XVM HARDWARE

XVM SOFTWARE

XVM UPGRADES

THIS DOCUMENT IS A GUIDE TO XVM PRODUCTS AND SERVICES.

PARTICULAR EMPHASIS IS PLACED ON SOFTWARE DESCRIPTIONS

AND UPGRADE OF EXISTING PDP-9 AND PDP-15 SYSTEMS.

XVM Upgrades

Table of Contents

Topic	Page

PDP-15 TO XVM UPGRADE

Your PDP-15 can be field-upgraded to XVM capabilities through the addition of the XM15 memory processor. The benefits of XVM include:

- performance comparable to 800ns MM/MK systems
- low prices for add-on MF15 memory
- all the functions of KM15/KT15 relocate/protect plus
  - even faster address translation
  - wide addressing
  - core sharing
  - reentrancy
- all the functions of KA15 automatic priority interrupt

  XVM also allows these growth capabilities:
- all the functions of MX15-B Unichannel interface plus
  - even faster address mapping
  - use of 11/40, 11/45 and other CPU's with memory management (not supported by PIREX)
  - more than 12K local memory; up to 124K total addresses
  - increased Unibus data bandwidth

- memory to 256K (requires custom software)
- Unichannels shared among two or more XVM CPU's
- ring - or master-slave arrays of two or more XVM CPU's

Included in the price of XM15 installation is the reinstallation on XVM of these options:

- ME15 980 ns. memory
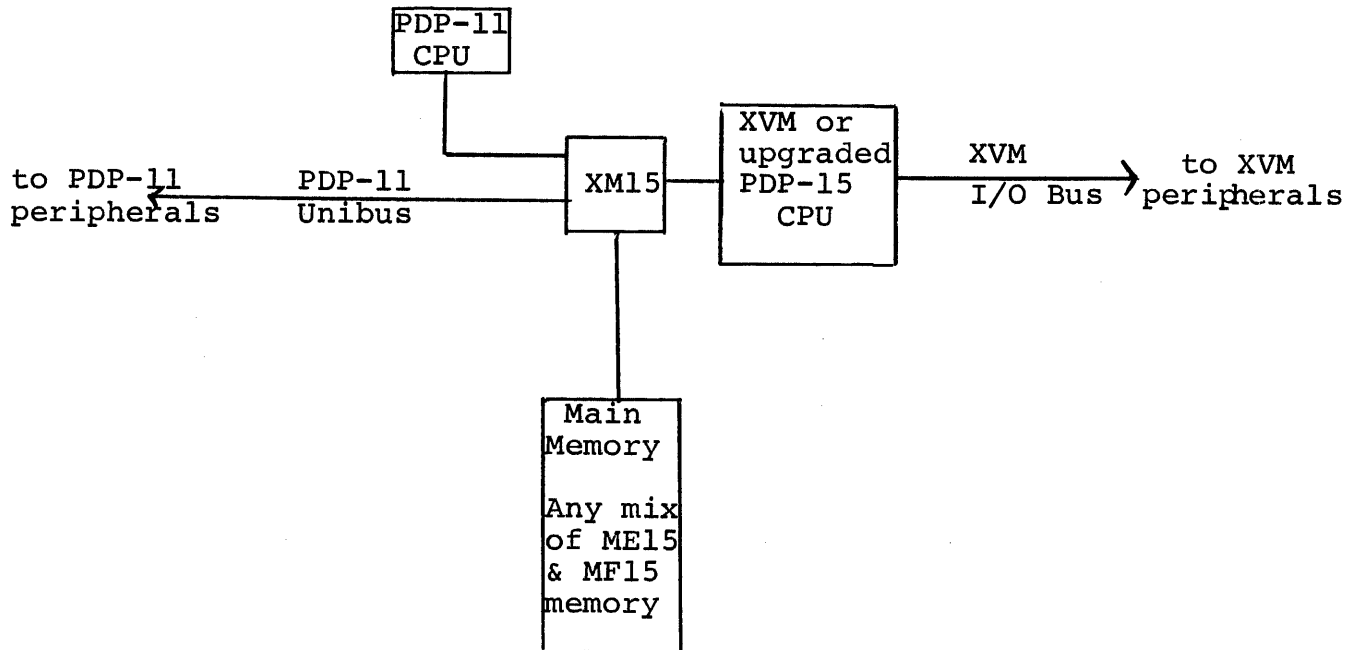- 11/05 or 11/10 Unichannel CPU and peripherals

and the deinstallation of these items which are replaced by the XM15:

- BB15/KA15/KM15/KT15 to priority interrupt/relocate/protect
- MM15/MK15   800 ns. memory
- MX15A/MX15B multiplexors

Because the XM15 represents a complete reengineering of the memory subsystem and certain complex CPU options, it is necessary that your PDP-15 CPU be upgraded to the very latest factory ECO level at the time of XM15 installation. If your system is covered by a DIGITAL maintenance contract, no extra charge will be made for this service.

XM15 and System Architecture

The XM15 memory processor relates to the overall system
in this manner:

```
                          ┌─────────┐
                          │PDP-11   │
                          │CPU      │
                          └─────────┘
                               │
                                              ┌─────────┐
                                              │XVM or   │
                                              │upgraded │   XVM
  to PDP-11        PDP-11      ┌─────┐         │PDP-15   │   I/O Bus    to XVM
  peripherals  ◄── Unibus ─────│XM15 │─────────│CPU      │──────────►  peripherals
                               └─────┘         └─────────┘
                                  │
                               ┌──────────┐
                               │Main      │
                               │Memory    │
                               │          │
                               │Any mix   │
                               │of ME15   │
                               │& MF15    │
                               │memory    │
                               └──────────┘
```
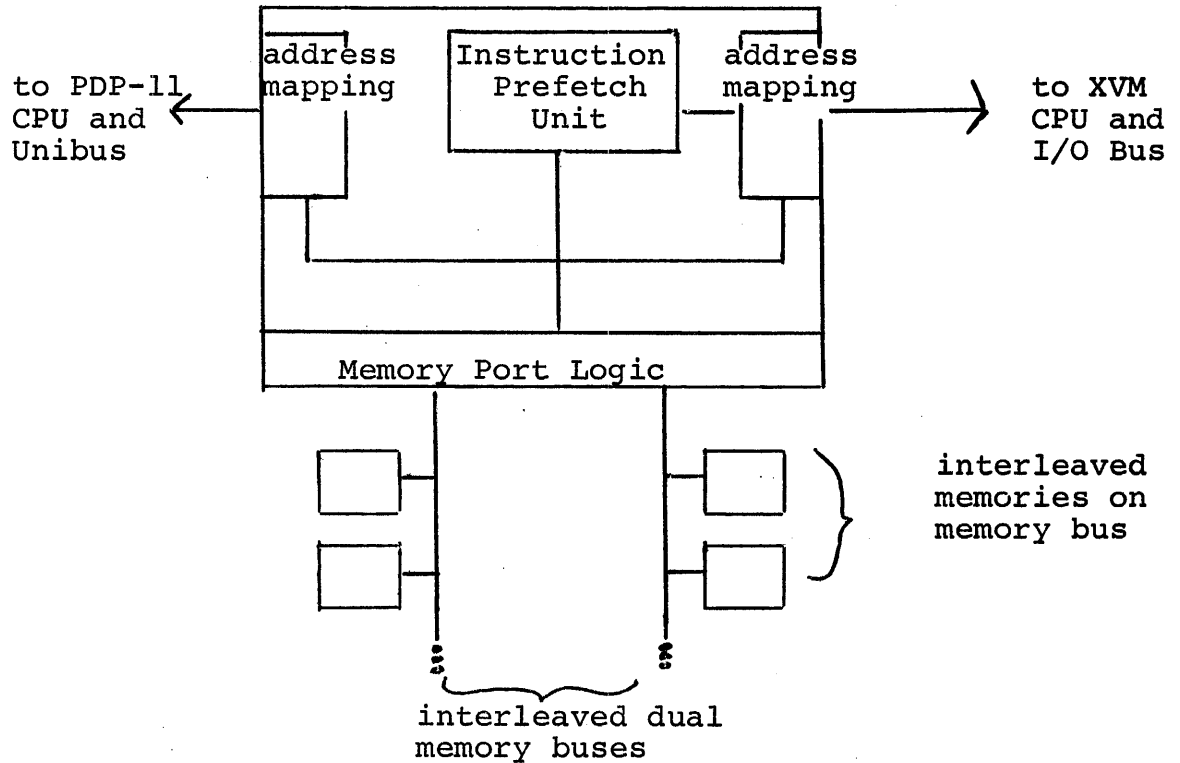
XM15 is designated as a "memory processor" because it has a
performance goal the rapid satisfying of memory requests
issued by these asynchrouously operating system elements:

- XVM central processor
- XVM peripherals
- Unichannel PDP-11
- Unichannel peripherals

To achieve this goal, the design of XM15 incorporates three
design elements:

1. Use of fast Schottky logic for all address mapping
    functions (protect/relocate; Unichannel).

2. Use of interleaved memories and dual interleaved
    memory busses to increase memory bandwidth by
    reducing the statistical likelihood of conflicts
    in memory access by the various system elements.

3. Exploitation of the increased memory bandwidth by
    means of an instruction prefetch unit that antici-
    pates   central processor instruction requests and
    buffers the instructions in a Schottky scratchpad
    memory.

XM15 Architecture

```
              ┌──────────────────────────────────────────────┐
              │ ┌─────────┐  ┌───────────┐  ┌─────────┐       │
to PDP-11     │ │address  │  │Instruction│  │address  │       │   to XVM
CPU and    ◄──┤ │mapping  │  │ Prefetch  │  │mapping  │       ├──►  CPU and
Unibus        │ │         │  │   Unit    │  │         │       │    I/O Bus
              │ └─────────┘  └───────────┘  └─────────┘       │
              │                                               │
              ├───────────────────────────────────────────────┤
              │            Memory Port Logic                  │
              └──────────┬──────────────────┬─────────────────┘
```

interleaved
memories on
memory bus

interleaved dual
memory buses

Memory Interleaving

XM15 has dual memory busses. Within each bus, the banks of
memory are interleaved so that sequential addresses are
assigned to successive memory banks. For example,

memory bus

Bank 0	addresses 0, 4, 8, 12...
Bank 1	1, 5, 9, 13...
Bank 2	2, 6, 10, 14...
Bank 3	3, 7, 11, 15...
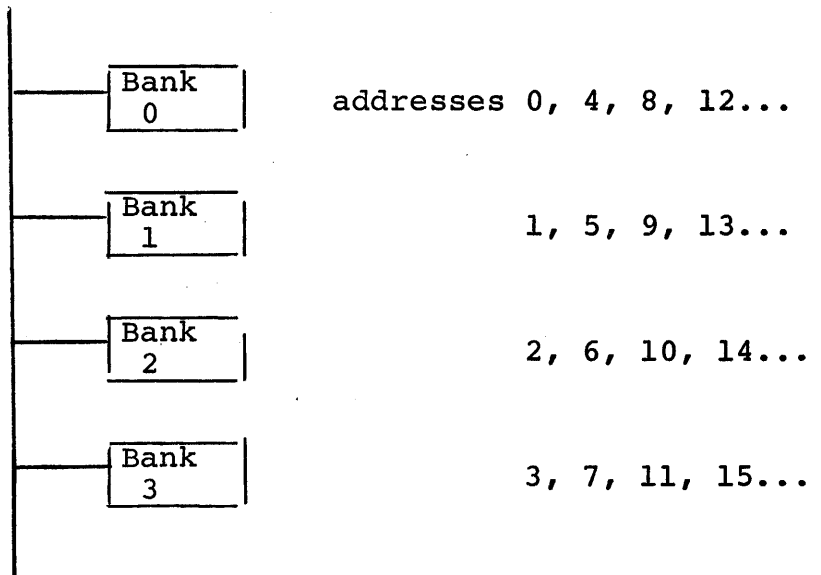
Interleaving increases throughput by maximizing the amount of time the memory bus is available for data transfers. A core memory passes through two distinct states when reading a location:

1. Read the word in question (destructively) and place the data on the bus.

2. Restore the data to the word read in step 1.

Interleaving allows the memory bus to be released as soon as step 1 is complete and so a word can begin to be read from one bank while another bank is completing its restore cycle.

The statistical likelihood that a given bank will be free when the system needs to access it is further increased by interleaving the memory busses themselves.

Instruction Prefetch

The instruction prefetch unit puts memory cycles to work that might otherwise be unused. Instruction words are pulled from memory before the XVM central processor actually asks for them. The prefetch unit remains up to 4 instructions ahead of the central processor. It contains logic which detects jumps, CAL's and other instructions which break sequence, and resynchronizes with the instruction stream in advance of sequence change.

For I/O intensive applications the instruction prefetch unit can be turned off under program control. A very useful aid to high-speed data collection, this slows down the central processor but makes the increased XVM memory bandwidth more available to the XVM I/O Processor.

When block I/O transfers are known to have been completed, the instruction prefetch unit can be turned on again under program control, allowing the CPU to run at its maximum rate.

Wide Addressing

XM15 also provides a number of indirect speed-ups to XVM systems because of its wide addressing feature.

In XVM/DOS and XVM/RSX, wide addressing increases system throughput by allowing data to be resident in memory beyond the 32K boundary. For applications requiring the inversion of very large matrices, for example, the increased amount of memory-resident data reduces the number of disk accesses required during the course of execution.

Under XVM/RSX, the XM15 option provides two more indirect
speed-ups in addition to that furnished by wide addressing.
The first is core sharing and the second is the task account-
ing timer.

Core Sharing under XVM/RSX

This core sharing facility allows several tasks to share a
global segment containing up to 8K of common data or instruc-
tions. The memory which is to be shared is reserved as a
separate partition which, strictly speaking, is not physically
part of any of the using tasks.

The using tasks have their own (local) partitions delimited
by the relocate/protect registers of XM15's memory management
logic. The main relocate/protect register creates a task
address space which begins at virtual address zero and runs
through the upper address corresponding to the length of the
task as determined by the Task Builder (which binds the re-
locatable compiler or assembler output into an executable
task core image).

Tasks normally address only their own local virtual address
space; this space is mapped into physical memory addresses
using the combination of relocation and boundary registers.

Attempts to reference outside this local virtual space are
normally illegal; however, with core sharing enabled, a task
has the additional privilege of including the virtual addresses
120-128K (partition size between 32 and 128K) or 24-32K (parti-
tion smaller than 24K) in the address space of the task for the
specific purpose of core sharing.

The auxillary global segment relocate/protect registers cause
these virtual addresses to refer to a partition of memory
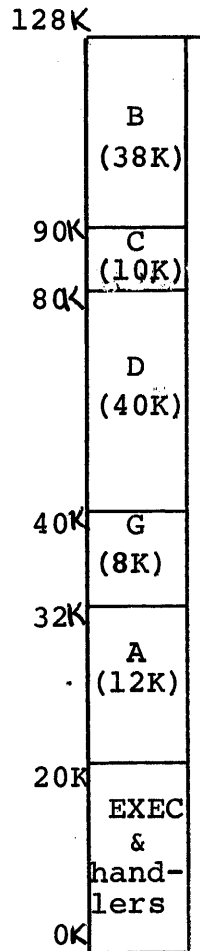maintained separately from the memory partitions of the using
tasks.

The separate virtual address spaces of tasks A, B, C, and D
are diagramatically represented in the following drawing.

Tasks C and D share an 8K common segment labeled G.
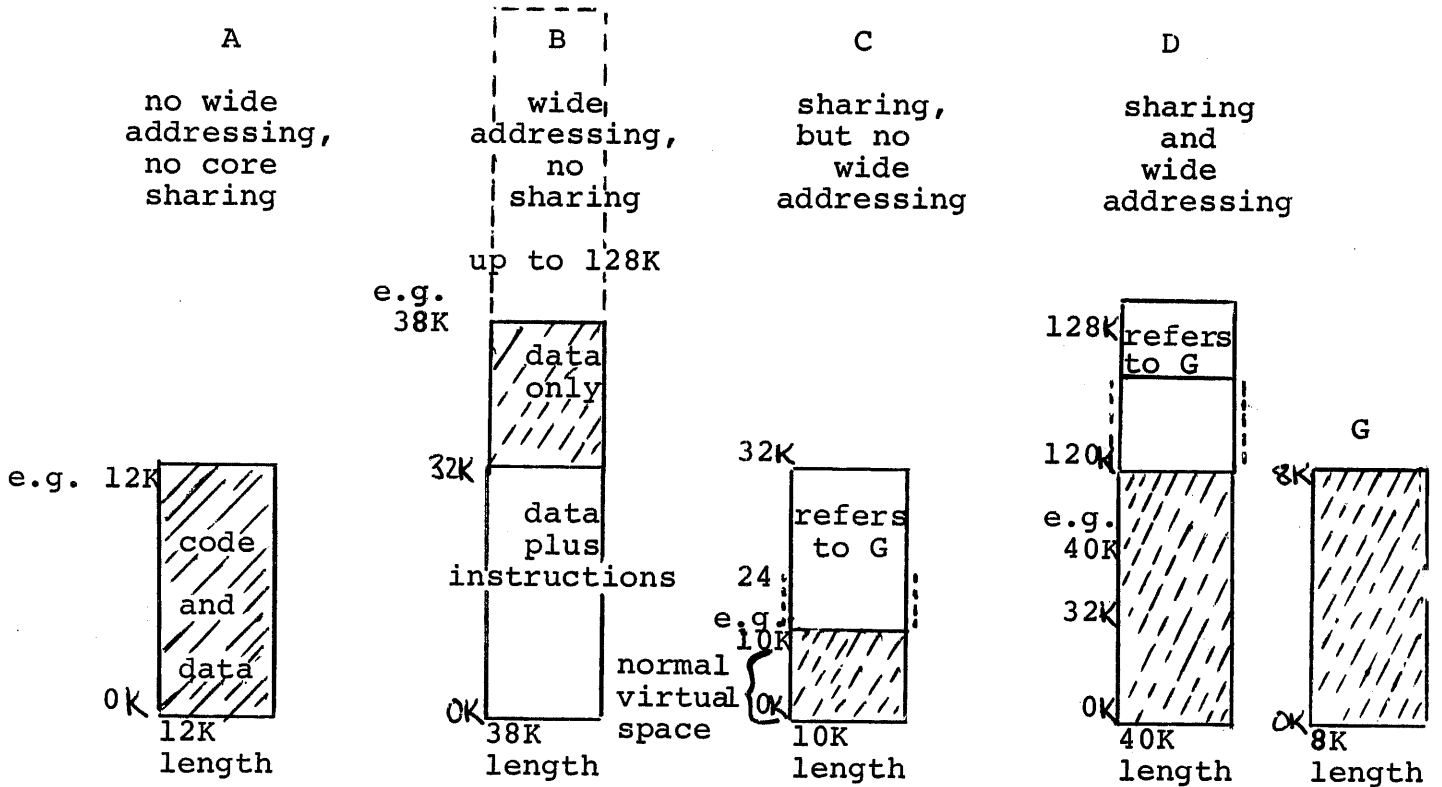
Physical memory is allocated only to those regions shaded in the diagram. The other regions are virtual addresses which are either global (refer to G) or illegal (not in any way included in the task's address space).

The diagram above depicts virtual addresses only. A diagram of physical memory follows.

```
128K ┌─────────┐
     │    B    │
     │  (38K)  │
 90K ├─────────┤
     │    C    │
     │  (10K)  │
 80K ├─────────┤
     │         │
     │    D    │
     │  (40K)  │
     │         │
 40K ├─────────┤
     │    G    │
     │  (8K)   │
 32K ├─────────┤
     │    A    │
   · │  (12K)  │
     │         │
 20K ├─────────┤
     │  EXEC   │
     │    &    │
     │  hand-  │
     │  lers   │
  0K └─────────┘
```

Reentrant Code

In cases where the global segment (G in the above diagrams)
occupies virtual addresses 24-32K, the global segment can
hold executable instructions.

This feature can be used to implement up to 8K of re-entrant
(sharable) code. To facilitate the writing of re-entrant
code, the first 256 words of the virtual shared space (addresses
24K to 24K + 256) refer to virtual addresses 0-256 within the
task's own address space, not the physical shared space.

By mapping the first 256 words back to the tasks own space
we have a mechanism whereby the sharable code can access,
through indirect addresses, the parameters, working storage,
etc. which reentrancy requires be kept in a task's own local.
memory. The technique on XVM is to use these 256 words to
hold indirect addresses which point to task-dependent para-
meters and working storage.

XVM/RSX Accounting Timer

This feature of the XM15 option measures, under XVM/RSX,
time spent in user mode and allows an installation to meter
not only task central processor usage, but also central pro-
cessor usage by those handler tasks which run in user mode.
This information on central processor use permits the know-
ledgable installation to administer and tune XVM/RSX for the
most productive service to the installation's user community.

PDP-9 TO XVM UPGRADE

You have a substantial investment in your PDP-9 system:

- purchase price of the system
- application software development
- special interfaces
- training and general familiarization

If you want to expand the productivity of your PDP9 facility, consider the possibility of an upgrade. An XVM upgrade is the best way to modernize your operation and protect your big investment because it will provide:

- Easy memory expansion to many times the size of your PDP9.
- More up-to-date software and hardware technology for increased uptime and decreased costs.
- Multi-user capability and other hardware and software extensions to increase machine productivity.
- Preservation of your current investments in both software and hardware.
- Minimum time and disruption for conversion.

The purchase price of an XVM upgrade consisting of

- XV100-A/-B   CPU package
- DW15-A   I/O bus converter
- XVM/RSX   multi-tasking executive
- XVM/DOS   disk operating system

is only slightly more than the list price of expanding your PDP-9 from 8K to 24K. And the XV100 comes standard with:

- XM15 memory processor
- 32K MF15 memory
- KE15 extended arithmetic element
- KW15 real-time clock
- PC15 reader/punch
- LA36 DECwriter II console terminal
- KF15 powerfail
- 3 training credits (spend them any way you like)

XVM is a superset of the PDP-9. It has all the PDP-9's instructions and features, and supports nearly every PDP-9 peripheral device but in addition, XVM removes the limits to growth. An XVM upgrade will allow you to phase-in use of the XVM features which are unavailable for the PDP-9:

- XVM/RSX for modern, economical multi-tasking operation
- FP15 hardware floating point for even faster computing
- Unichannel as 1.2M-word disk, spooler, and interfaceable Unibus
- memory to 128K at MF15 prices
- GT15 high-performance graphics
- RP152 10M-word disk
- BOSS option to DOS
- built-in DMA I/O processor for low-cost,high-performance interfaces

- dual or multi-processor operation
- built-in indexing

And you can phase them in because XVM will run any version of ADSS or B/F, or DOS version 3B (or earlier)--all without change. Since nearly every PDP-9 peripheral can be reinstalled on XVM, production work can continue during the transition to XVM/RSX.

Hardware Compatibility

The XV100 central processor discussed replaces these PDP-9 components in a software-compatible fashion:

- PDP-9 CPU
- MM09 memory and KG09 memory extension control
- KE09 extended arithmetic element
- KF09 automatic priority interrupt
- KX09 memory protect
- KP09 power fail
- PC09 reader/punch

In addition the following PDP-9 I/O devices can be reinstalled on the XVM system without modification and are fully supported by XVM/RSX and XVM/DOS:

- RF09/RS09 fixed-head disk
- TC02/TU55 DECtape
- TC59-D/TU10, 20, 30 magtape (7-or 9-track)
- CR03B card reader
- VP09 storage tube display
- LT19 teletype multiplexor
- LT09 teletype interface
- 350C plotter

Other DIGITAL devices are hardware-compatible but are not supported by XVM software:

- AA01 D/A
- AA05B D/A
- AF01B A/D
- AF02B A/D
- AF03B A/D
- AF04B A/D
- DP09A dataphone interface
- 647D 300 lpm printer
- 647E 600 lpm printer
- DB99A PDP-9/PDP-9 link
- DB98 PDP-9/PDP-8 link
- DB97 PDP-9/PDP-7 link
- DR09A relay buffer
- 34H display (control must be replaced with VP15)

Handlers for many of these devices using older versions
of RSX or DOS are available from DECUS or can be located
through the SIG-18 newsletter.

Most user-built interfaces and peripherals are similarly
hardware-compatible but not supported by XVM software.
The exceptions are those devices interfaced to PDP-9 memory
through a DM09A direct memory access multiplexor. These
are convertible as discussed subsequently.

Certain I/O devices were interfaced through DM09A DMA multi-
plexor and are not compatible with XVM:

- RM09 magnetic drum
- 339 display and options
- 340C display and options
- most customer DMA interfaces

These can, however, be converted by the user to run on XVM
by one of two methods:

1. Construction of a DM09A to XVM I/O bus adapter
2. Modification of the device interface to adapt
   it directly to the XVM I/O bus

The first method is recommended for converting two or more
devices when minimum cost is mandatory. The second method
is technically superior in that it allows use of the XVM
I/O Processor, but the cost is higher if two or more devices
must be converted.

What to Order

To upgrade to XVM you should purchase

- XV100 CPU package
- DW15-A I/O bus converter
- XVM/RSX and XVM/DOS

Trade-in allowances are available but vary with configuration.
Contact PDP-15/XVM Marketing for further information. Instead
of trading in the system you may wish to consider connecting
it to your XVM using a serial interface, or using it as a
stand-alone paper tape system for running COMPACT or FOCAL.

XVM UNICHANNEL PERIPHERAL PROCESSOR

This hardware/software product is a spooled input/output processor for PDP-15 and XVM systems. Spooling is a technique of increasing system throughput by buffering the data coming from or going to lower-speed peripherals on disk so that the XVM central processor runs at disk speed.

With Unichannel, the system can print concurrently on a line printer, read cards, plot on an incremental plotter, and read/write files on the Unichannel disk on behalf of four different jobs under either XVM/RSX or XVM/DOS. This can increase total system throughput by as much as 40%.

Unichannel also functions as a cartridge disk subsystem. Many users purchase it as their first disk without line printer or other peripherals attached. It is available for both PDP-15 and XVM systems. A particularly attractive offering is the RK15-LE/-LF which combines:

- XM15-UJ/-UK memory processor with 32K MF15 memory
- PDP-11/10 with 8K of MM11 memory
- RK11E/RK05 cartridge disk control and 1.2M-word drive

This equipment will upgrade any ADSS or B/F software-configured PDP-15 to XVM-200 system specifications.

Unichannel Hardware Architecture

The conceptual architecture of Unichannel hardware is shown below.
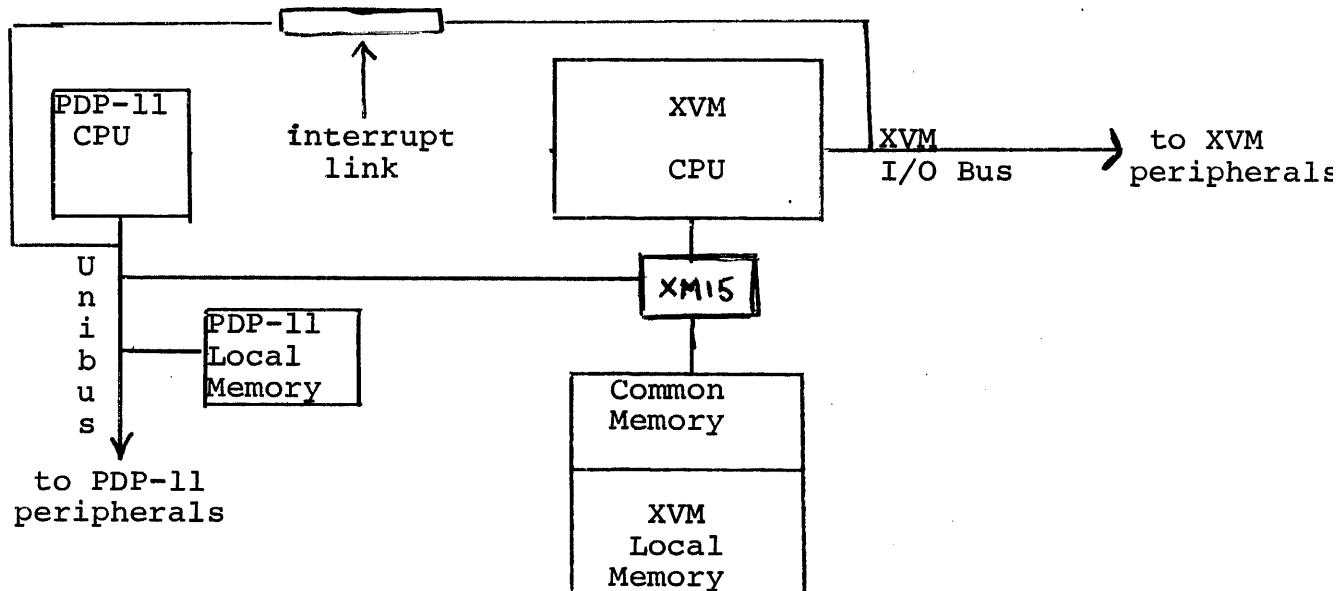
The systems are joined first by an interrupt link which
allows either CPU to interrupt the other, and second
through sharing of common memory. Common memory is 18
bits wide and the RK11-E Unichannel disk controller reads
and writes 18-bit data when transferring into or out of
common memory. This allows Unichannel to be used as a
system disk or file disk for the XVM system.

Common memory appears both on the memory bus of the XVM
CPU beginning at XVM address 0, and on the Unibus of the
PDP-11/10 CPU where it begins at the first address after
PDP-11 local memory (8K or 12K). As a result, the Uni-
channel can be operated as a stand-alone 28K PDP-11/10.
Contact PDP-15/XVM Marketing for information on how to
generate system disk cartridges for this mode of operation.

Unichannel Software Architecture

The Unichannel is driven by a PDP-11/10-based executive
called PIREX (PerIpheRal EXecutive) which is block-diagram-
med below:
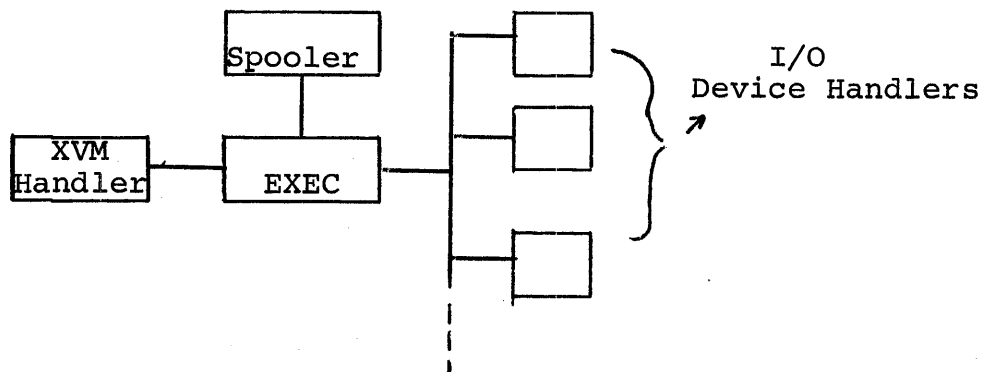
The executive is a multi-tasking monitor for the PDP-11/10
which interfaces identically through its XVM handler to
either XVM/RSX or XVM/DOS. This allows the XVM central
processor to be switched from either operating system to
the other without disturbing Unichannel operation.

PIREX and XVM/RSX or XVM/DOS synchronize their activities
by means of control blocks and data buffers maintained in
common memory. The requesting central processor fills out
fields in a control block and interrupts the other central
processor which decodes and executes the request.

When spooling is disabled (or if the spooler has not been
configured into PIREX at system generation time, data is
routed directly between common memory and the PIREX device
handlers. When spooling is enabled, data is buffered first
on Unichannel disk as fast as PIREX received it; simultaneous-
ly, it is de-spooled from disk at the speed of the destination
device.

If an output device is switched off-line or otherwise un-
available, the output data accumulates on disk till the
destination device becomes available. This allows the sys-
tem to run unattended during non-working hours while defer-
ring printing or plotting till an operator will be available
to tend to the paper supply. While the actual printing takes
place, the XVM can be switched to an unrelated use. For ex-
ample, many installations run batch computation or software
development jobs at night under XVM/DOS, switch to XVM/RSX
at the start of normal working hours and print the night's
output in parallel with the day's activities.

PIREX Device Support

PIREX supports the following I/O devices under XVM/RSX and
XVM/DOS:

- RK11-E/RK05 cartridge disk
- LP11 line printer (various models)
- LV11 (printer only)
- LS11 line printer
- CR11 card reader
- XY11 and XY311 incremental plotters (various models)

Available as separately priced software options for XVM/DOS
are:

- Quickscan (LV11 printer-plotter) for snapshot
  hard-copy of GT15 images
- 10/15 communications link for coupling XVM/DOS
  to DECsystem-10

Contact PDP-15/XVM Marketing for further information on these
XVM/DOS software options.

Custom Unichannel Configurations

The principal reasons for utilizing customized Unichannel
configurations include:

- use of PDP-11 peripherals not supported by PIREX.
- need to operate PDP-11 stand-alone as well as in
  conjunction with XVM/RSX or XVM/DOS.
- requirement for Unibus data rates in the range
  100-500 khz.
- requirement for more than 12K of local memory and/or
  more than 20K of common memory.
- requirement for sharing Unichannel among several XVM
  systems.

To meet these needs, five types of customization are possible. PDP-15/XVM Marketing should be consulted prior to placing any such order to verify technical feasibility, pricing, and delivery. These customizations are:
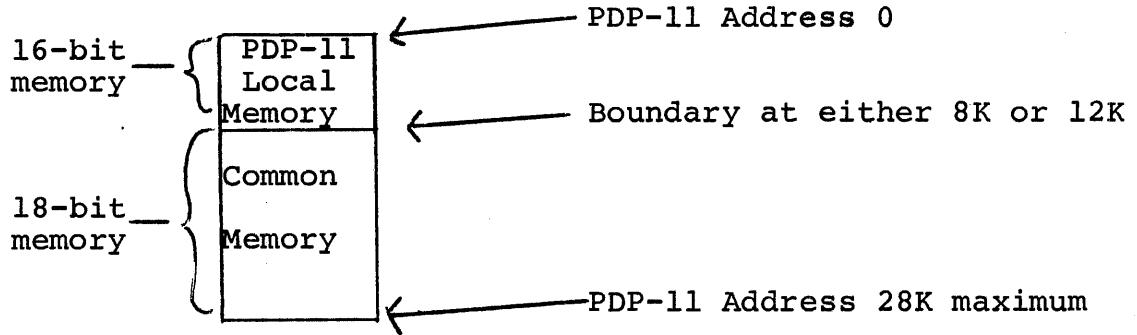
1. Use of standard 11/10 CPU and PIREX, but addition of standard Unibus peripherals or customer-built interfaces not supported by PIREX. Here the user simply writes a PIREX handler for the device in question. For more information, contact PDP-15/ XVM Marketing and request Unichannel-15 System Software Manual.

2. Larger-than-standard configurations for separate usage as both:

   * standard 8K or 12K Unichannel with standard PIREX peripherals
   * stand-alone 28K PDP-11 running some stand- alone PDP-11 operating system (e.g. RSX-11M or RT11) and using more peripherals than a Unichannel normally has.

Here PDP-15/XVM Marketing should be consulted on how to build the stand-alone PDP-11 software system on the 18-bit disk of Unichannel. Since standard PDP-11 disk cartridges cannot be read on Unichannel, the distribution medium must be PDP-15 or XVM 9-track magtape, PDP-11 DECtape on Uni- channel, or PDP-11 Floppy Disk.
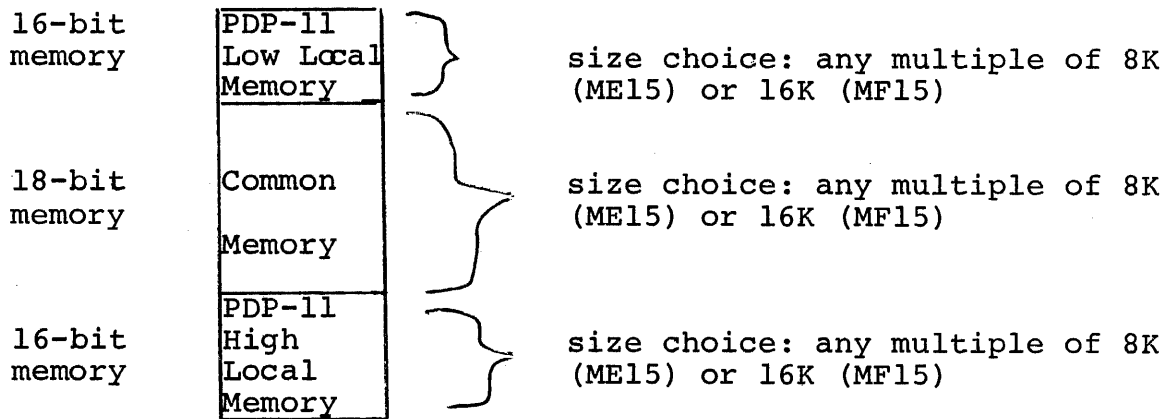
3. Use of PDP-11 central processors that have higher performance than the PDP-11/10 (requires XVM). This increases the available Unibus data bandwidth since the higher-performance PDP-11's arbitrate the Unibus faster.

In general, PDP-15/XVM Marketing should be consulted whenever the application requires Unibus I/O traffic peaking at 100 khz or more including disk transfers. Through suitable selection of PDP-11 central processor, local memory type, and programming strategies, this bandwidth limit can be raised to roughly 500 khz to/from common memory and roughly 2,000 khz to/from PDP-11 local memory.

4. Use of PDP-11 central processors having memory management (requires XVM). This offers many useful and interesting possibilities for the laboratory and classroom. Without memory management, the PDP-11's address space looks like this:
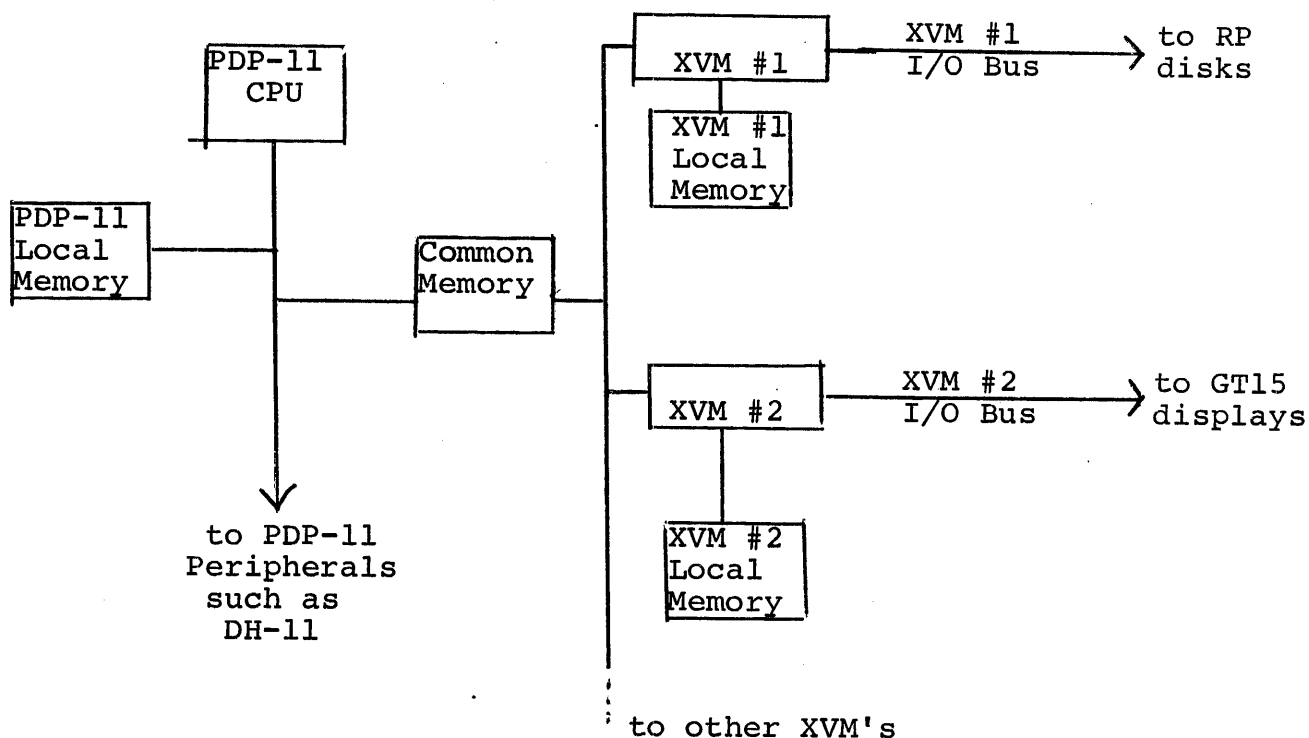
```
16-bit __  {  ┌─────────┐ <─────────── PDP-11 Address 0
memory       │ PDP-11  │
             │ Local   │
             │ Memory  │ <─────────── Boundary at either 8K or 12K
             ├─────────┤
             │ Common  │
18-bit __  { │         │
memory       │ Memory  │
             │         │ <───────────PDP-11 Address 28K maximum
             └─────────┘
```

Use of a larger PDP-11 having memory management removes
several constraints and allows the PDP-11 physical address
space to look like this:

```
16-bit     ┌─────────┐  }
memory     │PDP-11   │
           │Low Local│            size choice: any multiple of 8K
           │Memory   │            (ME15) or 16K (MF15)
           ├─────────┤
18-bit     │Common   │            size choice: any multiple of 8K
memory     │         │            (ME15) or 16K (MF15)
           │Memory   │
           ├─────────┤
           │PDP-11   │
16-bit     │High     │            size choice: any multiple of 8K
memory     │Local    │            (ME15) or 16K (MF15)
           │Memory   │
           └─────────┘
```

That is, the size of local memory can be configured over a
broad range; the starting address and size of common memory
can be similarly varied; and the balance of the central pro-
cessor's physical address space (124K maximum rather than
28K maximum as with the PDP-11/10 can be reassigned to Unibus
memory private to the PDP-11 (PDP-11 high local memory). In
turn, the PDP-11 memory management registers allow PDP-11
virtual addresses to be mapped onto physical addresses in a
variety of ways.

For such configurations, it is recommended that RSX-11M or RSX-11D be purchased and interfaced by the customer installation to XVM/RSX or XVM/DOS. Both RSX-11M and RSX-11D support PDP-11 memory management and a comprehensive set of PDP-11 peripherals.

5.   Use of PDP-11's with multiple XVM central processor. This allows configurations such as:

```
  ┌──────────┐                              ┌──────────┐   XVM #1      ⟩ to RP
  │PDP-11    │                              │ XVM #1   │   I/O Bus     ⟩ disks
  │CPU       │                              └──────────┘
  └──────────┘                                 ┌──────────┐
                                               │XVM #1    │
┌──────────┐                                   │Local     │
│PDP-11    │          ┌──────────┐             │Memory    │
│Local     │──────────│Common    │             └──────────┘
│Memory    │          │Memory    │
└──────────┘          └──────────┘          ┌──────────┐   XVM #2      ⟩ to GT15
                                            │ XVM #2   │   I/O Bus     ⟩ displays
         │                                  └──────────┘
         ▼                                     ┌──────────┐
     to PDP-11                                 │XVM #2    │
     Peripherals                               │Local     │
     such as                                   │Memory    │
      DH-11                                     └──────────┘

                                            to other XVM's
```

In this example XVM #1 serves as a disk-based multi-task computation system; XVM #2 serves as a controller for multiple GT15 interactive displays and joins these to the computational system, and the PDP-11 serves as a terminal concentrator and line printer and plotting subsystems.

UPGRADE TO XVM/RSX

Users of the ADSS and DOS PDP-15 operating systems should consider seriously an upgrade to XVM/RSX, even if an XVM hardware upgrade isn't planned for the immediate or near future. The major reasons are:

1. **XVM/RSX is our most productive system.**

   The process of adapting the computer to meet unique project or installation requirements calls for people, time and money. XVM/RSX allows more of these resources to be spent directly on productive activities because it greatly decreases the amount of time which must be spent adapting the computer.

   XVM/RSX is a resource-sharing executive and as such, directly boosts the productivity of the computer itself.

2. **XVM/RSX is PDP-15 Compatible.**

   XVM/RSX will run on PDP-15's having the prerequisite hardware. This is a significant advantage to the user who wishes to transfer to XVM/RSX prior to the purchase of XVM hardware.

3. **XVM/RSX is easy to use.**

   XVM/RSX has a more defined and useful architecture than previous software systems. For most installations, it is actually easier to use once the fundamentals have been learned. Its principal advantages are:

   - each task can address up to 32K without XVM (and more with XVM). In ADSS and DOS the total system (exec & application & background partition) was limited to a maximum of 32K.

   - the exec provides a number of useful services (task scheduling; storage allocation; resource sharing) not provided by ADSS or DOS.

4. **XVM/RSX is easy to extend.**

   Because of its excellent modularity and thorough documentation both in manuals and listings, new device handlers or software functions can be integrated into the system easily, without creating or encounting side effects.

5. <u>XVM/RSX is reliable and easy to maintain</u>.

    In spite of its power and functionality, XVM/RSX has
    the lowest bug rate per using installation of any
    of the 18-bit software systems. The modularity and
    isolation of functions tends not only to hold down
    the bug rate but also to make repairs easy to design,
    code and install. As a result XVM/RSX almost completely
    eliminates situations in which it was inadvisable
    to intergrate the latest DIGITAL software release
    with an installation's own version of a system. With
    XVM/RSX every installtion will now find it feasible
    to integrate the latest releases, thereby benefiting
    from improvements in reliability and features/benefits.

6. <u>Development emphasis will be on XVM/RSX</u>.

    XVM/RSX's ease of programming, modification, and
    support indicates a strong emphasis by DIGITAL on
    the further development and improvement of this
    operating system. Extended features, hardware devices,
    human use factors, and sales emphasis will receive
    considerable attention in 18-bit product development.

7. <u>Support for older software is being phased out</u>.

    The final updates to DOS V3 and RSX-PLUS III have
    already been made and distributed. A final release
    of ADSS and B/F will be made, probably during the
    Spring of 1976.

8. <u>Special offer expires 30 June, 1976</u>.

    Until that date, customers who have purchased soft-
    ware licenses or who obtained older systems with
    their computers can qualify for a special discount
    certificate good toward the purchase of any XVM
    software product.

    Application for the certificate is made by com-
    pleting an information form which will be mailed
    to all 18-bit users. The form should be returned to:

              Mike McCarthy
              XVM Product Manager
              Digital Equipment Corp.
              200 Forest Street
              Marlboro, Mass.   01752
              USA

    A discount certificate will be sent to you following
    receipt of the form. The certificate should accompany
    your software purchase order submitted to your Digital
    sales representative.

The table below expresses software prices in arbitary units (actual prices vary from country to country and are subject to change without notice). Only a written quotation from your local Digital sales representative expressed in local currency units consitutes a price committment by Digital.

	XVM/RSX Plus XVM/DOS	XVM/MUMPS Plus XVM/DOS	XVM/DOS Only
Product List Price in arbitary units	120	120	50
Service Price (discount cerfificates not applicable)	38	30	25
Maximum Net Price	158	150	75

The next table shows the number of credits in arbitary
units as a fraction of software previously purchased
(product credits) and of Plan B subscription service
(service credits). Please note that:

- product and service credits apply only to
  product purchases and not to software services

- to qualify for service credits you must be
  a Plan B or Software Update Service subscriber
  as of 27 December 1975 (if your subscription
  lapses before then you must renew   if you
  are not now a subscriber you must subscribe)

- the minimum purchase price of any XVM software
  product is 10 arbitary units no matter what
  the credit tables seem to indicate. You are
  not entitled to a rebate nor may you receive
  a product at no charge.

Product previously purchased	Product Credits	Service Credits	Total Credits
RSX-PLUS III, DOS Source	70	40	110
RSX-PLUS III, DOS Binary	60	40	100
RSX-PLUS	40	N/A	40
RSX-15	20	N/A	20
DOS V3B Source	30	20	50
DOS V3B Binary	20	20	40
DOS V3A Source	20	N/A	20
DOS V3A Binary	20	N/A	20
DOS V2 or Older	10	N/A	10
ADSS, B/F	20	10	30
ADSS	10	10	20
MUMPS V11	70	40	110
MUMPS V10	60	N/A	60
MUMPS V9 or Older	20	N/A	20

INTRODUCTION TO XVM SOFTWARE

The XVM/RSX, XVM/DOS and XVM/MUMPS software systems are management tools for maximizing the productivity of XVM and PDP-15 customer installations.

Advances in science, engineering, and production are facilitated by the availability of a powerful computer. Yet each round of advances carries an implication of computer application development, testing, installation, maintenance, and evolution.

XVM software systems are designed to get an application into production use quickly, and to maximize the amount of work the computer can do.

Productivity gains are realized in four ways:

1. Eliminating unnecessary software development. The DEC-supplied XVM software systems usually make it unnecessary for the user organization to develop its own operating system. This frees people and machine time, permitting the organization to focus on development of XVM applications rather than on XVM itself.

2. Facilitating application development. All XVM software systems contain application development aids-- software tools and procedures--which reduce the amount of effort required to develop an application. These take the form of language processors (both problem-oriented and machine-oriented) for developing code, run-time operating system services, and utility programs.

3. Maximizing system availability. XVM software systems contain features for maximizing access to production computer time through a mixture of strategies: Maximizing system uptime (XVM/RSX; XVM/MUMPS); sharing of system resources among a community of users (XVM/RSX; XVM/MUMPS); simplicity of use (XVM/MUMPS; XVM/DOS).

4. Maximizing system throughput. All XVM software systems allow the addition of hardware options which increase the amount of production work per unit time the XVM application system can accomplish. Reprogramming to take advantage of these options in general is not required.

XVM/MUMPS

XVM/MUMPS is a special-purpose timesharing system dedicated to on-line, interactive management of business or administrative data. It contains facilities for:

- entering alphanumeric information from keyboard terminals into the online data base.
- organizing the data base to reflect relationships among the data.
- integrating new information into that organization.
- interactive querying of the data base.
- performing calculations using elements of the data base as inputs and outputs.
- producing reports and extract runs both interactively and on system line printers or magtape.
- backing up the data base on magnetic tape or disk.

XVM/MUMPS contains a number of data security and data integrity features which make it ideal for high-volume production environments in which system availability is a primary concern.

XVM/RSX

XVM/RSX is a multi-purpose system for delivering simultaneous computing services to a diverse set of on-line users and applications. It can handle a mix of:

- single/multi-station graphic design automation.
- single/multi-terminal access to application programs.
- on-line software development.
- real-time data collection/device control.
- general purpose interactive computation.
- general purpose batch computation.

XVM/RSX can be used cooperatively with XVM/DOS. In fact, many installations operate under XVM/RSX during the day for on-line multi-user/multi-tasking operation, then switching to XVM/DOS for unattended batch processing of offline work at night.

XVM/DOS

XVM/DOS is a single-user system intended for general purpose computation, for software development, and as the foundation of certain dedicated XVM application systems.

The system has throughput maximization as its goal and it is most effective when operated under its batch processing option. XVM/DOS can concentrate all its resources on each job as it flows through the system. This permits:

- minimal job set-up time.
- automatic system control (hands-off operation).
- Unichannel spooling of card reader, line printer and plotter I/O.

XVM/DOS is used to assemble, configure, extend, and maintain XVM/RSX and XVM/MUMPS as well as XVM/DOS itself.

XVM/RSX and XVM/DOS function cooperatively in that:

- they can be co-resident on system bulk storage.
- files generated on either system can be read by the other.
- crossover commands exist for switching between systems.
- I/O spooling is not interrupted by system crossover.
- XVM/DOS can be used to develop XVM/RSX software.

HOW TO ORDER XVM SOFTWARE

Use the table below to select order numbers for the software
products and services you want. Then check the table of
"Minimum and Maximum Hardware Support" on page to be sure
you have all the hardware prerequisites. Check prevailing
prices with your local DIGITAL sales office and enclose the
necessary software source license agreements (signed by your
purchasing agent) with your purchase order.

Until 30 June 1976 customers of older 18-bit software products
are eligible for discounts toward the purchase of XVM software
products. See "Upgrade to XVM/RSX" for more information.

Software Products

Item	Order Number	Prerequisites	Description
XVM/DOS	QM060-YC	See SPD	XVM/DOS sources on DECtape; 2 training credits
	QM060-YD	See SPD	As above on 9-track magtape
	QM060-YF	See SPD	As above on 7-track magtape
XVM/RSX	QM070-YC	QM060-YC Plus See SPD	XVM/RSX sources on DECtape; 2 training credits
	QM070-YD	QM060-YD Plus See SPD	As above on 9-track magtape
	QM070-YF	QM060-YF Plus See SPD	As above on 7-track magtape
XVM/MUMPS	QM810-YC	QM060-YC Plus See SPD	XVM/MUMPS sources on DECtape
	QM810-YD	QM060-YD Plus See SPD	As above on 7-track magtape
	QM810-YF	QM060-YF Plus See SPD	As above on 7-track magtape

Software Maintenance Services

Item	Order Number	Prerequisites	Description
Software Information Service - all systems	Qm060-2Z	None	Monthly software dispatch - one-year sub-scription
Software Update Service for XVM/DOS	QM060-4C -4D -4F	QM060-YC -YD -YF	QM060-2Z and SPR's plus one source update to XVM/DOS on specified medi
Software Update Service for XVM/RSX	QM070-4C -4D -4F	QM060-4C -4D -4F	QM070-YC,YD,YF, QM060-2Z, Plus one source up-date to XVM/RSX on specified medium
Software Update Service for XVM/MUMPS	QM810-4C -4D -4F	QM810-YC -YD -YF	QM060-2Z and SPR's plus one source update to XVM/MUMPS on specified medium

INTRODUCTION TO XVM/RSX *

Uses of XVM/RSX

XVM/RSX is DIGITAL's software system recommended for maximizing productivity in many single-user and all multi-user/multi-processing applications. This Resource-Sharing Executive is especially suited for:

- laboratory data collection and reduction
- real-time process monitoring and control
- multi-scope interactive graphics/computer-aided design
- multi-user computing
- general-purpose batch computation/software development
- real-time command and control
- computer science research

XVM/RSX functions cooperatively with its prerequisite XVM/DOS and, while these are separate software systems, every XVM/RSX installation can be designed to take maximum advantage of both.

The system offers substantial expandability both in terms of user extensions or modifications for broader application, and in terms of Digital-supplied hardware for increased system throughput and capacity.

System Goals

The principal design goals of XVM/RSX are:

- rapid development of customer applications through
  - development of applications during production use of the system
  - choice of high-level or machine-oriented programming languages
  - variety of labor-saving executive service functions and utility programs

---

- maximization of both system and application avail-
  ability through provision of interlocks which, in
  the event of user program or system malfunction
  - guard the executive against damage caused
    by entrant user programs
  - guard other user programs against being
    similarly damaged
  - allow user programs themselves to sense and
    respond to a variety of transient and per-
    sistent error or alarm conditions

- high performance options which allow
  - time/space software tradeoffs
  - price/performance hardware tradeoffs
  - smooth capacity growth for flexible
    user budgeting
  - easy expansion into new application areas

- flexible techniques of system control and operation
  - full control by system operator
  - limited control by terminal users
  - automatic control by the exec and batch processor
  - blended control modes

- modular, multi-task architecture
  - isolation of functions (exec, handlers, etc.)
  - central processor scheduling strategies
    optimized for real-time
  - flexible allocation of core and disk space

XVM/RSX Application Development Features

XVM/RSX has a range of utility programs for addressing the
process of software development. The following diagram shows
each step in the process with a partial list of the utility
tasks supplied by DIGITAL as part of XVM/RSX which typically
are used at the indicated steps.

Because XVM/RSX is interactive, and because task development
can process during production use of the system, productivity
of both software developer and the computer are significantly
improved beyond more traditional card-oriented or single-user
systems. The diagram illustrates the iterative development
process.

```
                        ┌──────────┐
                        │   EDIT   │      EDIT      sources programs typed in
                        └──────────┘                from console terminal and
                                                     recorded in files on disk


FORTRAN ──────→   ┌──────────┐          ┌──────────┐  ← MACRO - runs at
runs at           │ COMPILER │          │ ASSEMBLE │
disk speed        └──────────┘          └──────────┘


TASK BUILDER →    ┌──────────┐
accepts reloc-    │  BUILD   │
atable modules    │   TASK   │
from compiler,    └──────────┘
assembler or
library - runs
at disk speed

                  ┌──────────┐
                  │ LOAD  &  │      MCR
                  │ EXECUTE  │      EXEC
                  └──────────┘      I/O Handlers


                  ┌──────────┐
                  │ TEST AND │      ODT - octal debugger
                  │  DEBUG   │
                  └──────────┘



                                  minor problem


                                  modify program in core
                                  with ODT



                                  major problem or

                                  accumulation of minor
                                  ones - fix source code
```

In addition to EDIT, FORTRAN, MACRO, TKB, and ODT shown
above, XVM/RSX includes other utility tasks, notably:

- FIN, FOU, etc. These are used for duplicating
  or moving files on disk, or for exchanging/mov-
  ing files on disk, or for exchanging/moving files
  between disk and media such as magtape or paper
  tape.
- SLIP--Source Language Information Processor.
  Card-oriented text editor especially well-suited
  for initial entry via XVM/RSX batch source pro-
  grams or alphanumeric data prepared off-line on
  cards.
- BTK--Batch Task Builder. A simplified subset of
  TKB which prepares tasks for execution under
  XVM/RSX batch.
- VTPRIM--VT15 Primitives. A library of FORTRAN-
  callable modules for drawing lines, points, and
  text with the interactive VT15 graphics processor.
  XVM/RSX will support up to four graphics work-
  stations for separate tasks.

In addition to utility programs and utility libraries, XVM/
RSX supplies a well-structured interface between user tasks
and the system EXECutive. A partial list of system services
includes:

- Inter-task communication
- Opening and closing files on disk
- Reading and writing data
- Calling-in overlays
- Scheduling one's own or other tasks for future
  re-execution by time-of-day, time-from-now, or
  upon occurence of one or more events
- Allocating and deallocating non-sharable resources
- Sensing various system events or alarm conditions

These and other directives increase installation development
productivity. First, by supplying in the RSX product itself
many commonly needed functions, the amount of new code that
must be written to do a job is decreased. Second, the use of
system directives tends to standardize both code and coding
techniques; this means that customer-developed or DIGITAL-
developed code can be largely or partially re-used in new
applications which are similar or related to their predecessors.

A third productivity-increasing technique is to use XVM/RSX as
a multi-terminal development or computational facility. Sev-
eral people can use the system concurrently to edit and submit
jobs to the batch processor of XVM/RSX. This allows remote
users to have tasks compiled, assembled, built, requested, and
so on.

XVM/RSX System Availability Features

XVM/RSX has several features which tend to maximize system availability. This means that malfunctions are either averted; detected and corrected; or reduced in their consequences. In particular, the EXEC embodies several strategies which, in priority order:

- defend the EXEC against damage by user tasks
- defend user tasks against damage by other user tasks
- defend user tasks against transient hardware malfunctions

In addition, if damage is incurred unavoidably, facilities exist for restoring damaged data and program files, for refreshing entire disk packs, etc.

In general, we can break availability into three constituent levels of safeguards:

- Security—limiting unauthorized access to memory, I/O devices, or files (prevention)
- Integrity—verifying the correctness of the I/O transfers, and of already-recorded data (detection)
- Recovery—repairing damage sensed by integrity checks (correction)

Prevention. In effect, XVM/RSX erects a firewall around each user task, preventing unauthorized access to:

- EXEC memory space
- Memory space of other tasks
- Unsharable I/O devices not reserved to the task
- Logical Unit Numbers (LUN's) not attached by the task

The first two preventive measures are implemented with memory protection hardware (see XM15, page 44 , and KS15, page 44) which validity checks the address of every task-generated memory reference. Violations cause an immediate abort to the offending task.

The last preventative measure is the result of validity checking by the EXEC of all I/O requests. Violations here can be sensed by the offending task via error codes passed from the EXEC; the task can then take corrective measures, detailed later in the "Detection" section.

In certain situations, operators may wish to change user or
system disk packs/cartridges while the system continues to
operate. To guard against any corruption of disk data during
the switchover period, XVM/RSX contains a DISmount function
which insures that all task-requested I/O for that unit has
stopped, and that all files are closed, and updates tables
on the disk used by the system (e.g. bulk storage bit maps).

Detection. Transient hardware errors (such as disk read/
write parity errors) are sensed by the EXEC and retried until
a threshold count is exceeded. At this point, the handler
informs the requesting task of a permanent error and dequeues
the I/O request.

The requesting task is free to try the operation again or
(as might be the case with a truly persistent error on disk)
close the current file and open another one.

In the case of suspect DECtapes, disk packs, cartridges, etc.,
the RSX FIN utility can be used to verify the integrity of
individual user files; this can be done also under DOS using
the PIP utility program.

Recovery. As mentioned above, the EXEC plus DIGITAL-distributed
I/O handlers contain error correction logic which is adequate
for transient I/O errors (a low occurrence rate is considered
normal). In the event of persistent errors (e.g. due to defects
in magnetic oxide coatings which wear-out over time), the using
installation usually has no choice but to reconstruct the in-
formation contained on the bad pack onto a good pack. This
reconstruction process is greatly simplified by the PIP utility
task, which can copy the good files over to a fresh pack.

Most installations periodically save an image of the entire
contents of their system and file packs onto backup media (mag-
tapes or other packs) so that, should a persistent error be
encountered, the entire new pack can be quickly refreshed by
DOSSAV without the operator having to   specify each file to
be transferred.

System Maintenance

A variety of system maintenance aids and services are available
for XVM/RSX systems.

As an aid to verifying the integrity of the CPU, memory,
I/O devices, the EXEC, and I/O handlers, XVM/RSX comes
with a Checkout Package (COP) which exercises the system
by creating a dummy user software load. COP operates I/O
devices singly and in combination.

XVM/RSX Performance Features

Certain XVM/RSX performance features involve software issues
such as time-slicing (see page 46); others involve optional
hardware. This section discusses how XVM/RSX makes use of
hardware options and the trade-offs in having, versus not
having any particular option.

XVM System Speed-ups

The XVM memory processor hardware option
is particularly effective on XVM/RSX systems and provides
several direct speed-ups beyond equivalent PDP-15 hardware.
In addition, XM15 provides new capabilities (wide addressing
and core sharing) which speed-up the system indirectly.

The direct speed-ups arise not only from instruction prefetch
and interleaved memory but also increased speed of operation
of memory protect/relocate and the interface to RK15 Unichannel.

With XVM hardware, an XVM/RSX system will operate at speeds
comparable to those of the original 800ns PDP-15. The addition
of memory protect/relocate or Unichannel to the PDP-15 resulted
in some performance loss. While substantially reducing the
price of RSX-configured systems and upgrades, XVM buys back
that performance.

Indirect speed-ups result from XVM addressing in three ways.
First, the wide addressing capability allows large arrays (such
as might be encountered in high-energy physics, crystallography,
and computer-aided design) to be retained in memory within the
address space of the application task. This makes the appli-
cation task run faster by reducing the number of disk accesses
required for bringing in data overlays. (In addition, the num-
ber of "passes" to be made over the data is reduced, thus cut-
ting down on central processor time, as well as disk I/O time.)

The second effect of wide addressing is a reduction in the num-
ber of code overlays that need to be retrieved when the situation
involves large amounts of code with relatively small amounts of
data (such as might be encountered in lens-design and other
complex problems involving lengthy sequences of calculations).

A third speed-up due to XVM addressing arises from core sharing (see page 5). Here intertask communication is facilitated by the ability to share as much as 8K of data among two or more tasks. Intertask communication via core sharing requires no routine executive intervention.

Core vs Disk Residency

Under **XVM/RSX**, the system operator can specify prior to execution whether a task is to be kept in memory or on disk. This decision doesn't affect the task's speed of execution (once in memory, it stays there till it has run to completion, though it will generally be interrupted). Rather, the decision affects how quickly the system can access the task in order to ready it for execution.

When specified to be memory-resident (FIXed) the task occupies its partition until forcibly cleared (UNFIXed) by the system operator. While this can result in idle memory, the system is guaranteed the fastest access to the task.

When disk-resident (default case), the task can be accessed for execution no faster than the combined access and transfer time of the system disk. As with the effect of XVM wide addressing speed-ups, the difference between core- and disk-residency must be measured in tens of milliseconds.

This choice of technique is up to the user installation. When disk speed is appropriate, a range of disk performances is available.

Interrupt Service Time

One of the prime goals of XVM/RSX is the conservation of central processor time needed to service time-critical functions such as I/O device completion interrupts. To facilitate meeting this goal, XVM/RSX encourages the division of all interrupt-related code (loosely termed "I/O handlers") into two sections:

- An Interrupt Service Routine (ISR). This section contains time-critical code (e.g. honoring a disk seek-complete interrupt). The minimum amount of code required to execute before the interrupt which triggered the ISR can be dismissed. When it completes, the ISR returns to the EXECutive but with a request for subsequent execution of the Handler Task proper, described below. An ISR can be interrupted only by an I/O device wired to a higher hardware priority level, i.e. only by a more important ISR.

- The Handler Task proper. The handler task contains the remainder of the interrupt-related code (e.g. decisions regarding which I/O request to issue next for the device in question and actually issuing that request). From the viewpoint of the EXEC, it is simply one of the several tasks in the system and has an Active Task List (ATL) priority which is not necessarily related to the hardware priority of the associated I/O device and ISR. This being the case, handler tasks can and will be interrupted by both ISR's and by any higher-priority task on the ATL.

Disk Types

XVM/RSX supports the full price/performance spectrum of XVM and PDP-15 disks, in any combination:

- RF15 disk control plus up to 8 RS09 fixed-head disk drives. This subsystem offers the fastest access of the three XVM disks. It is most effectively used strictly as a system device, in conjunction with one or both of the disks discussed below. Each drive holds 256K 18-bit words.

- RP15 control plus up to eight RP02 (new) or RPR02 (rebuilt) moving-head disk pack drives; each drive holds slightly more than 10,000K 18-bit words. Because the drives are dismountable, they are well-suited for holding user files, user programs, and less frequently used systems programs; however, a single RP drive can be both system and file device.

  For backup of information on RP02 packs, we recommend either an extra RP02/RPR02 drive (so that one pack can be copied onto another) or a TC59/TU10 magtape subsystem. Backup on magtape takes more time than disk-to-disk, but the price of a reel of magtape is lower than the price of spare RP packs.

- RK15 Unichannel plus up to eight RK05 moving-head cartridge disk drives; each drive holds approximately 1200K 18-bit words. In addition to functioning as a disk, Unichannel can dramatically increase system throughput by means of its spooling capabilities. Unichannel actually integrates a second computer into XVM and PDP-15 systems. This computer can be configured and programmed to extend its capabilities beyond those standard with XVM-200 systems. See page 12 for a discussion of both spooling and custom Unichannel systems. As mentioned above, XVM/RSX supports any mix of these three disk types. A particularly effective configuration is:

Powerful XVM/RSX Configuration

SPOOLING DISK;
LP11 AND XY11
OUTPUT TO HERE
BY MAIN SYSTEM
AND PDP-11 AT
DISK SPEED.

USER FILE DISK;
1200 K OF
REMOVABLE DISK
FOR SUPPORTING
APPLICATION
DEVELOPMENT

RK05

RK05

RK11

XY11

CR11

LP11

PDP-11
LOCAL
MEMORY

PDP-11
(UN-CHANNEL)
CPU

ACTS AS
I/O FRONT END
TO MAIN SYSTEM

INTERRUPT
LINK

COMMON
MEMORY
(PDP-11 AND
PDP-15)

PDP-15
LOCAL
MEMORY

XVM
PDP-15
CPU

MAIN SYSTEM,
HAS USER TASKS
AND DOES COMPUTATION

TOTAL OF 512K WORDS
HIGH-SPEED ACCESS
TO SYSTEM TASKS
AND FREQUENTLY
NEEDED APPLICATION
TASKS.

RS04

RS04

RF15

RP02

RP02

RP15

TC15

TU56

10,000K WORDS
(10M WORDS) FOR
STORING COLLECTED
AND REDUCED DATA,
LESS FREQUENTLY
NEEDED TASKS, ETC.

BACKUP FOR
FIRST RP02

LTA
D8
E's

USER
TERMINALS

FORTRAN Speed-ups

The FORTRAN system has two options for increasing execution speed.

The first is the FP15 floating point unit. Without floating point hardware, the FORTRAN compiler generates, for floating point operations, calls on sub-routines within the FORTRAN Object Time System (OTS). With FP15 available, the compiler instead generates in-line code which uses the FP15 directly. The resulting code yields a speed-up factor of ten in elementary FORTRAN operations (+, *, etc.); for programs overall, the speed-up is typically a factor of three to six. With FP15, the OTS is smaller (since it need not contain floating point subroutines); the resulting user program is smaller than its non-floating-point equivalent.

The second FORTRAN speed-up relates to subscript calculations. The compiler can be conditioned to generate either subroutine calls (which result in a compact object program) or in-line code (which yields a lengthier but faster-executing program). The in-line version of a typical two-dimensional matrix inversion program will usually be 15% larger but a factor of two to four faster than its subroutine equivalent.

These two speed-ups provide most of the benefits of a fully-optimizing compiler, but without the increased core and compile time requirements such compilers entail.

Multi-scope Interactive Graphics

Designer-machine interaction in complex problem situations is most effectively accomplished with high-performance interactive graphics. The GT15 display subsystem is a prestigious addition to XVM or PDP-15 systems.

XVM/RSX will support up to four VT04 or VT07 graphics stations in a configuration like this:

graphics stations
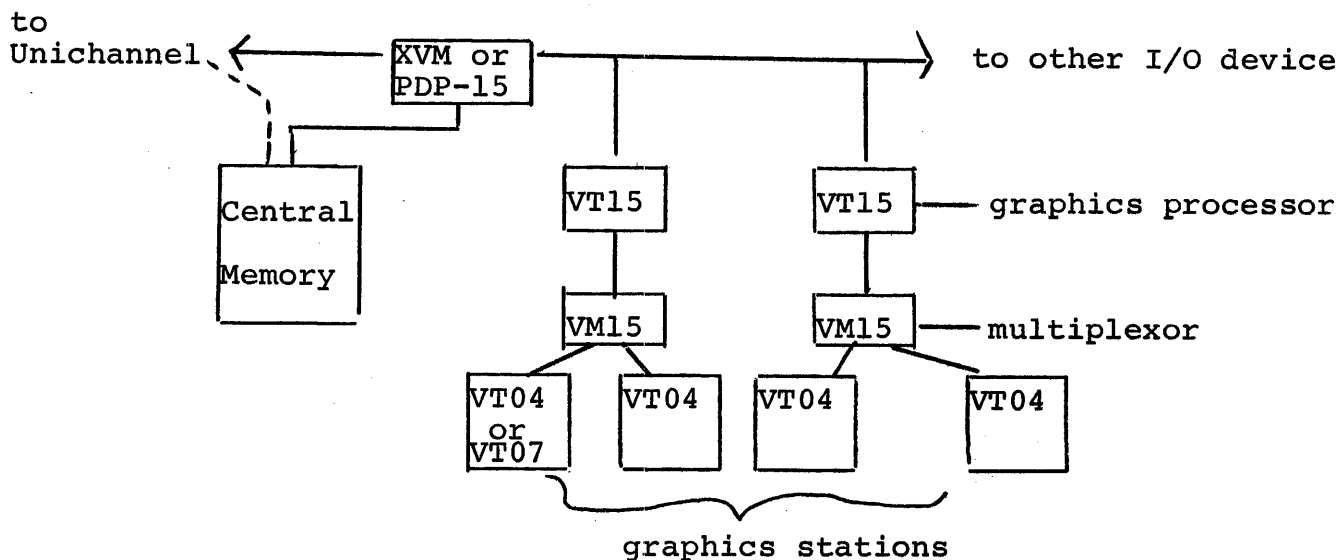
The display terminals are independently programmed from
four separate tasks. Each task uses its own copy of VTPRIM,
a set of FORTRAN-callable routines for drawing lines, text,
and other pictoral representations on a VT04 or VT07.

The configuration shown is recommended for multi-station
graphics; it provides an optimal balance between flicker-
free display and central processor access to memory cycles.
For specialized applications, other configuration types are
possible.

System Control Features

XVM/RSX allows several modes of system and task control
ranging from fully "manual" (operator commands) to fully
automatic (unattended system operation). Modes can be
blended so that some portions of the system are under oper-
ator control, while others are under the personal control
of users, and the remainder of the system is under auto-
matic control. The modes and their implementations are:

- Operator. Through the Monitor Console
  Routine (MCR) terminal, the system
  operator can issue a variety of direct-
  ives to the executive. Many of these
  are privileged functions unavailable
  to normal users or normal user tasks
  for reasons of system integrity. A
  partial list includes:

    - establishing or modifying number and
      size of partitions
    - installing tasks in partitions
    - modifying task priorities
    - modifying task execution criteria
    - regulating time-slicing
    - aborting tasks

The philosophy here is that the operator should be able to
exercise control over all matters which affect the behavior
of the total system--response times, throughput, relative
importance of tasks, etc. To assist in this control, the
MCR can supply the operator with printouts of various system
tables which collectively represent the state of the system
and its constituent tasks.

- User. In addition to system control via
  the MCR, applications software packages
  can be constructed that use specialized
  sets of commands designed for a specific
  end use.

XVM/RSX directives provide a convenient way for the application programmer to issue requests for user input of data, control parameters, etc., and to produce problem-oriented reports, messages, and so on.

Where the job at hand requires the coordinated application of a set of tasks (i.e. the process of developing software), XVM/RSX provides an extensible tool, the Command Dispatcher, which can present to one or more user terminals a fast means to invoke and execute a selected set of tasks. The TDV (Task DeVelopment) dispatcher within XVM/RSX is a good example of this facility. User installations frequently build their own dispatchers to address issues unique to their applications. Such dispatchers increase user productivity by cutting down on the amount of computer knowledge required to operate an application.

- Automatic. XVM/RSX offers two methods of automatic task control. The first is XVM/RSX Batch, which is intended for sequencing a usually unrelated set of low-priority, computation-oriented tasks through a single partition.

  The second method extends the concept of inter-task communication to include inter-task control, and allows an asynchronously operating collection of real-time tasks, often in separate partitions, to synchronize their activities.

- Mixed. A blended mode is also possible, since a batch task can invoke a real-time task. The converse is true and is one of XVM/RSX's most powerful features--a real-time data collection task for example, can leave its results in a file on disk and schedule a batch task for subsequent data reduction; the reduction task can run at a priority such that other real-time tasks get the service they require, while the reduction task uses machine time that might otherwise go to waste. Should the high-priority system task load decrease (e.g. as interactive users go to lunch, or as instruments are switched off-line), the amount of machine time available to the reduction task automatically is increased and it completes its work sooner.

- Batch. Under the Batch facility of XVM/RSX, task
  control commands come from files on disk, magtape,
  cards, etc., not from the MCR or user terminals.
  Using the EDIT task discussed earlier, the appli-
  cation programmer prerecords a dialogue of commands
  in a file; the task executes under the control of
  that file. The process requires no human attendance
  and runs at disk or magtape speed. The task in
  question can issue the kinds of commands discussed
  in the next paragraph.

- Intertask Control. With Intertask Control, an exe-
  cuting task (real-time, batch, or interactive) can
  instruct the EXEC regarding the execution of other
  tasks (again, real-time, batch, or interactive) by
  issuing directive for:

  - starting tasks
  - passing information to tasks
  - declaring events (on whose occurence other
    tasks conditionally will execute)

Multi-user Operation

This is an excellent example of quasi-automatic control
through a mixture of user, batch, and intertask control
facilities. In this mode, a developer interacts with the
user terminal, requesting a private copy of a utility like
the MACRO assembler. The request is queued to the Batch
processor by the terminal dispatcher, as are similar re-
quests by other users and subsequent requests by the same
user (e.g. for the Task Builder and then for execution of
the task).

When the Batch processor executes the MACRO assembler, it
obtains control statements from a Logical Unit (which can
be associated with a private or system UFD.) These state-
ments specify those user's files to be input to the assembler,
and which file is to receive the assembler output. Similarly,
when the Batch processor honors the user's queued request for
the Task Builder, it is automatically instructed regarding
which Logical Unit to use as input (e.g. one or more assembler
output or library relocatable binary files) and in which Log-
ical Unit to place the executable output.

The Batch processor has an input queue of tasks to be executed,
each having a Batch priority and other parameters. The Batch
processor can be conditioned to proceed through the list strictly
sequentially (analogous to pure pre-emptive scheduling), to favor
classes of tasks (i.e. those of interactive users, analogous to
time-slicing) or to defer execution of others until certain
criteria have been met. (Regarding the latter, data reduction
shouldn't start till data collection is finished and there are
no outstanding Batch requests by interactive users.

The selection of which job to run next involves looking at the
list of all jobs waiting to run and determining that job most
suitable for execution. Suitability is a function of resources
required by the job, estimated job run time, and operator system
control parameters.

Through the MCR, pre-emptive, time-slicing, and Batch priority
criteria can be tuned while the system runs to provide maximum
satisfaction to the user community in the face of a changing
of real-time, interactive, and batch system loading.

XVM/RSX Architecture

XVM/RSX is a multi-tasking, event-driven resource sharing oper-
ating system. Its major components and their functions are
illustrated subsequently. It is useful to think of XVM/RSX
as a collection of asynchronously operating software modules
(tasks) whose activities are synchronized by the executive in
much the same way a central processor synchronizes the acti-
vities of asynchronously operating I/O devices.
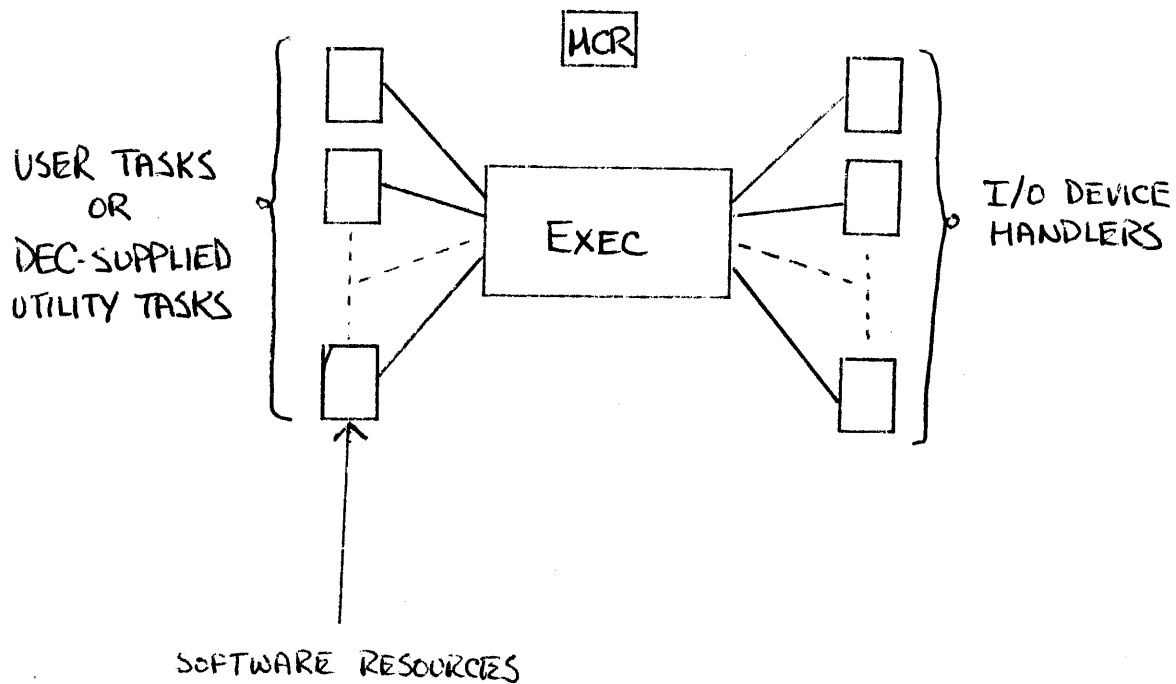
XVM/RSX Resource Allocation

A task is the basic computing entity within XVM/RSX. It is the smallest unit of execution capable of consuming or making a bid for system hardware resources. Like many useful application systems, XVM/RSX is a cooperating set of tasks.

Resource allocation generally is not performed through the MCR. Rather, the MCR is used to set limits, parameters, and other constraints relating to each active task.

The EXECutive actually implements resource allocation through a set of algorithms which are table-driven. The table parameter values are supplied or modified through the MCR; parameter values thus control the specific decisions made by the EXEC during the course of system execution.

The resource allocation strategies which the EXEC embodies are:

- Central processor time. Preemptive scheduling according to task priority. The EXEC has an Active Task List (ATL) which ranks in priority order all active tasks in which have been requested to run. Whenever the current task releases the central processor either voluntarily (e.g. to wait for I/O completion or some other event) or involuntarily (e.g. some kind of interrupt), the EXEC scans the ATL beginning with the first (highest-priority) task and selects for execution the first runnable task it encounters.

- I/O Device Time. This is scheduled pre-emptively according to the relative priorities of the issuing tasks. The EXEC inserts I/O requests for a given device into a queue which is scanned by the handler task for that device and executed in priority order. (Other I/O scheduling techniques can be implemented by the user).

- Central memory. This resource is reserved through the MCR by the system operator by dividing available memory into a variable number of partitions (contiguous regions of memory) of specifiable size. The EXEC does not control the size or number of partitions but it does regulate the flow of tasks through any given partition.

- Sharable bulk storage. In general, this resource is allocated on a first-come, first-serve basis, each disk I/O write request extending the space used by some previously opened file. The exception is contiguous files, space for which must be reserved before transmission of data to that file.

- Non-Sharable devices. These must be attached (reserved) for a given task by the task itself. Once attached, a device is unavailable to any other task until detached (released) by the owning task or under certain circumstances, via the MCR.

Relocation/Protection

XVM/RSX is a partitioned multiprogramming system. Through the memory relocation facilities of the XM15 memory processor option (KS15 on older systems), a task[1] can execute in any partition of sufficient length without regard to where in physical memory that partition begins.

This property affords significantly flexibility in managing system operations. It is not necessary to specify where the task must execute till run-time; therefore, the system operator can, without either taking down the system or having to reassemble a given task, reconfigure it from a condition such as:

```
 ___
|   |
| D |
|___|
|   |
| C |
|___|
|   |
| B |
|   |
|___|
|   |
| A |
|   |
|___|
|   |
|EXEC|
|___|
```
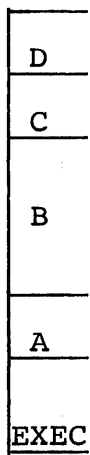
Partitions Created for Tasks A,B,C,D

---

[1]Relocation applies to user mode tasks only. Exec mode tasks are constrained to reside within the first 32K of physical memory; their exact locations are fixed at task build time and cannot be changed dynamically.

to a configuration such as. . .
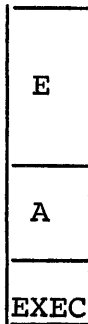
```
┌──────┐
│      │
│  E   │      partitions for B,C,D, have been consolidated
│      │      into one partition, allocated to new task E
├──────┤
│      │
│  A   │      task A remains in original partition
│      │
├──────┤
│EXEC  │
└──────┘
```

followed by. . .

```
┌──────┐
│      │
│  B   │      task B now in a new partition created
│      │      from top of E's partition
├──────┤
│      │
│  F   │      new task F partition created from
│      │      bottom of E's partition
├──────┤
│      │
│      │
│  A   │      task A remains in original partition
│      │
├──────┤
│EXEC  │
└──────┘
```

while the EXEC and task A continue to operate.

In addition to memory relocation, the hardware of both the XM15 and KS15 options provides protection among tasks, and between tasks and the EXEC. This is accomplished by providing a boundary register which is dynamically set by the EXEC to the partition   required by the task which is executing. (I/O buffers typically occupy the remainder of the partition.) Any attempt by the task to access memory outside its partition results in an interrupt to the EXEC. Input/output requests are similarly checked by the EXEC for requests that may violate a task's assigned partition.

In addition to protection and relocation, the XM15 memory
processor option provides several capabilities not included
in the KS15. Briefly, XVM/RSX allows a limited sharing of
core among tasks in different partitions to facilitate inter-
task communication.

Time-Slicing

As mentioned earlier, XVM/RSX allocates central processor
time on the basis of task priorities. An additional feature,
time-slicing, modifies the basic central processor scheduling
algorithm for interactive task response when executing a mix
of low-priority interactive and high-priority compute bound
jobs.

Without time-slicing, response time issues could develop into
real problems. Consider a situation in which the Active Task
List contains four tasks--A,B,C,D--in priority order:

```
+-----+
|  A  |        highest priority
+-----+
|  B  |
+-----+
|  C  |
+-----+
|  D  |        lowest priority
+-----+
```
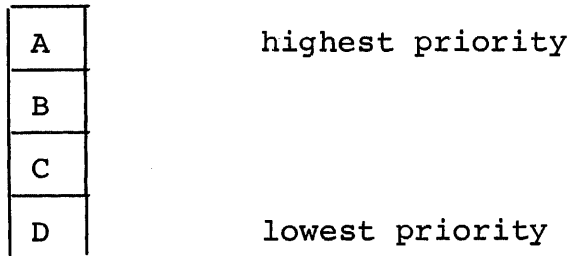
Let A be a task gathering instrument data in real-time; assume
task B is a compute bound data processing task; C constitutes
a set of interactive tasks which occasionally want the human
user to type in an input line of alphanumerics terminated by
carriage return; and let D be the XVM/RSX batch processor.

With purely preemptive scheduling we would find that task A
would get whatever central processor time it needed, but that
it would be possible for task B to dominate the system while
processing data. Tasks C and D would get no time at all,
since task B would not release the central processor voluntar-
ily, and could release it involuntarily only to the higher-
priority task A.

Time-slicing solves this problem by allowing a selected group of tasks to be scheduled on a round-robin basis - to be serviced sequentially for specified periods, with control reverting to the lowest priority group of tasks after a·specified number of round-robin service periods. In the example above, the effect of time-slicing would be to allow tasks to be serviced in the following order and without requiring human users to complete their input messages:

A   task A services instrument interrupt

B   task B runs (say for 0.17 seconds)

C   task C runs (also for 0.17 seconds)

B

D

B   task B uses a portion of its 0.17 second slice

A   task A takes control from B due to instrument interrupt

B   completes its slice

C   runs

D   gets control, say, every 3 turns of the round-robin scheduler

B   runs again

C   runs again

A   interrupts C

In general, if we consider the entire Active Task List,
time-slicing can be applied to a band of priorities:

A
B
C
D
E
F
G
H
I
J
K

highest priority tasks A-C get service as
usual through preemptive scheduling

time-slicing applied to priorities D-H;
service here is round-robin rather than
preemptive

lowest-priority tasks I-K get whatever
time is left over through preemptive
scheduling

Through MCR directives the system operator can turn time-slicing,
off or on, change the priority range over which it applies, and
adjust both the length of time any time-sliced tasks executes
and the distribution of time between the time-sliced set and the
lowest-priority set of tasks.

SOFTWARE COMPONENTS

The following software is available as part of XVM/RSX.

Monitor
Executive
System Configurator
MCR Task (console keyboard listener)
TDV Task (auxiliary keyboard listener)
IORD Task (I/O rundown)

Languages
FORTRAN IV
MACRO-XVM (assembler)

Task Builder
TKB (interactive version)
BTK (batch version)
DOSTKB (XVM/DOS version)

Text Editors
EDIT (interactive editor)
SLIP (batch line editor)

MCR Functions (for system status)
TASK LIST (print task list)
PARTITIONS (print partition definitions)
COMMON BLOCKS (print system common block definitions)
DEVICES & ASSIGNMENTS (print I/O assignments)
DEQUE (system DEQUE lister)
STROBE (system DEQUE lister)
UFD (print relationship between disk directories and I/O assignments)

MCR Functions (time-related)
ETIME (enter time and date)
TIME (print time)
DATE (print date and time)
REQUEST (request a task)
XQT (execute a task on a user disk)
RUN (run a task)
SCHEDULE (schedule a task)
SYNC (synchronize a task)
CANCEL (cancel task scheduling)

MCR Functions (debugging)
OPEN (examine core and disk)
ABORT (force a task to exit)

MCR Functions (reconfiguration)
ADV (add a device)
ASP (assign a task's partition and priority)
DTC (define terminal characteristics)
RCF (reconfigure system)
RCP (reconfigure only partitions)
REASSIGN (reassign I/O)
ACCESS (set shared access characteristics of memory blocks)

MCR Functions (miscellaneous)
INSTALL (install task on system disk)
CONSTRUCT (install task on user disk)
REMOVE (remove task from system)
FIX (fix task in core)
UNFIX (unfix task from core)
DISABLE (prevent task execution)
ENABLE (allow task execution)
RESUME (resume task execution)
MNT (mount a user disk)
DSM (dismount a user disk)
SLICE (turn time slicing on/off)
DOS (XVM/DOS bootstrap)

TDV Functions (miscellaneous)
DIR (list disk directory)
DTD (list DECtape directory)
NEW (write new DECtape directory)
FIN (file input to disk)
FOU (file output from disk)
LIST (list disk file)
RENAME (rename disk file)
DELETE (delete disk file)
INSTALL (install task in system)
CONSTRUCT (install task on user disk)
REMOVE (remove task from system)
REQUEST (request a task)
XQT (execute a task on a user disk)
MNT (mount a user disk)
DSM (dismount a user disk)

Batch Processing Components
BATCH (selects/sequences batch job file)
OPR (operator commands to control batch)
SCHED (sets batch scheduling parameters)
QUEUE (TDV, MCR, and OPR task to submit/stack jobs)
TLE (TDV function for overrun/time limit exceeded)
AB.OPR (aborts user tasks)
TDC (provides continuity in batch operations)
DECK (card file input to disk)
SPN (TDV task which initiates a program into a secondary
    partition for parallel processing)

I/O Handlers
AD (AD15 A-to-D Converter)
AF (AFC15 Flying Capacitor Scanner)
BD (Batch Processor)
CC (Common Communicator)
CD (CR15/CR11 Card Reader)
CP (Card Punch)
DK (Multi-disk Driver/Allocator)
DT (DECtape)
LP (LP15/LP11/LS11/LV11 Line Printer)
MT (Magtape)
PP (Paper Tape Punch)
PR (Paper Tape Reader)
RF (RF15/RK09 DECdisk File Handler)
RK (RK15/RK05 Cartridge Disk File Handler)
RP (RP15/RP02 Disk Pack File Handler)
TT (Terminals)
UD (UDC15 Universal Digital Controller)
VT (VT15 Graphics Displays)
VW VW01 Writing Tablets)
XY (XY11/XY311 Plotter)

Utilities (system, On-line)
RSX (XVM/DOS system program to load RSX)
TNTERM (task termination message printer)
SAVE (save core on disk)
NODCNT (count free nodes)
RSXODT (debugging aid)
POLLER (Unichannel device poller)

Checkout Package
RX (register integrity test)
DTRUN (DECtape exerciser)
TTYIO (terminal exerciser)
COPHM (print hour mark)
...COP (start checkout package)
SATCHK (disk bit map checker)
...KCH (stop checkout package)

MINIMUM HARDWARE REQUIRED:

XVM/RSX is packaged into a single kit, described later.

This kit requires the following basic XVM or PDP-15 hardware
(plus additional hardware listed later on):

XVM or KP15 Central Processor[2]
32K of 18-bit Core Memory
Two LA30C DECwriters, LA36 DECwriters, KSR35 Teleprinters, VT50
    Terminals, or VT05 Terminals (up to 2400 baud)[1]
PC15 High-Speed Paper Tape Reader/Punch
KE15 Extended Arithmetic Element
KW15 Real-time Clock (50 Hz or 60 Hz)
TC15 or TC02 DECtape Control with 1 TU56 Dual DECtape Transport
    or 2 TU55 Single DECtape Transports; or TC59 Magtape Control
    with 1 TU10, TU20, or TU30 Magtape Transport (7- or 9-track).

<u>XVM/RSX KIT (QM070)</u> for an RK05 Disk configuration:

Basic XVM or PDP-15 hardware plus,
RK15 Disk System with
8K of 16-bit Core Memory for 2-device spooling; 12K for 3-device
    spooling

Note that the XVM/RSX software does not support the PDP-11/05
or the XVM Power Fail hardware.

<u>RSX/XVM BASIC KIT</u> (QM070) for an RF15/RS09 Disk configuration:

Basic XVM or PDP-15 hardware, plus,
RK15 DECdisk Control
Two RS09 DECdisk Drives (256K words each)

<u>XVM/RSX BASIC KIT</u> (QM070) for an RP15/RP02 disk configuration:

Basic XVM or PDP-15 hardware, plus,
RP152 Disk System (which includes RP15 Disk Pack Control and
one RP02 10 million word Disk Pack Drive)

A line printer is not required; however, to utilize the system's
batch processing capabilities to the fullest, it is recommended.

---

[1]One terminal is normally used for the operator Monitor Console
Routines (MCR) and the other for Task Development Routines (TDV). At
least one hard copy terminal is strongly recommended. The second
terminal requires an LT15A Control. The LT19 Control can be used
with or in lieu of the LT15A to allow expansion to a 17 terminal sys-
tem. For input to TDV or MCR tasks the LA36 is supported as an 80
column device. Furthermore, for all FORTRAN I/O, only 80 column sup-
port is granted for the LA36.

[2]For Non-XVM systems, KS15 (which includes KA15, KM15 and KT15) is
required. The MX15A memory multiplexer may be required depending upon
the type of core memory used. The counterparts for these modules are
included in XVM systems and in XM15 options systems.

To utilize XVM/RSX Graphics software the following hardware
is also required:

GT15-S or GT15-L display systems (which include VT15 Graphic
Display Processor with VV15 Arbitrary Vector and VT04 or VT07 Scope)

OPTIONAL HARDWARE SUPPORTED:

XVM or PDP-15 Hardware:

AFC15[3] and AD15 A-to-D Converters
CP15[4] Card Punch
CR15 Card Reader
Core Memory (up to 128K 18-bit memory or 124K for UNICHANNEL-15
systems)
TC15 or TC02 DECtape Control plus up to 4 TU56 or 8 TU55 DECtape Drives
RF15 Disk plus up to 8 RS09 Drives
RP15 Disk Pack plus up to 8 RP02/RP03[5]
FP15 Floating Point
Up to 2 VT15 Graphics Display Processors each with VV15 Arbitrary
Vector plus up to 4 VT04 Displays plus up to 4 VL04 Light Pens
plus up to 4 LK35 Keyboards.
UDC15[3] Industrial Interface
LP15 Line Printer
TC59 Magtape Control plus up to 8 TU10, TU20, or TU30 Transports,
    either 8- or 9-track
Up to 4 LT19 Controllers plus up to 17 terminals: LA30C, KSR35,
    KSR33, VT05, LK35, LA36, or VT50[6]
Up to 4 VW01 Writing Tablets

PDP-11 Hardware:

For all PDP-11 hardware options the following which, defines the
KR15, is a prerequisite:

---

[3]Number of channels is optional.

[4]Hardware supported by Computer Special Systems.

[5]RP03 disk drives are supported only to RP02 capacity.

[6]The VT50's copier, cursor control commands, and hold screen
 mode are not supported.

UNICHANNEL
    PDP-11/o5 CPU
    8K of 16-bit core memory
    Two UNIBUS parallel interface DR11-C registers
    One parallel interface DR15-C register
XM15 Or MX15-B Memory Multiplexer
RK11E Disk Cartridge Controller
One RK05 Disk Cartridge Drive

Other PDP-11 options are:

CR11[7] Card Reader
Local Core Memory (up to 12K of 16-bit memory)
Up to 8 RK05 Disk Cartridge Drives
LS11, LV11, or LP11[7] Line Printers
XY11 or XY311[7] Plotter

PREREQUISITE SOFTWARE:

To run the XVM/RSX KIT (QM070) the XVM/DOS kit is required.

ORDERING INFORMATION:

XVM/RSX KIT - The kit contains all XVM/RSX components, including
the on-line task development and graphics features.

QM070-YC   Single use license*; binaries and sources
                on DECTAPE; manuals

QM070-YD   Single use license*; binaries and sources
                on 9-track magtape; manuals.

QM070-YF   Single-user license*; binaries and sources
                on 7-track magtape; manuals.

SUPPORT CATEGORY: C, plus SPR service for 90 days after delivery.
                     No on-site installation provided.

---

[7]The software will support only one card reader, one plotter,
and one line printer on the Unichannel System.
Note that the LV11 plotting capability is unused.

*This software is furnished under a license for use on a single
system and can be copied (with inclusion of DIGITAL's copyright
notice) only for use in such system, except as may otherwise be
provided in writing by DIGITAL. When a software license is ordered
without services, its category of support reverts to Category C.

UPDATE POLICY: Source update kits will be furnished to customers ordering the XVM/RSX software update service once per year.   The Source Program Update Service Subscriber will automatically receive the next update and may purchase replacement parts at prevailing handling and service charges.   XVM/RSX services are provided as follows:

QM060-2Z   Software Information Service

The customer receives a monthly newsletter containing announcements of new software developments, user techniques, outstanding software problems and problem solutions, where available.   SPR service is not included.

QM070-4 C, D, F   XVM/RSX Software Update Service

The customer receives XVM/RSX source module updates once per year.   This update includes fixes and enhancements. Also provided is a monthly newsletter as in QM060-2Z.   SPR service is included and is only available through this service.   Prerequisite for this service is a XVM/RSX source license (QM070-C,D,F), a XVM/DOS source license (QM060-C,D,F) and a DOS-15/XVM source update service subscription (QM060-4 C,D,F).

XVM/DOS (DISK BASED INTERACTIVE/BATCH OPERATING SYSTEM)
Version V1A000

XVM/DOS is a combination of interactive and batch operating systems designed to provide superior computing power to users of DIGITAL XVM and PDP-15 computer systems. XVM/DOS supports both single processor systems and dual processor UNICHANNEL systems.

The Disk Operating System (DOS) is an integrated set of software designed to meet the demands of research, engineering and industrial environments. It includes the software necessary for simplified programming and efficient use of the hardware. Since the operating system components reside on the disk until needed, XVM/DOS brings to the user the advantage of disk resident storage via rapid access to the system's resources.

The DOS Monitor, the heart of the system incorporates all the functions of DOS-15 and adds the power of 128K addressing. The user controls the operating system by instructions to the monitor. The monitor runs the jobs, supervises data and file manipulation, and interacts with the operator/user in a simple, conversational manner.

The Batch Operating Subsystem (BOSS) is a dedicated batch processing extension to the XVM/DOS software system. Batch is a method of serial computer program processing which maximizes the number of programs that can be run in a given time. XVM/BOSS is a general purpose scientific, commercial and applications programming package designed for computation centers in private industry and universities.

The XVM/BOSS subsystem is a disk-resident set of programs supervised by a monitor. The monitor tailors the system configuration according to the XVM/BOSS command language card, and then supervises operation of the system or user programs requested by control cards.

XVM/BOSS is intimately related to XVM/DOS since they share a common resident monitor which makes it possible to switch from one mode of operation to the other easily. XVM/BOSS has a unique and versatile job control language which can be modified and tailored to specific user needs.

XVM/BOSS FEATURES

All XVM/DOS software resides on either DECdisk, disk pack or disk cartridge until loaded and run. Additionally, user software may reside on backup storage such as DECtape or magtape.

Interactive Operation

An interactive keyboard/program monitor permits device-independent programming and automatic calling and loading of XVM/DOS system and user programs.

XVM/DOS Command Batching Operation

An alternative to interactive operation is a Batch Processing capability of the monitor permitting DOS user commands to be entered via paper tape, card reader, disk, DECtape, or Magtape. This allows many programs to be run without operator intervention or supervision. This capability is distinct from XVM/BOSS, the Batch Operating Software System, which uses XVM/DOS as a base.

XVM/BOSS Batch Processing Subsystem

XVM/BOSS provides another alternative to interactive operation with its card-oriented batch processing capability. The XVM/BOSS command language is designed for the job shop environment. XVM/BOSS utilizes CR11, CR15, or CR03B card readers to allow programs to be run without operator intervention. Program results are displayed on the LP15 or LP11 line printer. This capability is distinct from DOS Command Batch mode which utilizes an XVM/DOS command syntax.

Programming Languages

A choice of programming language is given:
FORTRAN IV, MACRO XVM, MAC11, or FOCAL (DOS only).

I/O Device Handlers

Data and file manipulating I/O device handlers are supplied for standard system peripherals, allowing device independent programming and overlapped computation with the simultaneous operation of asynchronous peripherals.

Input/Output Spooling

XVM/DOS systems using the RK15/RK05 UNICHANNEL Disk System provide spooling of card reader, line printer and XY plotter data.

Spooling is a method of storing (queuing) data to and from slow-speed devices on the high-speed RK05 Disk, dramatically improving the system utilization and throughput.

Spooling is provided for the following UNIBUS devices: CR11, LP11, LS11, LV11, XY311, and XY11. The LV11 is supported only as a line printer. (The LV11 utilized as a graphics output device is available as a separately priced software option).

Using any of the XVM/DOS systems (RF15, RP15, or RK15), spooling may be done to any one unit of the RK05 disk (unit 0-7).

XVM/DOS Programmed Monitor Commands

Input/output programming is simplified by the use of a set of system commands which are standardized for system-supported I/O devices.

XVM/DOS Conversational Programs

System utility programs interact with the Operator/User in a simple, conversational manner.

XVM/BOSS Job Accounting

A record of each job (run time, etc.) is kept in a disk-resident accounting file.

XVM/BOSS Job Control Language

XVM/BOSS has a unique job control language which has the following features:

- A compact command language where each command definition is stored on the disk as a "procedure file". Procedure files are in source language form.

- Procedure files can be modified and new ones can be added by the user.

- Cards form the primary input stream, but disk files can also be inserted into the stream.

Dynamic Storage Allocation

The available disk and DECtape storage is automatically allocated for optimum storage utilization.

Dynamic Buffer Allocation

Input/output core utilization is automatically optimized by the monitor. It allocates only that space which is required for the system and the user.

DECtape File Structure

Allows DECtape to be treated as a directoried (named, file oriented) device or as a sequential access (non-file oriented) device.

Disk File Structure

This allows the most efficient use of disk capacity and data retrieval for processing via:

- System supported DECdisk, disk pack, and cartridge disk devices, providing both economy and storage capacity.

- Virtually unlimited data capacity (disk pack = 83.7-million-words; DECdisk = 2.09-million-words; cartridge disk = 9.97-million-words).

- Random/Sequential file access and unstructured I/O.

- Virtually unlimited capacity for handling simultaneously open files, limited only by available buffer space and the number of logical device slots assigned to the disk.

- File protection through unique user directories and associated user identification codes. Files can be made invisible to

other users, but privileged access via a supervisory code is
allowed.

- User/User File Independence: identically named, unformatted
  Input/Output (FORTRAN IV).

- Random Access: Formatted/Unformatted Input/Output (FORTRAN IV).

System Tailoring by User
The user may easily incorporate installation software into or delete
software from the operating system, thereby tailoring the system
to special hardware and software needs.

BANK and PAGE Modes
The XVM/DOS system supports bank and page mode operation. In
page mode, user programs are loaded and relocated in 4K page units;
address modification via index register is also permitted. Bank
mode operation is available to users who prefer direct addressing
up to 8K. The use of the index register is not permitted in bank
mode.

XVM Addressing Mode
The XVM/DOS system supports wide address mode operation. In XVM
mode, user programs are able to indirectly address data stored in
COMMON blocks allocated above the traditional PDP-15 32K boundary.
The indirect addressing range of a user program is 128K in XVM
mode. In the traditional mode, indirect addressing is limited
to 32K.

SOFTWARE COMPONENTS
The following software is available as part of XVM/DOS:

Monitor
RESIDENT MONITOR
DOS NON-RESIDENT MONITOR
BOSS NON-RESIDENT MONITOR
SYSTEM LOADER
PIREX (Peripheral Processor Executive)

Languages
FORTRAN IV (F4X, FPF4X)
FOCAL (Interpreter)
MACRO XVM (DIGITAL XVM Assembler)
MAC11 (PDP-11 Assembler)

Text Editors
EDIT (Text/Source Program Editor)
EDITVP (Storage Scope Editor)
EDITVT (Graphic Display Editor)
B.PRE (BOSS Line Editor)

Loaders
Linking Loader
CHAIN and EXECUTE (overlay builder and loader)[1]

Debuggers
DDT (Dynamic Debugging Technique)
DUMP (Core Dump Lister)
QFILE (Store/Retrieve Core dumps)
UDMP11 (PDP-11 Core Dumper)
QDMP XVM (DIGITAL XVM Core Dumper)

Utilities (General)
MTDUMP (Magtape Dumper)
PIP (Peripheral Interchange Program)
SRCCOM (Source Compare)
UPDATE (Library File Manager)
8TRAN (PDP-8 to DIGITAL XVM Translator)
VP GRAPHICS LIBRARY
VT GRAPHICS LIBRARY

Utilities (System On-Line)
SGEN (System Generator)
PATCH (System Patcher)
SPOOL (Data Spooler)[2]
SPLOAD (Spooler Builder)[2]
SPLGEN (Spooler Disk Allocator)[2]
MCLOAD (MAC11 Builder)[2]

Utilities (System Off-Line)
DOSSAV (Disk Save/Restore)
RFBOOT (DECdisk Bootstrap)
RPBOOT (Disk Pack Bootstrap)
RKBOOT (Cartridge Disk Bootstrap)
ABSL11 (Absolute Loader for PDP-11)

I/O Handlers
CDB (CR11, CR15, CR03B card reader)
DKA, DKB, DKC, DKL (RF15/RS09 DECdisk)[3]
DPA, DPB, DPC, DPL (RP15/RP02 RP03 Disk Pack)[3,4]
DTA, DTC, DTD, DTE, DTF (DECtape)
LKA (LK35 Graphics Keyboard)
LPA (LP11, LP15 Line Printers)
MTA, MTC, MTF (Magtape)
PPA, PPB, PPC (Paper Tape Punch)
PRA, PRB (Paper Tape Reader)
RKA, RKB, RKC, RKL (RK15/RK05 Cartridge Disk)[3]
TTA (Teletype, LA30 DECwriter, LA36 DECwriter II)
VPA (Storage Scope)
VTA (VT15 Graphic Display)
VWA (VW01 Writing Tablet)
XYA (XY11, XY311 Plotter)

Checkout Package
RF.CHK (DECdisk Checkout)
RP.CHK (Disk Pack Checkout)
RK.CHK (Cartridge Disk Checkout)

MINIMUM HARDWARE REQUIRED

For an XVM System [13]:

KP15 Central Processor
24,576 Words of 18 Bit Memory
LA36, LA30C, KSR33, KSR35 or VT05 Console Terminal[5]
PC15 High-Speed Paper Tape Reader/Punch
KE15 Extended Arithmetic Element
KW15 Real-Time Clock[14]

XM15 Memory System including:[13]
  Wide Addressing
  Memory Protection and Relocation
  Instruction Prefetch
  Automatic Priority Interrupt
TC15 DECtape Control with 1 TU56 Dual DECtape Transport[7]; or
TC59 Magtape Control with 1 TU10, TU20 or TU30 (7 or 9-track)
  Magtape Transport
RK15 Cartridge Disk System With RK05 Cartridge Disk Drive and
  8,192 Words of 16-Bit Core Memory[8]; or
RP15 Disk Pack Control with one RP02 or RP03[4] Disk Pack Drive; or
RF15 DECdisk Control with two RS09 DECdisk drives.
CR11 Card Reader or CR15 Card Reader (Required Only to Utilize
  the XVM/BOSS Feature)[12]
LP11 Line Printer or LS11 Line Printer or LV11 Electrostatic
  Printer; or
LP15 Line Printer (Required only to utilize XVM/BOSS Feature)[12]

For a PDP-15 Computer System:

KP15 Central Processor
24,576 Words of 18-Bit Core Memory
KSR35, KSR33, LA30C, LA36 or VT05 Console Terminal[5]
PC15 High-Speed Paper Tape Reader/Punch
KE15 Extended Arithmetic Element
KW15 Real-Time Clock[14]
KA15 Automatic Priority Interrupt (required only for RK15 systems
  in certain configurations)[6]
TC15 DECtape control with one TU56 Dual-DECtape transport[7]; or
TCTC59 Magtape Control with one TU10, TU20 or TU30 (7- or 9-track)
  Magtape Transport
RF15 DECdisk Control with two RS09 DECdisk drives; or
RP15 Disk Pack Control with one RP02/RP03 Disk Pack Drive; or
RK15 Cartridge Disk System with one RK05 Cartridge Disk Drive
  and 8,192 Words of 16-Bit Core Memory[8]
CR15 Card Reader or CR11 Card Reader or CR03B Card Reader
  (Required only to utilize the XVM/BOSS Feature)[12]
LP15 Line Printer or LP11 Line Printer or LS11 Line Printer or
  LV11 Electrostatic Printer (Required only to Utilize the
  XVM/BOSS Feature)[12]

OPTIONAL HARDWARE SUPPORTED:

For the XVM system[13]

CR15/CR11/CR03B Card Readers
Core Memory 16-Bit (12K)[9]
Core Memory 18-Bit (ME15, MF15) (32K or 128K)
TC15/4 TU56 Dual DECtape or TC02/8 TU55 DECtape
RF15/8 RS09 Disk
RK15/8 RK05 Disk Cartridge[9]
RP15/8 RP02/RP03 Disk Packs[4]
FP15 Floating Point Processor
1 VT15/1 (VT04 or VT07)/1 VL04 Graphics Display
LP11/LP15/LS11/LV11[9] Line Printer
TC59/8 (TU10/TU20/TU30) Magtape
XY11/XY311[9] Plotter
VP15A Storage Scope
LA36/LA30C/KSR33/ASR33/KSR35/ASR35/VT05/LK35[10] Terminal
1 VW01 Writing Tablet

For the PDP-15 system

XM15 Memory Processing Unit including:
   Wide Addressing
   Memory Protection and Relocation
   Instruction Prefetch
   Automatic Priority Interrupt

KA15 Automatic Priority Interrupt
CR03B/CR15/CR11[9] Card Reader
Core Memory 16-Bit (12K)[9]
Core Memory 18-Bit (MM15/MK15/ME15/MF15) (32K to 128K)[11]
TC15/4 TU56 or TC02/8 TU55 DECtape
RF15/8 RS09 Disk
RK15/8 RK05 Disk Cartridge[9]
RP15/8 (RP02/RP03) Disk Pack[4]
FP15 Floating Point Processor
1 VT15/1 (VT04 or VT07)/1 VL04 Graphics Display
LP15/LP11/LS11/LV11[9] Line Printer
TC59/8 (TU10/TU20/TU30) Magtape
XY11/XY311[9] Plotter
VP15A Storage Scope
KSR33/KSR35/ASR33/ASR35/LA30C/LA36/VT05/LK35[10] Terminal
1 VW01 Writing Tablet

PREREQUISITE SOFTWARE:

None

ORDERING INFORMATION:

XVM/DOS KIT   The kit contains all XVM/DOS components, including
                 BOS XVM and graphics features.

QMO60-YC   Single use license*; binaries and sources on
                 DECtape; manuals.

QMO60-YD   Single use license*; binaries and sources on
                 9-track magtape; manuals.

QMO60-YF   Single-user license*; binaries and sources on
                 7-track magtape; manuals.

SUPPORT CATEGORY: C, plus SPR service for 90 days after delivery.
                     No on-site installation provided.

UPDATE POLICY: Source update kits will be furnished to customers
ordering the XVM/DOS software update service once per year. The
Source Program Update Service Subscriber will automatically receive
the next update and may purchase replacement parts at prevailing
handling and service charges. XVM/DOS services are provided as
follows:

     QMO60-28   Software Information Service

     The customer receives a montly newsletter containing announce-
     ments of new software developments, user techniques, outstanding
     software problems and problem solutions, where available. SPR
     service is not included.

     QMO60-4 C, D, F,   XVM/DOS Software Update Service

     The customer receives XVM/DOS source module updates once per
     year. This update includes fixes and enhancements. Also pro-
     vided is a monthly newsletter as in QMO60-28. SPR service is
     included and is only available through this service. Prere-
     quisite for this service is a XVM/DOS source license (QMO60-C,D,F).

---

*This software is furnished under a license for use on a single
system and can be copied (with inclusion of DIGITAL's copyright
notice) only for use in such system, except as may otherwise be
provided in writing by DIGITAL. When a software license is ordered .
without services, its category of support reverts to Category C.

NOTES:

1. EXECUTE is an improved version of the older EXECUTE overlay
    loader, which decreases program load and overlay times. It
    was developed from the FASTEX overlay loader available with
    DOS-15 V3B000.

2. Spooling is done only to the RK05 Cartridge Disk.

3. DKL, DPL , and RKL are integrally part of the System Loader.

4. The RP03 Disk Pack is supported as if it were an RP02 Disk Pack.

5. The LA36 and LA30C DECwriters and the VT05 Video Terminal are
    supported up to 300 baud.

6. API is required if the user has more than four PDP-11 options
    which need to interrupt the PDP-15.

7. The older style of DECtape can be used: TC02 Control with 2
    TU55 single DECtape transports.

8. The RK15 includes within it an 8,196 word UNICHANNEL-15
    peripheral processor. If the spooling software is to be used,
    the requirement for 16-bit core memory is raised from 8,192
    words to 12,288 words, if more than two devices are to be
    spooled.

9. Prerequisite hardware for devices like the CR11, LP11, LS11,
    LV11, XY11, and XY311, plus 16-bit core memory, is the RK15.
    The RK15 minimally contains a UNICHANNEL-15 with 8K words of
    16-bit memory, an RK11E Cartridge Disk Control and 1 RK05
    Cartridge Disk Drive. At any given time, only one line printer
    one card reader, and one XY plotter in the combined XVM/PDP-11
    or PDP-15/PDP-11 system is supported by the software. The
    LS11 or LV11 printer must be connected to the same vectors/
    priority as would the LP11.

10. The LK35 Graphic Keyboard is not a substitute for the console
    terminal; consequently, it also requires an LT15A Single
    Teletype Control. The paper tape facility on ASR Teletypes
    is not software supported. The LA30C and LA36 DECwriters
    and the VT05 Video Terminal can be run up to a baud rate of 300.

11. Memory configuration greater than 32K require the XM15
    options and can utilize ME15 or MF15 memory only.

12. The Card Reader and Line Printer requirements are waivered
    for those users not desiring XVM/BOSS operation.

13. The XM15 hardware option is incompatible with MM15 memory, KA15 automatic priority interrupt and KM/KT15 memory protect and relocate. The XM15 hardware functionally replaces these options. A trade-in policy has been established.

14. The KW15 Real-Time Clock is required on systems with UNICHANNEL-15 hardware.

15. While the minimum amount of memory required by XVM/DOS is 24,576 words of 18-bit storages, XVM configurations are available with a minimum of 32,768 words of 18 bit storage.

XVM/MUMPS

XVM/MUMPS is an interactive, single language, real-time, multi-user operating system which allows access to a common data base. The capabilities of the system are heavily oriented toward string manipulation using a high-level language. The system simplifies the use of tree-structured data bases by integrating the data base processing into the programming language.

Language processing by the system is in every sense interpretive. Each line of code undergoes identical processing each time it is executed (intermediate code is not generated). The MUMPS applications programmer is relieved of all the burdens associated with driving peripheral equipment or of programming in assembly language. Energies can be concentrated on the analytical aspects of the problem such as the logical hierarchy for the data base, and developing efficient logic for data base processing requirements.

The MUMPS language is supported by a stand-alone operating system. In addition to implementing the MUMPS language and providing all operating system capabilities, the system affords the user a unique data base structure and access method. Data referred to syntactically is automatically stored and linked in a tree structure. The physical allocation of mass storage for the tree-structured data base is accomplished by the operating system. The data base thus created is available to other users in the system on an interactive basis.

Minimum Hardware Required

XV-100 System[1]

TC15 DECtape Control with one TU56 DECtape transport or TC02 DECtape Control with two (2) TU55 DECtape transports or TC59 Magtape Control with one 7-track or 9-track Magtape Transport.

RF15 DECdisk Control with one RS09 DECdisk or an RP152 disk pack system.

---

[1]Alternatively a PDP-15 KP15 Central Processor with at least 24K words of 18-bit core memory, KE15 Extended Arithmetic Element, KW15 real-time clock, PC15 high-speed paper tape read/punch, and a console terminal (serial LA36, LA30C, KSR35, ASR35, KSR33, VT05, or ASR33).

Optional Hardware Supported

- Automatic Priority Interrupt (API) (included in XVM systems)
- DC01-ED (up to 8; each controls up to 8 lines for a maximum capacity of 64 lines)
- Additional core memory - up to 128K words total[2]
- RP15/Up to a total of 8 RP02 Disk Packs or 4 RP03 Disk Packs or a mixture of RP02s and RP03s (each RP03 can be replaced by up to 2 RP02s)
- RF15/Up to a total of 8 RS09 DECdisk Platters
- TC59/Up to a total of 8 TU10 Magtape Drives
- TC15/Up to a total of 4 TU56 Dual DECtape Transports or TC02/Up to a total of 8 TU55 DECtape transports
- CR03B or CR15 Card Reader
- LP15 Line Printers (up to a total of 2)

Prerequisite Software

XVM/DOS, DOS-15, or ADSS-15

Ordering Information

- QM810-AC single user license[3]; sources on DECtape; manuals
- QM810-AD single user license[3]; sources on 9-track magtape; manuals
- QM810-AE single user license[3]; sources on 7-track magtape; manuals
- QM810-GZ complete set of user manuals

---

[2]With a KP15 Central Processor core memory above 32K cannot be used for partition space.

[3]This software is furnished under a license for use on a single system and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL. When a software license is ordered without services, its category of support reverts to category C. Customer shall have the right to copy and use this source or any portion thereof, or the binaries/object codes generated therefrom, on any DIGITAL computer system located within the facility or institution specified for which (system) the Customer has purchased a separate binary license.

Literature

- Instruction Card
- Language Manual
- Language Manual Errata Sheet
- Operator's Guide
- Operator's Guide Supplement
- Operator's Guide Change Notice
- Version 10 Update Notice
- Version 11 Guide
- XVM/MUMPS Guide

Support Category

C, plus SPR Service for 90 days after delivery.

Update Policy

Any future updates will be available only through purchase
of the Source Program Update Service described below.

Additional Services and Subscription

It is recommended that 10 days of Software Service Installation
be purchased separately.

XVM/MUMPS source module updates are distributed on DECtape,
9-track magtape or 7-track magtape. The customer receives
a monthly newsletter plus a set of user manuals as part of
the starter kit. The customer also receives automatic updates
to software once per year. These updates include fixes
and enhancements. High priority SPR service is included
in this package.