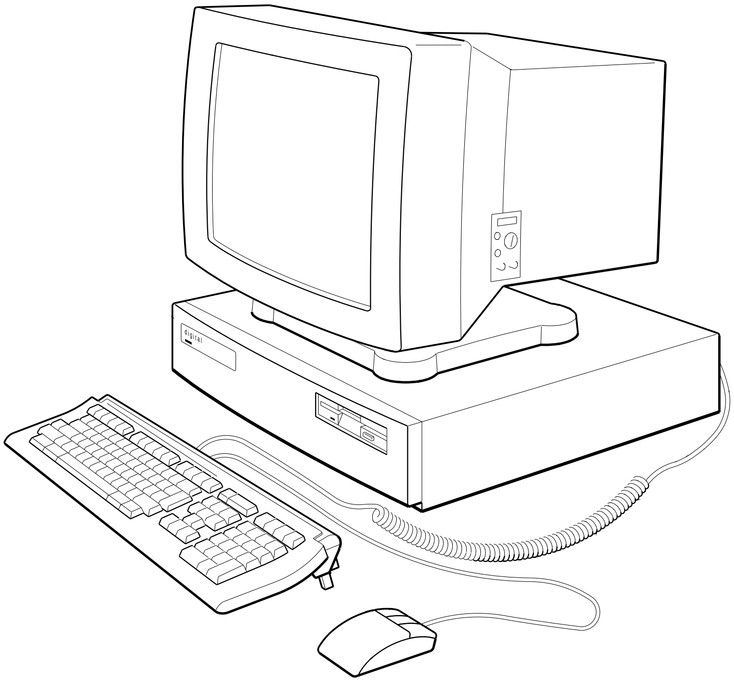
Personal DECstation Series Technical Overview

- **Personal DECstation 5000/20**
- **Personal DECstation 5000/25**



Worksystems Base Product Marketing
Digital Equipment Corporation
Palo Alto, California

Version 1.0
December, 1991

Digital believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. Digital is not responsible for any errors in the information given in this publication.

The software described in this document is furnished under license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

The following are trademarks of Digital Equipment Corporation: DEC, DECstation, DEC Ada, DEC FUSE, DEC GKS, DEC PHIGS, DEC VUIT, DECnet, DECwindows, DECsystem, the DIGITAL Logo, PATHWORKS, PixelStamp, TURBOchannel, ULTRIX, VAX, VAXstation, VMS, and XUI.

PostScript is a registered trademark of Adobe Systems, Inc. AT&T and System V are registered trademarks of American Telephone and Telegraph Company. Apple and Macintosh are registered trademarks of Apple Computer, Inc. Network Computing System is a trademark of Hewlett-Packard Company. Intel is a trademark of Intel Corporation. PC/AT is a trademark and IBM and OS/2 are registered trademarks of International Business Machines Corporation. Lotus is a registered trademark of Lotus Development Corporation. MIPS is a registered trademark of MIPS Computer Systems, Inc. MS-DOS is a registered trademark of Microsoft Corporation. NFS is a trademark of Sun Microsystems, Inc. Open Software Foundation is a trademark and OSF, OSF/1, OSF/Motif, and Motif are registered trademarks of Open Software Foundation, Inc. Open Desktop and SCO are registered trademarks of The Santa Cruz Operation, Inc. UNIX is a registered trademark of UNIX System Laboratories, Inc. X/Open is a trademark of X/Open Company, Ltd.

Printed in U.S.A.

Table of Contents

List of Figures

List of Tables

1 Introduction

The Personal DECstation 5000 Series provides a low-cost, functionally self-contained entry into Digital's RISC-based desktop workstation family. Because of its pricing and available application software, the Personal DECstation 5000 Series is well suited for office, business, and publishing applications that have traditionally been the domain of personal computers. But because it has the same RISC processor architecture, runs the same ULTRIX operating system, and is binary compatible with the rest of the DECstation line, the Personal DECstation 5000 Series can also run the same scientific, engineering, software development, and data processing applications when the highest CPU performance is not required. Even though it represents the low end of the DECstation 5000 line, the Personal DECstation still supplies a level of computing power that is impressive by previous workstation and minicomputer standards.

The Personal DECstation[1] comes in two models, the Personal DECstation 5000/20 and the Personal DECstation 5000/25. They differ only in the speed of the CPU subsystem, and provide an upgrade path by replacement of a CPU daughter card (see Section 2.2.3).

Very high levels of integration, both at the circuit level and the board level, contribute to the Personal DECstation's favorable price/performance, simplicity of configuration, and high reliability. The system module includes a color raster graphics system and an I/O subsystem based on a new I/O ASIC (Application Specific Integrated Circuit) that drives integrated controllers for a wide range of peripheral device types and communications interfaces.

The advanced TURBOchannel open interconnect provides communication between the CPU subsystem and the system memory, the on-board graphics, and the I/O subsystem. Through two TURBOchannel expansion slots, the Personal DECstation user has access to options, available from Digital or third parties, for accelerated graphics, multimedia, fiber-optic networking, and interfaces to other industry-standard buses. From its inception, the DECstation 5000 family has been committed to open industry standards for software and hardware interfaces. The newest industry standards relevant to RISC workstation technology are embodied in the ARC (Advanced RISC Computing) specification of the ACE (Advanced Computing Environment) initiative, now comprising over 200 companies. The ARC specification is based in large measure on DECstation 5000 technology, including the MIPS® R3000A processor architecture, the TURBOchannel I/O interconnect, and the ULTRIX operating system. While the Personal DECstation is not completely ARC compliant, it is indeed ARC compatible. Software that now runs on the Personal DECstation will be binary compatible with future ARC platforms.

[1] The term Personal DECstation will be used throughout this overview to refer to both the Personal DECstation 5000/20 and the DECstation 5000/25, unless otherwise noted.

Along with the Personal DECstation, Digital is introducing several new TURBOchannel-based graphics options and multimedia options. These new options are not specific to the Personal DECstation models, but can be used with the other models of the DECstation 5000 family as well. Some of these new graphics options are based on the architectures of previous graphics options, but use faster integrated circuits. Two of the new graphics options implement entirely new architectures, one intended particularly to support multimedia options.

This overview presents a technical description of the Personal DECstation base system, focusing on its advanced technical features. This description briefly reviews the fundamental benefits of the RISC philosophy and shows why the MIPS R3000A architecture is an especially effective implementation of that philosophy. The overview continues with a discussion of the system software, development software, and bundled applications.

Finally, this overview describes the numerous graphics and multimedia options that may be used with the Personal DECstation, as well as with other workstations in the DECstation family.

2 System Description

The Personal DECstation is a desktop workstation comprising a system unit in a low-profile desktop enclosure, external keyboard, mouse, and monitor. The system unit contains the base system module, including system memory, I/O subsystem, 8-plane color frame buffer, two TURBOchannel option slots, and a connector for the CPU subsystem daughter card, which runs on a different clock frequency from the system module. Two options are available for the CPU: the Model 20 CPU subsystem runs at 20 MHz, the Model 25 at 25 MHz. The system module clock, for TURBOchannel, system memory, and I/O subsystem, runs at 12.5 MHz. Section 2.2 provides more detail on the CPU subsystem.

The system unit contains space for one internal SCSI hard disk and one floppy disk. The I/O subsystem, presented in more detail in Section 2.3.3, supports a wide range of peripherals. Figure 2.1 is a block diagram of the Personal DECstation system.

2.1 Enclosure and Environmental Requirements

The exterior dimensions of the system enclosure are 16x17x3.6 inches. All I/O connectors, power switch, and line cord connector are located on the enclosure’s rear bulkhead.

The power supply is dual voltage capable, so the system will run on either 110 or 220 volt line voltage, and at a line frequency from 47 to 63 Hz.

The Personal DECstation can be used in any ordinary office environment. The operating limits for ambient temperature are 10^{o} to 40^{o} C (50^{o} to 104^{o} F).

2.2 CPU Subsystem

All DECstation 5000 models have a CPU based on the MIPS R3000A RISC architecture. This section begins with a review of the essential points of the RISC philosophy and its realization in the R3000A architecture. Then, Section 2.2.3 presents relevant details of the Model 20 and Model 25 CPU subsystems.

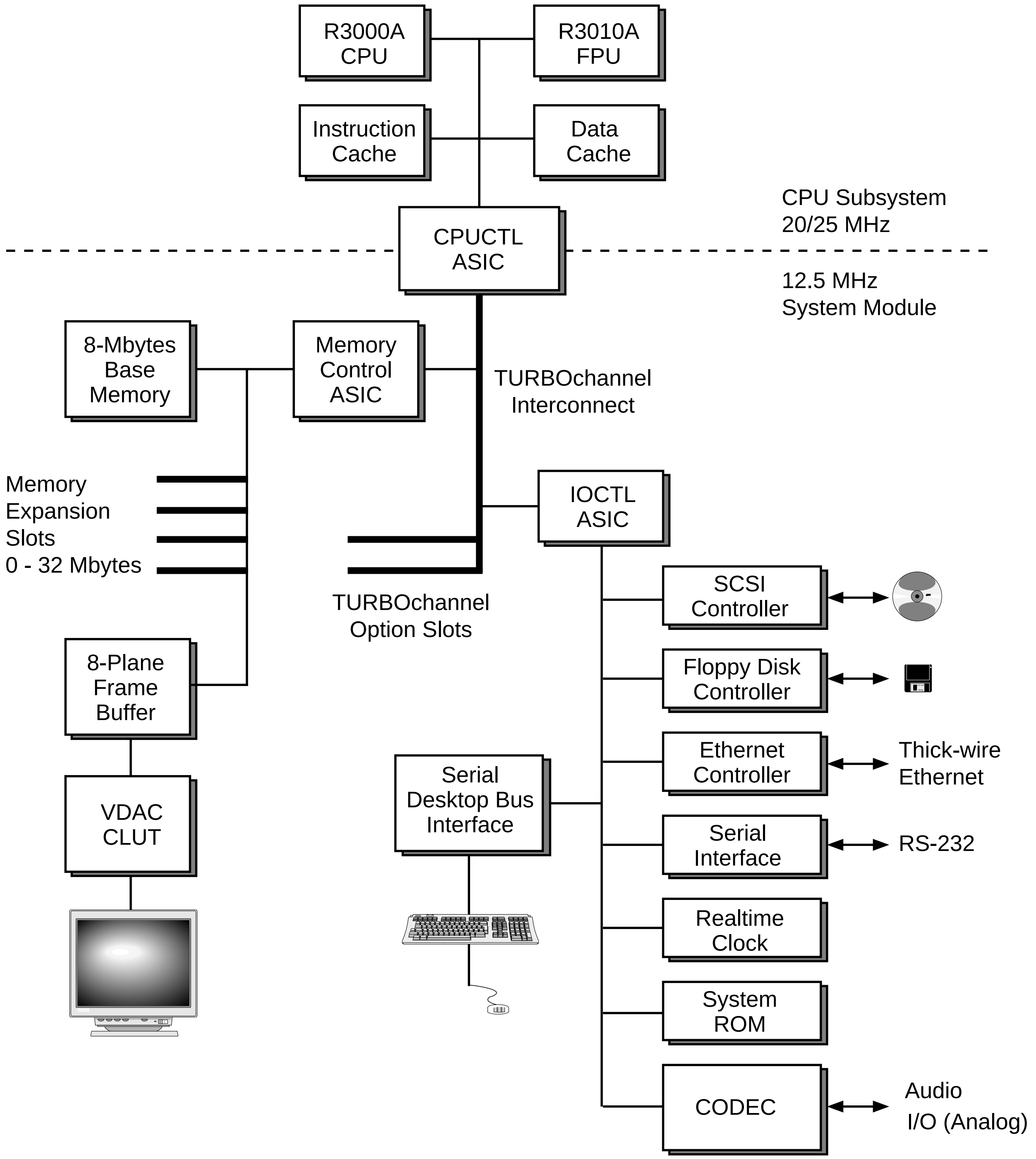


Figure 2.1 Personal DECstation 5000/20 and 5000/25 System Block Diagram

2.2.1 The RISC Philosophy

The approach to computer design called RISC, Reduced Instruction Set Computer, has gained increasing currency through the 1980s. RISC technology now appears to dominate the workstation domain and is making inroads into the mainframe market as well. The term CISC, Complex Instruction Set Computer, is now sometimes used to distinguish the older ideas of processor design from RISC. Some of the essential characteristics that distinguish RISC from CISC are:

- smaller instruction sets
- less complex instructions
- no microcoded instructions
- few instruction formats
- uniform instruction byte length
- uniform instruction cycle count
- load/store architecture
- pipelined instruction execution

The first four of these characteristics have the obvious benefit of speeding up the computer design cycle, leading to lower computer cost, to quicker implementation of advances in the underlying circuit technology, and so to better performance in any technological era. The first two features may seem liabilities relative to functionality, but this is mitigated if the omitted complex instructions are rarely used in practice. In fact, most complex instructions that had been added to instruction sets for special purposes were rarely used in compiler-generated code.

Practical performance questions concern almost exclusively the use of high-level languages. Therefore, performance can be enhanced if the processor's instructions are designed to perform functions that are known from studies of compiler-generated code to be frequently required and that are easy and straight forward for compilers to generate. Rarely used complex instructions can be replaced by sequences of simple instructions without significantly harming overall performance.

The meaning of the seventh feature, load/store architecture, is that the processor should contain an ample number of registers and that all arithmetic and logical operands and results should be in registers. The only memory references are load and store instructions, which reduces the number of memory references and makes caching more effective.

The fourth through seventh features enable the eighth, pipelined instruction architecture. It is this feature that yields the greatest performance benefit of the RISC philosophy. In a pipelined instruction architecture, the execution of successive instructions overlap in time. To say the pipeline has depth *d* means that every instruction executes in the same number *d* of clock cycles and there are up to *d* instructions in the pipeline at any time. The instructions in the pipeline are at different stages of execution, each one stage behind the preceding one and advancing one stage per clock cycle. The result of pipelined instruction execution is a peak instruction execution rate of one instruction per clock cycle.

Of course, it is not always possible to keep the instruction execution pipeline full. For example, when memory references are needed, there will be a delay before the referenced item is available for use in an operation. And branch instructions introduce uncertainty in the order in which instructions should be started in the pipeline, not completely resolved until the branch instruction is executed. However, it is usually possible to fill the pipeline slots caused by branch delays or load/store delays by judiciously reordering instructions while respecting the data-dependency constraints. Keeping the instruction pipeline as full as possible is a job for an intelligent compiler. Thus, one element of the RISC design philosophy is to shift complexity from the processor architecture to optimizing compiler design. The success of RISC design rests in part on the advanced understanding of compiler optimization attained over the past few decades.

2.2.2 The MIPS R3000A RISC Architecture

The MIPS R3000A architecture is a relatively mature implementation of RISC philosophy. The MIPS design was built on knowledge of earlier RISC efforts, especially the Stanford University RISC research work. MIPS started with the optimizing compilers from the Stanford effort, and studied the performance effects of variations in the instruction set through simulations of a range of applications compiled with the existing optimizers. The result of this approach is an instruction set that is very well tuned for high-level language use.

The basic R3000A CPU contains 32 general-purpose 32-bit registers and ALU, Shifter, Multiplier/Divider, Address Adder, and Program Counter Increment/Mux. These registers are used for integer data and address arithmetic. In addition, the R3000A includes virtual memory management functions on the chip, using a translation lookaside buffer (TLB) and associated registers to provided very fast virtual-to-physical address translation.

Addresses have 32 bits, providing a 4-GB virtual address space. Only the first 2 GB of virtual memory are accessible to user processes; access to the higher 2 GB of the address space is reserved to the privileged kernel mode of the processor.

The kernel-reserved space is divided into three segments. One 512-MB virtual segment is not mapped through the TLB, but is always mapped directly to the first 512 MB of physical memory and is always cached. It is typically used for kernel executable code and some kernel data.

Another 512-MB virtual segment is also directly mapped to the first 512 MB of physical memory but is never cached. The operating system typically uses it for I/O registers, ROM code, and DMA buffers.

The remaining 1-GB segment of the kernel-reserved virtual memory space is mapped through the TLB and may be cached or not cached selectively on a per-page basis according to a bit in the TLB entry. The operating system typically uses this segment for stacks, user-page tables, and other per-process data that must be remapped on context switches.

The 2 GB of virtual address space available in user mode is always mapped through the TLB, but also may be cached or not selectively on a per-page basis.

The TLB is a fully associative memory with 64 entries, each mapping a 4-KB memory page. Each TLB entry includes the high-order 20 bits of the address of a virtual page, the high-order 20 bits of the address of the corresponding physical page, a 6-bit process identifier, and other bits used in virtual memory management, including the bit that controls whether references to the page are to be cached. A virtual address page number resulting from a processor address computation is immediately translated to the corresponding physical page number by the associative memory, provided that the page is in the TLB. Otherwise a TLB-miss exception occurs, causing software to load the TLB from memory-resident page tables, perhaps after fetching the page from disk if necessary.

The process identifier in the TLB entry allows up to 64 simultaneous processes to use the same virtual address space. This feature enhances the performance of the processor in multitasking environments such as the ULTRIX operating system.

The R3000A has a coprocessor, the R3010 floating point accelerator (FPA), which performs single- and double-precision IEEE-standard floating point operations, using 16 64-bit floating point registers. The CPU, the memory management functions, and the floating point accelerator share local control logic, bus control logic, and a five-deep instruction pipeline. While the R3000A may be configured at boot time to operate with either byte order convention for memory words, the DECstations and the ACE/ARC specifications are exclusively little endian. That is, words are stored in memory with the least significant byte at the lowest address.

So that performance will not be limited by the bandwidth or access latency of the system memory, the R3000A relies on a fast cache memory. The R3000A architecture uses separate cache spaces for instructions and data. This division of cache has several advantages. The first advantage concerns locality of reference, that is, the probability that a series of successive memory references will all lie in a small region of the address space. Cache effectiveness depends essentially on locality of reference, and locality will clearly be greater for a sequence of instruction references only, or for a sequence of data references only, than for a mixture of instruction references and data references. A second advantage of the cache division is that it enables an architecture in which the processor can fetch one instruction and load one data word both in a single cycle, effectively doubling the maximum cache bandwidth. Further details on the Personal DECstation cache architecture and operation are given in Section 2.2.3.

2.2.3 The Personal DECstation 5000 CPU Subsystem

The Personal DECstation CPU subsystem is located on a daughter card measuring approximately 3x5 inches, which plugs into the system module. It comprises the R3000A CPU, the R3010 floating point accelerator, 64 KB each of instruction and data cache, and the CPUCTL custom ASIC, which provides the interface and buffering between the 20- or 25-MHz time domain of the CPU and the 12.5-MHz time domain of the system module. The isolation of the CPU time domain on the daughter card provides certain design and manufacturing advantages as well as the possibility of upgrading from the Model 20 to the Model 25, and possibly more powerful CPU subsystems in the future, including 64-bit R4000 processors, by replacing/swapping the daughter card.

The CPU subsystem contains 64 KB each of instruction cache and data cache. The caches are direct-mapped, write-through caches, each containing 16K word entries. A cache word entry contains 32 bits of instruction or data, 13 tag bits, a valid flag bit, and byte-parity bits. The tag bits hold the high-order part of the physical address in system memory of the cached word. The low-order bits of the system memory address of the cached word are the same as its address in the cache; they form the cache index. The dual cache is implemented in fast SRAM. The R3000A can fetch one instruction and load one data word in each cycle.

On a write to a cached location, the data and the 13 high-order address bits are stored in the data and tag bits of the data cache location indexed by the 16 low-order address bits. (Note that this potentially changes the cache tag; that is, it may overwrite a cache of a different physical memory location with the same cache index.) At the same time, the address and data are captured into a 6-deep FIFO in the CPUCTL ASIC, which writes from this FIFO to system memory. When the write FIFO contains five address/data pairs, the CPUCTL ASIC raises a *write busy* signal, which stalls the processor until there is more room in the write FIFO.

For a load or instruction fetch from a non-cached address, the processor puts out the address, together with a *read memory* signal, which causes the CPUCTL ASIC to perform a single-word read from the memory subsystem and pass the data back to the processor bus.

For a load or an instruction fetch from a cached address, the processor puts the cache index onto the bus and the cache responds by putting the data and tag contained in the indexed cache entry onto the bus. If the cache tag matches the tag that the processor seeks, then the load or fetch has been completed in one cycle. Otherwise, it is a case of cache miss, and in the next cycle the processor puts out the complete desired address on the tag and index lines, along with the *read memory* signal that causes the CPUCTL ASIC to initiate a cache load. The cache load involves a four-word block read of system memory to a read buffer on the CPUCTL, followed by transfer to the appropriate cache at one word per CPU cycle.

The CPUCTL ASIC also serves as TURBOchannel controller, providing all signals and cycle types necessary to perform TURBOchannel transactions.

2.3 System Module

The base system module includes on-board system memory and connectors for the expansion memory, the color frame buffer, the TURBOchannel interconnect with two option slots, and the I/O subsystem. It runs synchronously on a 12.5-MHz clock.

2.3.1 System Memory

The physical address space is 512 MB. The first 40 MB is for the system memory, and the higher addresses are either reserved or used for the on-board color frame buffer (1 MB), the TURBOchannel slots (64 MB each slot), system boot ROM, DMA buffers for both the SCSI and floppy disks, and various control registers of the rich I/O subsystem. A custom memory control ASIC provides memory timing and system arbitration. The memory access priority is fixed in the following order:

- DRAM refresh/Video refresh
- I/O subsystem DMA
- TURBOchannel slot 0 DMA
- CPU interface
- TURBOchannel slot 1 DMA

Memory is organized in 32-bit words with byte parity. The base board comes with 8 MB of DRAM and accommodates memory expansion via four connectors for Single In-line Memory Modules (SIMM). SIMMs must be added in pairs. Each SIMM contains either 2 MB or 8 MB, depending on whether it is populated by 1-Mbit or 4-Mbit DRAMs. The bank stride is determined by a single register on the system module, so it is not possible to mix 2-MB and 8-MB SIMMs, and the possible memory sizes are 8, 12, 16, 24, and 40 MB.

DRAM refresh occupies the memory system for 5 cycles of every 193 cycles. The single-word read latency is 5 cycles, the write latency is 3 cycles. The memory bandwidth for 64-word DMA transactions is 47 MB/s.

2.3.2 Graphics Frame Buffer

The base system module contains a 1024x768, 8-plane graphics frame buffer. It is organized as 256Kx32-bit VRAM, with each 32-bit word containing four 8-bit pixels. An Inmos IMS G332 Video DAC is used to scan the frame buffer and refresh the display. The VDAC contains a 256x24-bit color lookup table, which provides a palette of 16.7 million colors. The refresh frequency is 72 Hz.

The internal video frame buffer may be used with the VRM17, 17" 72-Hz monochrome monitor or the VRC16, 16" 72-Hz color monitor. See Section 4.1 for further discussion of the operation of raster graphics subsystems and Section 4.6 for graphics performance information on the base graphics system and graphics options.

2.3.3 I/O Subsystem

The system module contains integrated interfaces for:

- SCSI devices (Small Computer System Interface) including hard disk, tapes, and CD-ROMs
- Floppy disk
- Thick-wire Ethernet
- Serial ports (RS-232)
- Serial Desktop Bus (for keyboards and mouse)
- Analog audio

Each peripheral interface consists of a standard controller integrated circuit. The hub of this system is an IOCTL ASIC that is customized to communicate with this particular set of peripheral controllers. On one side, the IOCTL ASIC communicates with the several peripheral controllers via a 16-bit-wide data bus. On the other side, it communicates with the CPU subsystem and system memory via the TURBOchannel. The IOCTL ASIC provides DMA access for all the supported peripherals.

2.4 Peripherals

2.4.1 SCSI Devices

The SCSI interface conforms to the ANSI SCSI-2 standard, and so accommodates a wide range of devices available from Digital and third parties. It can perform synchronous or asynchronous data transfers at up to 5 MB/s. Up to seven SCSI devices can be connected on a daisy-chain cable up to six meters in length. Additional SCSI devices could be connected via a TURBOchannel SCSI adapter. Supported Digital SCSI devices include:

RZ23	104 MB	3.5-in hard disk
RZ23L	121 MB	3.5-in hard disk
RZ24	209 MB	3.5-in hard disk
RZ25	426 MB	3.5-in hard disk
RZ55	332 MB	5.25-in hard disk
RZ56	665 MB	5.25-in hard disk
RZ57	1.0 GB	5.25-in hard disk
RZ58	1.38 GB	5.25-in hard disk
RRD40	600 MB	CD-ROM disk
RRD42	600 MB	CD-ROM disk
TK50Z	95 MB	tape cartridge
TSZ05	1600 BPI	9-track tape
TZ30	95 MB	tape cartridge
TZK10	525 MB	QIC tape
TLZ04	1.2 GB	RDAT

•	TKZ08	2.2 GB	8mm tape
•	RX23	1.4 MB	3.5-in floppy disk
•	RX33	1.2 MB	5.25-in floppy disk

The system enclosure can accommodate an internal floppy disk drive and one internal 3.5" SCSI hard disk drive. As many as three SZ12 Dual SCSI Expansion Boxes may be connected to the base system SCSI adapter, each containing two devices. The internal SCSI drive can be one of the following:

•	RZ23L	121 MB	3.5-in hard disk
•	RZ24	209 MB	3.5-in hard disk
•	RZ25	426 MB	3.5-in hard disk

2.4.2 Internal Floppy Drive

The integrated floppy disk controller supports an internal RX26, extra-density, 3.5" double-sided, 135-track-per-inch, micro-floppy disk drive. The floppy drive can format, read, and write industry-compatible, extra-density floppy cartridge media with a formatted capacity of 2.88 MB. It is also compatible with standard 1.44 MB (HD) and 720 KB (DD) media. The RX26 connects to the floppy controller on the base system module via a 34-conductor flat cable.

2.4.3 Serial Desktop Bus

The desktop bus is a four-wire (data, clock, power, and ground) serial bus that runs at 100 KHz. The serial desktop bus hardware and protocol are based on the Philips/Signetics Inter-Integrated Circuit (I^{2}C) technology. The hardware conforms to the specification for the *ACCESS.bus* open desktop interconnect developed by Digital and adopted by the ACE initiative (see the ACCESS.bus Technical Overview). The Personal DECstation is the first workstation to incorporate this hardware. However, the software to support the full set of open bus features of ACCESS.bus will not be available when the Personal DECstations are released.

The serial bus on the Personal DECstation provides the means of connecting only the LK501 keyboard and VSXXX-BB mouse. This keyboard and mouse are special in that they have the ACCESS.bus interface, rather than the RS-232 interface, which is conventional for such devices. The keyboard connects to a 4-wire jack on the system unit back panel, and the mouse connects to a jack on the keyboard.

2.4.4 Serial Port

The serial port is provided for attaching a modem or other RS-232 peripheral. It supports baud rates between 50 baud and 19.2 Kbaud and the full set of modem control signals.

2.4.5 Audio I/O

The Personal DECstation includes a voice-grade audio I/O capability, provided by an encoder/decoder (codec) circuit in the I/O subsystem. The line-level analog audio input and output signals of the codec are connected to a modular jack on the rear panel; the output signal also runs to a small speaker within the system enclosure (located in the power supply).

The codec samples incoming audio at 8 KHz and quantizes the sampled analog values to 8-bit digital values using mu-law compression. This 64-Kbit/s data stream can then be processed and stored on magnetic media. Conversely, the codec can produce a pulse-code-modulated analog output signal from voice-grade digital audio data stored on magnetic media or CD-ROM.

These low-level audio capabilities are supported by the XMedia Tools Version 1.0 software, discussed Section 5.2. This kit provides means for application software to make use of the audio I/O capability, and includes some sample audio applications as well.

2.5 TURBOchannel I/O Interconnect

TURBOchannel is a high-performance, open I/O interconnect for desktop computers and servers, designed by Digital, and adopted by the ACE initiative as part of the ARC specification. TURBOchannel’s outstanding performance is the consequence of its innovative architecture, which is specialized and optimized for the I/O function. The TURBOchannel architecture departs from the pure bus topology that has previously been the norm for small computer interconnections. Instead, the TURBOchannel control signals have a radial point-to-point topology; a TURBOchannel-based system provides separate control lines for each of several peripheral slots. The benefits of this architectural innovation are:

- simple, efficient protocol
- low signal count per option module
- low-latency transactions
- high-bandwidth DMA block transfer
- simplified option module design

The TURBOchannel is a synchronous channel, transferring one 32-bit word of data or protocol overhead in each cycle. The TURBOchannel protocol implements two kinds of transactions, DMA transactions and I/O transactions, each transferring data in either direction between the option and the system. In DMA transactions, the option transfers data directly between its buffers and system memory in blocks whose size is determined by the option, up to a maximum block size determined by the system (128 words in the Personal DECstation).

TURBOchannel I/O transactions transfer single words of data between CPU registers and option memory locations. A load/store instruction referencing TURBOchannel address space causes the TURBOchannel control hardware to generate the TURBOchannel I/O transaction signal sequences for the appropriate TURBOchannel slot. I/O transactions are typically used for controlling the peripheral device, but they may also be used for data transfer by low-cost options lacking DMA buffers and logic.

The TURBOchannel definition is scalable, allowing implementation dependent cost/performance tradeoffs. See the TURBOchannel Technical Overview for more detailed discussion of the TURBOchannel architecture.

2.5.1 Personal DECstation 5000 TURBOchannel Implementation

In the Personal DECstation, the TURBOchannel runs at 12.5 MHz and provides the main data path on the system module. The CPUCTL ASIC on the CPU subsystem functions as the source of the TURBOchannel control signals. The system memory controller ASIC arbitrates TURBOchannel DMA contention on a fixed-priority basis.

The system module provides two TURBOchannel option slots for system expansion. Each TURBOchannel slot is allocated 64 MB of the physical address space for option data buffers, control registers, and ROMs that contain option-specific information used to configure the system at boot time.

The TURBOchannel Extender (PMTCE) provides a method of attaching a triple-width TURBOchannel option to the Personal DECstation or of attaching a double-width option without covering more than one internal TURBOchannel option connector. The PMTCE enclosure can also accommodate up to three SCSI devices -- two 3.5" hard drives and one 5.25" half-height removable media drive.

In DMA transactions, the TURBOchannel option transfers data between its own buffers and system memory in blocks of up to 128 words, at one word per memory cycle, yielding a 50 MB/s peak absolute bandwidth. The overheads that the simple TURBOchannel protocol imposes on the raw bandwidth are relatively small. Using the maximum 128-word DMA block size, and taking memory latencies and software overheads into account, the Personal DECstation can sustain average TURBOchannel DMA transfer rates approaching 40 MB/s.

The protocol-limited peak transfer rates for TURBOchannel I/O transactions are 16.5 MB/s for I/O writes and 12.5 MB/s for I/O reads. Taking memory latency and software overheads into account, TURBOchannel I/O transactions can move data between system memory and option buffers at sustained average rates around 5 MB/s.

2.5.2 TURBOchannel Options

Digital and third parties offer a growing family of TURBOchannel option boards. Available from Digital, in addition to the graphics and multimedia options described elsewhere in this overview are:

- SCSI controller
- Ethernet interface
- Fiber Distributed Data Interface (FDDI) controller
- VMEbus adapter

The growing set of TURBOchannel products available from third parties includes options for X.25 and other communications interfaces, RAID disk controller, film recorder, very high-resolution raster display, array processor, WORM drive controller, and others.

Now that TURBOchannel has been adopted by the ACE initiative, the universe of TURBOchannel devices can be expected to grow rapidly. Contact the Digital TRI/ADD Program for an up-to-date list of available TURBOchannel-based products.

2.6 Computational Performance

A number of different factors contribute to the system performance perceived by the user: processor clock speed, cache efficiency, memory bandwidth and latency, as well as software issues such as compiler optimization efficiency. To the user, the most relevant performance characterization is in terms of average instructions-per-second or operations-per-second as measured for a suite of benchmark programs that fairly represents the user’s intended workload, specified in the user’s preferred source language, and built with the supplied development software. The performance numbers given in this section result from industry-standard benchmark suites, using new compiler technology which will be available to DECstation users.

To facilitate the comparison of systems from diverse vendors, the System Performance Evaluation Cooperative (SPEC) has defined a suite of programs deemed typical of scientific and technical applications. There are separate programs for integer and for floating point performance characterization. The results are expressed as ratios of the performance to that of a VAX 11/780; the quoted numbers are geometric means of these ratios over the several programs in each suite.

The measured SPEC ratings for the Personal DECstations are:

	Model 20	Model 25
SPECmark	16.3	19.1
SPECint	13.5	15.7

The Dhrystone (Version 1.1) benchmark suite is somewhat older than the SPEC suite and tends to emphasize integer arithmetic. With this suite, the Model 20 rates at 21.6 MIPS and the Model 25 at 26.7 MIPS. Again, the reference machine is a VAX 11/780 which rates 1 MIP on this benchmark.

Another floating point benchmark, current as a standard test in supercomputer performance rating, is based on Linpack, a linear algebra package widely used in scientific applications. Floating point performance is characterized in terms of millions of floating point operations per second (MFLOPS), and both single-precision and double-precision floating point operations are of interest. The Linpack ratings for the Personal DECstations are:

	Model 20	Model 25
Single precision	5.3 MFLOPS	6.5 MFLOPS
Double precision	2.4 MFLOPS	2.8 MFLOPS

The graphics performance characteristics of the Personal DECstations with several of the available graphics options are given in Section 4.6.

3 System Software

The DECstation 5000 system software environment is based on open industry standards, and provides binary compatibility across the entire line. Adherence to open standards promotes portability of applications among systems from different vendors, thereby protecting the user's investment in software and training. Open standards also enable interoperability of systems from different vendors, which becomes more important as the connectivity of computing environments continually grows.

Digital is a leader in the development of new industry standards and integrates emerging standards into its software products. Digital develops system software according to the X/Open™, IEEE POSIX, and Open Software Foundation™ (OSF®) specifications. Digital is one of the founding members of the ACE initiative. The existing Digital system software provides a basis for a large part of the ARC specification under ACE. While the Personal DECstation does not fully implement the ARC specification, it is ARC compatible. Software that runs now on the Personal DECstation will be binary-compatible with future ARC-compliant platforms.

3.1 ULTRIX Operating System

The operating system for all DECstations is ULTRIX, Digital's enhanced native implementation of the industry-standard UNIX® operating system. ULTRIX is a general-purpose, multiuser, multitasking, interactive system conforming to IEEE standard 1003.1-1988 (POSIX) and to the Common Applications Environment as defined by the X/Open Portability Guide 3. The Personal DECstation requires ULTRIX Version 4.2A or higher.

ULTRIX includes the features of BSD (Berkeley Software Distribution) 4.3 and is source-level compatible with SVID (AT&T® UNIX System V® Interface Definition) Issue 2. But ULTRIX goes beyond these standards and offers many unique features, commands, and programming languages. For more information on ULTRIX functionality and conformance to industry standards, see the ULTRIX Handbook (EC-H0592-43).

The next major release of ULTRIX software will be based on the Open Software Foundation's OSF/1® operating system, which is also the basis for SCO® Open Desktop®, the UNIX specified for ARC/ACE. This system is specified to include an efficient virtual memory system, shared libraries, POSIX threads, a System V-compatible streams framework, dynamic subsystem loading, advanced security features, and comprehensive support for internationalization.

Of course, the ULTRIX operating system bundled with the Personal DECstation includes driver software for all the peripherals supported by the integrated I/O subsystem.

3.2 Software Development Environment and Compilers

A rich environment for software development has been a hallmark of UNIX since its inception. Throughout the evolution of UNIX, its editors, compilers, build utilities, and source-control systems have often represented the leading edge of software development technology. Digital's RISC/ULTRIX software development environment continues the tradition, and makes important new contributions to its further advancement.

3.2.1 Languages and Compiler Technology

Digital provides a full set of language compilers for the RISC/ULTRIX software development. Compilers developed by Digital include:

- DEC Ada
- DEC C
- DEC C++
- DEC FORTRAN
- DEC Trellis
- Pascal for RISC

The Digital-developed compilers conform to applicable language standards where they exist.

Compilers developed by third parties but supported by Digital as Digital Distributed Software include:

- pcc (MIPS C)
- MicroFocus COBOL/2
- Lucid Common LISP

For several of these languages and some other languages, including BASIC, alternate compilers for RISC/ULTRIX are available from third-party vendors.

Modern compilers are typically made up of two distinct parts: (1) a front end, which accepts the source code as input and parses the source language, but is independent of the target processor; and (2) a back end, which generates optimized machine code for a particular target processor, but is independent of the source language. The output of the front end, which is the input to the back end, is in an intermediate program specification language, which is independent of both the source language and the target machine architecture. A common intermediate language may be used for several different source languages and machine architectures. In this way, the benefits of optimizing the back end for a particular machine architecture can be enjoyed in several languages. Each language front end is used on several machine architectures, ensuring a common language definition and application portability. Figure 3.1 illustrates compiler architecture.

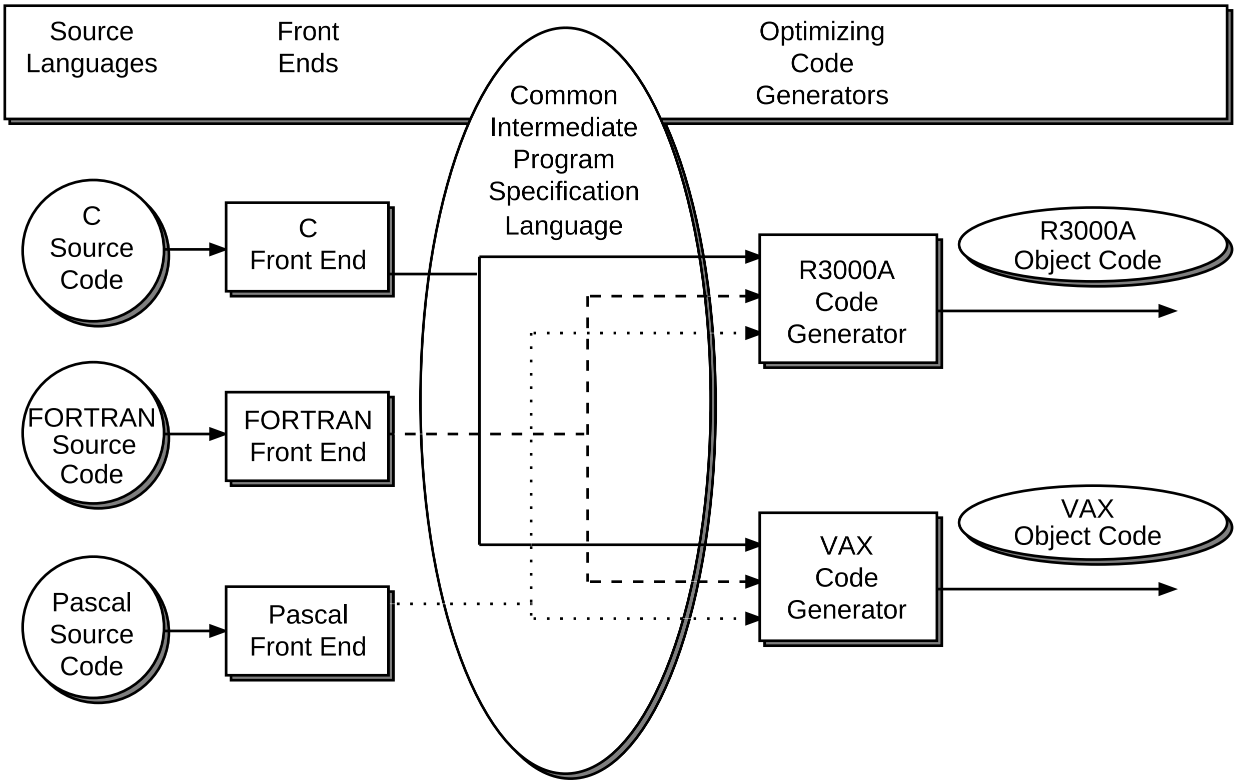


Figure 3.1 Compiler Architecture

Section 2.2.1 pointed out that compiler optimization is an essential element of the RISC philosophy, especially with regard to keeping the RISC instruction pipeline full. Optimizing generated code for the RISC processor is a function of the compiler back end, so it is natural that a number of compilers use the code generator that was developed at MIPS hand-in-hand with the design of the processor architecture. However, Digital has embarked on its own RISC compiler development program. The DEC FORTRAN and DEC Ada compilers for RISC use Digital's new proprietary R3000A code generator (and intermediate language), which generates substantially faster code, as much as 15% faster for some applications.

3.2.2 DEC C++ and Object-oriented Software Development

C++ is a programming language derived from the C language. It was developed at AT&T and has gained increasing favor in the last few years. C++ has a number of features that are improvements over C, but its most important distinguishing feature is its support of the *class* concept, which is central to the methodology of *object-oriented* programming. "Object oriented" refers to a complex set of ideas of software construction -- modularity, data abstraction, strong typing, encapsulation, messaging, information hiding, inheritance -- whose disciplined application can reduce programming errors and produce software that is more robust, understandable, extensible, and reusable.

DEC C++ implements the most advanced version of the C++ V3.0, as defined in the Annotated Reference Manual[1] written by the developers of the language. This version is the starting point for the ANSI standardization effort, in which Digital is an active participant.

DEC C++ is implemented as a native compiler, unlike the AT&T and other implementations that use pre-processors to translate C++ source to C code. It includes a C++ source-level debugger, which supports all the advanced features of the language and which can be used either with a *dbx*-like command line user interface or a with a graphical user interface within the DEC FUSE facility, described in the next section.

One of the most advantageous features of DEC C++ is its incremental linker, which processes only the modified modules on any link. Incremental linking can be three to seventy times faster than the conventional ULTRIX linker.

DEC C++ also comes with encapsulated reusable code in the form of class libraries, both the standard Class Libraries and additional class libraries developed by Digital. Digital has an ongoing program for extending the available class libraries for C++.

Digital offers an alternate object-oriented programming method in DEC Trellis. Trellis is a programming language, a development environment, and set of class libraries. One way in which the Trellis approach differs from that of C++ is that Trellis is a pure object-oriented language, while C++ is C with considerable object-oriented technology added to it. It is possible to use C++ without an object-oriented programming style, but the use of Trellis enforces an object-oriented discipline. DEC Trellis can exchange calls with DEC C++, so it is possible to have the best of both object-oriented worlds.

[1] *For more information on C++, the following books are recommended:*

The initial definitive specification of the language was given in the book by Bjarne Stroustroup, *The C++ Programming Language*, Addison Wesley, 1986.

The definition of Version 3.0, which is the basis for the standardization effort is in the book by Bjarne Stroustroup and Margaret Ellis, *The Annotated C++ Reference Manual*, Addison Wesley, 1991.

For an excellent discussion of the theory of object-oriented programming, independent of the C++ language, see the book by Bertrand Meyer, *Object-Oriented Software Construction*, Prentice-Hall, 1988.

3.2.3 DEC FUSE

DEC FUSE is an ULTRIX-based, integrated, graphically-oriented, multi-language software engineering environment. The introduction of the Personal DECstation is contemporary with the release of DEC FUSE Version 1.1. The supported languages include C, FORTRAN, Pascal, and C++. FUSE is based on UNIX commands and utilities traditionally associated with programmer productivity and provides a set of Motif®-based graphical tools. It makes extensive use of dynamic graphical capabilities to provide visual representations of a program's structure.

DEC FUSE includes its own editor, but also supports the GNU Emacs and vi editors. DEC FUSE includes a GUI-based debugger, a Program Builder, a Call Graph Browser, a Profiler, a Cross Referencer, a C++ class browser, a code management system, and extended annotation support. Another advanced feature is support for distributed build, that is, using several workstations on a network in parallel for compiling and linking large builds.

DEC FUSE EnCASE is a facility that allows developers and end-users to integrate additional development tools into DEC FUSE. The tools added to DEC FUSE by means of EnCASE can be third-party software development products or home-grown UNIX utilities. Tools added to the DEC FUSE environment can be linked with existing DEC FUSE tools or with other newly added tools to enhance existing work procedures. DEC FUSE EnCASE uses object-oriented messaging principles to manage the integration of tools.

3.3 DECwindows and Graphical User Interfaces

Virtually all users of workstations and personal computers are now familiar with the use of a window system to work with several applications at the same time. Each window serves as a virtual graphics terminal to receive the output of an application or utility. Most window systems provide the convenience of transferring data between applications by interactive "cutting and pasting" in the corresponding windows.

The window system must be common to all the concurrent applications, and so forms part of the system software, placed between the operating system and the applications. A central software element is the window manager, which resolves contention for display area among the different applications and gives the interactive user a measure of control over the window layout. The window manager is also concerned with associating operation of a single keyboard or pointing device with input for particular applications, identified with windows.

Window systems also serve as platforms for graphical user interfaces (GUI), in which interactive users enter input by directly manipulating screen objects -- buttons, menus, scroll bars, dialogue boxes -- by pointing and clicking with mouse and cursor. If a number of applications are constructed using the same GUI, that is, a common set of interactive screen objects and a common style of manipulating them, then they are all easier to learn and use.

3.3.1 DECwindows

The X Window System (also called X or X11) is an industry-standard networked window system developed at Massachusetts Institute of Technology with financial and technical support by an industry consortium of which Digital is a leading member. It is a networked window system constructed according to a *client/server* model. The server is the software that has control of the interactive graphics workstation -- the display and the associated interactive input devices. The client is an application that carries on interactive I/O through requests to the server. The client may run on the same processor as the server, or it may reside on a remote host and communicate with the server over a network.

The X client and server communicate through a precise protocol, the X Protocol, which is designed for communicating simple graphics and window control information. In case the client and server are on different hosts, the X Protocol is implemented over the underlying network protocol. The protocol is independent of the system architecture and operating system on both ends. Therefore, the X Protocol supports interoperability for transparent network graphics among diverse systems.

The X software architecture has two parts, the server and the client-side support. The graphical device dependence is all in the server; the client side is device independent. The basic application programming interface on the client side is *Xlib,* a library of almost 400 routines for 2D graphical primitives, window management functions, and input event management. Xlib is the lowest level access that an application programmer has to the X Window System.

The standard X Window System also provides a higher-level programming interface, the tool kit *Xtk,* which provides the programmer with tools for constructing and managing interactive screen objects, such as buttons and sliders, called *widgets* in X. Xtk comes with a set of widgets, called the *intrinsics,* but it defines no policy on look-and-feel standards. It can be used to implement graphical user interfaces with particular look and feel, some of which are discussed further below in Section 3.3.2. Figure 3.2 illustrates the architecture of the X Window System.

DECwindows is Digital's enhanced implementation of the X Window System. Some of Digital's enhancements to the standard system are: faster algorithms, international keyboard support, better security features, and additional language bindings. On the DECstation 5000 series workstations, DECwindows is an exceptionally fast and robust implementation of X. The version of DECwindows shipped with ULTRIX 4.2A embodies the X11R4 version of the X Window System.

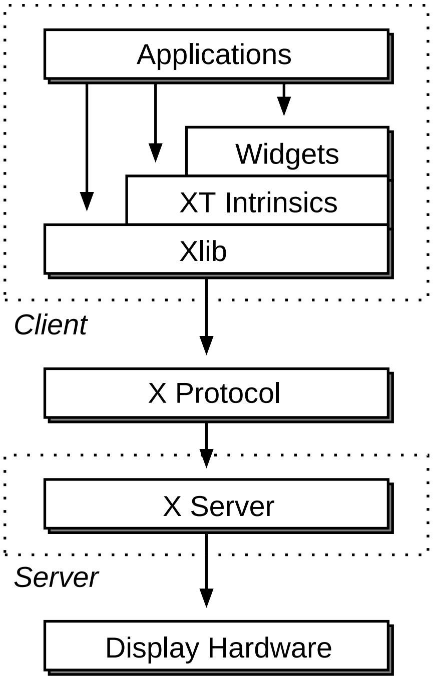


Figure 3.2 X Window System Architecture

3.3.2 OSF/Motif and VUIT

OSF/Motif® is an industry-standard graphical user interface adopted by the Open Software Foundation and based largely on XUI, the DECwindows GUI originally developed by Digital. OSF/Motif includes a tool kit for constructing interactive widgets, a user-interface language (UIL) for building applications with Motif user interfaces, and a style guide. Through adherence to the style guide, the application programmer can be sure that the application will have a standard look and feel, which will make it easy to learn and use by a user familiar with other Motif-based applications.

The ULTRIX operating system for the DECstations includes both OSF/Motif and XUI. The window manager for the former is called *mwm;* the window manager for the latter is called *dxwm.*

DEC Visual User Interface Tool (VUIT) is an interactive, WYSIWYG-style editor for building standard OSF/Motif application interfaces. VUIT provides an environment to rapidly develop, test, and modify graphical user interfaces for Motif-based applications. VUIT is a GUI for building GUIs; the application programmer uses VUIT to create his GUI by pointing, clicking, and dragging. VUIT writes and reads standard OSF/Motif UIL files, so it can be used to edit GUIs constructed without the use of VUIT. Interfaces designed with VUIT automatically conform to OSF/Motif Style Guide policies. VUIT is optionally available on all DECstation systems.

3.3.3 Extensions to the X Protocol: PEX, Display PostScript, and Xv

The standard X Protocol does not include semantics for addressing all conceivable functionality that a graphics server may provide. When new graphical functionality is added to the server side, then it becomes necessary to extend the protocol and the server code in order to be able to use the new functionality in the context of the X Window System. The design of the X Window System anticipated the need for such extensions and provides a standard way of making them.

For example, the graphical content of the original X Protocol is limited to two-dimensional graphics. However, some graphics subsystems, such as the PXG family of options discussed in Section 4.5, provide hardware and firmware for 3D graphics processing. The graphics subsystem is clearly on the server side of the X Protocol. In order to be able to use 3D hardware to make pictures in X windows, the X Consortium has defined *PEX*, a standard extension to the X Protocol. The name "PEX" stands for "PHIGS Extension to X" to reflect the fact that the semantics of PEX are based largely on PHIGS, the ANSI/ISO standard 3D graphics application programming interface (API), discussed in Section 3.4.2. However, PEX is just a protocol for communicating 3D graphics information between client applications and 3D graphics servers; it can also be used with other graphics APIs.

Implementation of a PEX server on 3D graphics hardware makes the hardware available for network-transparent, inter-vendor distributed applications. Digital has been a leading member of the PEX development and standardization effort, and produced the industry’s first PEX-based workstation in 1989. The Digital PEX server is bundled with the 3D acceleration options discussed below in Section 4.5.

A second extension of the X Protocol has been developed by Digital to support the Display PostScript® graphics API described below in Section 3.4.1. This extension has been adopted as a standard by the X Consortium.

Digital has worked with MIT to develop yet another extension to the X Protocol, called *Xv*, for support of the DECvideo multimedia options discussed in Section 5. Digital is actively working towards having Xv adopted as a standard by the X Consortium.

3.4 Graphics Programming Interfaces

The Xlib API, mentioned in Section 3.3.1, provides drawing primitives only for simple 2D figures such as lines, arcs, polygons, and text. To support development of both 2D and 3D applications, programmers need much richer and higher-level APIs for defining, manipulating, editing, displaying, and storing complex graphical objects. Digital offers several additional higher-level graphics APIs both for 3D graphics and for 2D applications that need more graphics primitives and richer graphical data structuring capabilities. For display I/O, all of these higher-level APIs work through the X Window System and its extensions. Figure 3.3 depicts the architecture of the graphics programming environment supported on the DECstations.

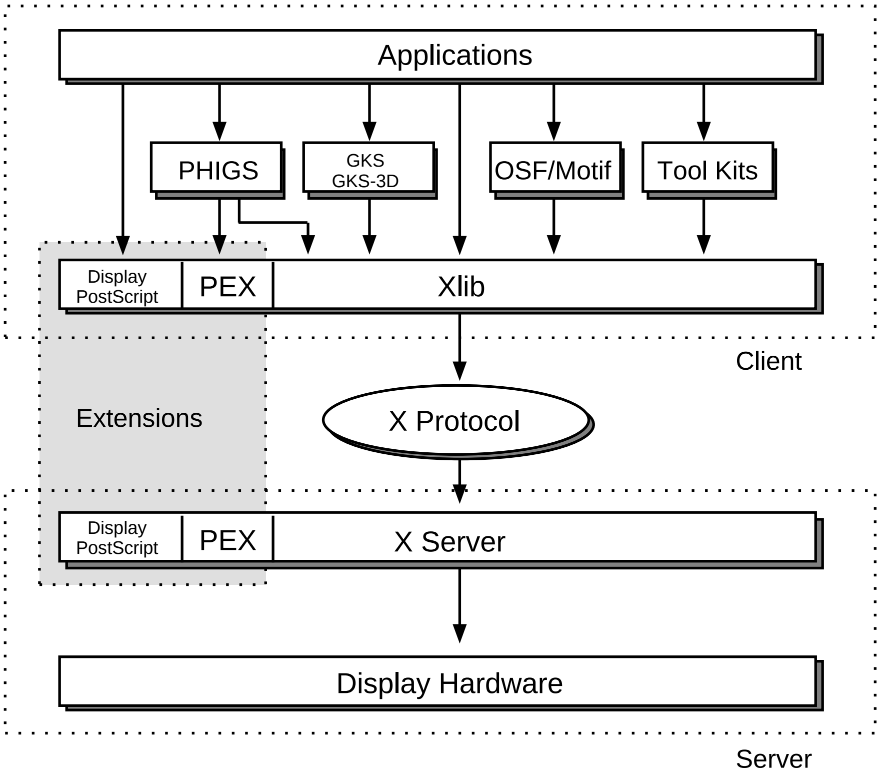


Figure 3.3 Graphics Programming System Architecture

3.4.1 Display PostScript

PostScript is a page description language, developed by Adobe Systems Inc., which has become the industry-standard means of access to the advanced graphical and typographical capabilities of laser printers. *Display PostScript* is a language for programming interactive raster displays, adapted from PostScript. Digital's implementation of the Display PostScript interpreter is licensed directly from Adobe and is complete and fully compatible.

As mentioned above in Section 3.3.3, Digital has implemented the Display PostScript interpreter as part of the X Server, and extended the X Protocol to access it. Server-side implementation of the Display PostScript interpreter has certain advantages concerning performance and compatibility. But, just as PostScript is frequently implemented by a PostScript engine within a laser printer, it is natural to anticipate the advent of Display PostScript acceleration hardware in workstation displays, which will then *require* server-side implementation and protocol extension. By taking the lead in developing server-side implementation and standardization of the protocol extension, Digital ensures future compatibility of Display PostScript applications written today.

3.4.2 DEC PHIGS and PEXlib

PHIGS (Programmer’s Hierarchical Interactive Graphics System) is the preferred ANSI/ISO standard API for 3D device-independent graphics. *PHIGS PLUS* is a proposed extension to the standard, which adds advanced surface primitives and lighting-and-shading rendering capability to the original PHIGS system. At the time of this writing, the PHIGS PLUS extension is close to final adoption; after adoption, the term "PHIGS" will be understood to include the PHIGS PLUS extension.

Standard PHIGS is a *structure-oriented* (as opposed to an *immediate-mode)* API. That is, through calls to the API library functions, the application builds data structures representing the picture to be drawn, structures which are retained and stored by the PHIGS runtime system and which can be redisplayed through subsequent calls. PHIGS structures admit hierarchical organization and structure editing, two important features that help distinguish PHIGS from the GKS and GKS-3D standards discussed in Section 3.4.3.

DEC PHIGS is a very complete implementation of PHIGS, including the proposed PHIGS PLUS functionality and several further extensions. The advanced PHIGS PLUS features of DEC PHIGS include lighting, shading, depth cueing, and non-uniform rational B-spline (NURBS) curves and surfaces. The further extensions include circle and arc primitives, an immediate-mode feature, and certain other extensions that are not part of the standard but advocated and informally encouraged by a group of CAD/CAM software companies. DEC PHIGS offers C and FORTRAN language bindings, but also supports PHIGS programming in several other languages.

DEC PHIGS is very well integrated with DECwindows and provides the programmer with several different choices for managing the interaction between PHIGS and X and for mixing PHIGS calls and Xlib calls. Further, DEC PHIGS can generate PEX protocol requests (see Section 3.3.3) to make use of hardware graphics acceleration. Alternatively, PHIGS can perform 3D graphics processing in software to generate drawing requests through the ordinary X Protocol. A few of the more compute-intensive capabilities of PHIGS PLUS are provided only when supported by 3D hardware acceleration. Still, the RISC processing power available in the DECstations enables practical 3D applications through DEC PHIGS without 3D acceleration, so long as such features as smooth shading and hardware Z buffering are not needed.

The PHIGS runtime support is bundled with the PXG, PXG+, and PXG Turbo+ graphics options and is optional on other configurations. The DEC PHIGS development system is optional on all DECstations.

PEXlib is a 3D graphics API that allows programmers to access the PEX protocol at a slightly lower level than PHIGS, analogous to the Xlib API to the X Protocol. PEXlib has no status as a standard, but it will be of interest to programmers who wish to avoid some of the overhead of PHIGS and have a little more direct control of the PEX server. PEXlib will ship for the first time with DEC PHIGS 2.3.

3.4.3 GKS and GKS-3D

GKS (Graphics Kernel System) is an ISO standard for a 2D graphics API, which was originally developed in Europe and predates PHIGS. *GKS-3D* is an extension of the standard to include 3D geometry. However, GKS-3D does not include the advanced primitives and lighting-based features of PHIGS PLUS.

Digital's implementation DEC GKS-3D supports full level 2c functionality of the standard, and is upward compatible with the DEC GKS implementation. GKS and GKS-3D are optional products for the DECstation 5000 family.

While GKS and GKS-3D have a kind of retained structures (called *segments)*, they lack the hierarchical structuring and structure-editing features of PHIGS, and the lighting and shading capabilities of PHIGS PLUS. Strictly 2D graphics applications can sometimes enjoy better performance when implemented in GKS than in PHIGS, because PHIGS is intrinsically 3D internally and therefore carries some 3D overhead even when used for 2D applications.

3.5 Networking Software

Connectivity, interoperability, open networking, and distributed applications have become normal and expected features for workstation installations in industrial, commercial, and academic environments. DECstations provide the full range of options for connectivity, both in Digital's proprietary network and in multivendor open networking situations. Ethernet support through the I/O subsystem is standard on the Personal DECstation. In addition, connections to Fiber Distributed Data Interconnect (FDDI) and IBM® Token Ring are available through TURBOchannel options. These networking possibilities are all supported by available software.

3.5.1 TCP/IP

The TCP/IP network protocol is a set of software communications protocols, which were developed in government and defense contractor installations, and are now widely used in many networking environments. In DECstation workstations, TCP/IP is the protocol by which the Ethernet networking hardware is used. TCP/IP has become a *de facto* standard in UNIX operating environments and is an integral part of the ULTRIX operating system, supporting such facilities as remote login (*rlogin)*, remote copy (*rcp*), and file transfer protocol (*ftp*), as well as the Network File System (NFS™), discussed in Section 3.5.2.

TCP/IP is not limited to UNIX systems and can be used with other operating systems. In particular, DECstations can use TCP/IP to communicate with VMS systems using the FUSION TCP/IP product.

3.5.2 Network File System

The ULTRIX operating system supports the NFS, an industry-standard facility that governs file sharing among networked systems. NFS allows a workstation user to access disk storage on a remote file server as if it were local, and provides a network administrative service called Yellow Pages. NFS comes standard with ULTRIX on all DECstation workstations. In the future, the OSF/Distributed Computing Environment (DCE) will provide NFS interoperability as well as significantly enhanced file services.

3.5.3 DECnet/OSI

DECnet is Digital's optional networking product that implements the Open Systems Interconnect (OSI) model. DECnet provides communication among both RISC- and VAX-based ULTRIX systems as well as VAX/VMS systems. DECnet offers task-to-task communications, establishing logical connection to other DECnet nodes in the network, remote file transfer, mail, coexistence with the Internet protocols on TCP/IP-based machines, and network-wide resource sharing and management.

3.5.4 Network Application Support

Network Application Support (NAS) is a backbone set of application services that allows integration of multiple applications in a distributed, multivendor environments. A NAS application can, for example, combine graphics from an Apple® Macintosh®, a Lotus® spreadsheet from a DOS PC, a drawing from a DECstation, data from an IBM mainframe, and a scanned image from a VAXstation all into a single report that can be sent electronically to others anywhere on the network. Figure 3.4 illustrates NAS architecture.

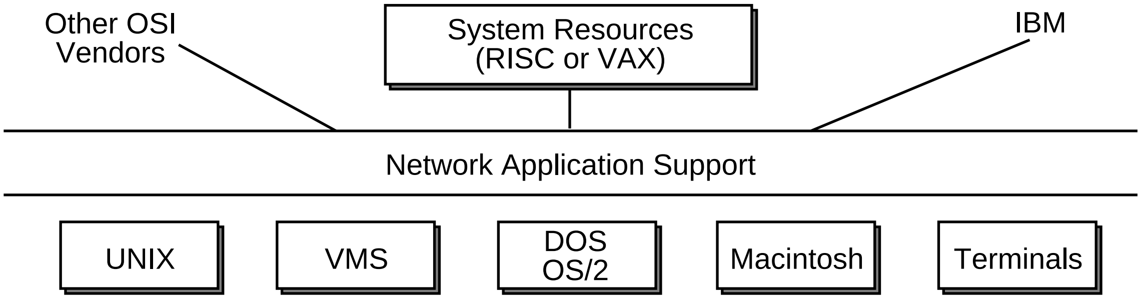


Figure 3.4 NAS Architecture

NAS consists of well-defined industry-standard programming interfaces, tool kits, and products that help developers build applications that are well integrated and easily portable across diverse systems. Some of the components of NAS are:

- ULTRIX/SQL, an industry-standard relational database management system with a Structured Query Language (*SQL*) interface. The ULTRIX/SQL runtime library and license are bundled with the ULTRIX operating system.
- Remote Procedure Calls (*RPC*), which allow remote execution over loosely coupled CPUs in a heterogeneous network. DEC RPC is based on the Network Computing System™ (NCS) from HP/Apollo, and is bundled with the ULTRIX operating system.
- Personal Computing Systems Architecture (PCSA), Digital's PC integration software that combines DOS or OS/2®-based PCs into a PC Local Area Network or corporate computing network.
- IBM 3270 terminal support and access to IBM environments through the DECnet/SNA family of IBM interconnect products.
- DECwindows, Digital's implementation of the X Window System (see Section 3.3.1).
- Compound Document Architecture, a method for creating, storing, and exchanging files that contain a number of integrated components including text, synthetic graphics, and scanned images.

3.6 Application Software

Because of its binary compatibility with the well-established DECstation 5000 family, the Personal DECstation enjoys a large repertoire of ready-to-run application software packages from many independent software vendors. See the ULTRIX Software Source Book (EC-J1284-43) for listings of available technical and commercial applications.

Because of its role as an entry-level system with pricing typical of personal computers, the Personal DECstation is an appropriate platform for the personal productivity applications that have formed the main stay of the PC industry. A wide range of the most popular PC applications -- spreadsheets, word processors, desktop publishing systems, presentation systems, drawing packages, database systems, and integrated packages -- have been ported to the DECstation 5000 architecture. Moreover, DEC SoftPC for ULTRIX is a software PC emulator which allows running the machine code of many MS-DOS® applications on the DECstation 5000. SoftPC emulates the MS-DOS file system and almost all the hardware features of the IBM PC/AT™ class of personal computer. The performance of typical PC applications running in the SoftPC emulation on a Personal DECstation is usually better than the performance on a 80286-based machine.

4 Graphics Options

Digital offers a wide range of TURBOchannel-based graphics options, which are applicable to all the DECstation 5000 models and may be used as an alternative to the Personal DECstation's on-board graphics. Several new options are introduced at the same time as the Personal DECstation. Two new options are higher-performance versions of earlier architectures and two have new architectures. All of the graphics options are TURBOchannel options, taking one to three TURBOchannel option slots. Table 4.1 summarizes the relevant characteristics of the TURBOchannel graphics options together with the Personal DECstation's on-board graphics.

The MX and PXG options were previously available. The new options are the HX, TX, PXG+, and PXG Turbo+.

Table 4.1 Graphics Options Summary

	Resolution	Depth	Refresh	Tc Slots	Acceleration Notes
Personal DECstation	1024x768 On-board	8	72/66 Hz	0	None
Options					
MX	1280x1024	1	72 Hz	1	None
HX	1280x1024 1024x864 1024x768	8	72/66/60 Hz	1	X primitives; Smart Frame Buffer; Fast Tc IF
TX	1280x1024	24+8+1	72/66 Hz	1	None; supports DECvideo
PXG	1280x1024	8/24+DB+24Z	72/66 Hz	2	Scan conversion; optional Z buffer (for 8-plane versions) and 3D geometry
PXG+	1280x1024	8/24+DB+24Z	72/66 Hz	2	Scan conversion; optional Z buffer (for 8-plane versions) and 3D geometry
PXG Turbo+	1280x1024	24+24+24+24	72/66 Hz	3	Scan conversion; two pixel stamps

4.1 Graphics Options Overview

This section sets the stage by introducing features common to all the graphics options and by discussing some of the ways in which graphics options differ. The subsequent sections present details on each of the available options.

Raster graphics systems are built around a *frame buffer*, which is a memory containing a pixel-level description of the displayed image and is used to refresh the raster display. Refreshing the display involves clocking out the frame-buffer data synchronously with the sweep cycle of a cathode ray tube (CRT) monitor, using the pixel values to produce the analog signals that drive the color guns of the CRT. Frequently, the frame buffer is implemented in a special kind of dual-ported RAM, called video RAM (VRAM), which allows the CPU or graphics processing hardware to draw in the picture by writing to the frame buffer without interfering with the refresh cycle. Figure 4.1 depicts a "generic" raster graphics system.

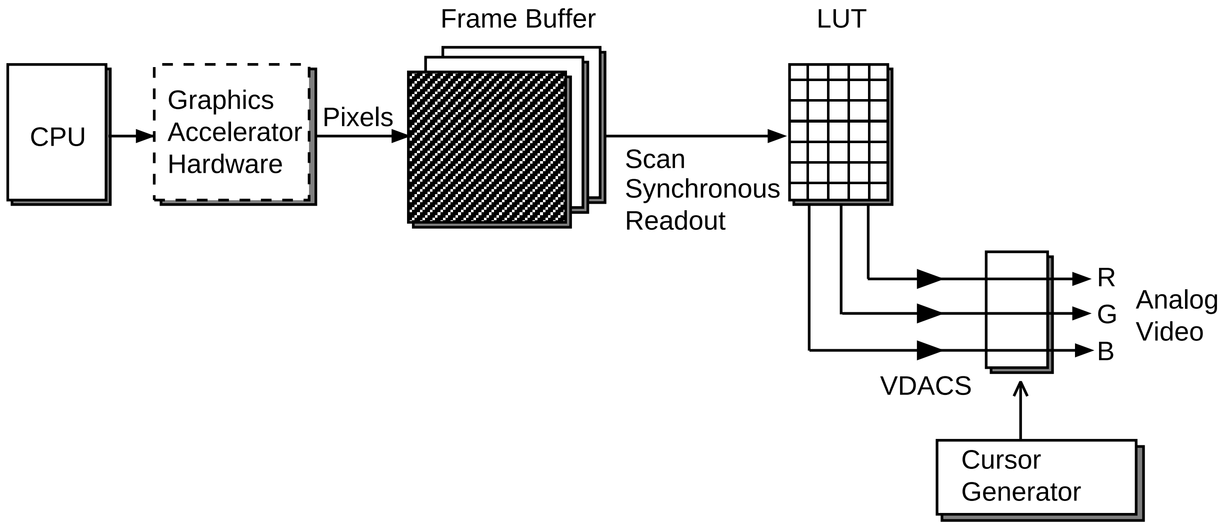


Figure 4.1 Generic Raster Graphics System

The number of bits per pixel in the frame buffer is called its *depth.* If the depth is *d*, then the maximum number of different colors or intensities that the display can simultaneously contain is 2^d: specifically, 2 colors when *d*=1, and 256 colors when *d*=8. About 16.7 million different colors are possible when *d*=24, but the maximum number of simultaneous colors is, of course, limited by the number of pixels, and this is somewhat less than 16.7 million for all of the graphics options.

On most 8-bit color systems and many 24-bit color systems, the pixel values are used as indices into a color lookup table whose entries determine the red, green, and blue (RGB) components of the color actually displayed. Typically, the lookup table entries each contain 24 bits, eight bits for each of the red, green, and blue components. The color lookup table entries are themselves under software control. Thus, when *d*=8, the lookup table has 256 24-bit entries, so the 256 different colors that can be simultaneously displayed at any moment are selected from a palette of 16.7 million colors.

A 24-bit system that has no color lookup table (such as the TX option) is called a *true-color* system. The 24 bits at each pixel are partitioned into three 8-bit fields, which directly determine the RGB values for the pixel. In 24-bit systems that have color lookup tables (such as the PXG family options), there are three 256-entry x 8-bit lookup tables, which allow remapping of each of the color components independently. Such remapping of color components is frequently referred to as *pseudo-color*.

Many frame buffers have additional *bit planes* (bits per pixel), for purposes other than encoding the displayed color information -- for double buffering, for Z buffering, and for pixmap storage.

Video digital-to-analog converters (VDAC) convert the digital RGB values from the lookup table to the analog signals for driving the CRT guns. One relevant parameter characterizing the raster system is the refresh frequency -- the number of times per second that the entire CRT screen is scanned. A display refresh frequency that is too low will produce annoying flicker. The bandwidth of the VDACs is a major limiting factor for the maximum refresh frequency. The TURBOchannel-based options discussed below all will provide 66-Hz refresh, some will also refresh the monitor at 72 Hz. Frequently, the color lookup table and VDAC circuitry are integrated on a single chip. Hardware cursor generation may also be included on the same circuit or on a different circuit. Mixing the hardware cursor signal with the image signal in the analog domain allows the cursor to move about the display without having to disturb the underlying image values in the frame buffer.

The frame buffer and VDACs are elements common to all raster graphics system. The options differ in the frame buffer parameters -- its resolution (rows and columns of pixels), depth (bits per pixel), and off-screen planes (double buffer, Z buffer). Options differ in details of the VDAC/lookup table/hardware cursor implementation and the refresh frequency.

The DECstation 5000 graphics options also differ in graphics acceleration -- the graphics processing implemented in hardware or firmware on the option cards. The MX and TX provide no graphics processing acceleration. The new HX option provides acceleration for a set of 2D line drawing, polygon filling, and image copying operations, which are heavily used in X11-based applications. The PXG, PXG+, and PXG Turbo+ options are based on a common architecture providing hardware acceleration both for the 2D scan conversion and for the 3D geometry processing -- the most compute-intensive parts of the 3D rendering pipeline (see Section 4.5). Graphics acceleration implemented in hardware or in dedicated high-performance processors running firmware can speed up applications in at least three ways. First, the graphics acceleration hardware is specialized to perform a particular graphics processing task and so can perform it faster than the general-purpose CPU working with system memory. Second, graphics acceleration hardware usually reduces the amount of data that has to be communicated over the I/O channel between the CPU and the graphics subsystem. For example, if the hardware does line drawing, then only the endpoints of the line need be sent to the graphics acceleration hardware, whereas if the CPU is doing the line drawing then all the pixels in the line have to be sent. Third, the graphics acceleration hardware frees the CPU from routine graphics processing to do other work on the application simultaneously with the offloaded graphics processing.

The following sections present more details on the graphics acceleration features.

4.2 MX Graphics Option

The MX is a single-plane (1 bit-per-pixel), high-resolution (1280x1024) monochrome frame buffer. A Bt455 RAMDAC is used to drive the monitor, with a 72-Hz refresh frequency. A Bt43 Cursor chip provides a hardware cursor.

The MX uses one TURBOchannel slot and is used with either the 19-inch VR319 or 17-inch VRM17 monitor.

The MX is suitable for software development, desktop publishing, database management, and other applications that do not require color or greyscale but can benefit from flicker-free high resolution.

4.3 TX Graphics Option

The TX graphics option has been designed especially for use in conjunction with the DECvideo multimedia options, discussed in Section 5. It provides a socket for the DECvideo/PIP live-video daughter card.

The TX resolution is 1280x1024 and its pixel depth is 24+8+1. More specifically, the TX contains a 24-bit true color frame buffer and an independent 8-bit, color-mapped frame buffer. The contents of the single bit plane determines, for each pixel, which of the two frame buffers determines the color of that pixel. Therefore, either the 24-bit image or the 8-bit image may be used as an overlay on the other. The 24-bit frame buffer may display realtime video from the DECvideo/PIP card, as discussed in Section 5.1.2.

The TX, which occupies one TURBOchannel slot, uses a Bt463 RAMDAC to drive a monitor at 66 or 72 Hz. A pair of Bt431 Cursor generator chips provide a three-color hardware cursor. The TX frame buffers are mapped into TURBOchannel I/O space. The CPU writes to the frame buffer using TURBOchannel I/O transactions.

4.4 HX Graphics Option

The HX graphics option takes one TURBOchannel slot and has 8 bit planes with 1280x1024 resolution. On the video output side, it also uses the Bt459 RAMDAC/Cursor chip, including a 256-entry, 24-bit color lookup table. The primary refresh rate is 72 Hz. The HX option is available in several versions, each having a secondary mode with lower resolution and lower refresh rate, so that a wide range of monitor types can be supported.

The unique feature of the HX graphics option is the Smart Frame Buffer (SFB) custom ASIC, which provides acceleration for 2D line drawing, stippled polygon filling, and pixel copy and boolean operations (raster operations).

The SFB implements a fast integer Bresenham algorithm for the line drawing. The polygon fill and pixel copy operations use considerable parallelism, processing 32 pixels at a time. The pixel copy operations provide for applying any of 16 boolean operations on the source and destination pixel. Such pixel copy operations have been called *raster-ops*. It is worth noting that the particular graphical functions that the HX accelerates are all very heavily used by the DECwindows and Motif window managers, as well as many DECwindows applications.

The HX is suitable for all graphics applications that need only 8-bit color and do not require hardware 3D acceleration. Former PX applications requiring hardware double buffering will have to migrate to the PXG.

4.5 Accelerated 3D Graphics Options

The DECstation graphics options offering 3D acceleration are the PXG, the PXG+, and the PXG Turbo+, which are different implementations of a common graphics acceleration architecture, depicted in Figure 4.2. The graphics acceleration is applied to two compute-intensive stages of the *3D rendering pipeline*. Section 4.5.1 discusses the rendering pipeline in general and the common architecture of these three options. Section 4.5.2 goes on to delineate the parameters that distinguish these options.

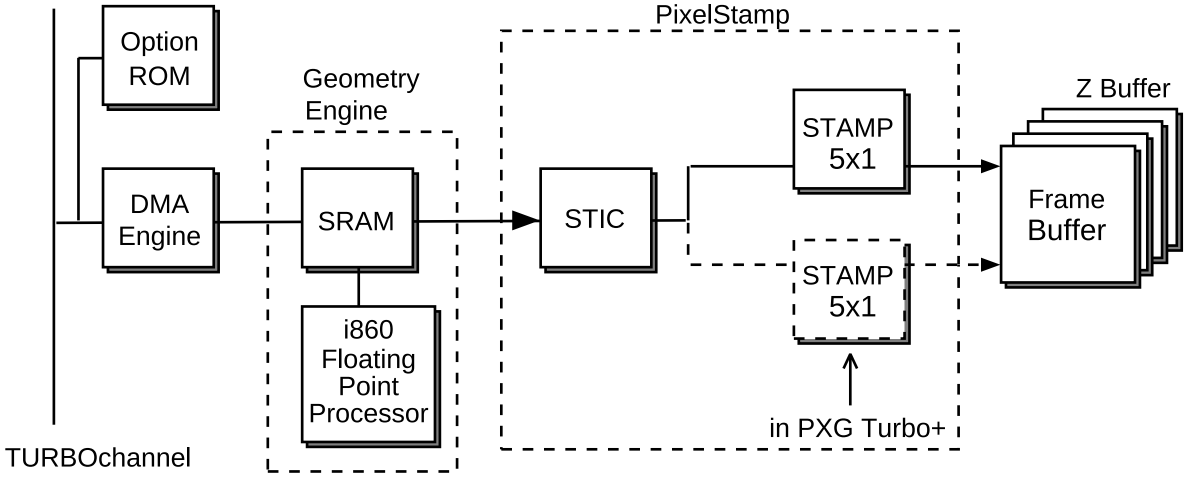


Figure 4.2 Graphics Acceleration Architecture

4.5.1 The Rendering Pipeline in the DECstation 3D Options

The rendering pipeline is the sequence of processing steps for producing the pixel-level description of the image, which is to be written into the frame buffer, from the application-level representation of the 3D scene to be depicted. This is called a *pipeline* because it consists of a sequence of steps to be applied identically to a stream of graphical elements, the same processing to each element. In 3D acceleration options, it is implemented as a hardware pipeline, that is, as a sequence of processing stages, each performing one of the rendering functions, operating simultaneously in assembly-line fashion on the incoming stream of elements.

Applications represent the 3D scene by data structures whose elements are the primitives and attributes defined by the application programming interface (API). The primitives are 3D geometric objects, typically lines and polygonal areas in an abstract 3D model space, and more complex objects made up of such lines and polygons. Typical attributes are line style, surface color, light-source color, and surface-reflectance properties. For smooth shading and other advanced features, the application must supply further data with each primitive, such as vertex colors or vertex normal vectors.

The rendering pipeline begins with traversal of the application data structures defining the scene. The traversal is performed either by the API runtime software (in a structure-oriented API), or by the application itself (with an immediate-mode API). In either case, the API runtime software running in the client CPU generates PEX requests and sends them to the PEX server running in the CPU to which the graphics acceleration option is attached. The PEX requests themselves are in terms of geometric primitives, very similar to those of the API, so generation of PEX requests amounts essentially to reformatting the data structures.

The PEX server converts the high-level 3D PEX requests to 3D request packets acceptable to the 3D acceleration hardware on the graphics option. This conversion consists essentially of breaking complex primitives into simpler components, such as line segments, convex polygons, and triangle strips, still in the 3D model space of the application.

The first stage of the hardware pipeline on the 3D acceleration graphics option is the DMA engine, which fetches from a FIFO in system memory the 3D request packets generated by the PEX server and stores them in a FIFO in the fast SRAM memory on the option.

The first computational stage on the graphics acceleration option consists of the *geometry engine,* implemented by an Intel™ i860 high-performance, 64-bit floating point processor, running firmware stored in the fast SRAM. The function of geometry processing is to map the 3D lines and polygons to the plane of the display surface. This processing involves a *projection* from 3D space to 2D space, as well as a transformation of coordinates from the device-independent 3D coordinates suitable for the application to the device-specific 2D coordinates that locate elements of the image on the display surface. This processing consists mostly of matrix multiplication and requires floating point arithmetic. Geometry processing may also include *clipping,* which is the elimination of primitives and parts of primitives that lie outside of a defined region of interest.

The results of geometry processing are stored in another FIFO in the option RAM. These results include the description of 2D lines and triangles in device coordinates. The results of geometry processing may also include some per-vertex or per-polygon data, such as colors or depth values, which carry information about the original 3D arrangement of the scene. These residual 3D data are needed for certain advanced rendering features such as smooth shading, depth cueing, or Z buffering, which are computed using interpolation at the scan conversion-stage described below.

Scan conversion is the process of determining which pixels must be set and to what values they must be set in order to draw a given 2D primitive (e.g. lines and polygons).[1]

In the DECstation 3D options, scan conversion is performed by the *PixelStamp,* which computes the pixel addresses and pixel values for drawing lines and triangles to the frame buffer. Although the PixelStamp computations are in a 2D image space, they include the interpolation in 2D space of certain quantities carrying information about the original 3D scene, such as colors or depth values. These interpolations are needed for some rendering features, such as smooth shading or Z-buffer hidden-surface removal. The PixelStamp design also supports antialiasing of lines.

The PixelStamp scan conversion works by computing the linear equations defining a given primitive for each pixel in a set of pixels determined to have a chance of intersecting the primitive. Therefore, the architecture is scalable in that it can be implemented to perform the computations for a number of pixels simultaneously in parallel.

Some advanced graphics terms:

Smooth Shading is the gradual variation of color across a primitive element used to make more realistic pictures by hiding the facets when polygonal meshes are used to approximate smooth surfaces. Such color variation is also used for data mapping in data visualization applications.

Depth Cueing is the gradual variation of color across a primitive element as an indication of the variation of depth in the 3D scene.

Z Buffering attaches a depth value to every pixel as a means of determining which parts of the objects in a scene must be discarded from the processing because they would be hidden from view by other opaque objects (hidden surface removal).

Antialiasing is a filtering operation applied to the rendered image to smooth the objectionable jagged appearance of straight lines that is caused by the finite pixel size.

Double Buffering means rendering a picture into an off-screen frame buffer, then presenting the viewer with the completed picture all at once, either by very rapidly copying the completed raster image to the on-screen buffer or by making the drawing buffer the on-screen buffer by switching. Double buffering is relevant when the time to draw a picture is noticeable, which is frequently the case when the picture is complex. Some applications, animation effects in particular, are much more effective if the viewer does not see the drawing in progress, but only completed frames.

[1] This is the meaning of the term "scan conversion" as used in synthetic computer graphics. But in the video domain, the term *scan conversion* has a wider meaning, which includes conversion from one live video format to another.

In the DECstation 5000 graphics options, the PixelStamp is implemented in two custom ASICs: the STAMP, which is essentially the linear solver and interpolator, and the STIC (for Stamp Interface Chip) which provides its interface with the incoming 2D request packets left in the SRAM by the geometry engine. The STIC chip performs certain set-up operations for the STAMP and also is the source of video timing signals for the VDACs.

The STAMP operates simultaneously on five consecutive pixels in a scan line. STAMP chips can be ganged to operate in parallel on several consecutive scan lines, as in the PXG Turbo+, discussed in the next section.

4.5.2 PXG, PXG+, and PXG Turbo+ Options

All three of these options have 1280x1024 frame-buffer resolution, and all three implement the just described 3D acceleration architecture. The difference between the graphics acceleration in the PXG and the PXG+ is that the latter uses faster versions of both the i860 and STIC/STAMP chips. The i860 runs at 40 MHz on the PXG and at 44 MHz on the PXG+. The STIC/STAMP in the PXG+ are 33% faster than in the PXG. Both options contain 128 KB of fast SRAM.

The PXG Turbo+ uses the same i860 and STIC/STAMP parts as the PXG+, but contains two STAMP chips and 256 KB of SRAM. The two STAMPs work in parallel on two consecutive scan lines, each in parallel on five consecutive pixels.

The PXG and PXG+ are available in 8-plane and 24-plane versions, each with a full screen double buffer. The 8-plane version can be upgraded to 24-planes. A 24-plane Z buffer is standard on the 24-plane version and optional on the 8-plane version. Both versions are double-width TURBOchannel options.

Both the 24-plane version and 8-plane versions can be color mapped through the table in the Bt459 RAMDAC/Cursor chip. The 24-plane version contain three of these Bt459, in order to have three 256-entry x 24-bit lookup tables, one for each of the RGB components of the pixel value.

The PXG Turbo+ option is a triple-width TURBOchannel option, so it cannot be used with the Personal DECstation without a PMTCE TURBOchannel Extender (see Section 2.5.1). The PXG Turbo+ option is available only with 96 raster planes:

- 24 image planes
- 24-plane double buffer
- 24-plane Z buffer
- 24-plane buffer for off-screen storage of pixmaps

All three of these options can be used with either the 16-inch VRT16 or the 19-inch VRT19 66-Hz or 72-Hz Trinitron monitors.

4.6 Graphics Performance Characteristics

Characterizing graphics performance in ways which can help a user discriminate among competing systems is not a straight-forward problem. The relative performance of two different systems will differ from benchmark to benchmark, depending not only on the sort of graphics processing inherent in the application, but also on the way in which it was coded. Systems which favor line drawing, for example, may perform relatively poorly for polygon fill. On some systems the performance may be greatly enhanced by aggregating objects into large primitives, while other systems may reward such aggregation very little. Some systems pay a heavy penalty for frequent attribute changes while others may not.

Certain gross drawing speed measures have become current in the industry: pixels per second for area filling, vectors per second for wireframe drawing, and polygons per second for shaded surface rendering. Such quoted numbers usually represent peak rates obtained with artificial timing programs that create most favorable conditions not necessarily always pertaining in typical applications.

In an attempt to provide a more relevant means of performance characterization, an industry committee has defined a Picture Level Benchmark (PLB) suite. These benchmarks consist of drawing complete pictures deemed to be more typical of applications than the instruments which are used to determine the peak rates. Section 4.6.1 reports a variety of peak speed measurements for the on-board graphics and several graphics options available for the Personal DECstation and Section 4.6.2 presents the first PLB results.

4.6.1 Peak Performance Measurements

This section reports on measurements of several of the peak graphics speed parameters. In each case, the measurements were made using the appropriate application programming interface, Xlib for the 2D parameters, and DEC PHIGS for the 3D. Therefore, the measurements reflect the overhead costs of the supplied APIs as well as the CPU and graphics hardware. The 2D numbers were obtained using programs from *x11perf*, the performance measurement suite supplied with the X System.

Area-fill performance is especially relevant in routine window system operation. The prototypical area fill operation is simply copying a rectangular pixmap from system memory to the frame buffer. This simple copy reflects performance characteristics which will affect many windowing and image-manipulation functions. The *x11perf* pixmap copy test uses a 500x500 pixmap. The speed of the operation is expressed in megapixels per second (Mpixel/s), where 1 Mpixel = 2^{20} = 1,048,576 pixels. The results for the Personal DECstation with the several graphics options are:

	On-board	MX	HX	PXG	PXG+	
Model 20	5.7	8.0	14.6	13.9	18.3	Mpixel/s
Model 25	6.4	8.0	14.7	13.9	18.3	Mpixel/s

Beyond area-fill, the next most interesting gross 2D performance parameter is line-drawing speed, measured in vectors per second. This test also was provided by one of the standard *x11perf* benchmarks. Vectors are 10-pixel line segments, randomly oriented, grouped in 10-segment polylines:

	On-board	MX	HX	PXG	PXG+	
Model 20	153	108	274	260	338	x 10^3 vectors/s
Model 25	183	115	285	260	339	x 10^3 vectors/s

In 3D graphics, line drawing speed remains important, but the speed for rendering shaded 3D polygons replaces the simple area fill criterion. The performance measurements are obtained with benchmarks written in DEC PHIGS. Vectors are now 3D vectors, still 10-pixel-long, randomly oriented, grouped into 10-segment polylines and clip-checked. 3D polygons are are triangles, 100 pixels in area, combined into 10-triangle strips, shaded, illuminated with two light sources, Z buffered, and clip-checked. Z buffering and smooth shading in good interactive time requires hardware acceleration, so these results are given only for the PXG and PXG+.

	PXG	PXG+	
Model 20	288	308	x10^3 vectors/s
Model 25	288	312	
Model 20	51	68	x10^3 polygons/s
Model 25	51	68	

4.6.2 PLB Measurements

The Graphics Performance Characterization (GPC) committee is a volunteer group of vendors, users, and consultants that provide and support standardized benchmarks for measuring performance of computer graphics systems. GPC operates under the administration of the National Computer Graphics Association (NCGA). The first performance measurement instrument developed by GPC is called the Picture-Level Benchmark (PLB). PLB characterizes performance in terms of the time to draw specified complete pictures. The initial set of PLB pictures adopted by GPC are:

- "pc_board" - a typical 2D electrical CAD application
- "sys_chassis" - a 3D wire frame model of a computer chassis
- "cyl_head" - a 3D solid model of an automobile engine’s cylinder head
- "head" - a 3D model of a human head based on laser scanner data
- "shuttle"- an example of low-end 3D simulation

The PLB results reported here are for the "literal" version of the benchmarks. That is, the drawings were produced exactly as specified, without attempts to improve performance by manipulating the definition of the picture. The units are in "GPCmarks", obtained by dividing a normalizing constant by the drawing time; larger values indicate better performance.

	pc_board	sys_chass	cyl_head	head	shuttle
Model 20 PXG	9.8	10.2	14.4	18.8	17.6
Model 25 PXG	9.8	10.5	14.4	18.8	17.6
Model 20 PXG+	11.2	10.8	16.2	20.9	20.0
Model 25 PXG+	11.2	11.9	16.2	20.9	20.0

4.7 Multiscreen Support

The DECstation 5000 family supports multiscreen configurations for the MX and HX options. A multiscreen configuration, as shown in Figure 4.3, consists of up to two graphics options (via TURBOchannel) of the same type, in addition to the on-board graphics, installed in a single DECstation 5000 workstation. Presently, you cannot mix TURBOchannel graphics options in a multiscreen configuration. This is a limitation of the Motif software, which could, in principle, change in the future.

Software support for the multiscreen capability resides in the X Window server code, so is available to all X-based applications and APIs. Windows must lie all on one screen, but the cursor tracks across all the screens.

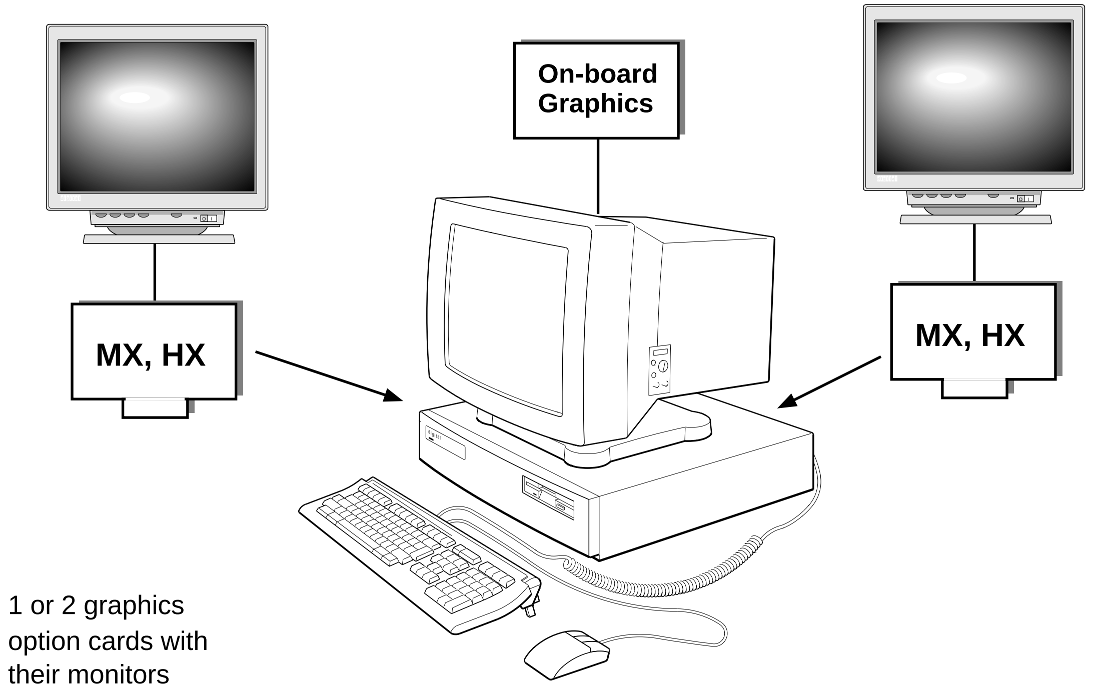


Figure 4.3 Multiscreen Configuration

5 Multimedia Options

Along with the introduction of the Personal DECstation, Digital is introducing a new line of TURBOchannel-based multimedia hardware options and corresponding multimedia software products. The audio I/O provisions of the Personal DECstation system are themselves multimedia features. This section describes further DECstation 5000 multimedia hardware and software products.

In the workstation context, the term *multimedia* refers to a set of new technologies that allow for the capture, manipulation, presentation, and integration of information involving diverse time-dependent data types, such as animated graphics, scanned still images or video, and audio, as well as ordinary text or spreadsheet data. Existing or anticipated future multimedia configurations connect the workstation with various sources and sinks of these data types: video cameras, tape recorders, laser disks, analog audio circuits, digital audio recording equipment or musical instrument interfaces, digital or analog telephone equipment, FAX, and other media. Training, education, presentation, and conferencing technology are obvious multimedia application areas. But multimedia techniques have potential for enhancing the general interactive use of workstation in many ways. Conversely, the introduction of the computer, with its programmability, data conversion capabilities, and interactive control features, enables many innovations in traditional audio/visual applications.

The high information content and communication bandwidth requirements of typical audio/visual media demand corresponding high-processing performance and storage capacities of the multimedia host workstation. The RISC-based DECstation 5000 workstations, with their high-performance TURBOchannel I/O interconnect, are ideal platforms for desktop multimedia. Generally, multimedia applications depend on the availability of medium-specific hardware provisions beyond the processing, storage, and peripheral functions traditionally associated with workstations. Section 5.1 presents relevant details of the DECaudio and DECvideo/PIP hardware options, which can be used with any DECstation 5000 series workstation.

Of course, a full realization of the potential of multimedia will depend on the development of appropriate software -- both innovative multimedia application software and the infrastructure of system software needed to support distributed access to the multimedia hardware. Digital's first offerings of multimedia software emphasize robust runtime support of the multimedia hardware and tools for application development. With the XMedia Tools software, Digital is taking a leading role in defining a multimedia software architecture based on the client/server model.

Multimedia is a relatively new feature of workstations, and the existing repertoire of multimedia applications is small in relation to what is envisioned for the near future. Digital's first multimedia software offerings include several useful layered applications, as well as some small control applications and other demonstration applications included with the basic XMedia Tools kits. In addition, several independent software vendors have announced forthcoming applications and application development products supporting the DECmedia hardware and making use of the XMedia Tools software.

Section 5.2 discusses XMedia Tools and other multimedia software. Some multimedia applications are entirely software based and can run on any DECstation, without needing multimedia hardware options.

5.1 DECmedia Hardware Options

The first DECmedia hardware offerings include both audio and video options.

5.1.1 DECvideo/PIP Option

The DECvideo/PIP live-video-in option requires the single-slot TX graphics option described in Section 4.3. DECvideo/PIP (Picture-InPicture) is a daughter board that plugs into the TX graphics option. Its primary function is to scan-convert live video signals into the TX's 24-bit true color frame buffer. The PIP daughter board also contains a 1280x1024 single-bit plane for video selection; its contents control which pixels in the TX true color buffer are set by the video signal. The image written to the 24-bit buffer may range from full size (640x480) to a single pixel, under program control. The decoded signal may appear anywhere within the 1280x1024 image.

DECvideo/PIP accepts several formats for the input video, including the three standard composite (1-wire) formats -- NTSC, PAL, and SECAM, as well as S-video (2-wire) and RGB. Decoders for the composite and S-video are included on the board. Registers on the board, accessible in the TURBOchannel address space, control the positioning and scaling of the image in the true color frame buffer. The video signal is scaled down from the full size by dropping pixels.

Using the 1-bit selection plane in the TX, the live video window can be overlayed with an 8-bit graphics image.

5.1.2 DECaudio Hardware Option

The DECaudio hardware option includes a single-slot TURBOchannel board and an associated Distribution Box. The DECaudio board duplicates the voice-grade audio functionality that is already present on the Personal DECstation system board, and may be used to bring this capability to other DECstation 5000 models. Further, the DECaudio board includes a Motorola 50001 digital signal processor (DSP) and supporting SRAM memory. The DSP has the potential for providing a variety of high-bandwidth audio functions, such as music-quality stereo, audio synthesis, voice synthesis, text-to-speech conversion, speech recognition, speaker phone echo cancellation, modem simulation, and others. However, the DSP-based features of DECaudio are not yet supported by software. Until software support for the DSP is developed, there is no reason for using DECaudio with a Personal DECstation, and so the DECaudio option is not yet supported by the XMedia Tools software on the Personal DECstation. Rather, the audio features of the XMedia Tools software are applied to the Personal DECstation's on-board audio I/O.

5.1.3 Multimedia Memory, Storage, and CD-ROM

Multimedia is memory-intensive -- at least 16 MB is recommended for video, and 24 MB or more is required when it is necessary to capture video frames. Applications combining audio and video can require 32 MB. Equal attention must be given to the disk storage requirements when applications are to store video frames on disk.

CD-ROM is supported by the I/O subsystem as a SCSI device. Because of the extreme storage demands of multimedia applications, CD-ROM is recommended for any multimedia-capability workstations.

5.2 Multimedia Software

XMedia Tools Version 1.0 is audio/video enabling software for ULTRIX workstations. It consists of an Audio/Video Developer Kit and an Audio/Video Runtime Kit. The Developer Kit provides tools for constructing distributed multimedia applications or for adding audio and video components to existing applications and documentation. The Runtime Kit contains the drivers, servers, and the client-side routines needed to run applications built with the Developer Kit. The Runtime Kit includes small demonstration audio and video clip libraries. In addition, the Developer Kit provides an extensive CD-ROM-based audio clip library, useful for constructing audio applications.

XMedia Tools is based on the client/server model for managing applications’ access to shared resources in a network. The server controls the shared resources -- peripheral equipment, data, or applications, -- and the clients request access to the resources through messages to the server. The application of the client/server model to the management of peripheral equipment is familiar to many because of its use in the X Window System.

The client/server model provides several kinds of benefits. First, it enables distributed access to resources by clients communicating over a network. The server localizes the mechanisms for arbitration of contention for the resources among client applications and for synchronization of access when the clients are cooperating. Another important benefit is that the server contains all the particular device dependencies of the peripheral equipment. The client applications communicate with the server through a precisely defined protocol whose semantics are based on abstractions characteristic of the generic nature of the service, but are independent of device peculiarities. The use of such a protocol can promote portability of applications over diverse platforms and interoperability between systems from diverse vendors.

Because on-screen video shares the workstation display with X-based graphics, the video server must work closely with the X Server, and clients communicate with it through an extension to the X Protocol, called Xv, mentioned above in Section 3.3.3.

Similarly, for providing access to DECaudio equipment, Digital has developed an audio server and corresponding audio server protocol, analogous to the X Window System. However, the audio services have no necessary dependency on the window services, so the audio server and protocol are not implemented as extensions to the X Window System.

Section 5.2.1 describes the XMedia Audio Server, its protocol, the corresponding client-side library, and a few audio utilities and simple audio applications included with the packages. Section 5.2.2 describes the video server extension and corresponding basic client-side software. The remaining sections describe further layered multimedia applications -- Desktop Video Conferencing and the DECstation FAXserver.

5.2.1 XMedia Tools Video Software

The DECvideo/PIP live-video-in capability is addressed through the Xv extension to the X Server and Protocol. The basic capability is to input one channel of analog full-motion video (of any of several formats -- see Section 5.1.2), digitize it, and display it within a specified window. Programmers access the video facilities through the Xv protocol by calls to the Xvlib program library, analogous to Xlib.

The Runtime and Developer Kit also include a small set of demonstration video utilities and applications:

- *dxgrabframes* enables storing on disk individual digitized frames of the video input to the DECvideo/PIP option from a CAV laser disk
- *vset* is a control panel application that allows the user to set various preferences concerning the live video picture, such as brightness, contrast, hue, and saturation

5.2.2 XMedia Tools Audio Software

In order to provide applications with a device-independent interface to audio hardware, the audio server is structured into device-dependent and device-independent layers. The device-dependent layer contains drivers for the physical audio devices. The device-independent layer contains *virtual audio devices*, which are software abstractions that provide the higher levels of software with an implementation-independent view of audio resources. Virtual audio devices belong to classes describing their generic function, such as *input*, *output*, *speaker*, *player*, *recorder*, *telephone*, *mixer*, and others. Virtual devices can be organized into more complex structures called *logical audio devices*.

The audio server protocol provides transparent distributed functionality on a TCP/IP network. The protocol supports several types of messages: *request, reply event, error*. A separate command queue is associated with each virtual device that the server drives. The server provides synchronization primitives necessary to sequence commands and devices, as well as to prioritize audio resource utilization among different clients. In addition, the server provides for seamless play of a sequence of commands and permits audio mixing. Events are provided to allow the development of sound editors and advanced client-side synchronization facilities. The runtime software includes an application-side Audio Manager, analogous to the window manager in a window system. It sets and enforces policy on resolving contention among clients for the audio resources.

The application programmer access to the audio server and protocol is through an audio library *Alib*, analogous to Xlib. At a slightly higher level than Alib, the developer kit includes an *Audio Tool Kit* and *Audio Widgets*, which offer support for constructing audio user interfaces.

- *aset* is a utility by which the user can set parameters of the logical audio devices (e.g. gain and sound library path)
- *play* is a widget that provides for playing digital audio recorded in files (on magnetic disk or CD-ROM) to selected analog audio output
- *record* is a widget that provides for recording input analog audio to digital files

In addition, the XMedia Tools kits contain a *dxsoundbrowser* application, which enables the user to view a list of the sound libraries and to select, play, or record each sound. The *dxplayview* application is a sort of audio editor, allowing a user to compose a sound from a number of different sound fragments.

5.2.3 Software Motion Pictures

XMedia Tools includes Software Motion Pictures, which is an effective means for displaying animated sequences of images on fast workstations with raster displays. It does not require any special video hardware for the playback functions. Any source of digital images may be used in producing a Software Motion Pictures clip, including video images grabbed from the DECvideo/PIP input with the *dxgrabframes* utility mentioned in Section 5.2.1.

Software Motion Pictures uses a proprietary image compression scheme that permits realtime decompression in software on typical workstation CPUs. This algorithm provides image compression ratios as high as 200, which permits storing 2 minutes of video in 16 MB, or 37 minutes on a compact disk. The maximum frame rate for realtime decompression depends on the image size and the CPU.

In addition to *dxgrabframes*, the Software Motion Pictures production tools include *dxsmpcomp*, which is an image compiler that performs the compression, and *dxsmpcat*, which is an editor that allows cutting and splicing of Software Motion Pictures sequences.

5.2.4 DECstation FAXserver

The DECstation FAXserver software provides FAX capabilities to ULTRIX workstations on a network in which at least one node is attached to an external FAX modem via the serial port. The server software supports any FAX modem that is compatible with the proposed EIA/TR29.1 Service Class 2 standard, such as the EverFAX 2394D.

The DECstation FAXserver enables the transmission of ASCII text, image files in G3 format, or pages defined by PostScript files. The client can send FAXes through an extension of the ULTRIX *lpr* line printer command or an extension of the ULTRIX mailer. A receiving daemon handles incoming FAX documents, which can be stored in G3, DDIF, or Encapsulated PostScript formats. Incoming FAXes can be autoprinted, sent to a dispatcher as an electronic mail message (in DDIF format), or simply stored in a directory as files.

5.2.5 Desktop Video Conferencing

Digital's Desktop Video Conferencing software provides a capability for multistation video conferencing among DECstations, using standard network protocols. Each workstation must be equipped with appropriate multimedia hardware and networking options. In order to serve as a video conferencing station, a Personal DECstation needs a TX frame buffer with DEC-video/PIP, a video camera, and microphone. Ethernet can support video conferencing with 320x240 resolution at 10 frames/sec in black-and-white and at 4 frames/sec in color. FDDI networking can support 17 frames/sec in black-and-white and 7 frames/sec in color. Moreover, FDDI allows distribution of the conferencing stations over a 40 km distance.