digital

VAX/VMS

PRESENTATION HANDOUTS

DECUS - CHICAGO

APRIL 22-25, 1980

VAX11

VAX/VMS

PRESENTATION HANDOUTS

DECUS - CHICAGO

APRIL 22-25, 1980

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

AN OVERVIEW OF VAX/VMS V2.0

QUOTAS AND PRIVILEGES

- DISK QUOTAS
- CPU TIME QUOTAS
- NEW PRIVILEGES
- SET PRIVILEGES

PERFORMANCE ENHANCEMENTS AND SYSTEM MANAGEMENT TOOLS

- ACP ENHANCEMENTS
- AUTOMATIC WORKING SET ADJUSTMENT
- NEW SYSGEN PARAMETERS
- ENHANCEMENTS TO SYSGEN UTILITY
- LOGIN/AUTHORIZE ENHANCEMENTS

CUSTOMIZING YOUR SYSTEM

- USER WRITTEN SYSTEM SERVICES
- MESSAGE FACILITY
- LOADABLE TERMINAL DRIVER
- USER WRITTEN PRINT SYMBIONTS

NEW NATIVE MODE UTILITIES

- RMS WILD CARD SUPPORT
- DIRECTORY
- LIBRARIAN
- EDT
- MAIL

MEMORY MANAGEMENT

- PFN MAPPING

- SHARED MEMORY GLOBAL SECTIONS,

  MAILBOXES, AND COMMON EVENT FLAGS

- AUTOMATIC USER STACK EXPANSION

- SHAREABLE IMAGE CHANGES

REAL TIME ENHANCEMENTS

- MAP TO I/O SPACE
- USER WRITTEN SYSTEM SERVICES
- CONNECT TO INTERRUPT
- REAL TIME USER'S GUIDE

- FULL DUPLEX TERMINAL DRIVER

- UNIBUS POWERFUL RECOVERY

- ENHANCEMENTS TO

    - SDA (SYSTEM DUMP ANALYZER)

    - DEBUGGER

    - LINKER

UNSUPPORTED PROGRAMS

- SFA (SYSTEM FILE ANALYZER)
- INFO
- USERS
- TALK

UPGRADING

- AUTOMATED UPGRADE PROCEDURE -

  USER FILES ON SYSTEM DISK

  REMAIN INTACT

- COMPATIBILITY

VAX/VMS LIBRARIAN AND LINKER

BENN SCHREIBER

VAX/VMS DEVELOPMENT

THE VAX/VMS LIBRARIAN IS MADE UP OF TWO FUNCTIONAL PARTS:

1) LIBRARY COMMAND PROCESSOR

2) SHAREABLE IMAGE (LBRSHR) OF ACCESS

PROCEDURES

DCL LIBRARY COMMAND PROCESSOR

- INTERFACES WITH CLI TO GET THE COMMAND REQUEST

- INTERFACES WITH LBRSHR TO MANIPULATE LIBRARIES

FEATURES

- COMPLETELY REPLACES VMS V1.0 LIBRARIAN

- AUTOMATICALLY CONVERTS V1.0 LIBRARIES BEFORE MODIFICATION

- PROCESSES OBJECT, MACRO, HELP AND TEXT LIBRARIES

- CROSS REFERENCE OF OBJECT MODULE LIBRARIES

- ACCEPTS WILD-CARDS FOR MOST OPERATIONS

- 18 SHAREABLE PROCEDURES, 17 ARE TYPE-INDEPENDENT

FEATURES

- SUPPORT ASCIC OR 32-BIT BINARY KEYS

- FUNCTIONS SUPPORTED:

    - OPEN OR CREATE LIBRARY
    - LOOKUP A KEY
    - READ MODULE TEXT
    - WRITE MODULE TEXT
    - INSERT, REPLACE, AND DELETE KEYS
    - RETURN CONTENTS OF AN INDEX
    - SEARCH AN INDEX
    - RETRIEVE HELP TEXT

HELP LIBRARIES

- TOP LEVEL KEY ENTERED IN LIBRARY INDEX

- MANY DIGITAL SUPPLIED UTILITIES HAVE HELP NOW:

  DEBUG, DISKQUOTA, EDT, MAIL, SYSGEN, DCL

- AVAILABLE TO ANY PROGRAM

- HELP SOURCE FILE FORMAT IS SIMPLE AND EASY TO UNDERSTAND.

TEXT LIBRARIES

- CAN STORE ANY VARIABLE LENGTH RECORD-ORIENTED FILE.

- PLI INCLUDE FILES

LINKER FEATURES FOR V2.Ø

- MODIFY PSECT ATTRIBUTES

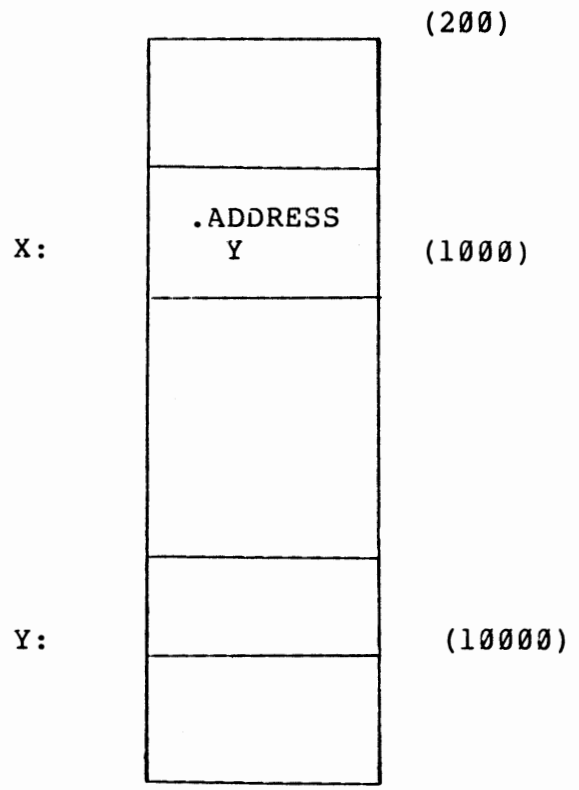
- DEFINE GLOBAL, ABSOLUTE SYMBOLS

- COLLECT GLOBAL PSECTS INTO CLUSTERS

- 31 CHARACTER NAMES

- COERCION OF POSITION INDEPENDENCE IN SHAREABLE IMAGES

- USER-SPECIFIED DEFAULT OBJECT LIBRARIES



IF THE ADDRESS OF Y (10000) WAS STORED AT LOCATION X (1000), THEN Y WOULD ALWAYS HAVE TO BE AT 10000. THIS IMAGE IS NOT POSITION INDEPENDENT.

BY DEFERRING THE STORING OF Y'S ADDRESS UNTIL THE SHAREABLE IMAGE IS PLACED IN THE VIRTUAL ADDRESS SPACE OF AN EXECUTABLE IMAGE, THE SHAREABLE IMAGE IS KEPT POSITION INDEPENDENT.

VAX/VMS VERSION 2.0

FIELD TEST PANEL

BENN SCHREIBER

VAX/VMS DEVELOPMENT

- WHAT IS A "FIELD TEST"?

  - TESTING OF NEW SOFTWARE, HARDWARE, AND DOCUMENTATION BY SELECTED CUSTOMERS

- WHY FIELD TEST?

  - COMPLEXITY OF PRODUCT

  - INFINITE DIVERSITY OF USES

- HOW ARE FIELD TEST SITES CHOSEN?

  - HARDWARE/SOFTWARE CONFIGURATION

  - AGREEMENT AND COOPERATION OF LOCAL SUPPORT OFFICE

  - HARDWARE/SOFTWARE TO BE TESTED

- WHAT DOES DIGITAL DO TO SUPPORT FIELD TEST SITES?

  - CLOSE LIAISON WITH LOCAL OFFICE
  - PERIODIC UPDATES THROUGHOUT FIELD TEST
  - "EMERGENCY" FIXES
  - QARS

- WHAT ARE BENEFITS TO FIELD TEST SITES?

  - DIRECT INPUT TO DEVELOPMENT
  - PRE-RELEASE VERSIONS OF SOFTWARE AND DOCUMENTATION

- FIELD TEST DIAL-IN QAR SYSTEM

HOW TO

READ A

VAX/VMS

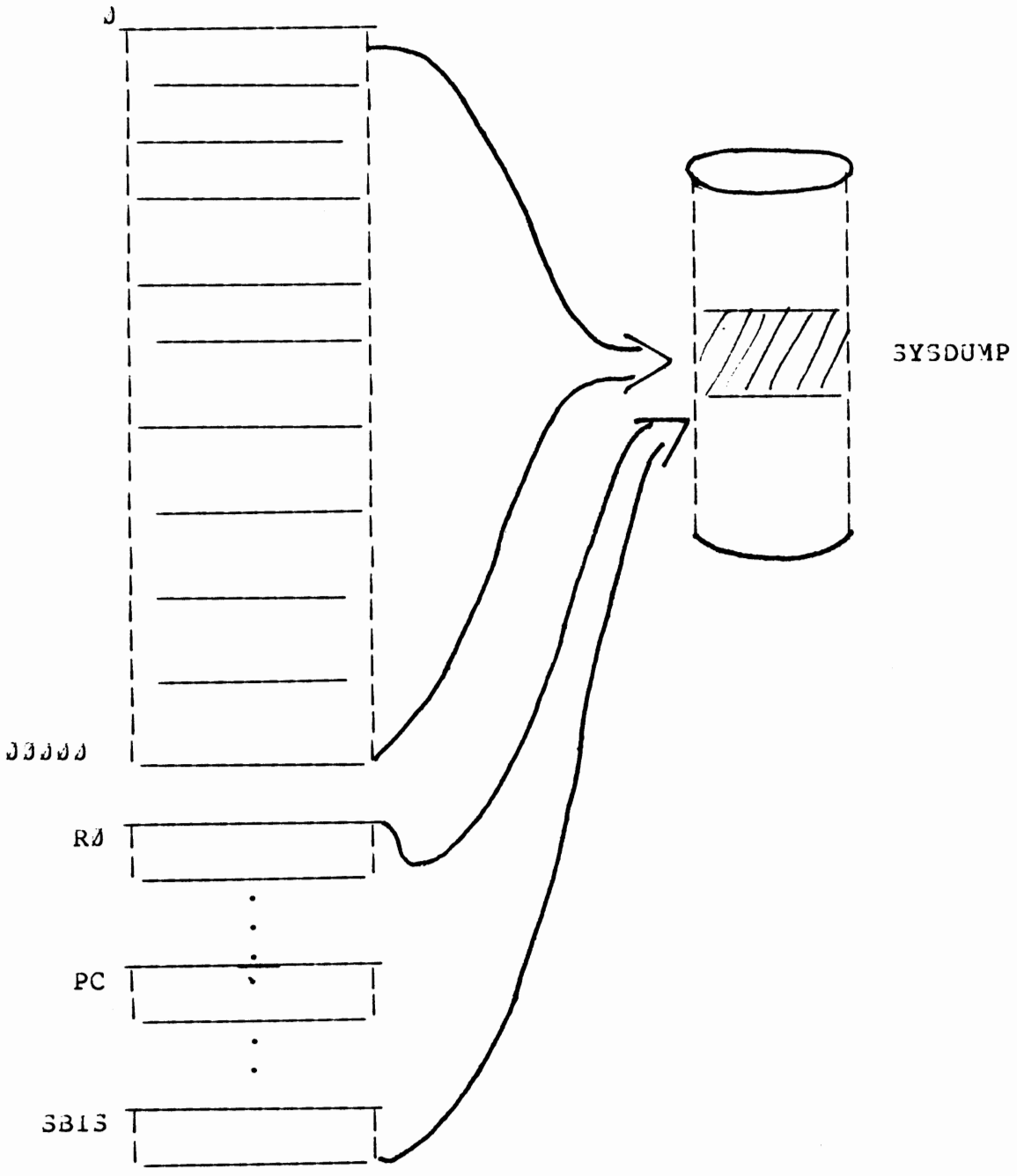
SYSTEM DUMP

WHY SYSTEM DUMP ANALYSIS?

- I/O DRIVER DEBUGGING
- PRIVILEGED PROGRAM DEBUGGING
- USER WRITTEN SYSTEM SERVICE DEBUGGING
- VMS PROBLEM ANALYSIS
- PROCESSOR PROBLEM ANALYSIS

WHAT IS A SYSTEM DUMP?

- COPY OF ALL PHYSICAL MEMORY
- ERROR LOG BUFFER CONTENTS
- SAVED CONTENTS OF ALL PROCESSOR REGISTERS
- WRITTEN TO SYS$SYSTEM:SYSDUMP.DMP



SETTING SYSTEM DUMP FILE SIZE

- [SYSUPD] SWAPFILES

- BLOCKS = PHYSICAL-MEMORY-SIZE-IN-PAGES+4

- TAKES EFFECT NEXT BOOTSTRAP

WHAT CAUSES A SYSTEM DUMP?

- "BUG CHECK" - INTERNAL CONSISTENCY ERROR
  - EXCEPTION WITH ELEVATED IPL
  - EXCEPTION ON INTERRUPT STACK
  - MACHINE CHECK IN KERNEL MODE
  - BUG-CHECK PSEUDO INSTRUCTION
  - HALT INSTRUCTION RESTART
  - INTERRUPT STACK INVALID RESTART
  - UN-HANDLED KERNEL OR EXECUTIVE MODE EXCEPTION

FIRST STEP - EXAMINE CONSOLE DISPLAY:

- CURRENT PROCESS
- CURRENT IMAGE
- CURRENT PSL
  - MODES (CURRENT/PREVIOUS)

    (KERNEL/EXECUTIVE)
  - IPL
- STACK CONTENTS

SYSTEM DUMP ANALYZER (SDA)

- EXAMINE VIRTUAL MEMORY BY

  ADDRESS OR SYMBOLIC NAME

- FORMAT AND DISPLAY PROCESS

  DATABASE

- FORMAT AND DISPLAY I/O DEVICE

  DATA STRUCTURES

- FORMAT AND DISPLAY MEMORY MANAGEMENT

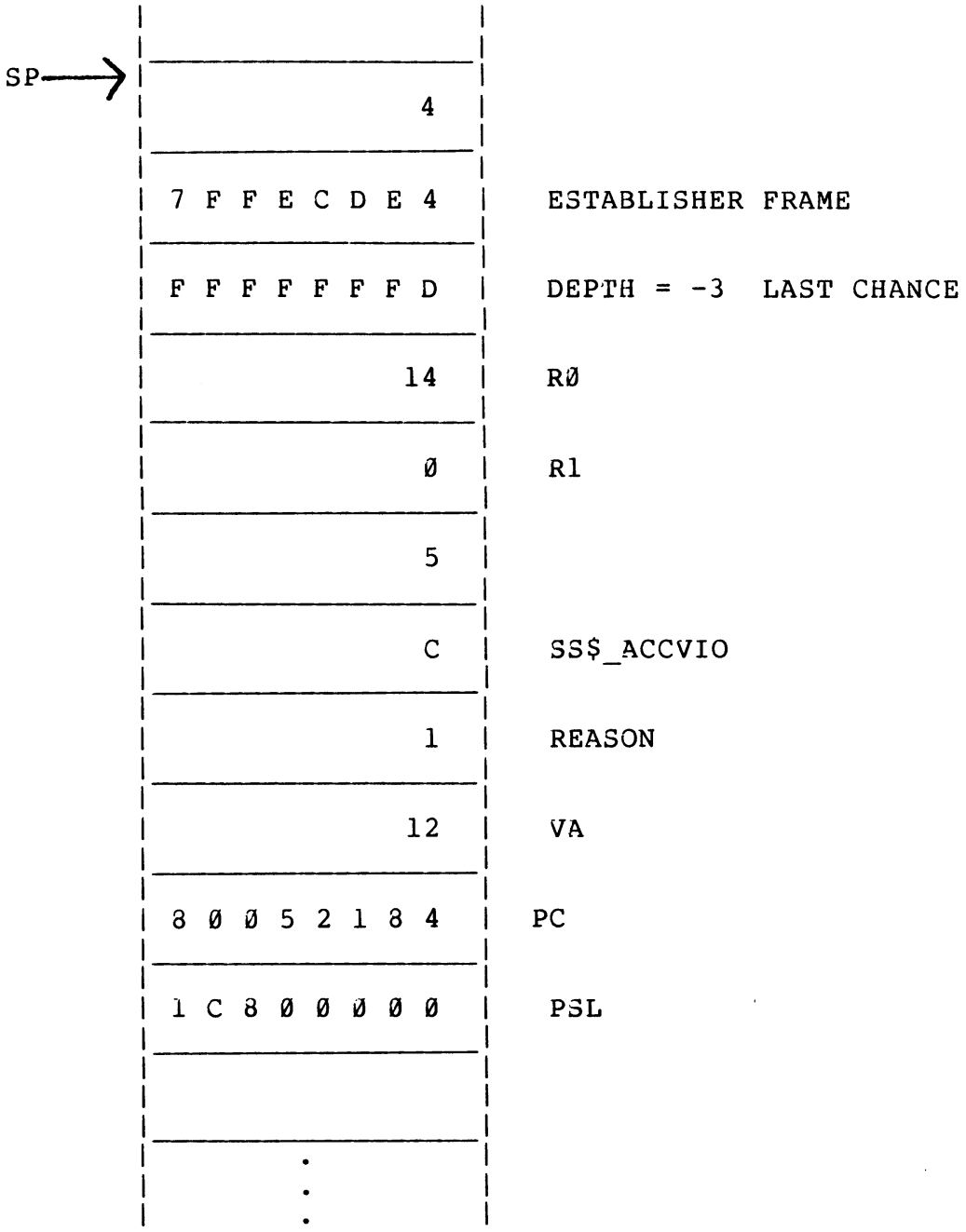
  DATA STRUCTURES

- EXAMINE PROCESSOR REGISTERS + STACKS

MOST COMMON PROBLEMS:

- EXCEPTIONS -
  - ACCESS VIOLATION
  - PAGE FAULT ABOVE IPL 2
  - RESERVED OPERAND
- MACHINE CHECKS -
  - CPU TIMEOUT
- CORRUPTED DATA STRUCTURE CONSISTENCY CHECK

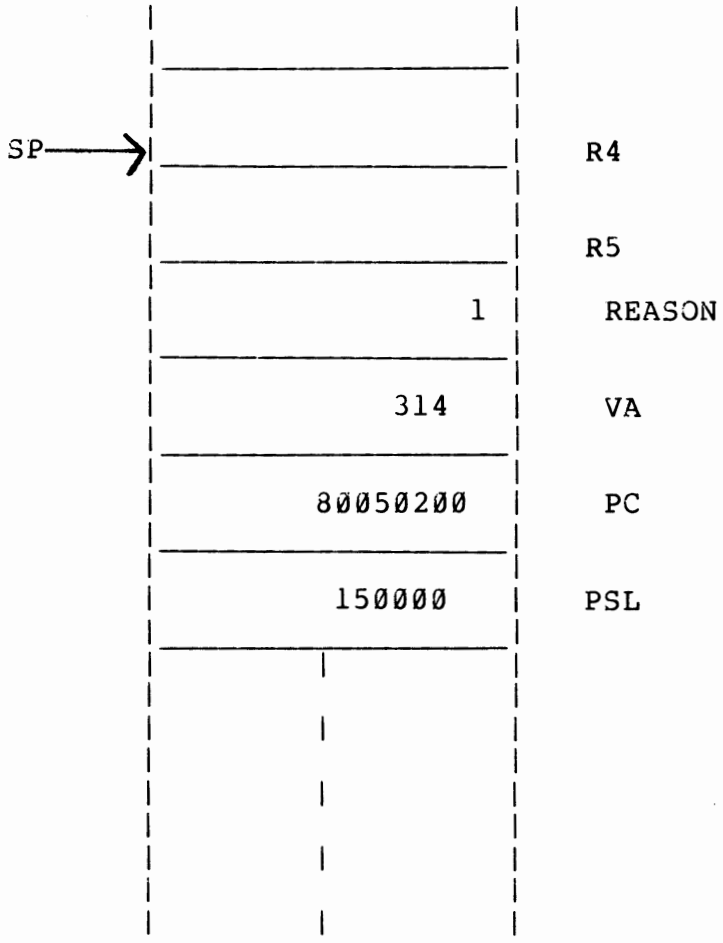
ACCESS VIOLATION



PROBABLE CAUSES:

- BLOWN REGISTER
- INCORRECT DATA STRUCTURE FIELD
- IMPROPER SYNCHRONIZATION

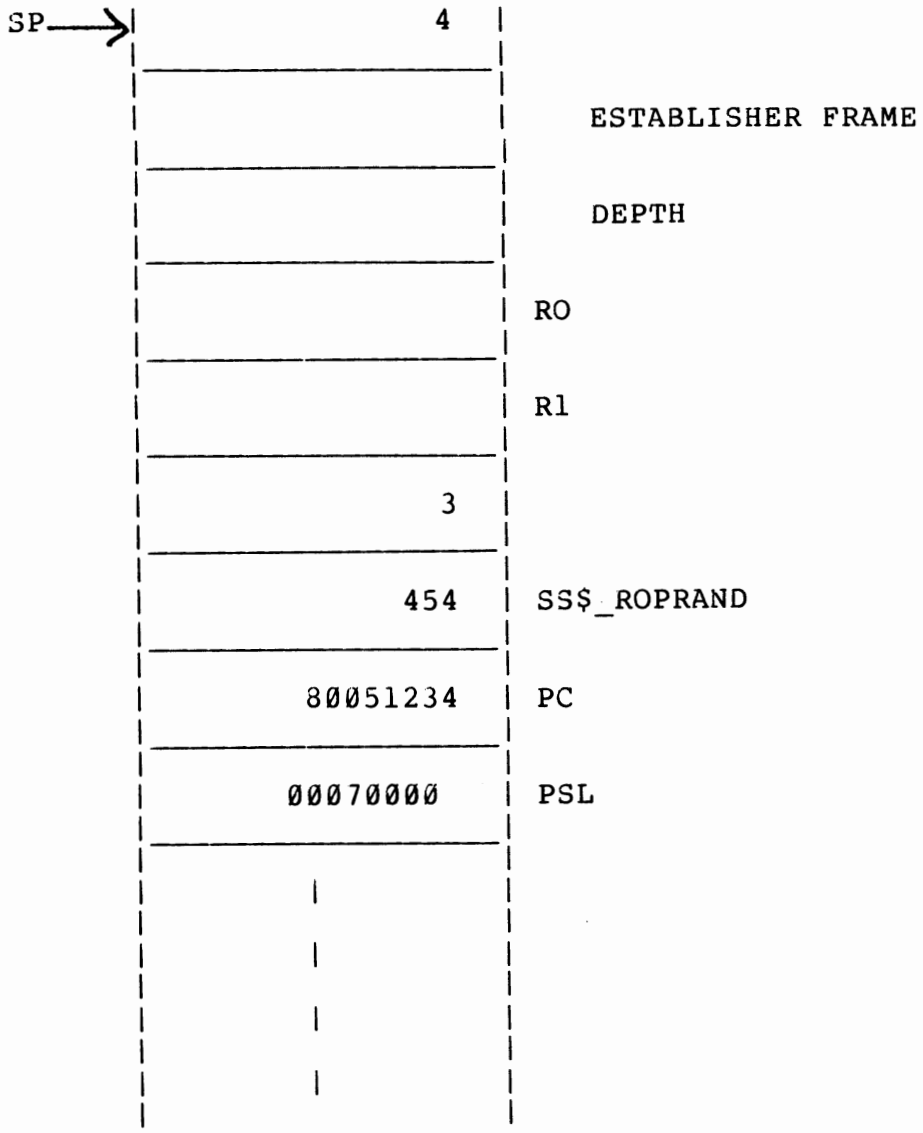
PAGE FAULT ABOVE IPL 2



PROBABLE CAUSES:

- BLOWN REGISTER IN FORK/INTERRUPT ROUTINE
- IMPROPER START I/O ROUTINE DESIGN

RESERVED OPERAND

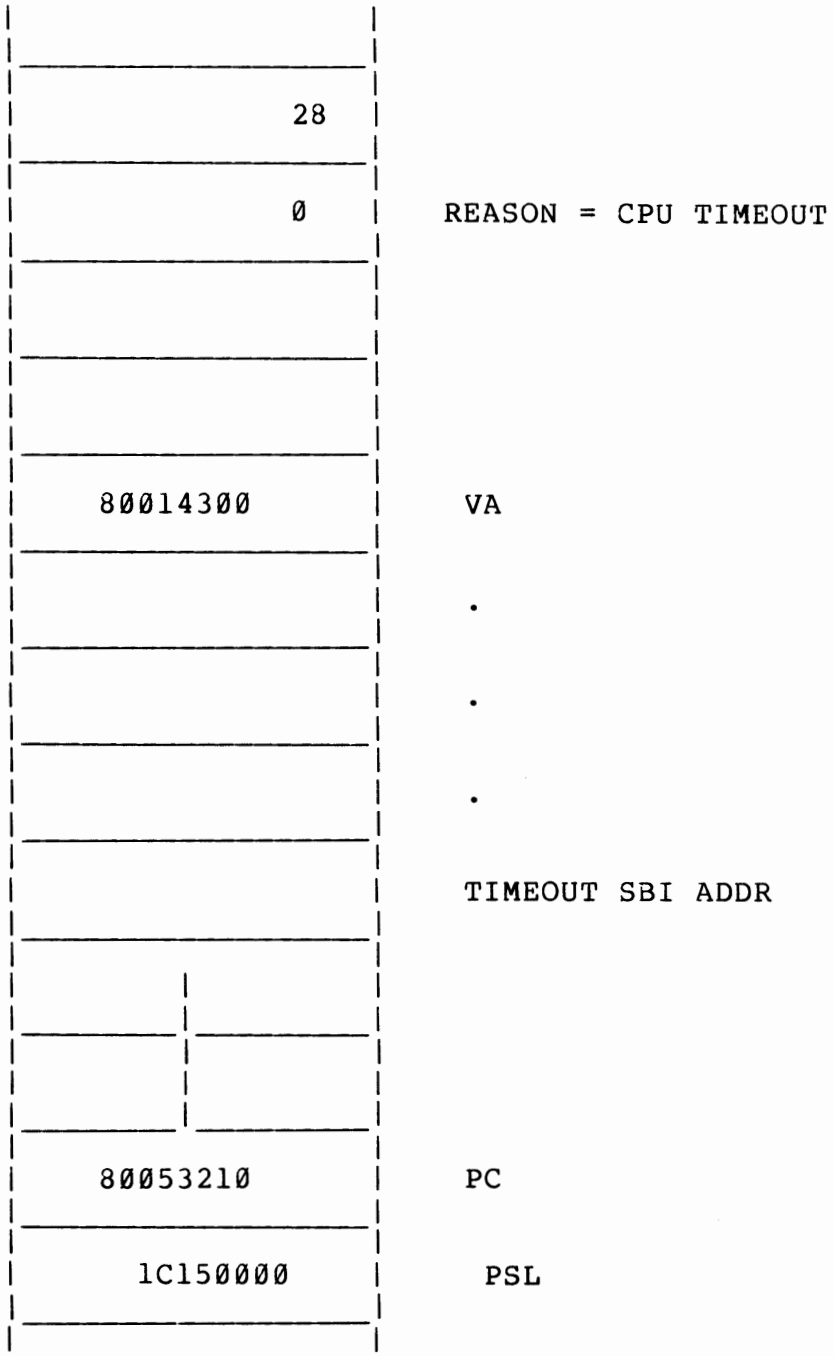


PROBABLE CAUSES:

- REI FAILURE
  - IPL PROBLEMS (ALLOCATE MEMORY AT WRONG IPL)
  - BLOWN STACK
- RET FAILURE

MACHINE CHECK IN KERNEL MODE

(CPU TIMEOUT)



REASONS:

- ACCESSING NON-EXISTANT UBA OR SBI ADDRESS ADDRESS
- CORRUPTED PAGE TABLES
- PROCESSOR DEVICE OR BUS FAILURE

"HUNG" PROCESS

- MWAIT - MISC. WAIT
  - REASONS -
    - MUTEX NEVER RELEASED

      OR DEADLOCKED
    - RESOURCE USED UP
    - I/O NEVER COMPLETED
  - CLUES -
    - EVENT FLAG WAIT MASK

      = SMALL POSITIVE # ⟹ RESOURCE NUMBER

      1 : AST WAIT

      2 : MAILBOX

      3 : NON-PAGED POOL

      = LARGE NEGATIVE # ⟹ MUTEX ADDRESS

- LEF/CEF - EVENT FLAG WAIT
  - REASON -
    - I/O REQUEST NEVER COMPLETED
  - CLUES -
    - EVENT FLAG WAIT MASK
    - DIRECT/BUFFERED I/O COUNT
    - BUSY DEVICE DRIVERS
    - CHANNEL CONTROL BLOCKS

"HUNG" SYSTEM

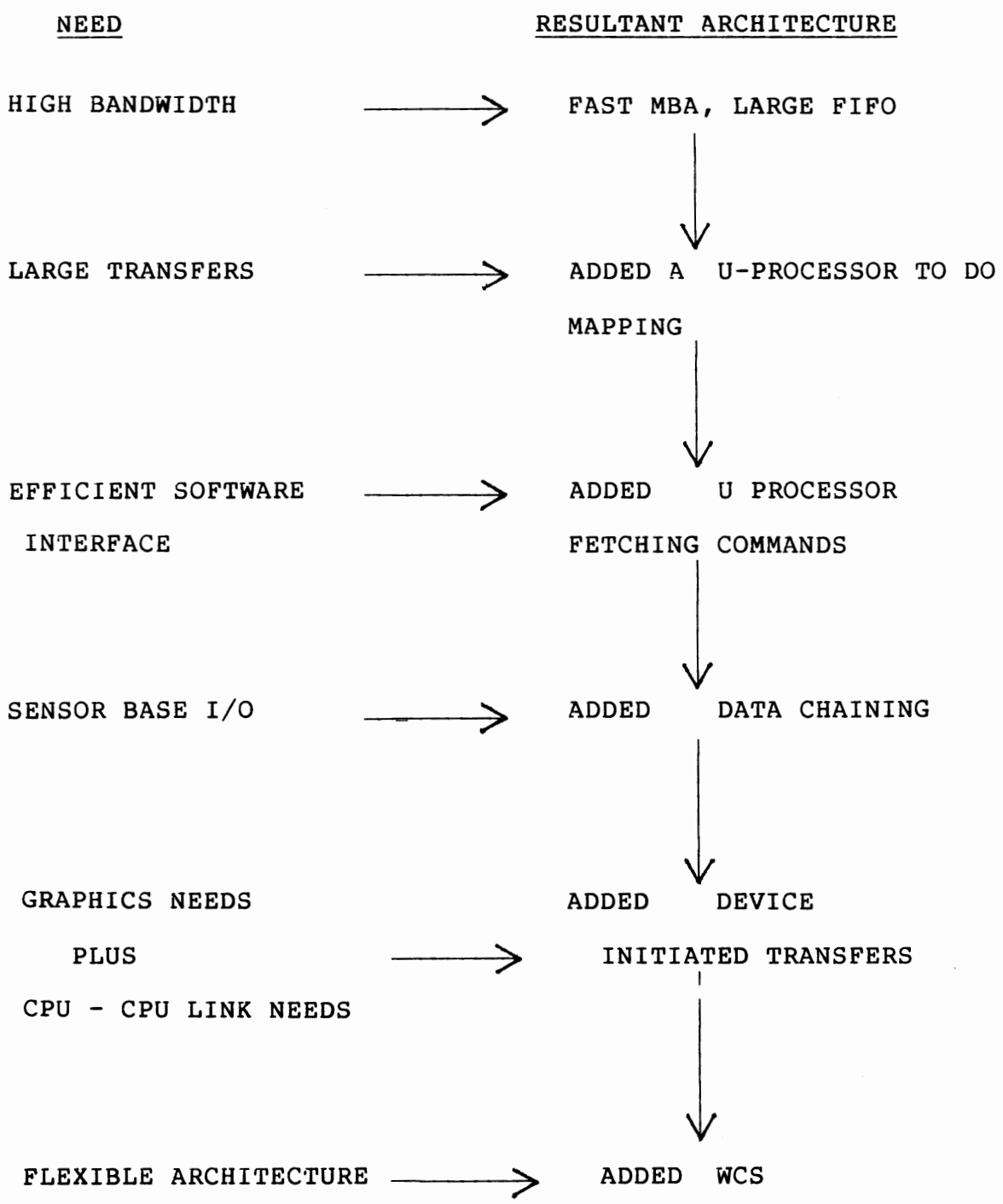
- ALL PROCESSES ARE HUNG
- INFINITE LOOP AT HIGH IPL
- INFINITE INTERRUPT LOOP
    - FORGOT TO DISABLE DEVICE INTERRUPTS
- DEBUG WITH XDELTA
- EXECUTE CRASH COMMAND PROCEDURE

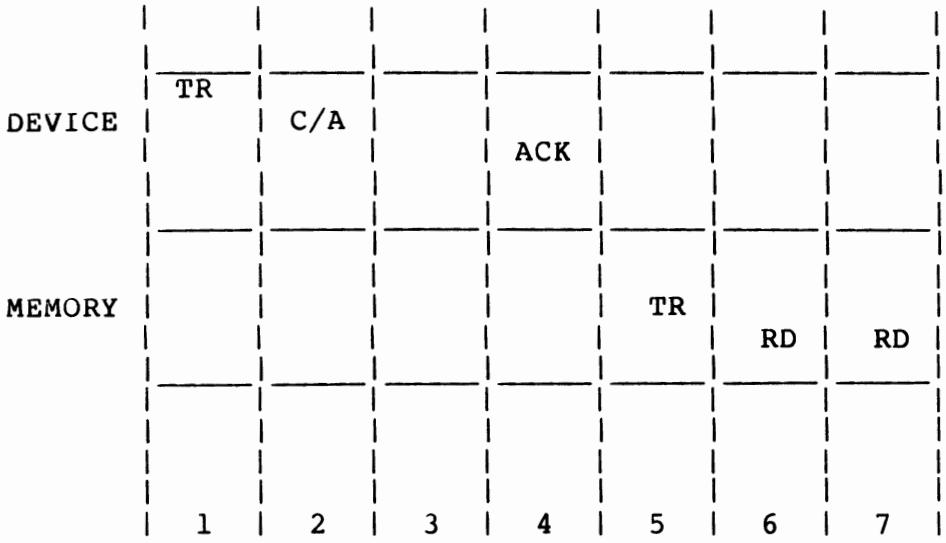
EVOLUTION

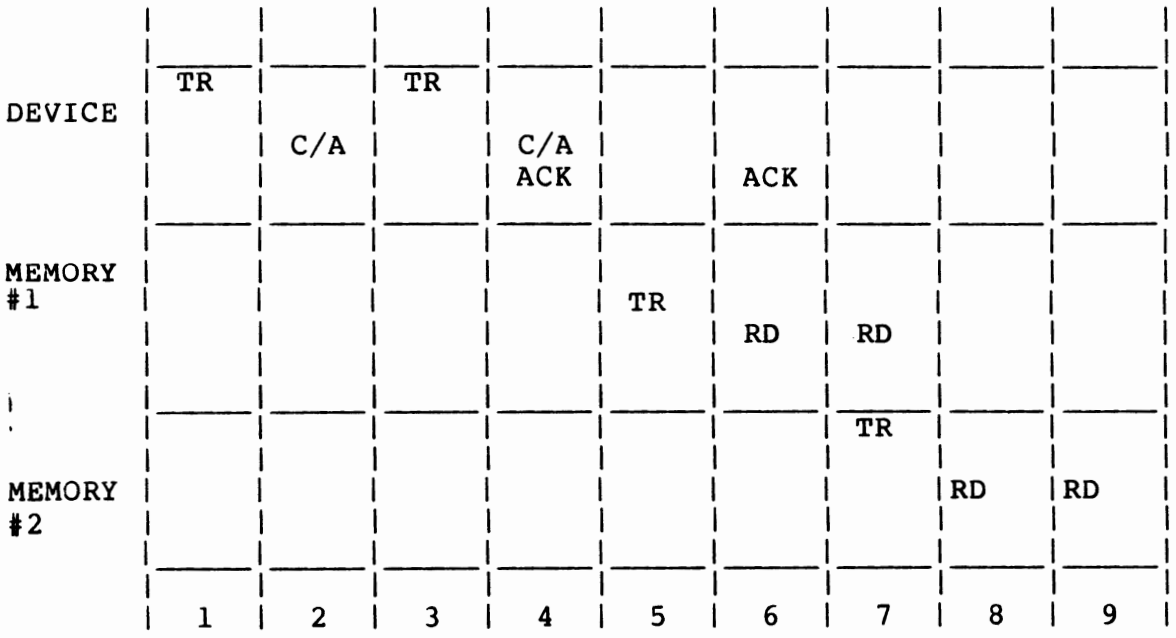
OF THE

DR32 ARCHITECTURE

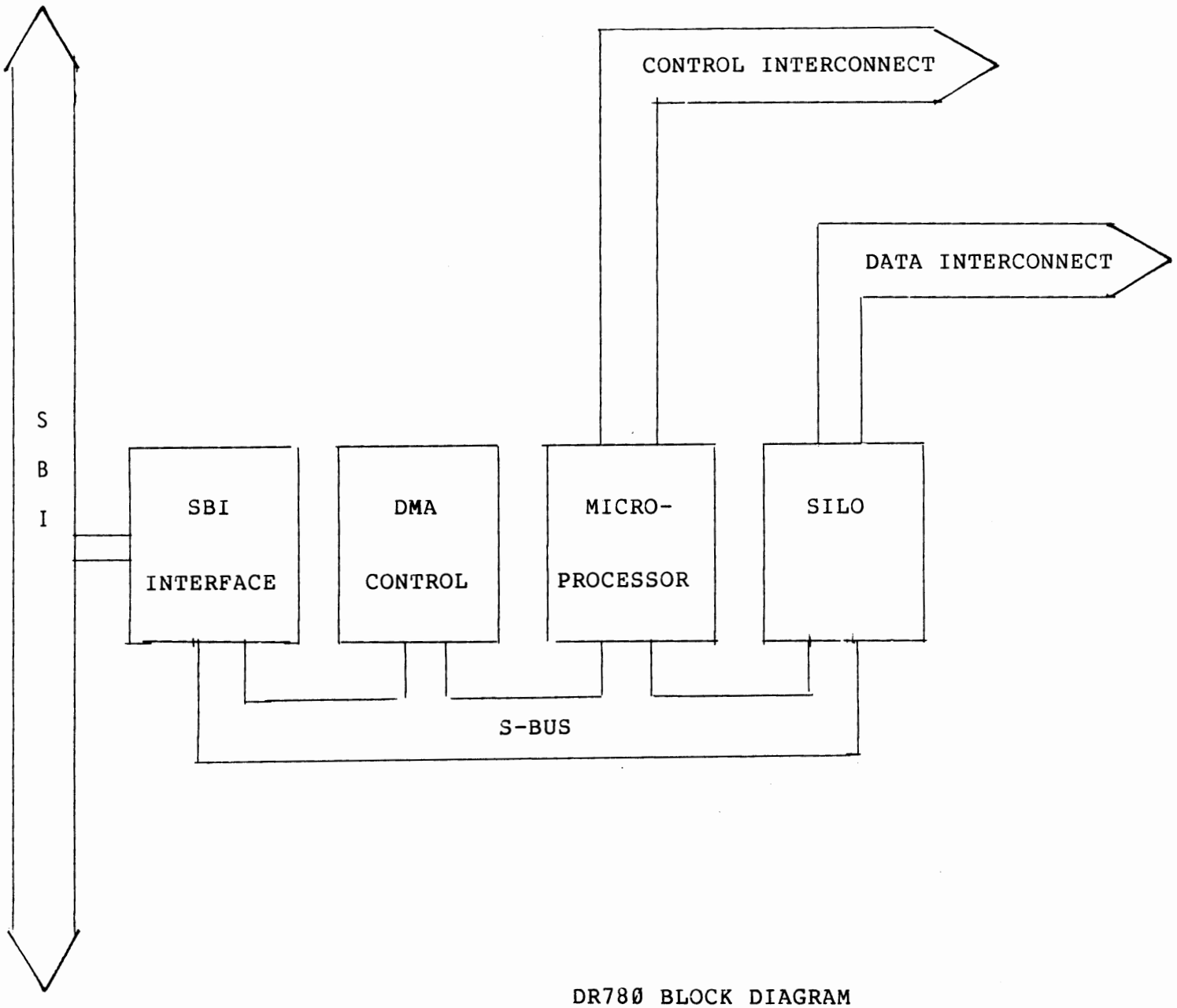
EVOLUTION OF THE DR32 ARCHITECTURE



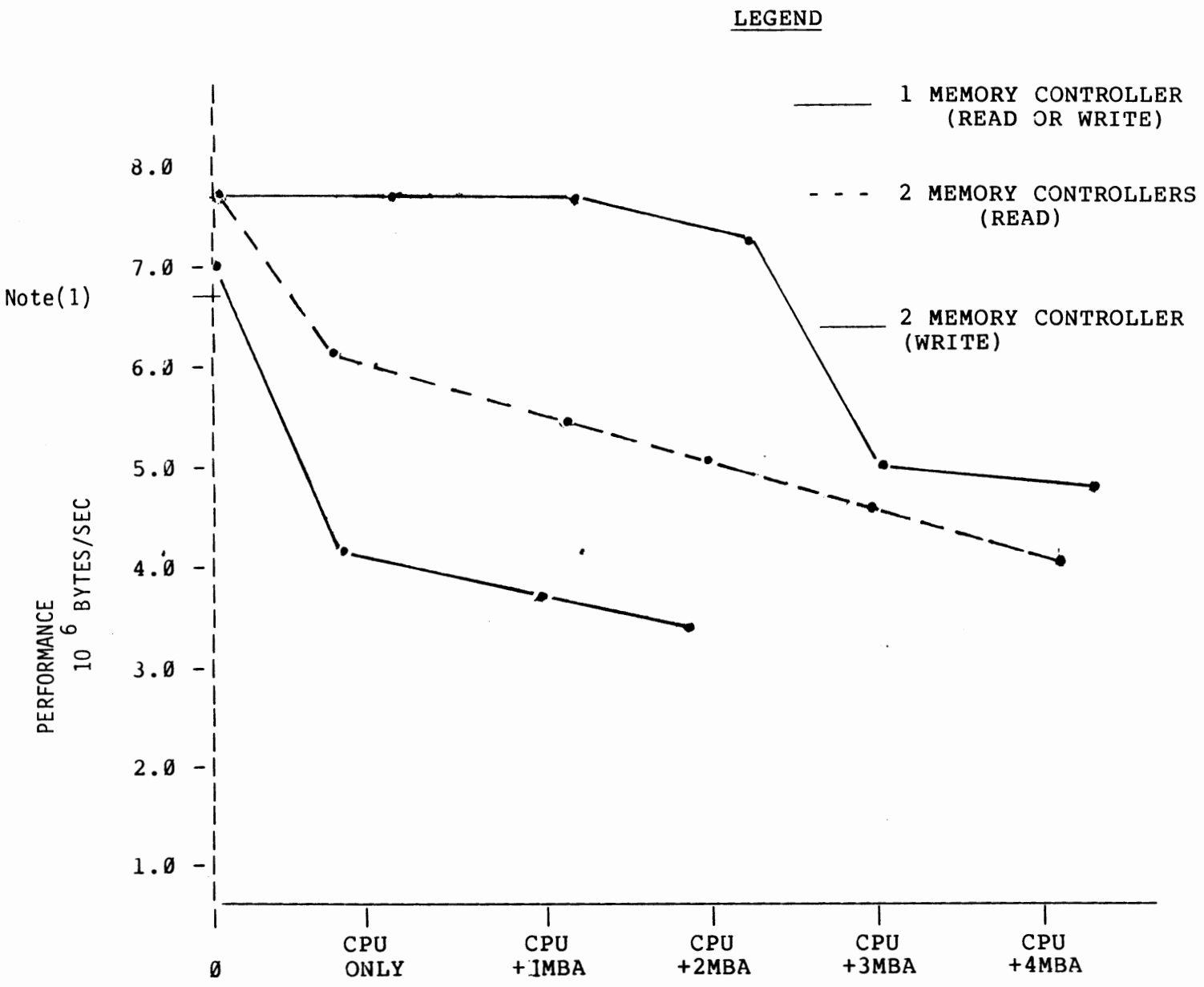




COMPARISON OF SB1 PERFORMANCE MEMORY READS

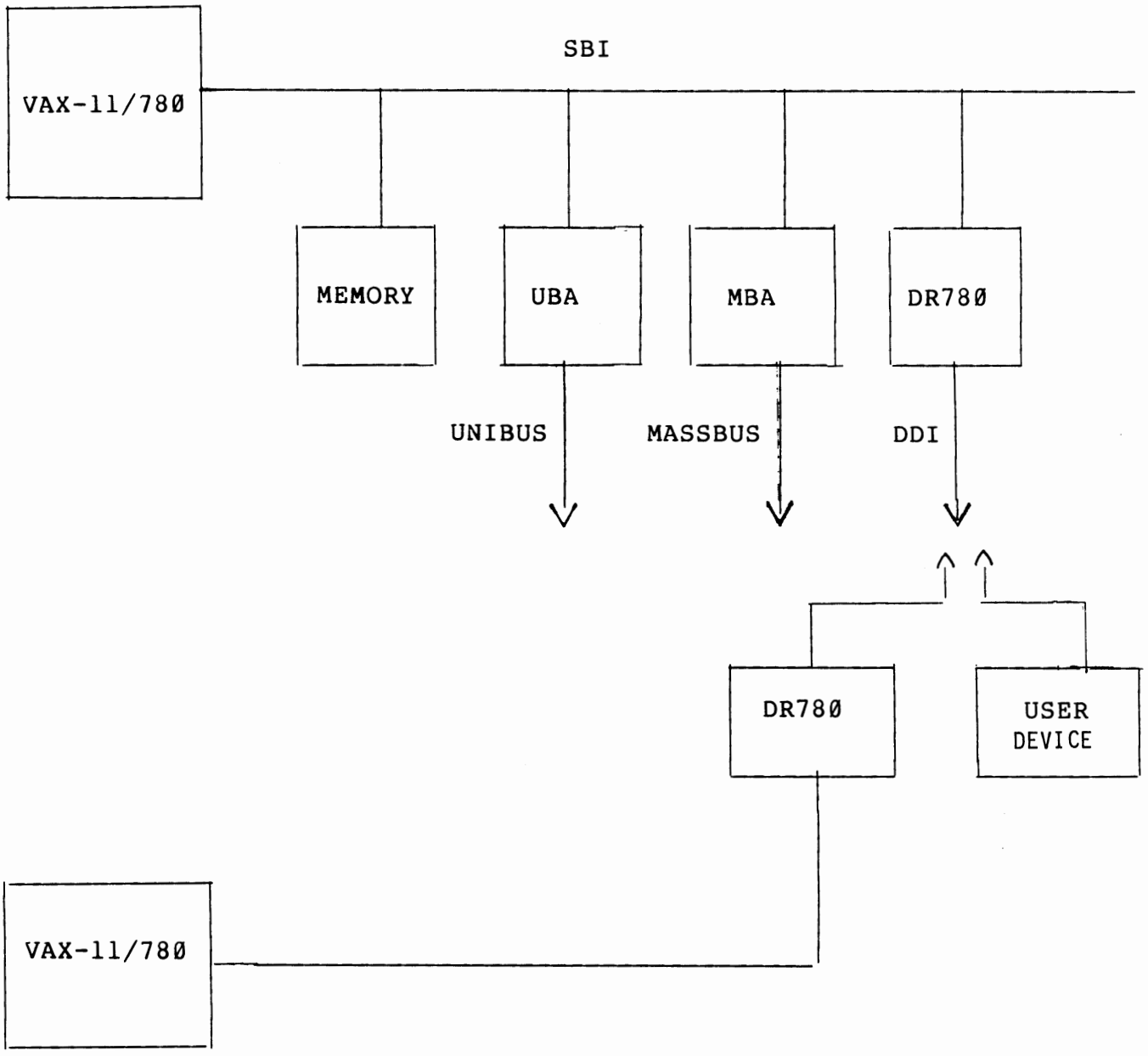


DR78Ø BLOCK DIAGRAM



NOTE (1) - Transfer rates above 6.67 Mbytes/second require a user supplied clock.

DR780 CONFIGURATION



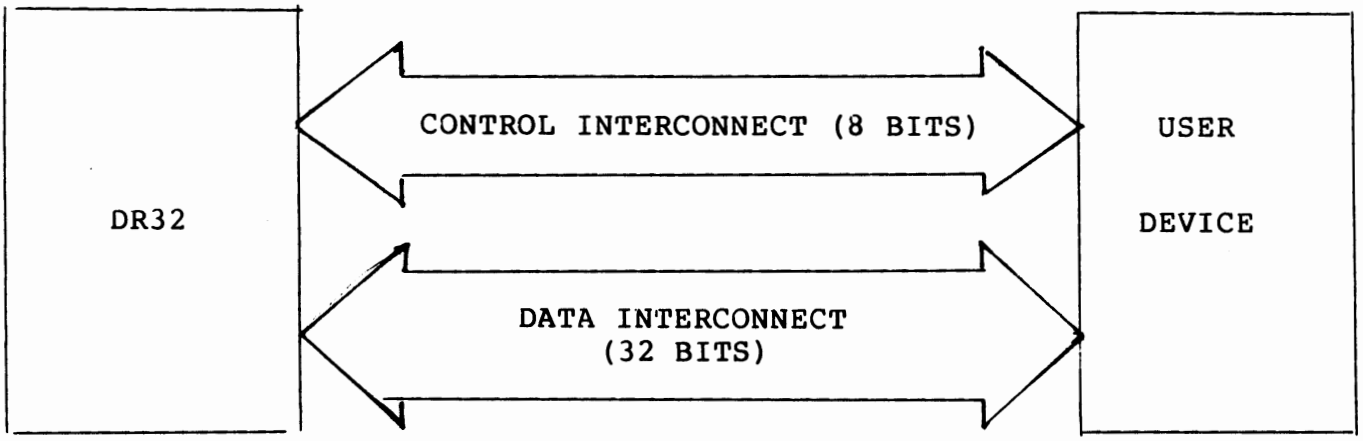
DR32 DESIGN GOALS

- HIGH BANDWIDTH (6 MBYTE/SEC)
- DIRECT COMMUNICATION WITH USER PROCESS (WITHOUT GOING THROUGH DRIVER)
- COMMAND CHAINING
- DATA CHAINING
- DEVICE INITIATED TRANSFERS
- BYTE ALIGNED DATA
- LARGE TRANSFERS (> 65K BYTES)
- VAX-11 CPU-CPU LINK

DR32 APPLICATIONS

- DATA COLLECTION
- INTERFACE ARRAY PROCESSORS
- INTERFACE GRAPHICS PROCESSORS
- CPU-CPU LINK

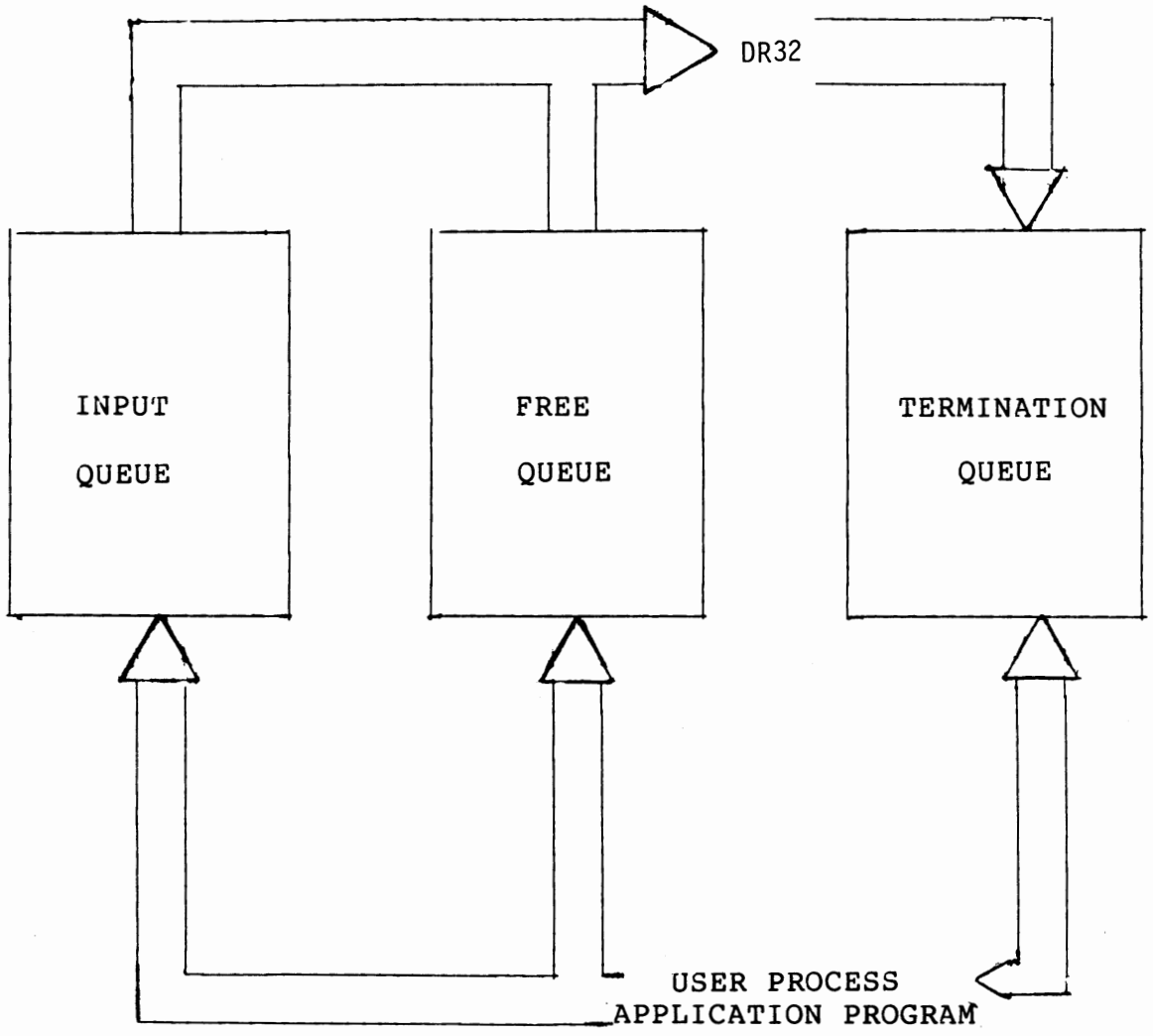
DR32 DDI



+ MISC CONTROL SIGNALS

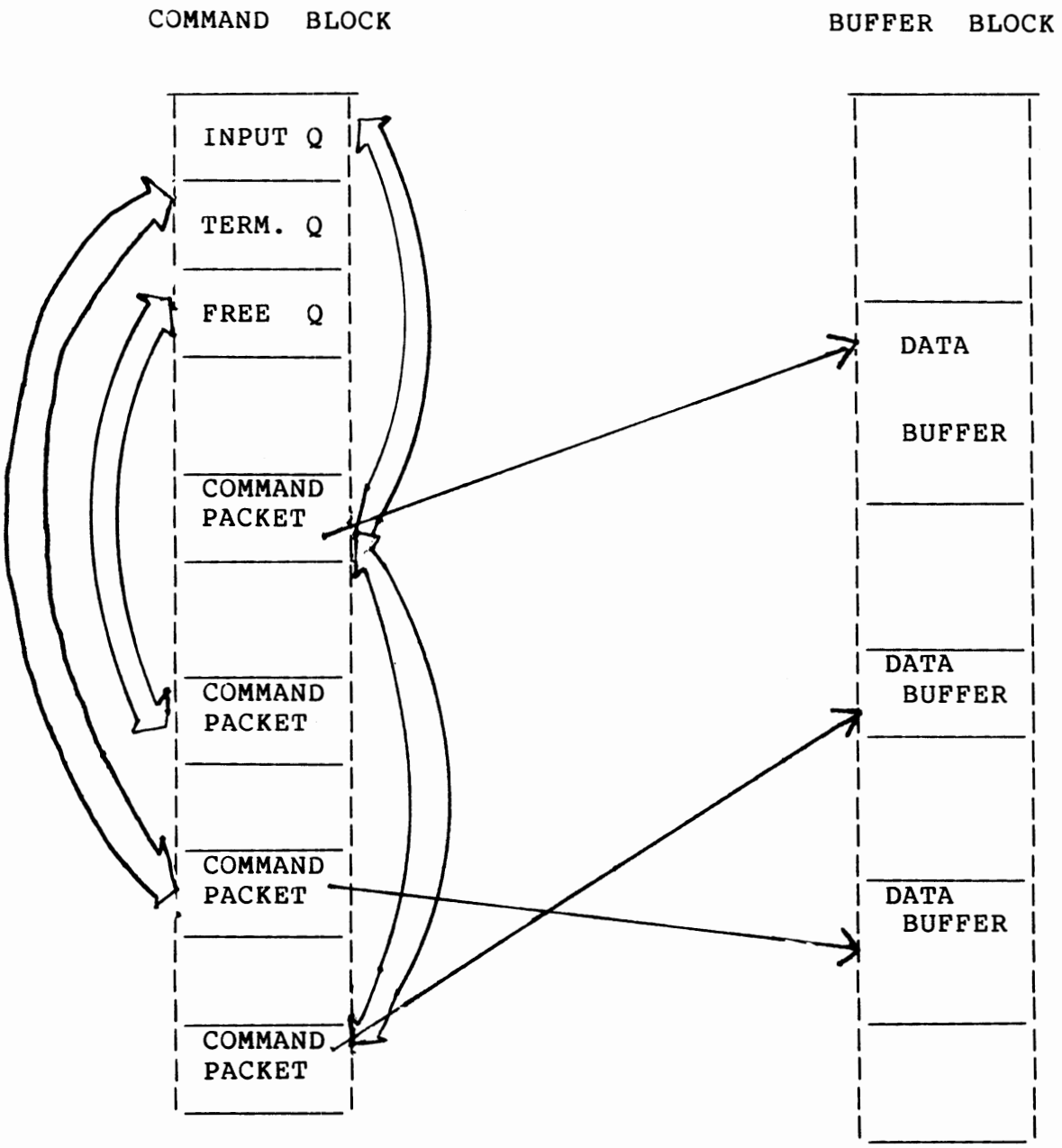
- UP TO 80 FEET
- MODIFIED MASSBUS CABLE
- CAPABLE OF OVERLAPPING CONTROL AND DATA TRANSFERS

DR32 COMMAND PACKET FLOW

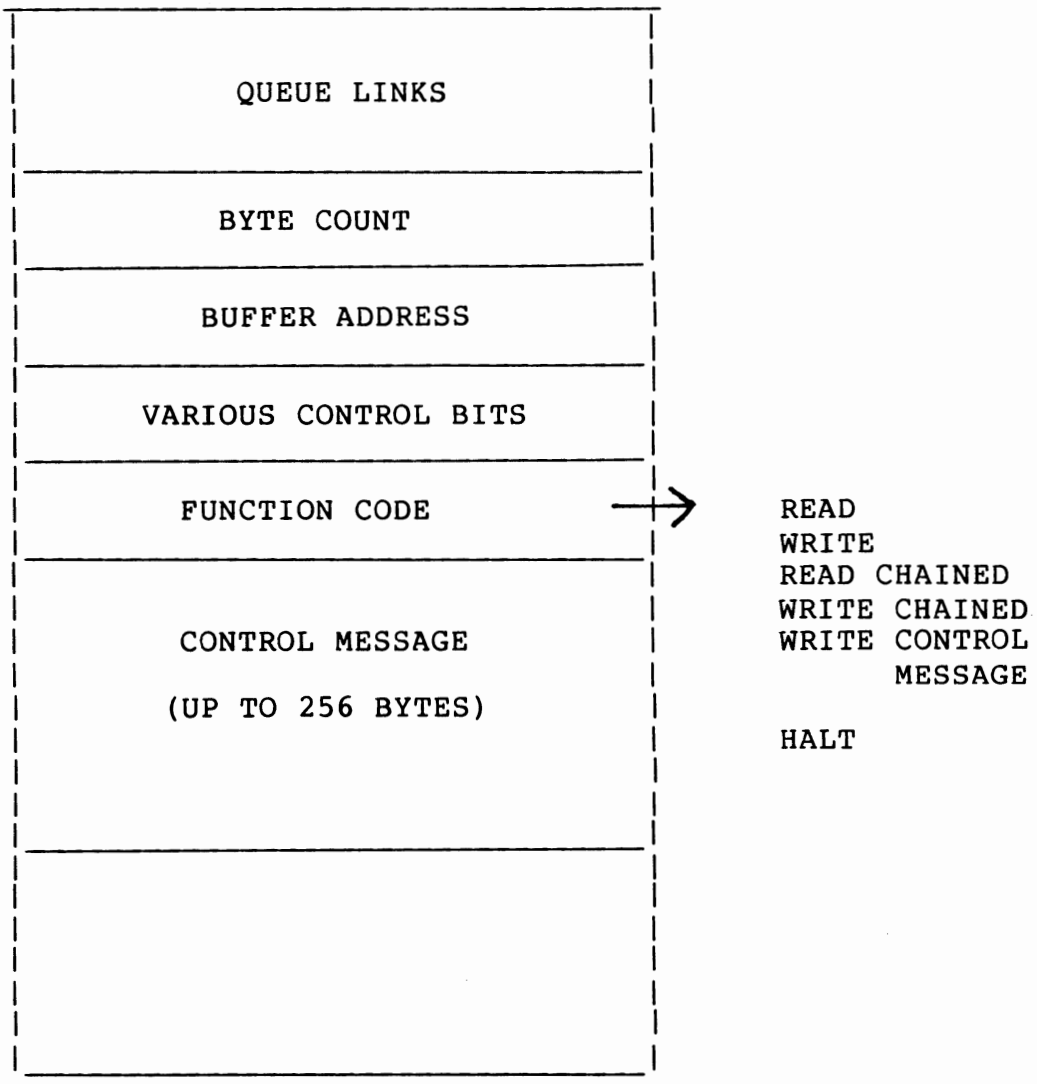


DR32 COMMAND AND BUFFER BLOCKS

(IN USER PROCESS VIRTUAL ADDRESS SPACE)



DR32 COMMAND PACKET



DR32 DEVICE DRIVER

- STARTDATA QIO
  - ADDRESS AND SIZE OF COMMAND BLOCK
  - ADDRESS AND SIZE OF DATA BLOCK
  - PACKET AST ADDRESS

- DRIVER PROBES AND LOCKS REGIONS IN MEMORY

- DRIVER FIELDS PACKET INTERRRUPTS AND QUEUES ASTs.

DR780 PERFORMANCE

- DDI ITSELF - 8 MEGABYTES/SEC.

- SINGLE MEMORY CONTROLLLER - 2-4 MBYTES/SEC.

- INTERLEAVED MEMORY _ 4-7 MYBTES/SEC.

DR32 HIGH LEVEL LANGUAGE SUPPORT ROUTINES

- CALLABLE FROM FORTRAN OR OTHER HIGH LEVEL LANGUAGES

- ROUTINES TO:
  - SET UP
  - START DR32
  - RELEASE PACKETS ONTO FREE QUEUE
  - BUILD PACKETS AND INSERT ONTO INPUT QUEUE
  - GET PACKETS FROM TERMINATION QUEUE

NEW VAX-11 INTERLOCKED QUEUE INSTRUCTIONS

INSQTI, INSQHI REMQTI, REMQHI	INSQUE, REMQUE
SELF RELATIVE	ABSOLUTE
EMPTY QUEUE IS Ø	EMPTY QUEUE CONTAINS ADDRESS
POSITION INDEPENDENT (PIC)	NON-PIC
MUST BE QUADWORD ALIGNED	NO ALIGNMENT NECESSARY
INTERLOCKED FOR MULTI-PROCESSOR ACCESS	NOT INTERLOCKED
MAY FAIL - NEEDS RETRIES	ALWAYS WORKS
4 INSTRUCTIONS	2 INSTRUCTIONS

U S E R

W R I T T E N

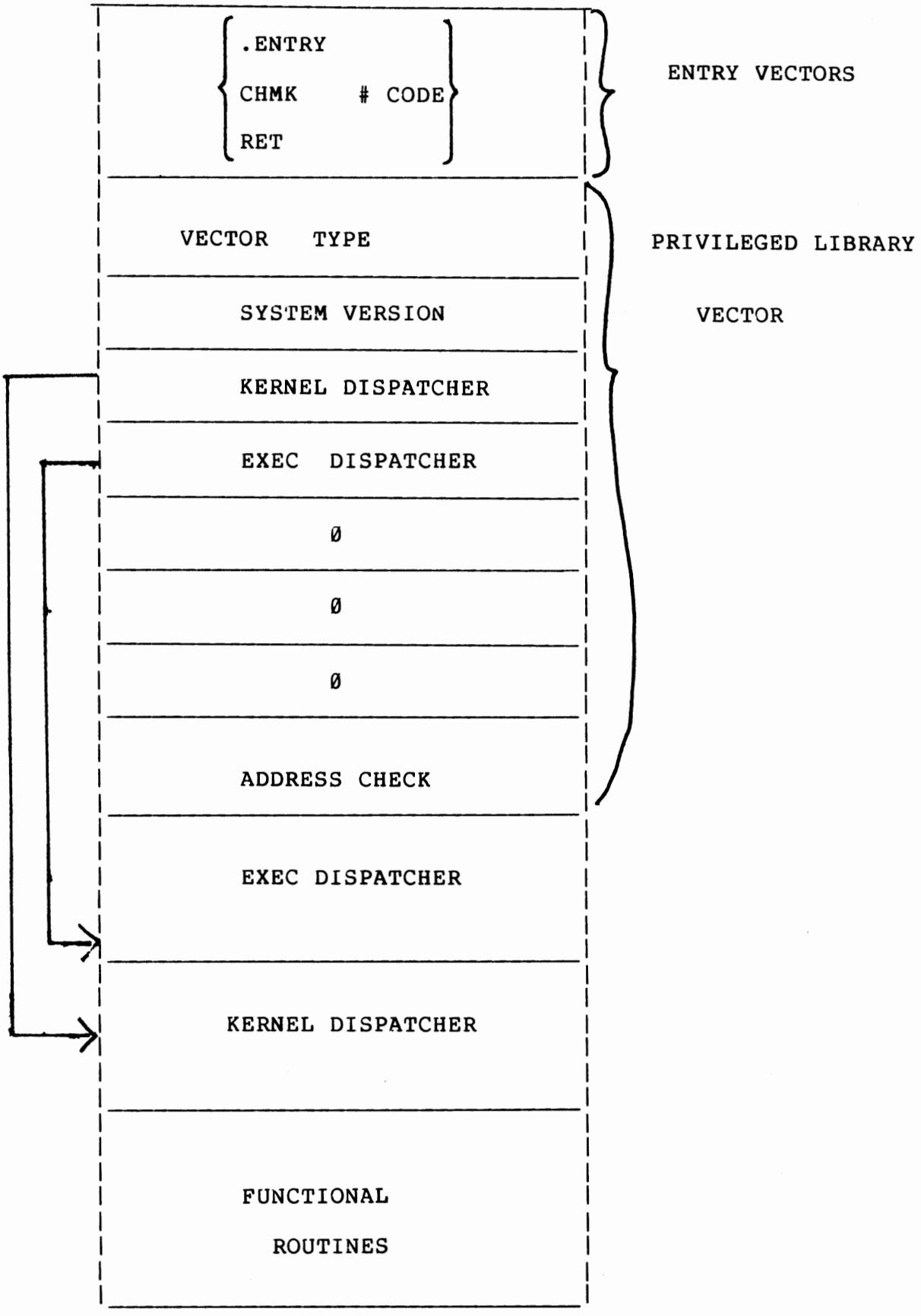
S Y S T E M

S E R V I C E S

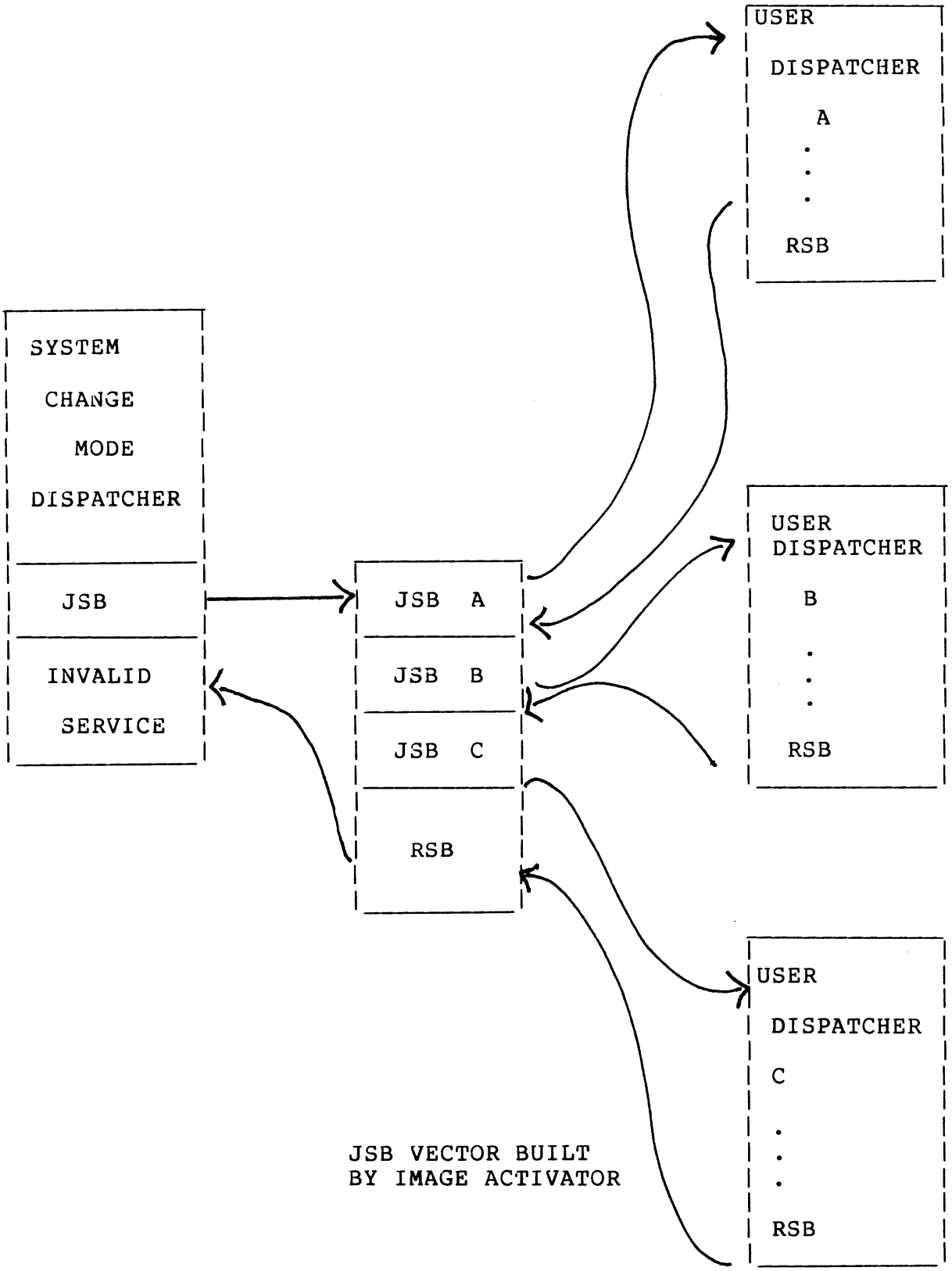
USER WRITTEN SYSTEM SERVICES

- CONTAINED IN PRIVILEGED SHAREABLE IMAGES
- CALLED AND LINKED LIKE NORMAL SHARABLE IMAGES
- INSTALLED WITH /PROTECT
- CONNECTED TO CHANGE MODE DISPATCHER BY IMAGE ACTIVATOR
- ENTERED FOR UNDEFINED CHANGE MODE CODES (NEGATIVE VALUES RESERVED FOR USERS)
- PROVISIONS FOR:
  - MULTIPLE PRIVILEGED SHAREABLE IMAGES
  - COMBINED PRIVILEGED AND NON-PRIVILEGED SHAREABLE IMAGES

STRUCTURE OF PRIVILEGED SHAREABLE IMAGE



OVERVIEW OF CHANGE MODE DISPATCHING



EXEC/KERNEL DISPATCHER STRUCTURE

- INVOKED VIA JSB TO JSB BY SYSTEM CHANGE MODE DISPATCHER

- RETURNS VIA RSB IF CODE NOT HANDLED BY THIS DISPATCHER

- CHECKS FOR REQUIRED ARGUMENTS

- PROBES REQUIRED ARGUMENT LIST

- DISPATCHES TO FUNCTIONAL ROUTINE VIA CASE ON CODE VALUE

FUNCTIONAL ROUTINE STRUCTURE

- BEGIN WITH .ENTRY
- VALIDATE ARGUMENTS AFTER FETCHING/COPYING
- PERFORM SPECIFIED FUNCTION
- RETURN STATUS CODE IN RØ
- EXIT VIA RET INSTRUCTION

ENTRY VECTOR

- FIRST IMAGE SECTION IN PRIVILEGED SHAREABLE IMAGE
- PERMITS RELINKING SHAREABLE IMAGE WITHOUT CALLING IMAGES
- CONTAINS QUADWORD VECTORS OF THE FOLLOWING STRUCTURE:
  - ALIGN QUAD
  - TRANSFER NAME
  - MASK NAME
  - CHMK/CHME # CODE
  - RET

USER SYSTEM SERVICE TEMPLATE

- SOURCE PROVIDED ON BINARY KIT

  [SYSHLP.EXAMPLE]USSDISP.MAR

- LINK COMMAND FILES PROVIDED ALSO

- CONTAINS TEMPLATE DISPATCHERS FOR EXEC AND KERNEL MODES

- MACRO TO EASILY ADD SERVICES

```
DEFINE_SERVICE      NAME 1,  3,  KERNEL

DEFINE SERVICE      NAME 2, 2,   EXEC
```

- ALL REQUIRED DATA STRUCTURES GENERATED BY MACRO:

  - ENTRY VECTOR
  - CASE TABLE
  - ARGUMENT CHECK TABLE

HOW TO WRITE FUNCTIONAL ROUTINES

- READ I/O DRIVERS MANUAL

- BORROW IDEAS FROM EXISTING SERVICES

- CAREFULLY OBSERVE SYNCHRONIZATION RULES:

    - STAY AT IPL$_ASTDEL TO AVOID AST/DELTE
      - BUT NO SYSTEM SERVICES

    - STAY AT IPL$ _ SYNCH TO AVOID RESCHEDULING
      - BUT NO PAGE FAULTS

- DEBUG IN STANDALONE ENVIRONMENT

    - CAN USE $CMEXEC OR $CMKRNL
      TO TEST ROUTINES

    - DEBUG WITH XDELTA (JSB INI$BRK)

PRIVILEGE AMPLIFICATION

- $SETPRV EQUATES EXEC/KERNEL MODE WITH SETPRV PRIVILEGE

- PRIVILEGES CAN BE RAISED DURING SERVICE EXECUTION

- SERVICE IS RESPONSIBLE FOR RESTORING ORIGINAL PRIVILEGES

VAX/VMS

MAGTAPE

TUTORIAL

VAX/ VMS MAGTAPE TUTORIAL

- WHY USE MAGNETIC TAPE?
- TYPES OF MAGTAPES IN VAX/VMS
- GENERAL ANSI LABELS
- MOUNTING ANSI TAPES
- DETAILED ANSI LABELS
- ANSI TRANSPORTABILITY: VAX - DEC
- FOREIGN MAGTAPE FORMAT
- MOUNTING AND USING FOREIGN TAPES

WHY USE MAGNETIC TAPES?

- MASS STORAGE
- EASY, INEXPENSIVE TO TRANSPORT
  - VAX - VAX
  - VAX - DEC
  - VAX - WORLD

MAGNETIC TAPES IN VAX/VMS

- ANSI LABELED MAGNETIC TAPE
- FOREIGN OR UNLABELED TAPE

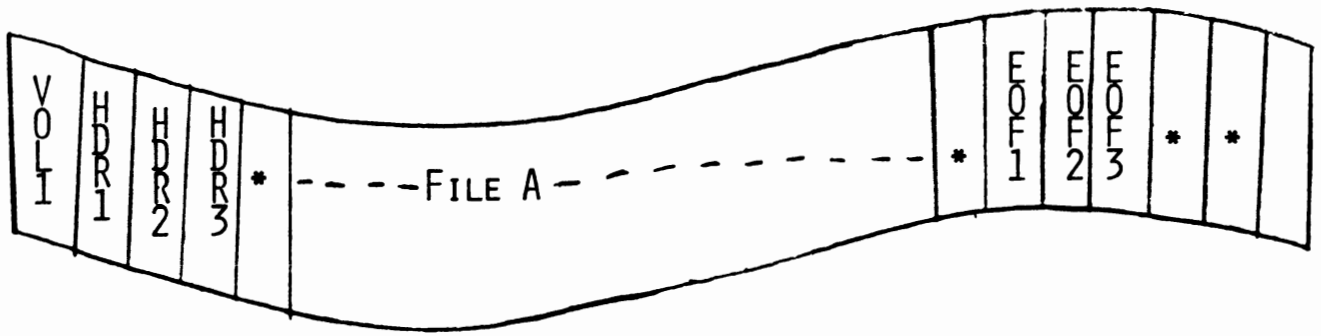
ANSI Magnetic Tape

- Information Interchange
- File Structured Device
- File Integrity
- Volume set/File Access protection
- Multi-File Volumes/Multi-Volume Files
- Blocked variable length records

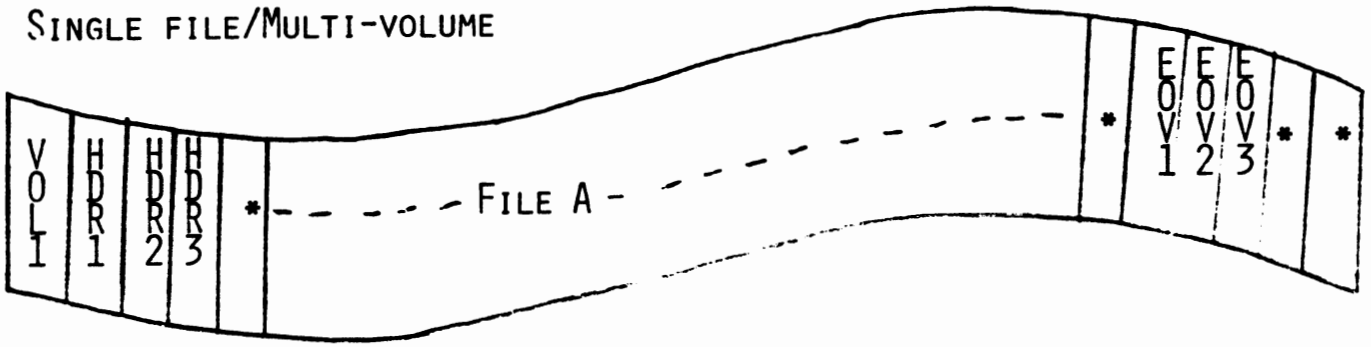
Standard Labels written by VAX/VMS

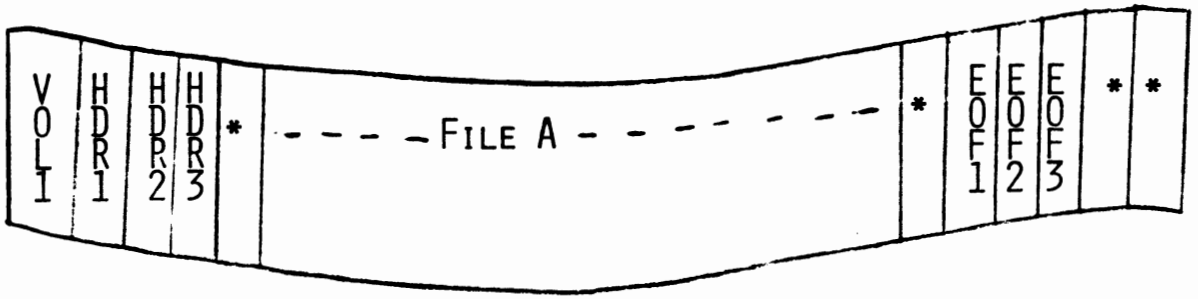
- VOL1
- HDR1
- HDR2
- HDR3
- EOF1 (EOV1)
- EOF2 (EOV2)
- EOF3 (EOV3)

SINGLE FILE/SINGLE VOLUME

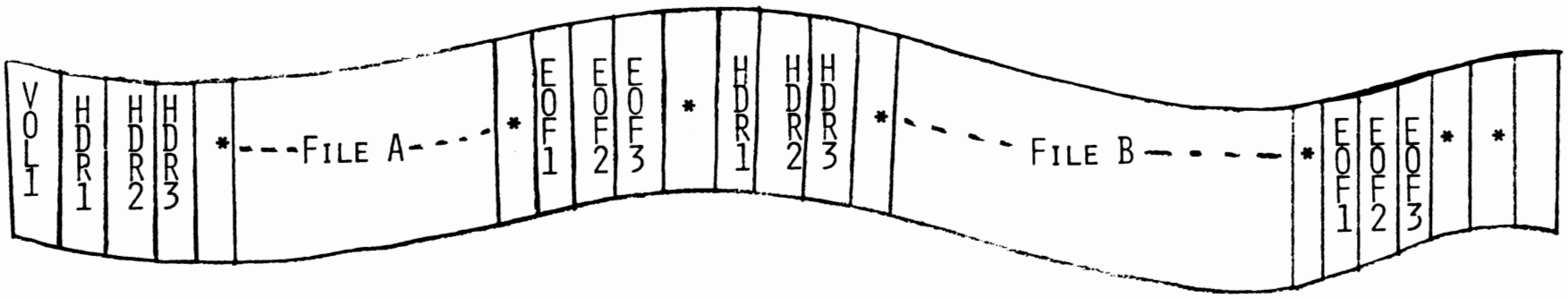


SINGLE FILE/MULTI-VOLUME

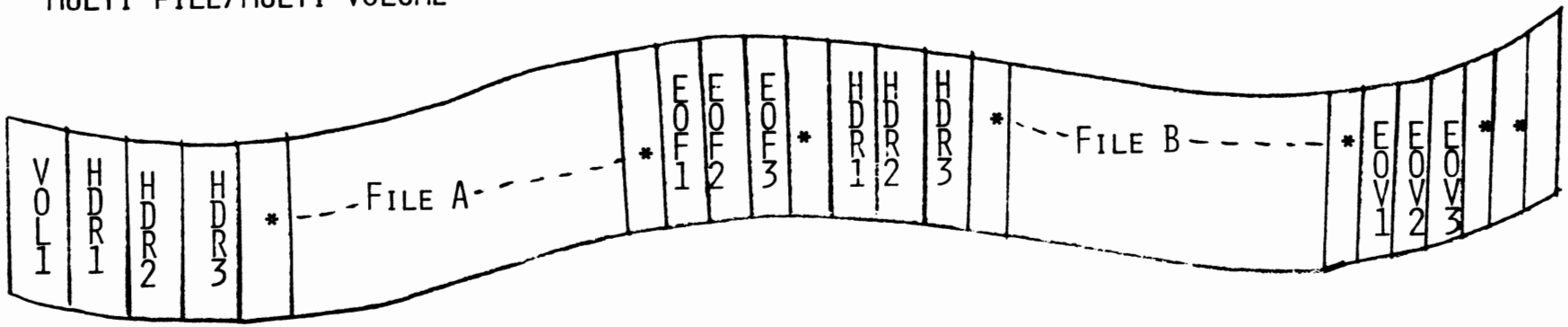


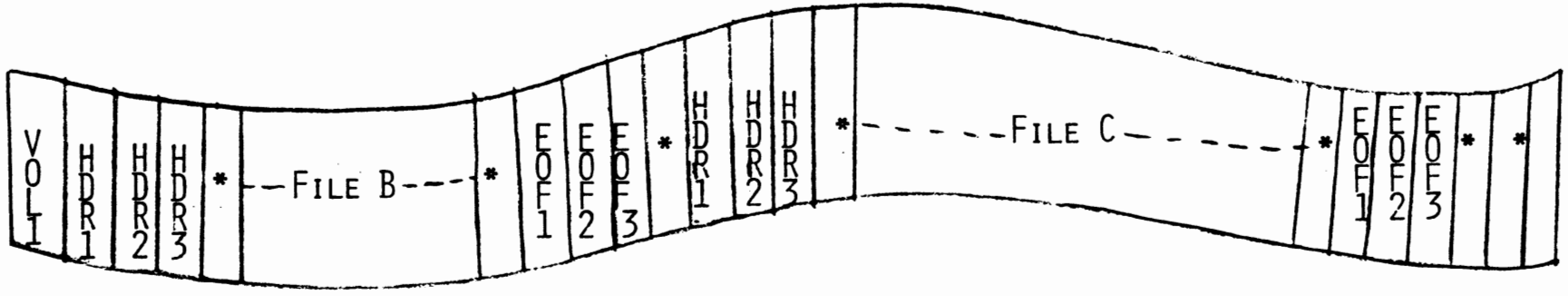


MULTI-FILE/SINGLE VOLUME



MULTI-FILE/MULTI-VOLUME





INITIALIZE MTx: LABEL

/DENSITY = N

/PROTECTION = CODE

/OWNER_UIC = UIC

/OVERRIDE = OPTIONS

. ACCESSIBILITY

. EXPIRATION

MOUNT MTx:,... LABEL,... [LOGNAME]

/BLOCKSIZE = N

/RECORDSIZE = N

/DATA_CHECK [=OPTIONS]

. READ

. WRITE

/LABEL

MOUNT MTx: (CONT.)

/OVERRIDE = OPTIONS

. ACCESSIBILITY
. EXPIRATION
. IDENTIFICATION
. SETID

/OWNER_UIC = UIC

/PROTECTION = CODE

MOUNT MT x: (CONT.)

/PROCESSOR = OPTION

. UNIQUE

. SAME: DEVICE

. FILE-SPEC

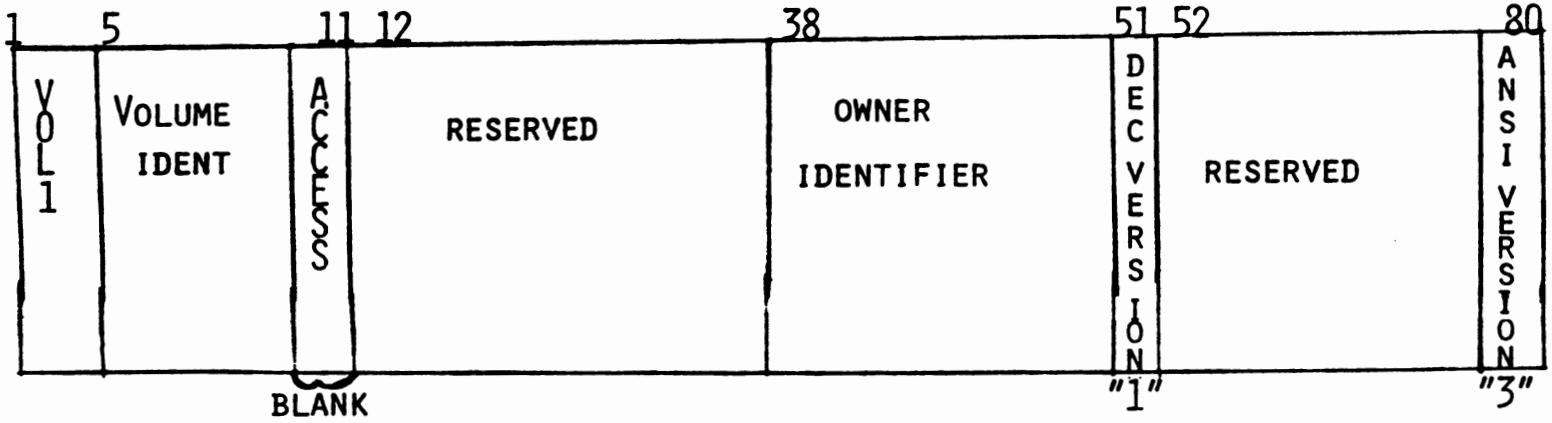
/WRITE

/NOWRITE

/HDR3

/NOHDR3

VOLUME LABEL 1



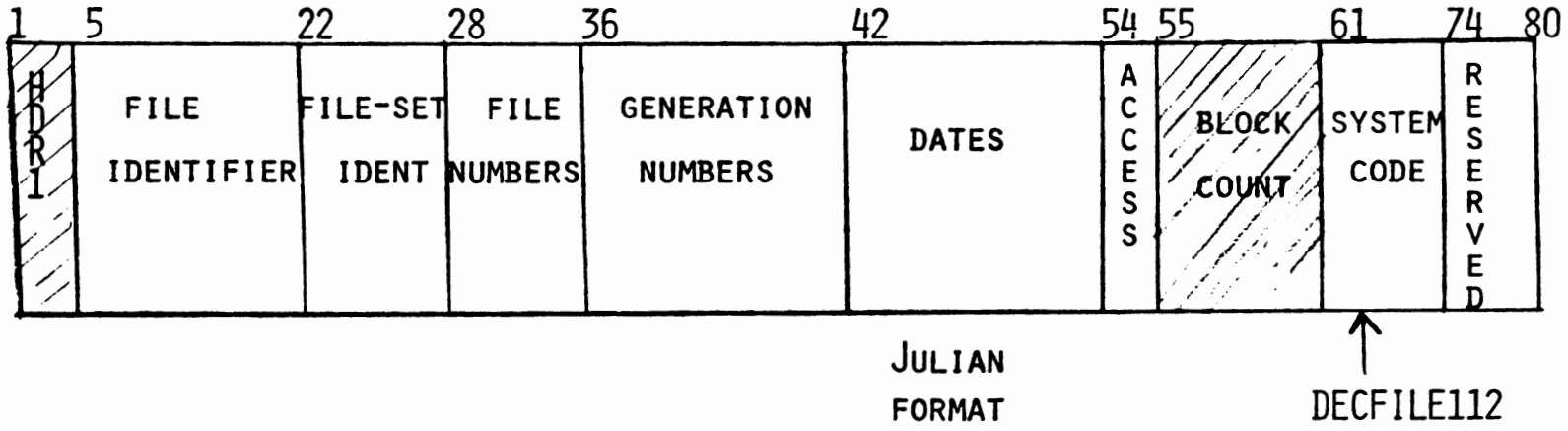
D% M

8 - PDP-8
A - PDP-10
B - PDP-11
C - VAX-11
F - PDP-15
K - DECSYSTEM-20

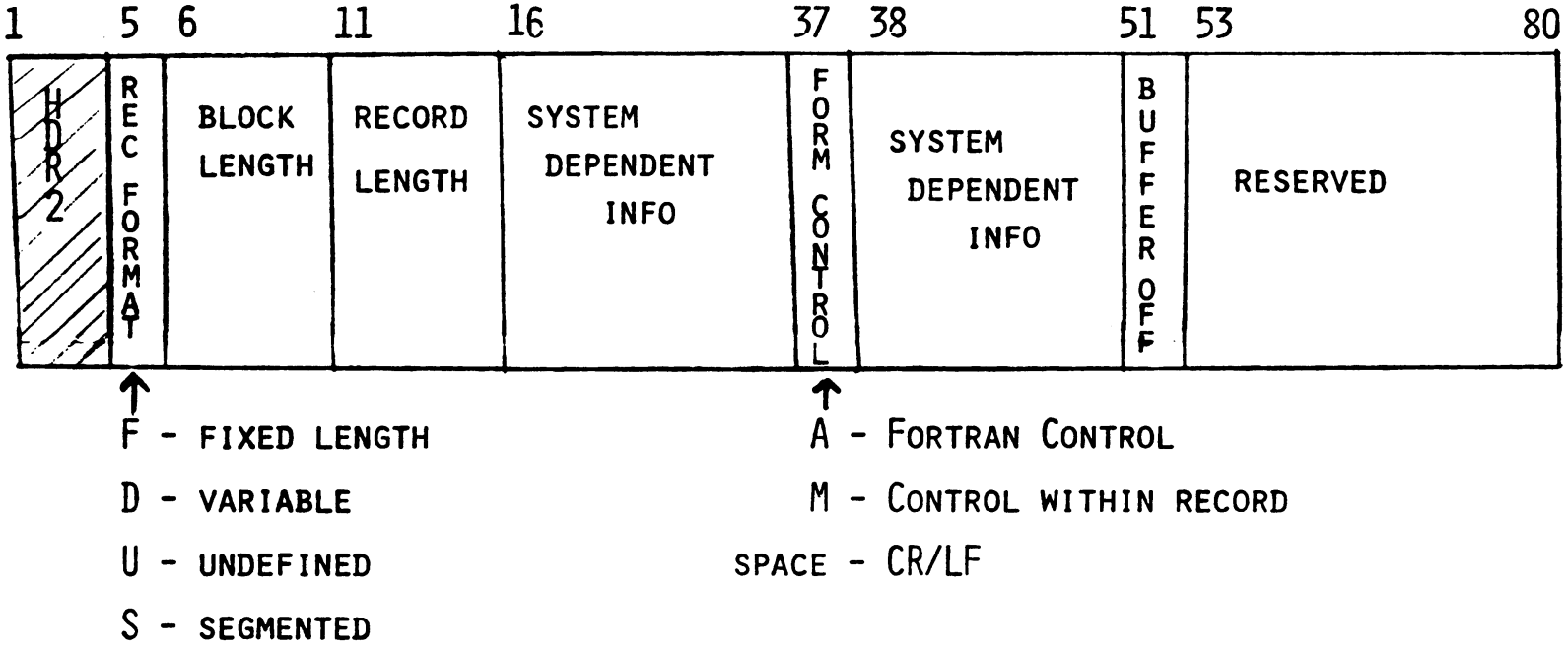
&

VOLUME ACCESS INFO.

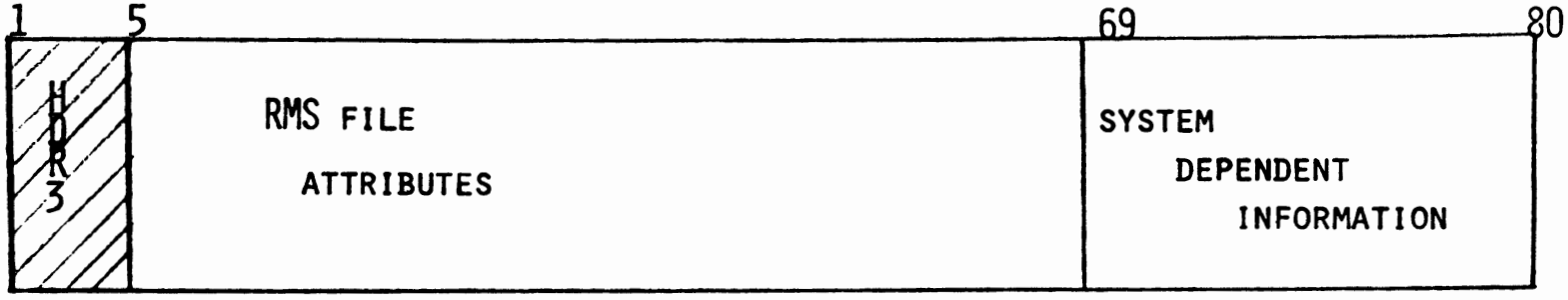
FILE HEADER LABEL 1



File Header Label 2



FILE HEADER LABEL 3

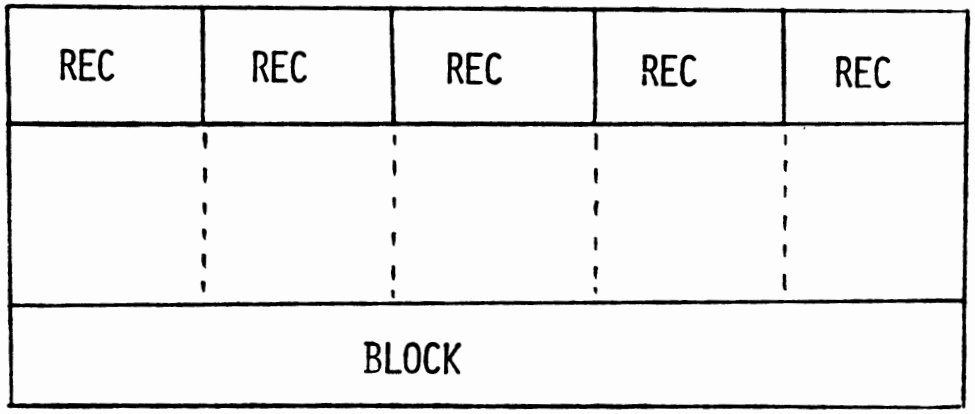


DATA FORMATS

- FIXED LENGTH RECORDS
  - BLOCKED RECORDS
  - UNBLOCKED RECORDS
- VARIABLE LENGTH RECORDS

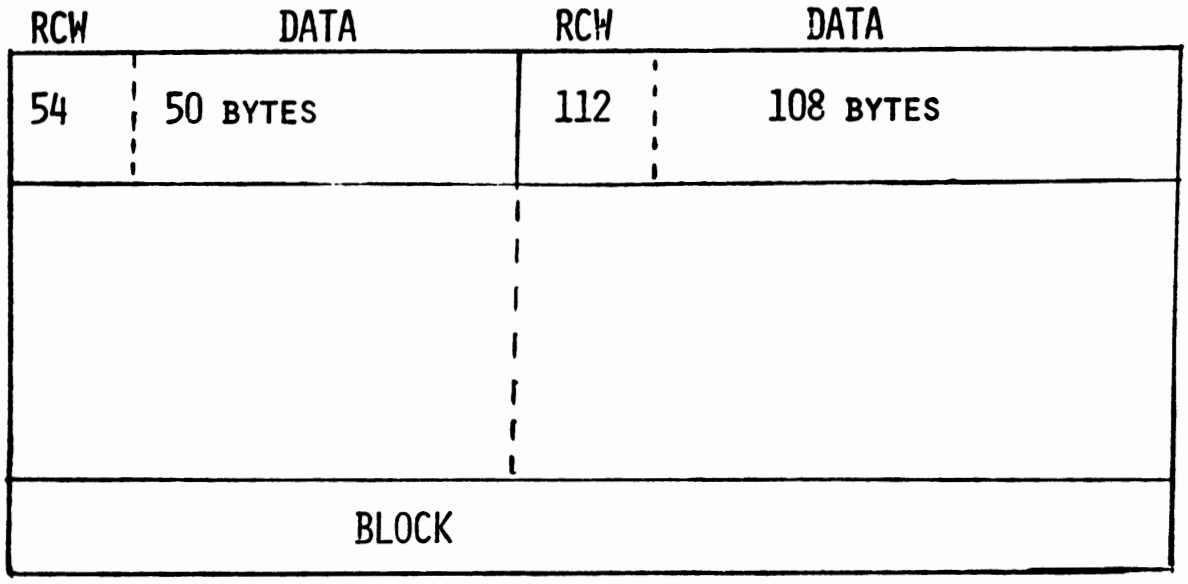
FIXED LENGTH RECORDS

- No physical record boundaries
- Blocks are not padded
- No indication of length within record



VARIABLE LENGTH RECORDS

- REQUIRES RECORD CONTROL WORD
- BLOCKS ARE PADDED TO MAXIMUM LENGTH



TRANSPORTABILITY: VAX-DEC

- MAXIMUM BLOCK SIZE: 2048 BYTES
- NO UNDEFINED FORMAT RECORDS
- NO VFC FILES
- SOME RMS ATTRIBUTES LOST
- DOS FORMAT / FLX

VAX - RT

- SUBSET OF ANSI LEVEL 1
- NO HDR2
- NO MULTIVOLUME
- FIXED LENGTH RECORDS ONLY
- 512 BYTES MAXIMUM BLOCK

VAX - RSTS/E

- FILE NAME TRUNCATION

VAX - PDP-20

- ANSI LEVEL 4
- SPANNED RECORDS
- STANDARD FILENAMES
- NON-BLANK ACCESSIBILITY

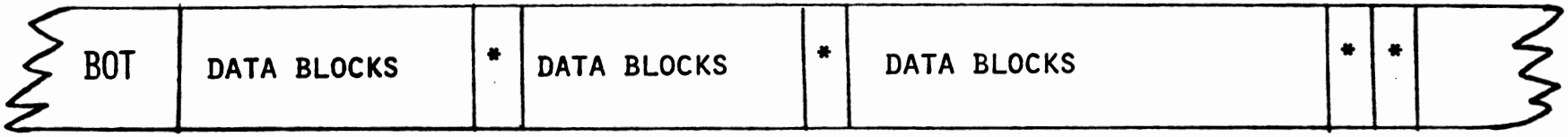
FOREIGN TAPES IN VAX/VMS

- NON-FILE STRUCTURED DEVICE
- ALL NON-ANSI STANDARD TAPES
- LOGICAL I/O OPERATIONS
- SPECIAL PROGRAM NEEDED TO PROCESS LABELS, IF ANY
    - FLX
    - EBCDIC

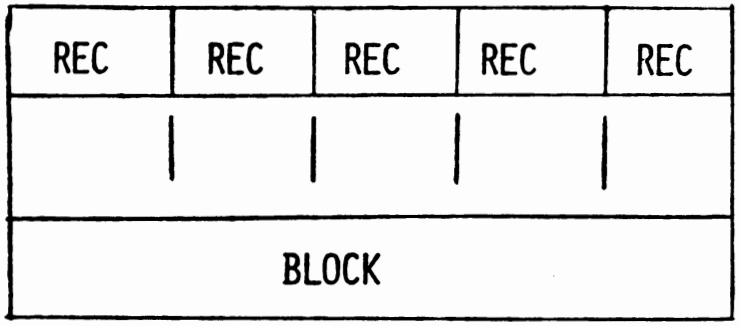
Foreign Tapes (COPY / RMS)

. Tape Format

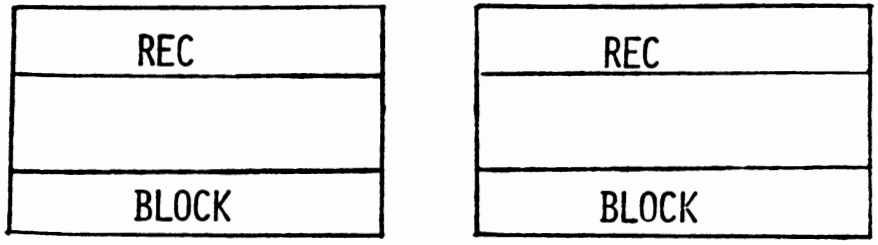
- No tape mark precedes first file
- Tape mark between each file
- Two tape marks at end of volume



Foreign Tapes - Data Format



Blocked Fixed Length Records



Unblocked Records

MOUNT/FOR MTx: OR MOUNT/NOLABEL MTx:

OPTIONS:

/BLOCKSIZE = N

/RECORDSIZE = N

/DATA_CHECK [=OPTIONS]

. READ

. WRITE

/DENSITY = N

/FOREIGN

/NOLABEL

/OWNER_UIC = UIC

/PROTECTION = CODE

/WRITE /NOWRITE

RMS Magnetic Tape FOP options

- * . CIF - create if not found
- * . NEF - not at end of file
- * . POS - current position
- . RWC - rewind on close
- . TWO - rewind on open

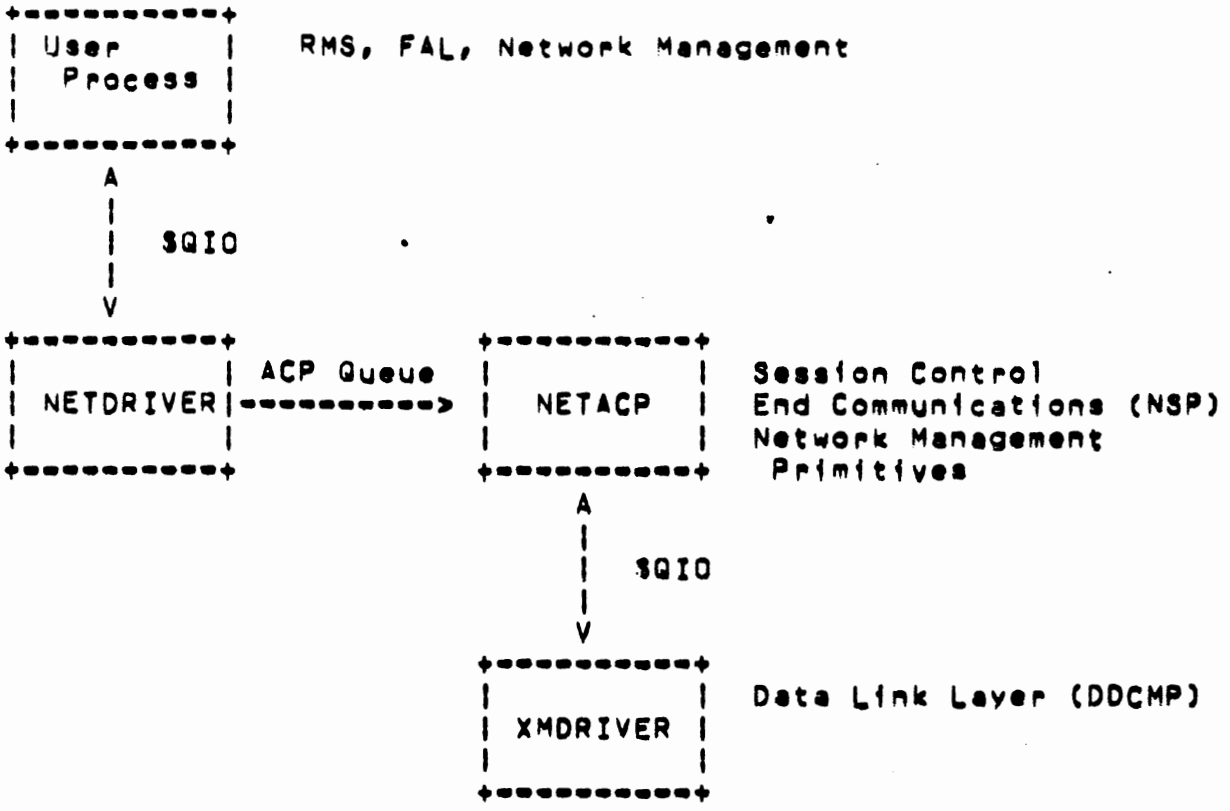
* ANSI magtape only

DECNET - VAX

ARCHITECTURE OVERVIEW

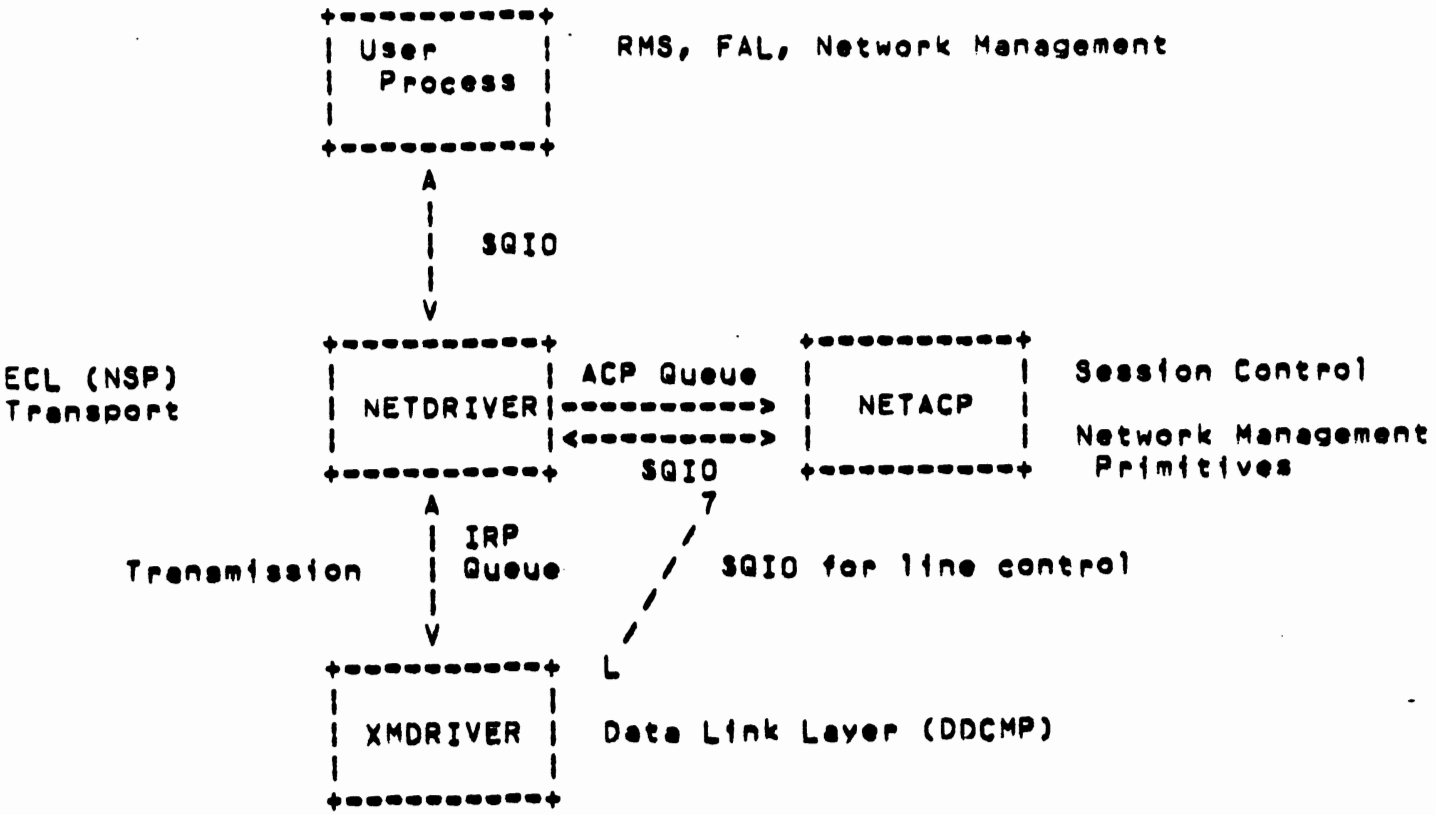
DECNET - VAX

ARCHITECTURE OVERVIEW



DECNET - VAX

ARCHITECTURE OVERVIEW



DECNET - VAX

VMS FACILITIES USED BY DECNET

* $ASSIGN System Service Performs Assign "_NET:" and NETACP Access Function if "NODE::" Present in Device String.
* $ASSIGN Dynamically Creates UCB if Template Bit is Present in UCB of Device.
* LOGINOUT.EXE has Special Internal Interface to Allow Access Control to be Passed to Log in a Network Job.
* Internal IRP Queue - Alternate Mechanism to Queue IRP to a Driver Without Processing At FDT Time. Also Used by $BRDCST, Pager, Swapper Use Similar Mechanism.
* IOCSIOPOST - Call System Address if PID Field of IRP is Negative. Second Mechanism of Inter-driver Communication.

DECNET - VAX

PROCESS OF ACTIVATING A NETWORK OBJECT

* NETACP Receives Connect Request
* Creates Process Running LOGINOUT.EXE
* Passes Access Control To LOGINOUT.EXE
* Passes Command File To Execute
* Defines SYSSNET as NCB for Connect Accept

DECNET - VAX

FORMAT OF NCB

NETWORK CONNECT BLOCK

User Logical Name SYSSNET Contains the NCB

NODE"USER PASS"::	Target Node and Password
"TASK=PROG	Target Identifier
/XX	Internal Link Address
Optional Data	Counted String in 17 Bytes
Destination Descriptor	0 Object number 1 Object number, Object name 2 Object number, Group, User

"

DECNET - VAX

REQUESTING A LOGICAL LINK

(CONNECT INITIATE)

* Assign Channel to "_NET:".
* Access Function to NETACP with NCB Containing Link Address of Zero.
* Optional Data May be Passed.

DECNET - VAX

ACCEPTING A LOGICAL LINK REQUEST

(CONNECT CONFIRM)

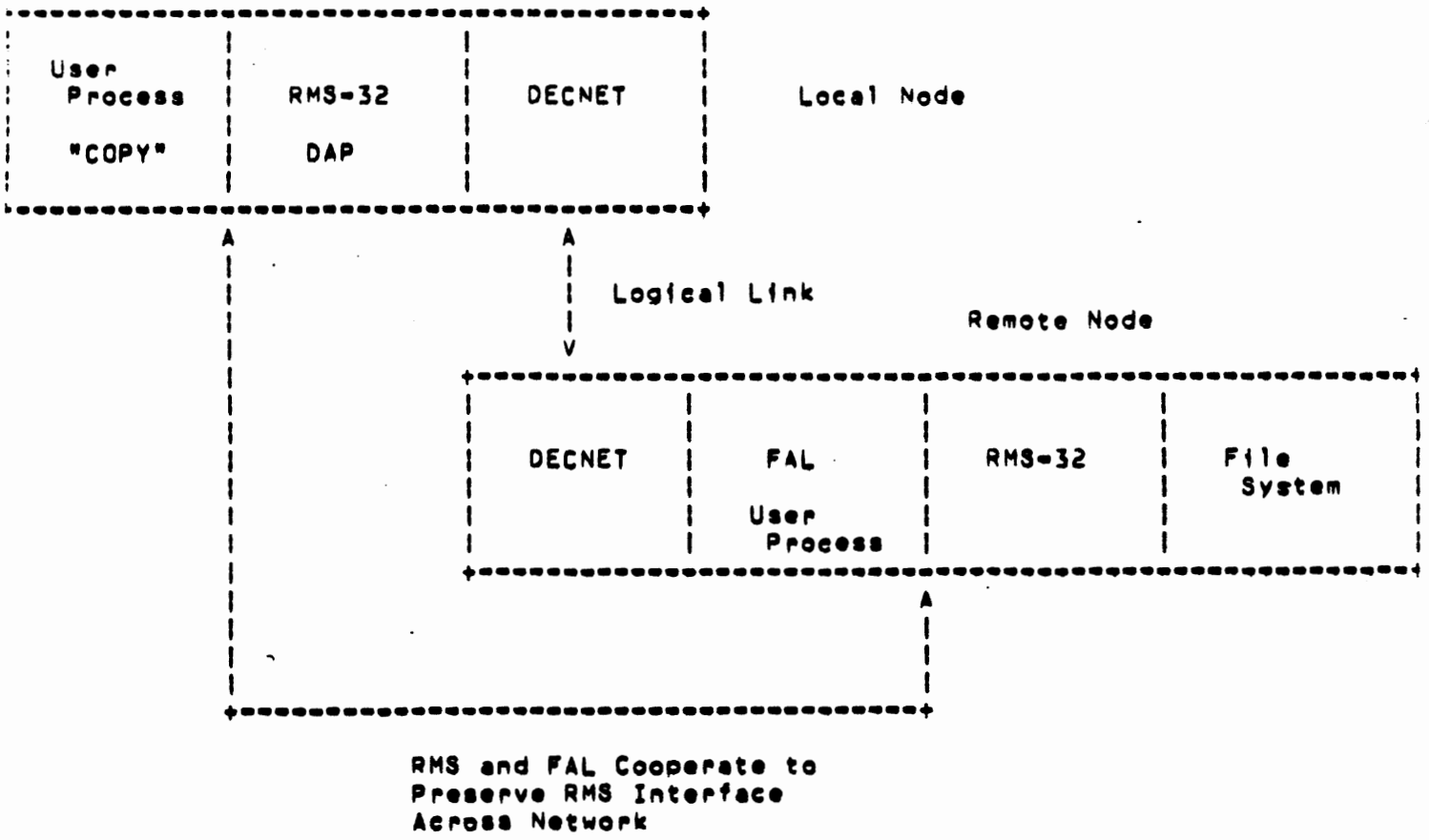
* Assign Channel to "_NET:".

* Access Function to NETACP with NCB from SYSSNET.

* NETACP Checks Link Address for Matching Sequence Number and Process ID.

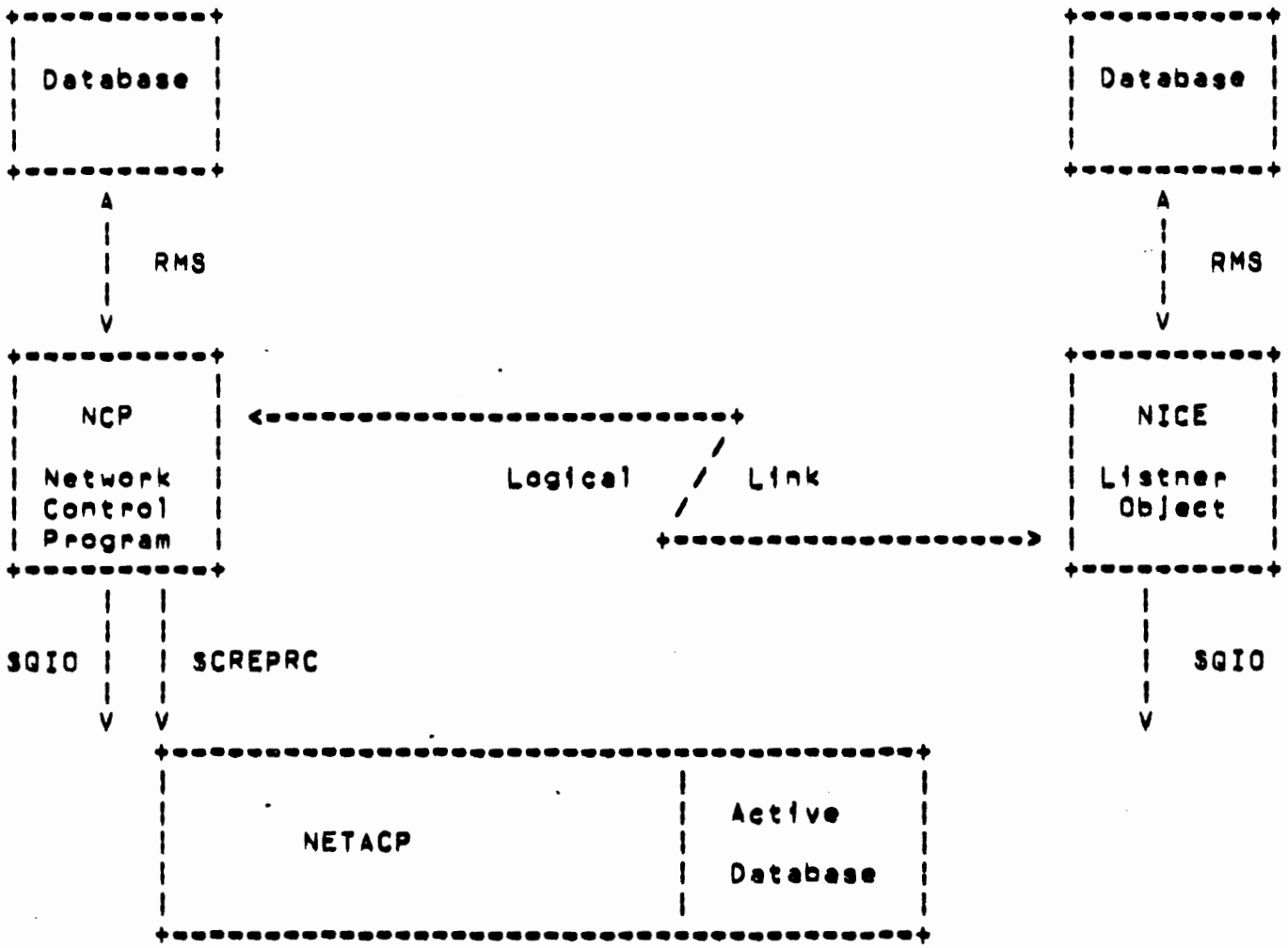
DECNET - VAX

NETWORK FILE ACCESS



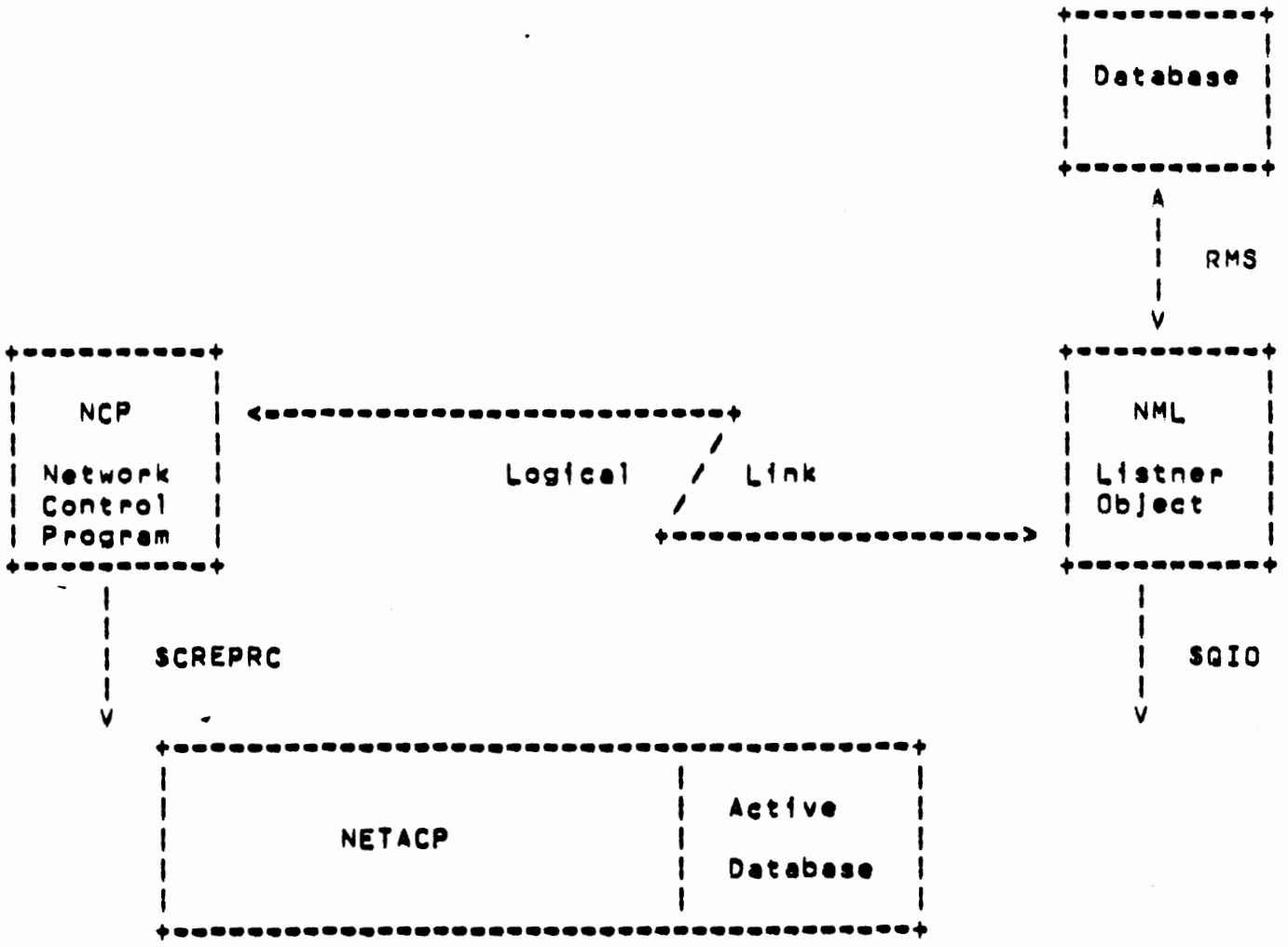
DECNET - VAX

NETWORK MANAGEMENT STRUCTURE



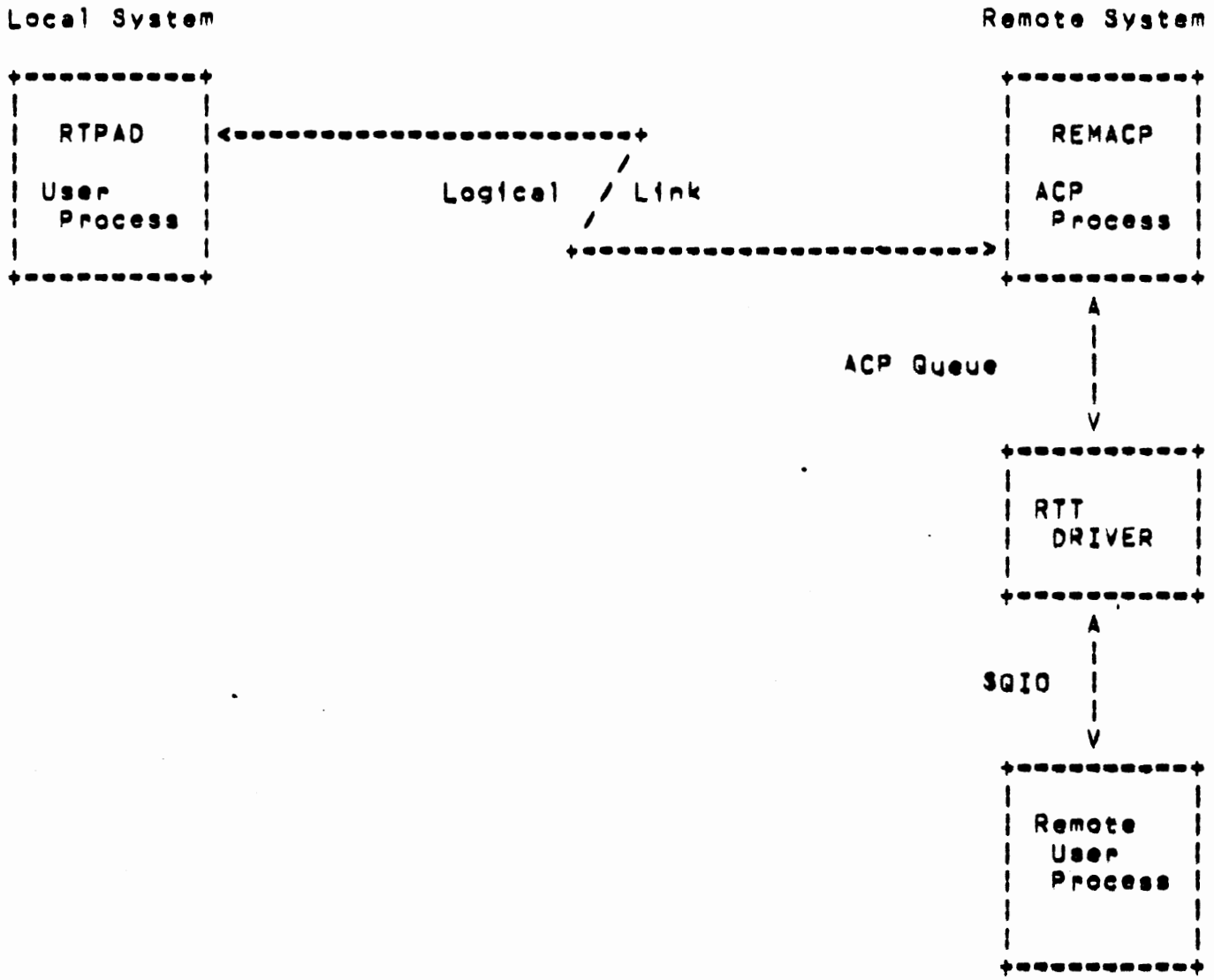
DECNET - VAX

NETWORK MANAGEMENT STRUCTURE



VAX / VMS

REMOTE COMMAND TERMINALS



DECNET - VAX

BRINGING UP THE NETWORK

* SYSGEN Used to Load NETDRIVER.

  ```
  SYSGEN> CONNECT NET /NOADAPTER -
                      /DRIVER = NETDRIVER
  ```

* NCP Used to Build Configuration Database.

  NCP Writes the Files Itself. NICE Not Involved.

  @CONFIGNET.COM

* NCP Mounts and Starts NETACP.

  NCP> SET STATE LOCAL ON

  Create VCB (Volume Control Block)
  Create AQB (ACP Queue Block)
  $CREPRC for NETACP

* NETACP Reads Configuration Database for Local Node.

* NCP Sends Message to NICE to Write Database to NETACP.

DECNET - VAX

BRINGING UP THE NETWORK

* SYSGEN Used to Load NETDRIVER

  ```
  SYSGEN> CONNECT NET /NOADAPTER -
                      /DRIVER = NETDRIVER
  ```

* NCP is started. If NETACP is Not Mounted,

  NCP Starts NETACP with SCREPRC.

* NETACP Mounts and Starts Itself.

  NCP Can Communicate with NML at This Time.

* NCP and NML Communicate to Build Configuration

  Databases. NCP Does Not Write Files Directly.

* NCP and NML Communicate to Load Databases.

  ```
  NCP> SET EXECUTOR ALL
  NCP> SET KNOWN NODES ALL
  NCP> SET KNOWN LINES ALL
  NCP> SET STATE LOCAL ON
  ```

DECNET - VAX

SHUTTING DOWN THE NETWORK

* NCP and NICE Communicate Shutdown Request.

  NCP> SET EXECUTOR STATE OFF
  RESTRICTED
  SHUT

* NETACP Informs All Declared Objects Via Mailbox Message of Shutdown.

* NETACP Dismounts Itself and Exits When State Goes to Off.

* Any Process Having OPER Privilege Can Request or Confirm a Link Until OFF is Entered.

DECNET - VAX

SHUTTING DOWN THE NETWORK

* NCP and NICE Communicate Shutdown Request.

  NCP> SET STATE LOCAL OFF

* NETACP Informs All Declared Objects Via Mailbox Message of Shutdown.
* NETACP Dismounts Itself and Exits When All Links Have Disconnected.
* NCP Cannot Reestablish Link to NICE After Shutdown Request.

DECNET - VAX

DECLARED OBJECT PROCEDURE

* Open Channel to "_NET" with Mailbox.
* Attempt to Declare Name and/or Number.
  On Failure, Exit to Prevent Race for
  First Confirm.
* Do Not Use SYS$NET for Connect Confirm.
* Read Mailbox to Obtain Connect Requests.
* Exit Upon Network Shutdown Notification.
* Ignore Unknown Mailbox Message Codes.

VAX

ARCHITECTURE DEVELOPMENTS

DILEEP P. BHANDARKAR

VAX/PDP-11 ARCHITECTURE MANAGEMENT

DIGITAL EQUIPMENT CORPORATION

VAX ARCHITECTURE DEVELOPMENTS

TOPICS

- ARCHITECTURE MANAGEMENT
- ARCHITECTURE ENHANCEMENTS
  - INTERLOCKED QUEUES FOR MULTIPROCESSORS
  - EXTENDED RANGE FLOATING POINT
  - FLOATING FAULTS

VAX ARCHITECTURE MANAGEMENT

GOALS

- ENSURE CONSISTENT IMPLEMENTATION OF VAX FAMILY ARCHITECTURE.
- PROVIDE RATIONAL PROCESS FOR EVOLUTION

VAX ARCHITECTURE MANAGEMENT

ACTIONS

- CENTRALIZED ORGANIZATION RESPONSIBLE FOR ARCHITECTURE MANAGEMENT PROCESS.
- CREATE/MAINTAIN ARCHITECTURE CONTROL DOCUMENT
- ADMINISTER WELL-DEFINED CONTROL PROCESS
- VALIDATE IMPLEMENTATIONS FOR COMPLIANCE

VAX ARCHITECTURE MANAGEMENT

EVOLUTION OF ARCHITECTURE

- CREATION/DEFINITION
- DOCUMENTATION
- APPROVAL/RATIFICATION
- IMPLEMENTATION
- VALIDATION
- CONTROL
- MEASUREMENT AND EVALUATION
- EVOLUTION
- RETIREMENT

VAX ARCHITECTURE MANAGEMENT

MAJOR ACTIVITIES

- MAINTAIN ARCHITECTURE CONTROL DOCUMENTATION
  - ANSWER QUESTIONS
  - RESOLVE AMBIQUITIES
  - CORRECT ERRORS
  - DOCUMENT MISSING ASPECTS
- ADMINISTER PROCESS FOR ARCHITECTURE CONTROL
- ARCHITECTURE VALIDATION

VAX ARCHITECTURE MANAGEMENT

ARCHITECTURE SPECIFICATION

- ARCHITECTURE DOCUMENT
  - "PRINCIPLES OF OPERATION"
  - COMPLETE DEFINITIVE SPECIFICATION
  - USED BY HARDWARE IMPLEMENTORS TO BUILD COMPATIBLE MACHINES.
  - USED BY SOFTWARE DESIGNERS TO WRITE COMPATIBLE SOFTWARE.
- SPECIFICATION TECHNIQUE
  - COMBINATION OF PROSE AND PSEUDO-ALGOL
  - REQUIRED BEHAVIOR SPECIFIED
  - IMPLEMENTATION SPECIFIC ASPECTS CLEARLY DEFINED.

VAX ARCHITECTURE MANAGEMENT

ARCHITECTURE CONTROL PROCESS

- REVIEW BOARD CONSISTS OF REPRESENTATIVES OF ALL MAJOR HARDWARE AND SOFTWARE DEVELOPMENT GROUPS.

- PROPOSALS TO CHANGE OR EXTEND ARCHITECTURE
    - SUBMITTED IN WRITING
    - BENEFITS/RATIONALE
    - IMPACT, COSTS
    - PLAN FOR MANAGEMENT OF CHANGE

- DECISION BY REVIEW BOARD CONSENSUS

- CHAIRMAN OF REVIEW BOARD LEGISLATES ACTION IN THE ABSENCE OF CONSENSUS.

VAX ARCHITECTURE MANAGEMENT

ELEMENTS FOR SUCCESSFUL OPERATION

- VISIBILITY
- CREDIBILITY
- MANAGEMENT SUPPORT/EMPHASIS
- RESPONSIVENESS
- GOOD WORKING RELATIONSHIP WITH IMPLEMENTORS
- DEVELOPMENT ENVIRONMENT
- TRACK RECORD

VAX ARCHITECTURE MANAGEMENT

MAJOR EXTENSIONS APPROVED

- SELF-RELATIVE INTERLOCKED QUEUE INSTRUCTIONS
- EXTENDED RANGE FLOATING POINT
- FLOATING FAULTS

INTERLOCKED QUEUES

- MOTIVATION
- DESCRIPTION
  - DATA STRUCTURE
  - OPERATIONS
- ARCHITECTURAL DETAILS

INTERLOCKED QUEUES

MOTIVATION

- SYNCHRONIZATION IN MULTIPROCESSOR SYSTEM
  - MULTIPORT MEMORIES
  - SYMMETRICAL MULTIPLE PROCESSORS
  - CPU AND INTELLIGENT I/O ADAPTER
- ORIGINAL VAX QUEUES INADEQUATE
  - GLOBAL INTERLOCK REQUIRED FOR MULTIPROCESSOR SYNCHRONIZATION.
  - REASONABLE FOR INTER-PROCESS SYNCHRONIZATION IN A UNIPROCESSOR SYSTEM.

INTERLOCKED QUEUES

SOLUTION

- NO CHANGES TO OLD QUEUES
- ADD NEW INTERLOCKED QUEUES
  - NEW INSTRUCTIONS
  - RETROFIT ALL VAX-11/780'S

INTERLOCKED QUEUES

DATA STRUCTURE

- DOUBLY LINKED LIST WITH QUEUE HEADER
- LINKS SELF-RELATIVE TO QUEUE ELEMENT
  - HEADER WITH ZERO LINKS ⟶ EMPTY QUEUE
  - POSITION INDEPENDENT
  - INDEPENDENT PLACEMENT IN COOPERATING PROCESSES
- QUADWORD ALIGNED

INTERLOCKED QUEUES

OPERATIONS

- RESTRICTED TO HEAD AND TAIL OF QUEUE
- REMOVE ENTRY FROM HEAD
- REMOVE ENTRY FROM TAIL
- INSERT ENTRY AT HEAD
- INSERT ENTRY AT TAIL

HARDWARE INTERLOCKS

- WHEN SET, OTHER INTERLOCKED READS TO SAME DATUM DISALLOWED BY HARDWARE.
- NON-INTERLOCKED READS MAY BE ALLOWED
- INTERLOCK SET BY INTERLOCKED READ OPERATION
- INTERLOCK RELEASED BY INTERLOCKED WRITE OPERATION
- INTERLOCKED READ IS ALWAYS FOLLOWED BY INTERLOCKED WRITE
- INTERLOCKED WRITE IS ALWAYS PRECEDED BY INTERLOCKED READ
- IMPLEMENTATIONS MAY INTERLOCK MEMORY ENTITY LARGER THAN DATA BEING ACCESSED.
- INSTRUCTIONS USING INTERLOCKS
    - BBSSI, BBCCI
    - ADAWI
    - REMQHI, REMQTI, INSQHI, INSQTI

INTERLOCKED QUEUES

SIMPLIFIED DESCRIPTION

- READ QUEUE HEADER INTERLOCKED
- IF SECONDARY INTERLOCK IS SET

  THEN RELEASE HARDWARE INTERLOCK
  SET CONDITION CODES
  TERMINATE INSTRUCTION

  ELSE SET BIT Ø OF HEADER TO SET SECONDARY INTERLOCK
  WRITE MODIFIED HEADER INTERLOCKED
  PERFORM QUEUE OPERATION
  RELEASE SECONDARY INTERLOCK USING HARDWARE
  INTERLOCK.
  SET CONDITION CODES
  DONE

INTERLOCKED QUEUE

INSTRUCTIONS

OPCODE	MNEMONIC	OPERAND SPECIFIERS
5C	INSQHI	ENTRY.AB, HEADER.AQ
5D	INSQTI	ENTRY.AB, HEADER.AQ
5E	REMQHI	HEADER.AQ, ADDR.WL
5F	REMQTI	HEADER.AQ, ADDR.WL

- ONE BYTE OPCODES CHOSEN TO FACILITATE RETROFIT TO VAX-11/780.
- CONDITION CODE SETTINGS CHOSEN TO EXPLOIT BRANCHES ON COMBINATIONS.

INTERLOCKED QUEUES

CONDITION CODES

		N	Z	V	C
INSERTION					
	SUCCEEDED	Ø	FIRST ENTRY	Ø	Ø
	FAILED	Ø	Ø	Ø	1
		Ø	SUCCESSFULLY INSERTED FIRST ENTRY	Ø	INSERTION FAILED DUE TO SECONDARY INTERLOCK
REMOVAL					
	SUCCEEDED	Ø	QUEUE EMPTY	NO ENTRY TO REMOVE	Ø
	FAILED	Ø	Ø	Ø	1
		Ø	QUEUE EMPTY	NO ENTRY TO REMOVE	REMOVAL FAILED DUE TO SECONDARY INTERLOCK

INTERLOCKED QUEUES

APPLICATIONS

- SET SOFTWARE INTERLOCK REALIZED WITH A QUEUE

```
1$:  INSQHI    . . .        ; WAS QUEUE EMPTY?
     BEQL      2$           ; YES
     BCS       1$           ; TRY INSERTING AGAIN
     CALL      WAIT         ; NO, WAIT

2$:  ENTER CRITICAL SECTION
```

- RELEASE SOFTWARE INTERLOCK REALIZED WITH A QUEUE

```
3$:  REMQTI    . . .        ; REMOVED LAST?
     BEQL      4$           ; YES
     BCS       3$           ; TRY AGAIN
     CALL      ACTIVATE     ; ACTIVATE WAITERS

4$:  CONTINUE
```

INTERLOCKED QUEUES

APPLICATIONS

- REMOVE AND PROCESS QUEUE ENTRIES

```
1$:   REMQTI    . . .        ; ANYTHING REMOVED?
      BVS       2$           ; NO
      .
      .
      PROCESS REMOVED ENTRY
      .
      .
      BR        1$

2$:   BCS       1$           ; TRY REMOVING AGAIN
      QUEUE EMPTY
```

INTERLOCKED QUEUES

VAX/VMS USAGE

- MULTIPORT MEMORIES - MA78Ø SUPPORT
- HIGH SPEED I/O BUS ADAPTER - DR78Ø SUPPORT

FLOATING POINT ENHANCEMENTS

- MOTIVATION
- DESCRIPTION
    - NEW DATA TYPES
    - OPERATIONS
    - EXCEPTION CONDITIONS
- COMPATIBILITY ISSUES
- IEEE FLOATING POINT STANDARDS SUBCOMMITTEE

FLOATING POINT ENHANCEMENTS

MOTIVATION

- INADEQUATE RANGE OF DOUBLE FLOATING FOR FORTRAN PROGRAMS MOVED OVER FROM MAINFRAMES:
  - CDC 6600 10^{+307}
  - IBM 370 10^{+75}
  - VAX/PDP-11 10^{+38}
- INFLUENCED BY INTERACTION WITH IEEE FLOATING POINT STANDARDS SUBCOMMITTEE.

FLOATING POINT ENHANCEMENTS

SOLUTION

- IMMEDIATE PROBLEM SOLVED BY ADDING NEW EXTENDED RANGE DOUBLE PRECISION DATA TYPE.

- FURTHER RANGE EXTENSIONS PROVIDED BY NEW QUAD PRECISION DATA TYPE.

- FORMATS OF NEW DATA TYPES SIMILAR TO EXISTING FLOATING DATA TYPES.

FLOATING POINT ENHANCEMENTS

NEW DATA TYPES

	G FLOATING	H FLOATING
SIZE	64 BITS	128 BITS
EXPONENT SIZE	11 BITS	15 BITS
FRACTION	53 BITS	113 BITS
SMALLEST NUMBER	0.56×10^{-308}	0.84×10^{-4932}
LARGEST NUMBER	0.9×10^{308}	0.59×10^{4932}
PRECISION (DECIMAL)	15 DIGITS	33 DIGITS

FLOATING POINT ENHANCEMENTS

OPERATIONS

- ALL OPERATIONS CURRENTLY DEFINED FOR D_FLOATING ARE ALSO AVAILABLE FOR G_FLOATING.
- EQUIVALENT SET OF OPERATIONS FOR H_FLOATING
- NEW SET OF CONVERTS FROM/TO H, AND BETWEEN G AND H
- NO CONVERTS BETWEEN D AND G
    - ARCHITECTURE ALLOWS D AND G TO COEXIST IN A PROGRAM
    - USERS ANTICIPATED TO USE ONE OF THE TWO AVAILABLE FORMATS.
    - FORTRAN DOES NOT ALLOW MIXING WITHIN SINGLE SUBROUTINE.
- 56 NEW INSTRUCTIONS

FLOATING POINT ENHANCEMENTS

EXCEPTION HANDLING

- VAX-11/780 TRAPS ON FLOATING POINT EXCEPTIONS
  - DEFAULT RESULT STORED MAY HAVE OVERWRITTEN SOURCE OPERAND.
  - SOURCE OPERANDS NOT ALWAYS AVAILABLE FOR DEBUGGING
  - PROGRAM COUNTER POINTS TO NEXT INSTRUCTION
  - DIFFICULT TO DETERMINE INSTRUCTION CAUSING EXCEPTION

FLOATING POINT ENHANCEMENTS

EXCEPTION HANDLING

- DESIRE AMONG NUMERICAL ANALYSTS TO AUTOMATICALLY CONTINUE COMPUTATION BEYOND EXCEPTIONS.
    - SYMBOLS FOR OVERFLOW, UNDERFLOW, INFINITY, ETC.
    - RULES FOR OPERATIONS INVOLVING SYMBOLS
- ENHANCED ARCHITECTURE IMPLEMENTS FAULTS ON FLOATING POINT EXCEPTIONS.
    - FAULT CONDITION HANDLERS CAN BE WRITTEN TO PROCEED BEYOND EXCEPTIONS.

FLOATING POINT EHHANCEMENTS

FLOATING FAULTS

- NO RESULT IS STORED
- ALL SIDE EFFECTS OF INSTRUCTION ARE RESTORED
- SAVED PROGRAM COUNTER ON EXCEPTION STACK POINTS TO INSTRUCTION CAUSING EXCEPTION.
- EXCEPTION CODE IDENTIFYING FAULT CONDITION PUSHED ON STACK
- UNDERFLOW FAULT CAN BE DISABLED, ZERO STORED AS RESULT

CONDITION HANDLER

- EXCEPTION CONDITION HANDLER CAN:
    - SIMULATE INSTRUCTION AND USE A RESERVED OPERAND TO ENCODE SYMBOL TO BE STORED AS RESULT.
    - UPDATE SAVED PC TO POINT TO NEXT INSTRUCTION
    - RETURN
- RESERVED OPERAND FAULT IF SYMBOL IS USED AS SOURCE OPERAND

FLOATING POINT ENHANCEMENTS

COMPATIBILITY ISSUES

- NEW INSTRUCTIONS
  - AVAILABLE ONLY IF OPTIONAL EXTENDED FLOATING POINT FEATURE IS INSTALLED.
  - SOFTWARE EMULATION OF NEW INSTRUCTIONS SUPPORTED IN RUN-TIME LIBRARY.
- FORTRAN USERS
  - DEFAULT FOR REAL*8 IF D_FLOATING
  - COMPILE TIME SWITCH ALLOWS G_FLOATING FOR EACH SEPARATELY COMPILED PROGRAM.
  - COMPILER GENERATES INSTRUCTIONS FOR H_FLOATING IF REAL*16 IS USED.

FLOATING POINT ENHANCEMENTS

COMPATIBILITY ISSUES

- FAULTS VERSUS TRAPS
    - ALL FUTURE VAX FAMILY PROCESSORS WILL IMPLEMENT FLOATING FAULTS.
    - DIFFERENT EXCEPTION CODES FOR FAULTS AND TRAPS
    - RUN TIME LIBRARY INCLUDES CONDITION HANDLER TO CONVERT FAULTS TO TRAPS.
- NO CHANGES FOR PROGRAMS THAT DO NOT SPECIFY TRAP HANDLERS

IEEE FLOATING POINT STANDARD

- WORKING GROUP FORMED IN 1977
- THREE PROPOSALS
  - KAHAN, COONEN, STONE (KCS) ——> INTEL 8087
  - PAYNE, STRECKER (PS) ——> ENHANCED VAX
  - FRALEY, WALTHER (FW) ——> HP RESEARCH LAB
- WORKING GROUP MEETINGS MOSTLY IN SAN FRANCISCO BAY AREA
- MAJORITY OF WORKING GROUP PREFERS KCS

IEEE FLOATING POINT STANDARD

SIMILARITIES BETWEEN VAX AND KCS

- SIZE OF EXPONENT AND FRACTION FIELDS
- NORMALIZED BINARY FRACTION WITH HIDDEN BIT
- SIGNED MAGNITUDE
- ROUNDING RULES SIMILAR THOUGH NOT EXACTLY SAME
- VAX HAS SLIGHTLY WIDER RANGE FOR NORMALIZED NUMBERS

IEEE FLOATING POINT STANDARD

MAJOR DIFFERENCES BETWEEN VAX AND KCS

- KCS USES TWO RESERVED OPERAND
  - DENORMALIZED NUMBERS ON UNDERFLOW
  - SYMBOLS
- KCS SPECIFIES INFINITY ARITHMETIC ON OVERFLOW
  - NO CONSENSUS WITHIN WORKING GROUP
  - PROPOSED ARITHMETIC UNPROVEN
- EXTENDED REGISTERS INCLUDED IN KCS AS OPTIONAL FEATURE
  - HIGH LEVEL LANGUAGE IMPLICATION NOT ADDRESSED
  - UNNECESSARY INTRODUCTION OF ADDITIONAL DATA TYPES
- VAX ALLOWS USER-DEFINED EXCEPTION HANDLING
  - FLEXIBLE ACCORDING USER AND PROBLEM NEEDS
  - DOES NOT PREMATURELY FREEZE UNPROVEN IDEAS INTO ARCHITECTURE.
- THREE LEVELS OF PRECISION AND RANGE PROVIDED BY VAX

FLOATING POINT ENHANCEMENTS

NEW CAPABILITIES

- THREE LEVELS OF RANGE AND PRECISION WITH F, G, AND H
  - EACH SUCCESSIVE LEVEL OF PRECISION ALSO PROVIDES ADDITIONAL RANGE.
  - RANGE AND PRECISION CAN BE TRADED OFF AGAINST SIZE AND SPEED.
  - INTERMEDIATE RESULTS CAN BE COMPUTED IN NEXT HIGHER LEVEL.
  - SIMPLE SOLUTION TO OVERFLOW/UNDERFLOW PROBLEMS
  - RANGE AND PRECISION OF H IS MORE THAN ADEQUATE FOR MOST PROBLEMS IN THIS DECADE.
- EXCEPTION HANDLING
  - ENCOURAGE EXPERIMENTATION AND RESEARCH
  - DOES NOT LOCK IN USERS TO A PREMATURE STANDARD
  - SATISFACTORY SIMPLE DEFAULT FOR MAJORITY OF USERS

V A X / V M S

RECORD MANAGEMENT SERVICES

V A X / V M S

RECORD MANAGED I/O SYSTEM

- o DEVICE INDEPENDENT
- o FIXED AND VARIABLE FORMATS
- o USED BY ALL LANGUAGES
- o AUTOMATIC AND CONTROLLABLE
- o PROTECTION AND SHARING
- o IMPLEMENTED AS SYSTEM SERVICES

VAX-11 RMS RECORD PROCESSING

- UNIT RECORD DEVICES
  - READ AND WRITE VIRTUAL (QIO)
  - TERMINALS, MAILBOXES, FOREIGN VOLUMES
- BLOCK STRUCTURED DEVICES
  - RECORD FORMATTING, BLOCKING
  - ANSI MAGTAPE, DISK FILES
- RANDOM ACCESS DEVICES
  - RANDOM RECORD RETRIEVAL, INSERTION
  - INDEXING BY KEY VALUE (ISAM)

VAX-11 RMS FILE PROCESSING

- PARSE, SEARCH, OPEN, CREATE, EXTEND
- FILENAME PARSING
  - DEVICE, DIRECTORY DEFAULTING
  - DIRECTORY AND FILENAME WILDCARDING
  - RECURSIVE LOGICAL NAME TRANSLATION
- CHANNEL ASSIGNMENT AND FILE ACCESS
- FILE CREATION, ALLOCATION, AND EXTENSION

VAX-11 RMS RECORD OPERATIONS

- GET, PUT
  - SEQUENTIAL ACCESS FOR ALL DEVICES
  - RANDOM ACCESS FOR MOST DISK FILES
- UPDATE - REWRITE RECORDS, DISK ONLY
- DELETE - REMOVE RECORDS, RELATIVE AND INDEXED SEQUENTIAL FILES ONLY

F U L L Y I N T E G R A T E D I N S Y S T E M

- SYSTEM CODE OPERATING IN PROCESS CONTEXT
- NO TAILORING AT LINK TIME
- PAGES AGAINST SYSTEM WORKING SET
- GLOBAL SHARED FILE DATABASE

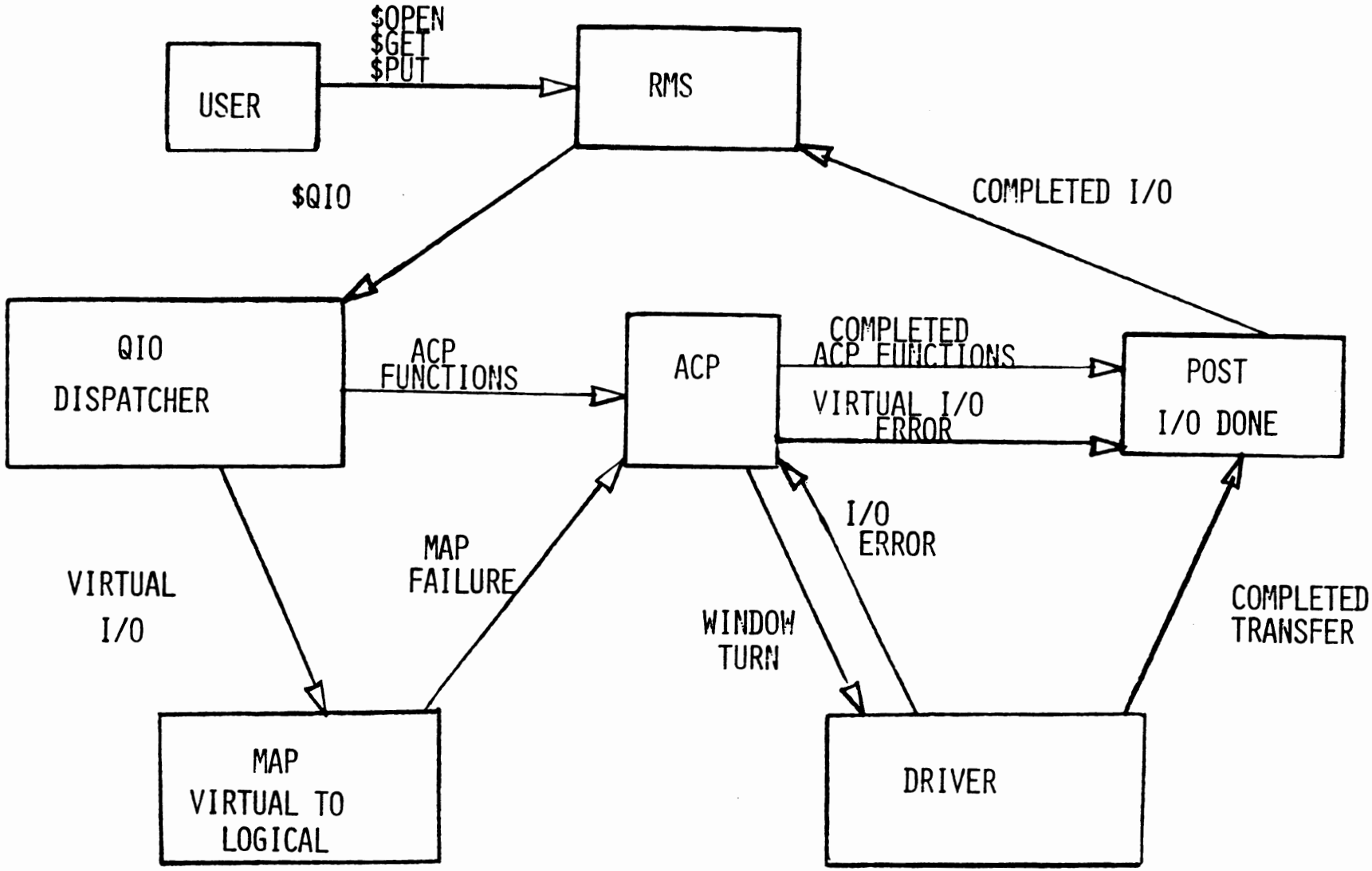
  ALLOWS:

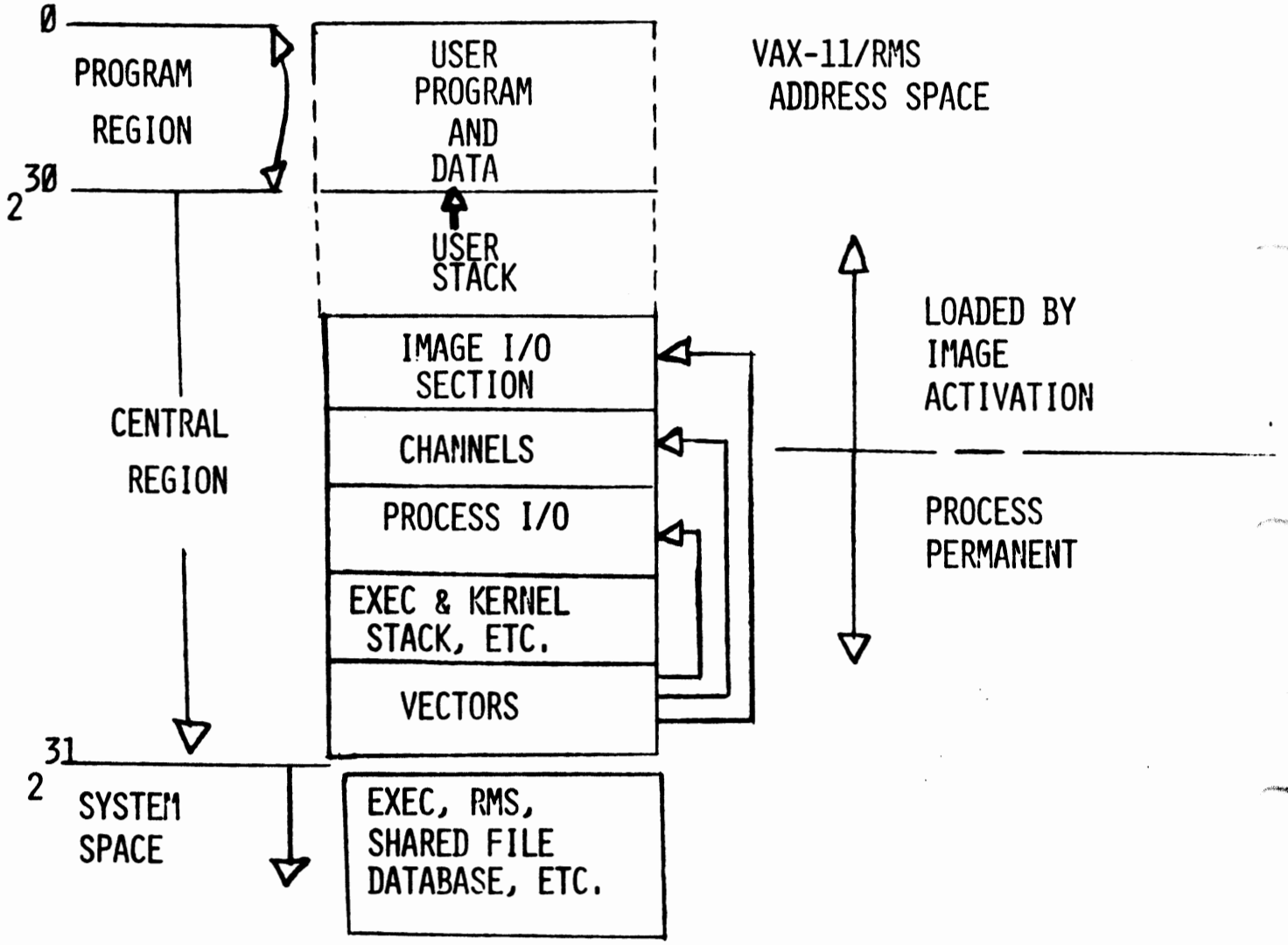
  - TRUE RECORD LOCKING
  - BUCKET CACHING FOR SHARED FILES

INTEGRITY/ PROTECTION

- PROTECTED I/O DATA BASE, BUFFERS, CHANNELS
- OPERATIONS COMPLETED AT I/O RUNDOWN
- FILE INTEGRITY MAINTAINED AT IMAGE EXIT

FILE I/O CONTROL FLOW




0
PROGRAM
REGION
2 30
CENTRAL
REGION
2 31
SYSTEM
SPACE
USER
PROGRAM
AND
DATA
USER
STACK
IMAGE I/O
SECTION
CHANNELS
PROCESS I/O
EXEC & KERNEL
STACK, ETC.
VECTORS
EXEC, RMS,
SHARED FILE
DATABASE, ETC.
VAX-11/RMS
ADDRESS SPACE
LOADED BY
IMAGE
ACTIVATION
PROCESS
PERMANENT

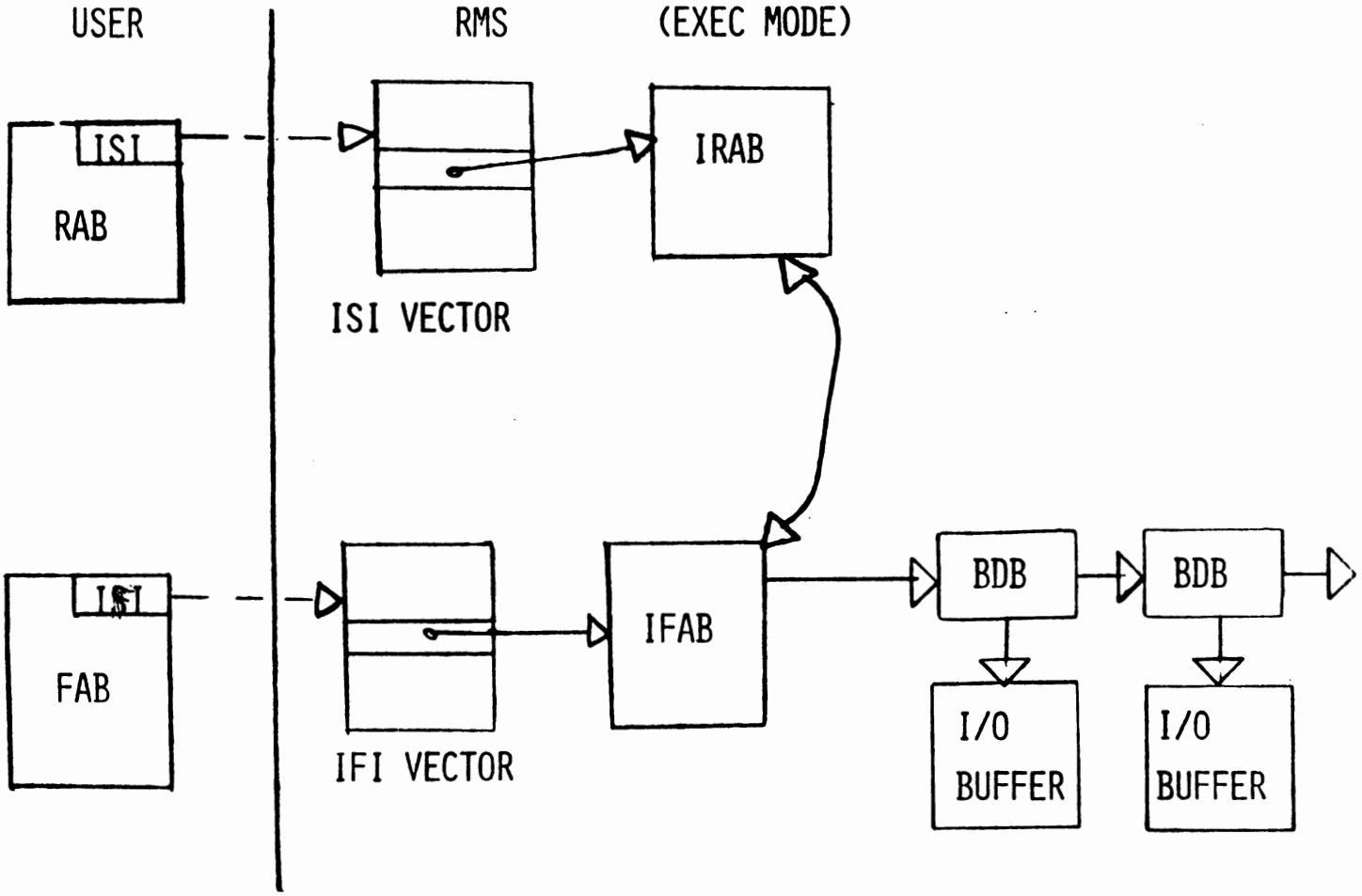
PER PROCESS DATA STRUCTURE

- 1 PAGE/FILE BASIC ALLOCATION STRATEGY
- CONTAINS PROTECTED COPY OF FAB AND RAB INFORMATION (IFAB + IRAB)
- USER FAB + RAB USED ONLY DURING EXECUTION OF RMS CALLS
- IFAB + IRAB:

  ALL FILE AND RECORD CONTEXT

  CONTAINS ALL INFORMATION NEEDED FOR I/O RUNDOWN

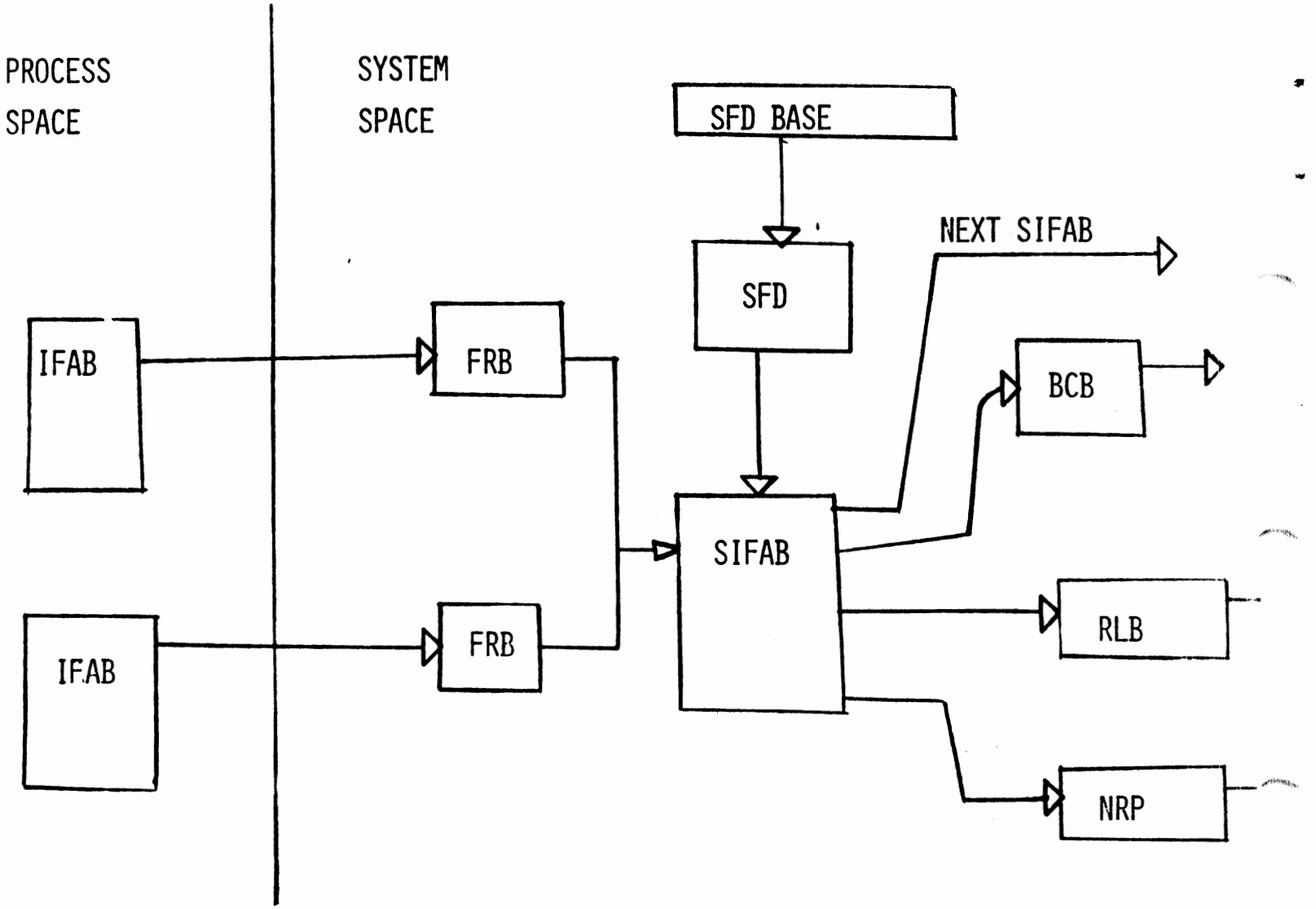
VAX-11 /RMS CONTROL BLOCKS



SHARED FILE DATABASE

- ALLOCATED FROM SYSTEM PAGED POOL - SO SPACE
- DESCRIBES ALL GLOBAL STRUCTURES
  - FILE RESOURCE BLOCKS (FRB'S)
  - SHARED FILE LIST (SIFAB'S)
  - BUCKET CONTROL BLOCKS (BCB'S)
  - RECORD LOCK BLOCKS (RLB'S)
  - NEXT RECORD POSITION (NRP'S) - ISAM ONLY

SHARED FILE DATABASE



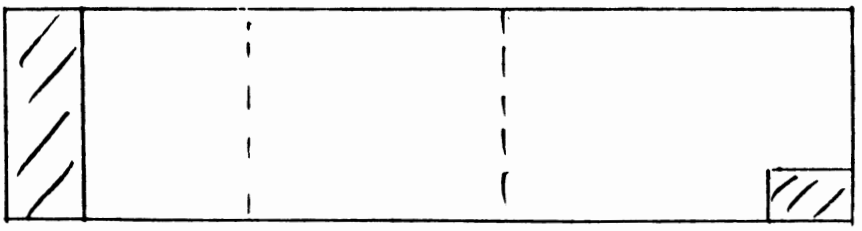
INDEXED FILE ORGANIZATION
(MULTI-KEY)

- DATA RECORDS ARE STORED IN ASCENDING COLLATING SEQUENCE BY PRIMARY KEY
- RECORDS MAY BE RETRIEVED RANDOMLY VIA PRIMARY OR ALTERNATE KEYS
- RECORDS MAY BE RETRIEVED SEQUENTIALLY VIA PRIMARY OR ALTERNATE KEYS AND ARE PRESENTED IN ASCENDING COLLATING SEQUENCE

G O A L S

- A CONTENT ADDRESSABLE RECORD ACCESS CAPABILITY
- ALTERNATE KEY CAPABILITY
- UNIFORM RANDOM ACCESS TIME TO RECORDS
- GOOD SEQUENTIAL PERFORMANCE ON PRIMARY KEY
- REASONABLE SEQUENTIAL PERFORMANCE ON ALTERNATE KEYS
- UNIQUE ADDRESS FOR A RECORD FOR THE LIFE OF THE FILE
- GOOD SPACE UTILIZATION
- PROVIDE FOR SYSTEMS FAILURE WITHOUT LOSS OF USED DATA RECORDS

B U C K E T S



- BUCKETS ARE A USER-DEFINED NUMBER OF VIRTUAL BLOCKS
- ALL BUCKETS ARE FORMATTED WITH 15 BYTES OF RMS INFORMATION
- ALL BUCKETS CONTAIN RECORDS
- THE TYPE OF RECORD IN A PARTICULAR BUCKET DEPENDS ON HOW THE BUCKET IS BEING USED IN THE LOGICAL STRUCTURE OF THE FILE

ISAM FILE INTEGRITY

- BUCKET INTEGRITY CHECKS
- DATA AND INDEX LEVELS LINKED HORIZONTALLY
  - WILL SURVIVE INDEX UPDATE FAILURE
- DATA RECORD INTEGRITY CHECKS
- RECORD OPERATION FAILURES DON'T COMPROMISE FILE STRUCTURE

VAX-11 RMS ISAM PROCESSING

- RECORD AND BUCKET LOOKS ARE DISTINCT
  - RFA IS BASIS OF RECORD LOCK
  - BUCKETS ONLY LOCKED FOR PROCESSING
  - BUCKET ACCESS IS QUEUED
- BUCKETS ARE SPLIT 50/50
  - IMPROVES AVERAGE BUCKET LOADING
  - ASCENDING ORDER SPECIAL CASES

VAX-11 RMS PERFORMANCE FEATURES

SEQUENTIAL FILES

- MULTI-BLOCK COUNT (MBC IN RAB)
  - CONTROLS SIZE OF I/O TRANSFERS
- READ-AHEAD (RAH ROP OPTION IN RAB)
  - ASYNCHRONOUSLY READ NEXT BUFFER
- WRITE-BEHIND (WBH ROP OPTION IN RAB)
  - ASYNCHRONOUSLY WRITE MODIFIED BUFFERS
- MULTI-BUFFER COUNT (MBF IN RAB)
  - NUMBER OF BUFFERS FOR RAH/WBH

VAX-11 RMS PERFORMANCE FEATURES

RELATIVE AND INDEX SEQUENTIAL FILES

- o DEFERRED WRITE (DFW FOP OPTION IN FAB)
  - WRITE OF MODIFIED BUCKET DEFERRED UNTIL NECESSARY (NON-SHARED ONLY)
- o MULTI-BUFFER COUNT (MBF IN RAB)
  - CACHE COMMONLY REFERENCED BUCKETS
- o ASYNCHRONOUS OPERATION (ASY ROP OPTION IN RAB)
  - OVERLAP I/O WITH COMPUTING
  - OVERLAP I/O ON DIFFERENT DEVICES

SOME VMS SPECIFICS

RMS IS A PIC IMAGE EXECUTING IN THE PROCESS CONTEXT. IT IS A SYSTEM SERVICE AND EXECUTES AT EXEC LEVEL OR EXEC AST LEVEL AFTER STALLING FOR I/O. I/O COMPLETION IS WAITED FOR IN USER MODE SO THAT USER LEVEL AST'S MAY BE PROCESSED DURING SYNCHRONOUS OPERATIONS, OR USER LEVEL PROCESSING MAY OCCUR DURING ASYNCHRONOUS OPERATIONS.

RMS CODE RESIDES IN SYSTEM S0 SPACE AND PAGES IN THE SYSTEM WORKING SET ALONG WITH PAGEABLE EXECUTIVE CODE, SYSTEM MESSAGE FILE, AND PAGED DYNAMIC MEMORY (INCLUDING SHARED FILE DATABASE).

- SYSGEN PARAMETER SYSMWCNT CONTROLS SIZE OF SYSTEM WORKING SET.
- SYSPAGING CONTROLS PAGING OF PAGEABLE EXECUTIVE CODE - NOT RMS.
- MONITER PAGING BEHAVIOR WITH PA OPTION OF DISPLAY - WATCH FOR SYSTEM PAGE FAULT RATE.

RMS CODE USES PSECTS TO GROUP CODE BY FUNCTIONALITY, FILE ORGANIZATION, AND USAGE. WIDE VARIETY OF OPERATIONS OCCURING SIMULTANEOUSLY WILL REQUIRE LARGEST WORKING SET.

RMS INTERNAL CONTROL BLOCKS (IFAB'S, IRAB'S, BDB'S, RLB'S, NRP'S) AND I/O BUFFERS ARE ALLOCATED FROM THE PROCESS ADDRESS SPACE.

- INITIALLY TAKEN FROM PRE-ALLOCATED ADDRESS SPACE IN THE P1 REGION. LINKER OPTION IOSEGMENTS (DEFAULT = 32) DETERMINES NUMBER OF PAGES ALLOCATED TO THIS REGION. WHEN P1 SPACE IS USED UP, RMS WILL EXTEND P0 REGION UNLESS DISALLOWED.

RMS BUFFER MANAGER MAINTAINS LIST OF FREE PAGES AND SPACE WITHIN PAGES. ATTEMPTS TO KEEP ALL CONTROL STRUCTURES FOR A GIVEN FILE ON A SINGLE PAGE TO IMPROVE LOCALITY OF MEMORY REFERENCES. I/O BUFFERS ARE ALWAYS PAGE ALIGNED.

PROCESS I/O FILES (TYPICALLY SYS$INPUT, SYS$OUTPUT, SYS$COMMAND) REMAIN OPEN ACROSS IMAGE ACTIVATIONS. INDIRECT OPENER MERELY RETRIEVES IFI OF IFAB FROM LOGICAL NAME. NO I/O TAKES PLACE ON $OPEN. CURRENTLY LIMITED TO SEQUENTIAL FILES ONLY.

SHARED FILE CONTROL STRUCTURES (SFD, SIFAB, BCB, RLB, NRP) ARE ALLOCATED FROM SYSTEM PAGED POOL. I/O BUFFERS ARE STILL ALLOCATED PER PROCESS. BUCKET WILL NOT BE READ FROM DISK FOR SHARED FILE IF PROCESS HAS VALID COPY AND ANOTHER PROCESS HAS NOT MODIFIED IT.

RECORDS, NOT BUCKETS, ARE LOCKED FOR SHARED FILES. ACCESS TO BUCKETS IS QUEUED WHEN CONFLICT OCCURS DURING RECORD OPERATIONS.

MAINTENANCE OF NEXT RECORD POSITION FOR SEQUENTIAL GET'S FROM ISAM FILES IN SHARED DATABASE ALLOWS MORE EFFICIENT RE-POSITIONING AND RECOVERY OF SPACE.

TIME TO REORGANIZE?

1) IMPROVE ACCESS TIMES

BEST ACCESS TIMES WHEN:

FILE EXTENTS ARE CONTIGUOUS
RECORDS ARE ADDED TO FILE IN ASCENDING KEY VALUE

BEST CHANCE THAT SUCCESSIVE I/O REQUESTS WILL BE ON THE SAME CYLINDER.

BETTER LOADING OF BUCKETS MEANS LESS I/O ON SEQUENTIAL ACCESS OF RECORDS.

RANDOM ADDITION OF NEW KEYS CAUSES:

LOSS OF ADJACENCY IN PHYSICAL STRUCTURE.

POTENTIAL GENERATION OF RRV'S (PRIMARY DATA LEVEL).

ADDITIONAL I/O FOR ALTERNATE KEY ACCESS.

$UPDATE AND $DELETE RE-ACCESS CURRENT RECORD VIA RRV.

2) LOSS OF USEFUL SPACE WITHIN THE FILE

CURRENT IMPLEMENTATIONS RECYCLE FREE SPACE WITHIN BUCKETS.
BUCKETS ALWAYS CONTAIN SAME RANGE OF KEY VALUES.

SPACE IS RECOVERED BY $DELETE AND $UPDATE (WHEN RECORD SHRINKS).

SPACE EFFECTIVELY RECOVERED WHEN NEW KEY VALUES (FROM $PUT OR $UPDATE) FALL INTO BUCKETS THAT RECORDS HAVE BEEN DELETED FROM. LARGER BUCKET SIZES MAY IMPROVE THIS SITUATION.

BAD SCENE WHEN NEW KEY VALUES ARE INCREASING WITH TIME AS OLDER (LOWER KEY VALUE) RECORDS ARE DELETED.

DELETED DATA RECORDS LEAVE 2 BYTE RRV RECORDS (PRIMARY DATA LEVEL).

USES 1 OF 255 POSSIBLE ID'S FOR BUCKET. VERY LARGE BUCKET SIZES MAY AGGRAVATE THIS (YOU CAN'T WIN!).

MAY BE PREFERABLE TO $UPDATE RECORDS INSTEAD OF $DELETE THEM.

SOME BITS AND BYTES THAT YOU SHOULD KNOW ABOUT.

DEFERRED WRITE - DFW (IN THE FAB FOP FIELD)

SET THIS AT $OPEN OR $CREATE.

DEFERS WRITING PRIMARY AND ALTERNATE DATA BUCKETS TO DISK UNTIL BUFFER IS NEEDED FOR ANOTHER BUCKET.

BIG IMPROVEMENT WHEN PERFORMING SEQUENTIAL $PUT, $UPDATE, OR $DELETE OPERATIONS ON RELATIVE OR ISAM FILES WITH MORE THAN ONE RECORD PER BUCKET.

SEQUENTIAL OPERATIONS ONLY - SQO (IN THE FAB FOP FIELD)

USEFUL FOR NETWORK ACCESS ON VMS - CAUSES DAP TO USE FILE TRANSFER MODE INSTEAD OF RECORD TRANSFER MODE. REDUCES TRAFFIC ON NETWORK LINK.

ASYNCHRONOUS OPERATION - ASY (IN THE RAB ROP FIELD)

ALLOWS CONTROL TO RETURN TO CALLER BEFORE RECORD OPERATION IS COMPLETE.

CAN BE USED TO 'HIDE' RECORD I/O TIME IN SOME APPLICATIONS AND IMPROVE INTERACTIVE RESPONSE TIME.

SUPPOSE AN APPLICATION RETRIEVES 3 RECORDS RANDOMLY FROM AN ISAM FILE WHICH ARE THEN DISPLAYED ON A VIDEO TERMINAL. CODE MIGHT READ AS FOLLOWS:

```
L:  $GET     FROM ISAM FILE
    FORMAT OUTPUT LINE
    $PUT     TO TERMINAL
    BRANCH TO L IF NOT DONE
```

A 9600 BAUD TERMINAL OUTPUTS ABOUT A CHARACTER A MILLISECOND, THEREFORE SAY A FULL 80 COLUMN LINE IS ABOUT 80 MS.

SAY THE RANDOM $GET FROM THE ISAM FILE TAKES ABOUT 80 MS ALSO.

3 TIMES THROUGH LOOP THEN TAKES 3 X 160 OR 480 MS (ABOUT ½ SEC).

THE SAME SEGMENT CODED TO PERFORM THE TERMINAL I/O ASYNCHRONOUSLY MIGHT READ AS FOLLOWS:

```
L:  $GET     FROM ISAM FILE
    $WAIT    FOR TERMINAL I/O TO COMPLETE
    FORMAT OUTPUT LINE
    $PUT     ASYNCHRONOUS TO TERMINAL
    BRANCH TO L IF NOT DONE
    $WAIT    FOR TERMINAL I/O ON LAST PASS
```

THIS LOOP OVERLAPS THE RANDOM $GET FROM THE ISAM FILE WITH THE COMPLETION OF THE TERMINAL I/O. TOTAL ELAPSED TIME FOR 3 TIMES THROUGH LOOP IS 3 X 80 + 80 (WAITS FOR TERMINAL THE LAST TIME THROUGH) OR 320 MS. THIS IS 2/3'S THE TIME FOR THE SAME SEQUENCE PERFORMED TOTALLY SYNCHRONOUSLY.

FAST DELETE - FDL (IN THE RAB ROP FIELD)

WILL OMIT REMOVING ALTERNATE KEY POINTERS TO DATA RECORD FOR ALTERNATE INDICES WHERE DUPLICATES ARE ALLOWED ON $DELETE OPERATIONS (ISAM ONLY).

WILL CAUSE ADDITIONAL I/O ON $GET VIA ALTERNATE KEY TO DETERMINE THAT RECORD IS DELETED.

LOCATE MODE - LOC (IN THE RAB ROP FIELD)

AVOIDS MOVEMENT OF DATA RECORD FROM RMS I/O BUFFER TO USER BUFFER UNDER SOME CONDITIONS.

RECORD WILL BE MOVED ANYWAY IF:

1) RECORD CROSSES BLOCK BOUNDARIES (SEQUENTIAL FILES)
2) UPDATE ACCESS SPECIFIED IN FAC FIELD OF FAB
3) MULTI-STREAM ACCESS SPECIFIED IN SHR FIELD OF FAB
4) INDIRECT OPENER OF PROCESS PERMANENT FILE, E.G., SYS$INPUT (VMS ONLY)

READ AHEAD - RAH (IN THE RAB ROP FIELD)

CAUSES BLOCKS IN FILE TO BE READ BEFORE THEY ARE REQUESTED SO THAT THEY ARE ALREADY IN RMS I/O BUFFER WHEN NEEDED.

CURRENTLY ONLY IMPLEMENTED FOR SEQUENTIAL DISK FILES ON VMS.

WRITE BEHIND - WBH (IN THE RAB ROP FIELD)

PERFORMS WRITE OF BUFFER TO DISK ASYNCHRONOUSLY. RECORD OPERATION DOES NOT WAIT FOR DISK I/O TO COMPLETE.

CURRENTLY ONLY IMPLEMENTED FOR SEQUENTIAL DISK FILES ON VMS.

MULTI-BLOCK COUNT - MBC (IN THE RAB)

DETERMINES SIZE IN BLOCKS OF RMS I/O BUFFERS FOR SEQUENTIAL FILES.

LARGER BUFFERS MEAN LESS I/O.

MULTI BUFFER COUNT - MBF (IN THE RAB)

IMPLICATIONS ARE ORGANIZATION DEPENDENT.

FOR SEQUENTIAL FILES ON VMS, THIS DETERMINES THE NUMBER OF BUFFERS USED IN THE READ AHEAD/WRITE BEHIND RING. THE APPROPRIATE ROP OPTION (RAH OR WBH) MUST BE SET TO TAKE ADVANTAGE OF THE EXTRA BUFFERS.

FOR RELATIVE AND ISAM FILES, EXTRA BUFFERS FORM AN LRU (LEAST RECENTLY USED) CACHE OF BUCKETS. WILL AVOID DISK I/O IF A SUBSEQUENT OPERATION REFERENCES A BUCKET THAT HAS ALREADY BEEN READ IN. 'RANDOM' OPERATIONS EXHIBITING SOME LOCALITY IN THEIR REFERENCES MAY BENEFIT.

ADDITIONAL BENEFITS OF EXTRA BUFFERS FOR ISAM FILES:

ROOT BUCKETS OF INDICES WILL BE CACHED FOR RANDOM ACCESS.

ON VMS, THE FOLLOWING BUCKETS HAVE 'PERMANENCE', I.E., A 'NON-PERMANENT' BUCKET WILL BE USED FIRST:

1) ROOT BUCKET OF ANY INDEX ON A KEYED READ REQUEST.
2) ROOT BUCKET OF INDICES WITH A KEY OF REFERENCE LESS THAN MBF - 2 ON $PUT AND $UPDATE OPERATIONS.
3) ALTERNATE INDEX DATA LEVEL BUCKET WHEN PERFORMING SEQUENTIAL READ REQUESTS VIA ALTERNATE KEY.

ALSO NOTE THAT IF ENOUGH BUFFERS ARE PRESENT TO CACHE THE ENTIRE INDEX TREE (NUMBER OF INDEX LEVELS + 1), RANDOM ACCESS WITHIN SMALL RANGES OF KEY VALUES MAY BENEFIT. DRAMATIC IMPROVEMENT WHEN LOADING SINGLE KEY FILE IN ASCENDING KEY ORDER IN CONJUNCTION WITH DEFERRED WRITE.

BLANK SPACE:

VAX

SYSTEM UPDATE

WHAT CAN YOU EXPECT?

- EASE OF USE
- RELIABILITY
- PERFORMANCE

WHAT ARE THE NEW PRODUCTS?

- VAX/VMS V2.0
- VAX-11 COBOL V1.0
- VAX-11 BASIC V1.0
- VAX-11 FORTRAN V2.0
- VAX-11 FMS V1.0
- DATATRIEVE/VAX V2.0

VAX/VMS V2.0

- o PRODUCTIVITY AIDS
  - NEW LANGUAGE SUPPORT
  - EDT EDITOR
- o MANAGEMENT TOOLS
  - UTILITIES
  - SYSTEM FUNCTIONS
- o ENHANCED NUCLEUS FUNCTIONALITY
  - PERFORMANCE ENHANCEMENTS
  - REAL-TIME EXTENSIONS
  - NEW DEVICE SUPPORT

NEW DEVICES ON VAX

- MA780 MULTIPORT MEMORY
  - CONNECT UP TO 4 VAX-11/780 CPUS
  - UP TO 2 MA780s PER CPU
  - 256KB TO 2MB PER MA780
  - SHARE DATA, PASS MESSAGES
- DR780 HIGH-PERFORMANCE PARALLEL INTERFACE
  - CONNECT DEVICES TO VAX-11/780 INTERNAL MEMORY BUS (SBI)
  - TRANSFER DATA TO AND FROM MEMORY AT 6.67 MB/SEC

NEW DEVICES ON VAX

- TS11 MAGNETIC TAPE
  - 45 IPS
  - 1600 BPI ONLY
  - UNIBUS ADD-ON DEVICE
- RL02 CARTRIDGE DISK
  - 10 MB
  - UNIBUS ADD-ON DEVICE
- RX02
  - 512 KB
  - UNIBUS ADD-ON DEVICE
  - NO NEED NOW TO USE RX01 CONSOLE AS A DATA DEVICE

WHAT DOES ALL THIS MEAN TO YOU?

- LANGUAGE OF YOUR CHOICE
- EASE OF USE
- POWERFUL INTERACTIVE
  BUSINESS DATA PROCESSING
- REAL-TIME POWER

IT MEANS _ _ _ _ _

"THE HIGH-PERFORMANCE LANGUAGE

OF YOUR CHOICE"

COMMON LANGUAGE ENVIRONMENT

- SYMBOLIC TRACEBACK AND DEBUGGER
- COMMON RUN TIME LIBRARY
- VAX CALLING STANDARD
- COMMON EXCEPTION HANDLING
- I/O STANDARD (RMS)

IT MEANS _ _ _ _ _

"VAX IS FRIENDLY

EASY TO USE"

EASE OF USE FOR:

- PROGRAMMERS
- OTHER USERS
- SYSTEM MANAGERS

EASE OF USE FOR PROGRAMMERS AND OTHER USERS:

- HELP FACILITIES
- ONE COMMAND LANGUAGE
- WILD CARDS
- LEARNING AIDS
- DATATRIEVE

EASE OF USE FOR PROGRAMMERS:

- PROGRAM DEVELOPMENT AND MAINTENANCE FACILITIES
- PRODUCTIVITY AIDS
- LOGICAL NAMES

AND

LARGE ADDRESS SPACE

EASE OF USE FOR SYSTEM MANAGERS:

- SELF-CONFIGURATION
- RESOURCE CONTROLS (DISK, CPU, ETC.)
- ENHANCED UTILITIES

IT MEANS _ _ _ _ _

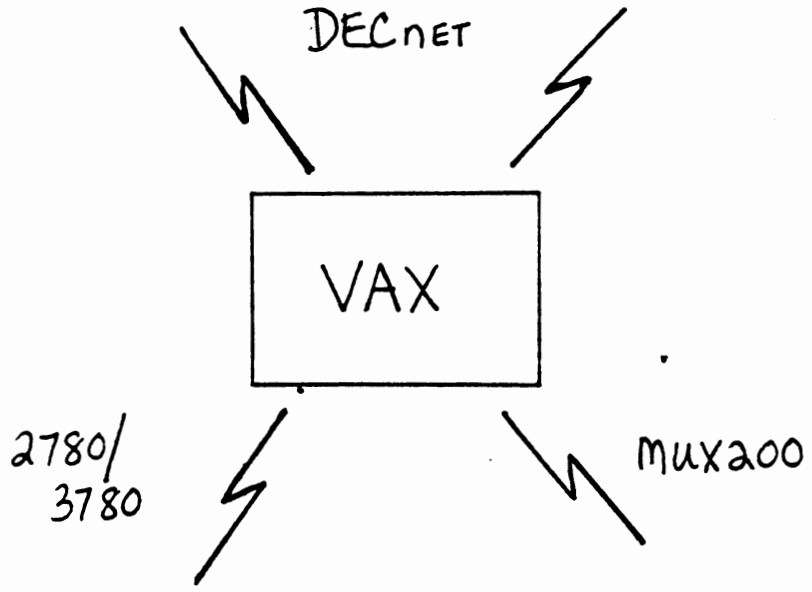
"VAX PROVIDES FOR POWERFUL INTERACTIVE DATA PROCESSING

FOR VARIETY OF BUSINESS APPLICATIONS"

FOR INTERACTIVE BUSINESS DATA PROCESSING:

- COBOL AND BASIC
- EXTENSIVE UTILITIES
- BATCH AND INTERACTIVE
- VAX-11 RMS
- FORMS PROCESSOR (FMS)

NETWORKS



IT MEANS _ _ _ _ _

"VAX OFFERS REAL-TIME POWER

AND EASE OF USE"

REAL-TIME POWER:

- TUNING FOR PERFORMANCE
- DR780 HIGH PERFORMANCE INTERFACE
- MA780 HIGH SPEED COMMUNICATION
- REMOTE DIAGNOSIS

REAL-TIME EASE OF USE:

- NEW REAL-TIME SYSTEM FUNCTIONS
- EASY TO WRITE DRIVERS AND DIAGNOSTICS
- EASY TO INTERFACE DEVICES
- TRANSPARENT MULTIPORT MEMORY SUPPORT

HOW TO FIND MORE DETAILS AT DECUS?

- APPROXIMATELY 50 VAX-RELATED SESSIONS
- OVERVIEW SESSIONS
  - VAX LANGUAGES PANEL
  - VAX/VMS V2.0 FEATURES
  - VAX/VMS V2.0 FIELD TEST PANEL
  - VAX FUTURE DIRECTIONS
  - VAX ARCHITECTURAL DEVELOPMENTS
  - VAX FOR COMMERCIAL USERS PANEL

HOW TO FIND MORE DETAILS AT DECUS?

- OPTIONAL SOFTWARE PRODUCT SESSIONS
  - VAX BASIC INITIAL EXPERIENCES
  - FEATURES OF COBOL ON VAX
  - INTRODUCING VAX-11 FORTRAN V2
  - FMS FOR VAX
  - DATATRIEVE ON VAX/VMS
  - DECNET WORKSHOP
  - HOW TO TALK TO VAX USING PASCAL

WHAT'S HAPPENING IN

SYSTEM QUALITY ASSURANCE?

VAX/VMS QUALITY ASSURANCE

OVERVIEW

- DEVELOPMENT PROCESS
- DEVELOPMENT TEST
- FIELD TEST
- PRODUCT ASSURANCE
- USER ENVIRONMENT TEST
- PERFORMANCE MEASUREMENT

VAX/VMS QUALITY ASSURANCE

DEVELOPMENT PROCESS

- MANAGEMENT
  - PLANNING
  - INVOLVEMENT
  - CONTROL
- IMPLEMENTATION
  - "BUILD/TEST" CYCLES
  - CHANGE CONTROL

VAX/VMS QUALITY ASSURANCE

DEVELOPMENT TEST

- SYSTEM BUILT ON ITSELF
- IN-HOUSE TEST SYSTEMS
- WIDE INTERNAL USE
- ENGINEERING NETWORK

VAX/VMS QUALITY ASSURANCE

FIELD TEST

- 11 TEST SITES, INTERNAL AND EXTERNAL
- 8 MONTH FIELD TEST
- VALIDATES TOTAL SYSTEM IN USER ENVIRONMENT
- PROBLEM REPORTING SYSTEM

VAX/VMS QUALITY ASSURANCE

PRODUCT ASSURANCE

- NEW FOR VAX/VMS V2.0
- ENSURES A COMPLETE "SYSTEM"
- TESTS INSTALLATION AND USABILITY OF VAX/VMS AND ITS OPTIONAL PRODUCTS

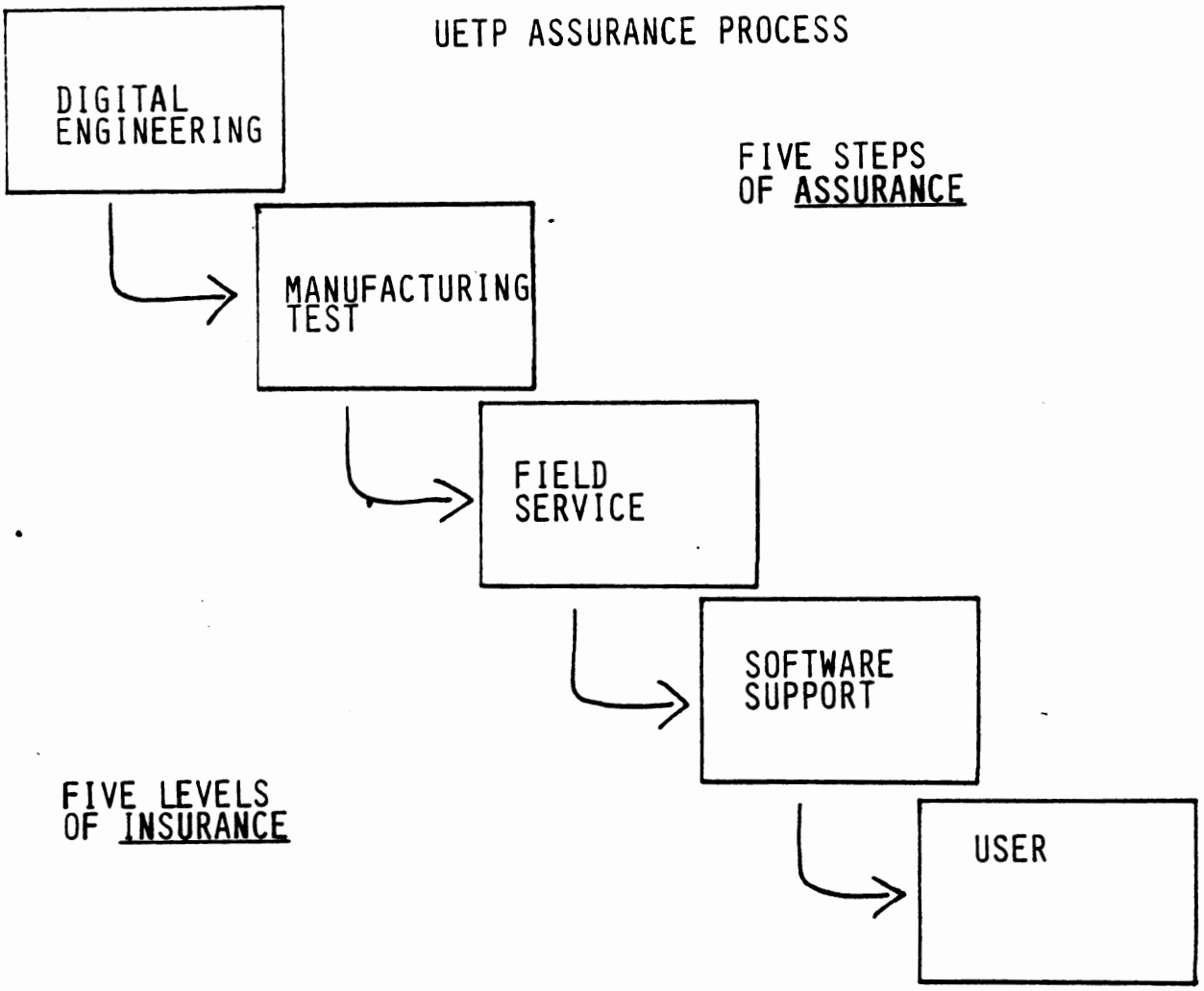
VAX/VMS QUALITY ASSURANCE

USER ENVIRONMENT TEST PACKAGE (UETP)

- A COMPREHENSIVE TEST TO VALIDATE THAT A SYSTEM HAS BEEN INSTALLED PROPERLY
- AUTOMATIC AND EASY TO USE
- DOCUMENTED AND SUPPORTED
- PART OF VAX/VMS

VAX/VMS QUALITY ASSURANCE

UETP ASSURANCE PROCESS



FIVE STEPS OF **ASSURANCE**

FIVE LEVELS OF **INSURANCE**

VAX/VMS QUALITY ASSURANCE

PERFORMANCE MEASUREMENT

OBJECTIVES

- SET USER EXPECTATIONS
- HELP USER TAILOR SYSTEMS
- HELP US ENHANCE SYSTEM PERFORMANCE

VAX/VMS QUALITY ASSURANCE

PERFORMANCE ACTIVITIES

- MODELING
- MEASUREMENT
  - REAL-TIME
  - RMS
  - MULTI-USER
- ANALYSIS
  - TUNING

WHAT'S NEW IN

VAX DOCUMENTATION?

VAX SOFTWARE DOCUMENTATION

- VAX/VMS VERSION 2.0 DOCUMENTATION
  - 30 MANUALS

- OPTIONAL SOFTWARE DOCUMENTATION
  - USUALLY 2 OR 3 MANUAL PER PRODUCT

VAX/VMS V2.0

OVERVIEW OF DOCUMENTATION CHANGES

- ALL EXISTING MANUALS REVISED; 6 NEW MANUALS ADDED
- AIDS FOR THE READER:
  - INDEX TO DOCUMENT SET
  - NEW ORGANIZATION OF MANUALS
  - "SUMMARY OF TECHNICAL CHANGES"
- SOME TOPICS MOVED TO DIFFERENT MANUALS

VAX/VMS V2.0

NEW MANUALS

- REAL-TIME USER'S GUIDE
- GUIDE TO USING COMMAND PROCEDURES
- UTILITIES REFERENCE MANUAL
- PATCH UTILITY REFERENCE MANUAL
- SYSTEM DUMP ANALYZER REFERENCE MANUAL
- EDT EDITOR REFERENCE MANUAL

VAX/VMS V2.0

MANUAL TITLE CHANGES

- VAX-11 INFORMATION DIRECTORY AND INDEX
- VAX/VMS SUMMARY DESCRIPTION AND GLOSSARY

VAX/VMS V2.0

CHANGES TO SELECTED MANUALS

- SYSTEM MANAGER'S GUIDE
- SYSTEM SERVICES REFERENCE MANUAL
- I/O USER'S GUIDE
- LINKER REFERENCE MANUAL
- SYMBOLIC DEBUGGER REFERENCE MANUAL
- GUIDE TO WRITING A DEVICE DRIVER
- RMS MANUALS

WHAT'S HAPPENING IN

VAX SERVICES?

VAX WORKS

SUPPORT FOR FIELD SPECIALISTS

- TELEPHONE SUPPORT ("HOTLINE)

- INFORMATION FOR SPECIALISTS

- SYSTEM SEMINAR

- UPDATE SEMINARS

HELP TO ENSURE THE QUALITY OF THE NEXT RELEASE

- TEST NEW FEATURES

- CHECK "BUGS" FIXED IN NEW RELEASE

- REVIEW PROBLEM AREAS FROM PREVIOUS RELEASES

- REVIEW DOCUMENTATION FOR NEW RELEASE

PROVIDE INPUT TO SOFTWARE DEVELOPMENT

- COMPONENTS WHICH DON'T WORK

- AREAS WHICH NEED IMPROVEMENT

- FEATURES WHICH ARE NEEDED

NEW

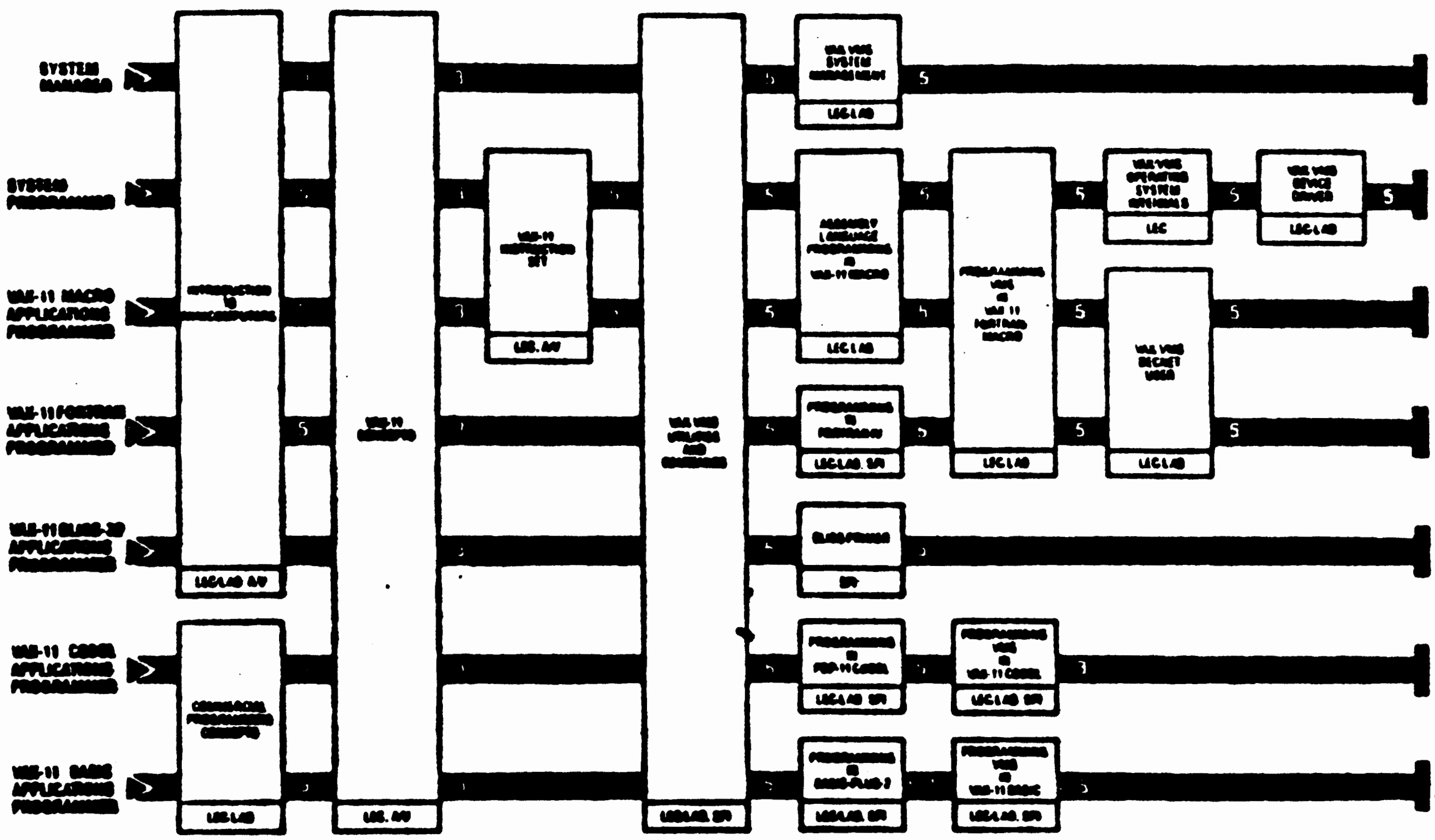
FROM

EDUCATIONAL SERVICES

EDUCATIONAL SERVICES

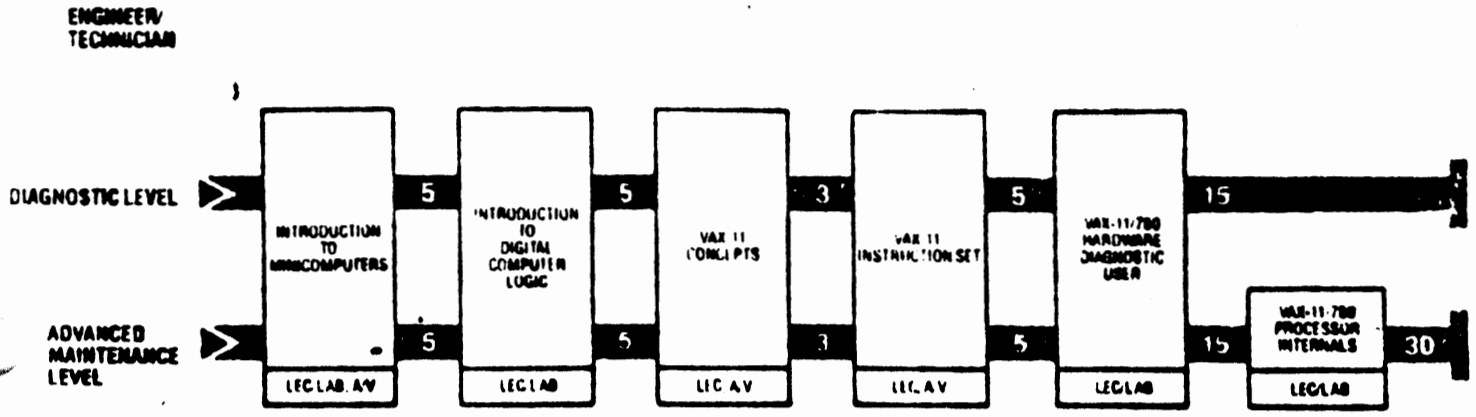
- NEW COURSES - SOFTWARE
- NEW COURSES - HARDWARE
- NEW LOCATIONS
- NEW MEDIA

NEW COURSES - SOFTWARE



VAX/VMS SYSTEM TRAINING

NEW COURSES - HARDWARE



VAX-11/780 MAINTENANCE TRAINING

NEW LOCATIONS

- BEDFORD
- LANHAM
- LOS ANGELES
- NEW YORK
- ROLLING MEADOWS
- SANTA CLARA

NEW MEDIA

- SELF PACED COURSES
  - VMS UTILITIES AND COMANDS
  - PROGRAMMING VMS IN VAX-11 COBOL
  - PROGRAMMING VMS IN VAX-11 BASIC
  - EDT CAI

WHAT'S HAPPENING IN

VAX PERFORMANCE MEASUREMENT?

PERFORMANCE

- EXTENSIVE PERFORMANCE TESTING IN MULTI-USER ENVIRONMENT
- CUSTOMERS DO NOT RUN STAND-ALONE SYSTEMS
- WE OPTIMIZE ON MULTI-USER SYSTEM PERFORMANCE
- RESULTS ARE ENVIRONMENT DEPENDENT
- TESTING ACCOMPLISHED THROUGH A REMOTE TERMINAL EMULATOR

REMOTE TERMINAL EMULATOR

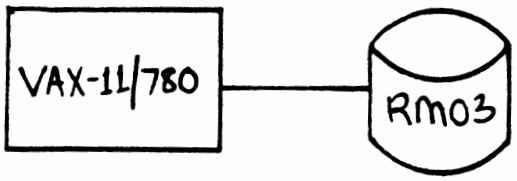
- o IMPLEMENTED ON ANOTHER CPU
- o EMULATES TIMESHARING USERS
- o - EACH TERMINAL REPRESENTED BY A SCRIPT OF COMMANDS
  - DIFFERENT TYPES OF USERS CAN BE REPRESENTED BY CHANGING SCRIPTS (COMMAND TYPE AND SEQUENCE, TYPING RATE, THINK TIME)
- o TESTING IS REPEATABLE

PERFORMANCE

- THREE ENVIRONMENTS TESTED
  - EDUCATIONAL TIMESHARING
  - ENGINEERING/SCIENTIFIC TIMESHARING
  - COMMERCIAL ENVIRONMENT
- FIELD TEST SOFTWARE USED FOR TESTING
- REASONABLE SYSTEM TUNING PERFORMED

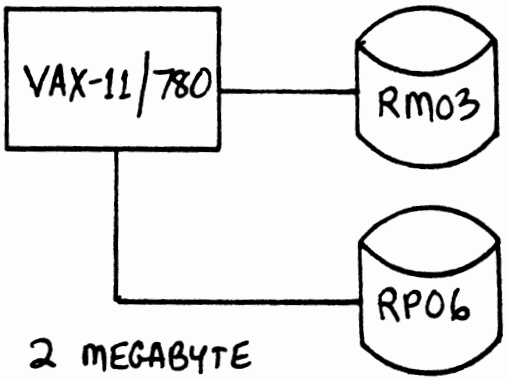
CONFIGURATIONS

EDUCATIONAL AND ENGINEERING/SCIENTIFIC TIMESHARING

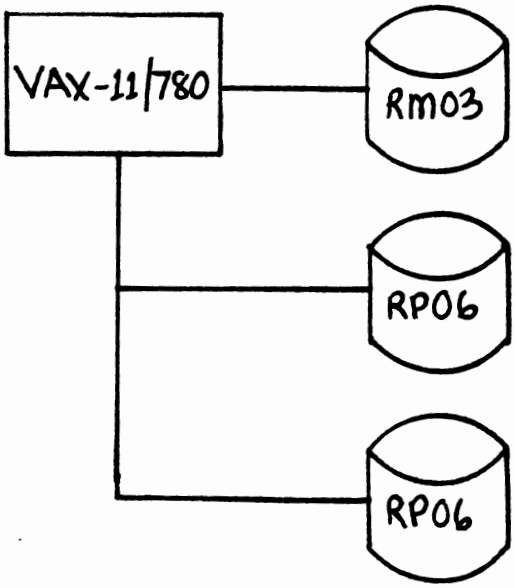


1/2 MEGABYTE

1 MEGABYTE



2 MEGABYTE



3 MEGABYTE

4 MEGABYTE

EDUCATIONAL TIMESHARING WORKLOAD

BATCHSTREAM: ADMINISTRATIVE PROCESSING

SCRIPT 1: INTERACTIVE ADMINISTRATIVE FILE UPDATING

SCRIPT 2: INSTRUCTIONAL PROGRAM DEVELOPMENT

SCRIPT 3: EXECUTION OF COMPUTATIONAL PROGRAMS

- USES COBOL, BASIC AND FORTRAN
- RUN FOR MULTIPLES OF 8 USERS

EDUCATIONAL TIMESHARING WORKLOAD

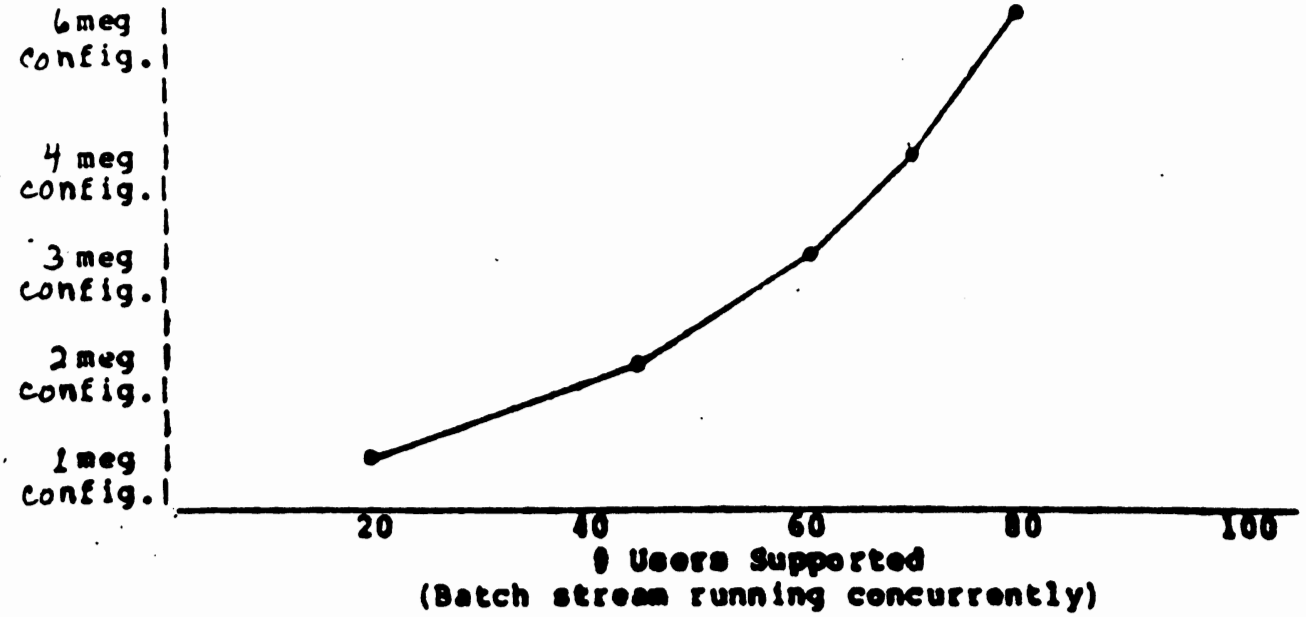
TEST #1

# USERS	TYPE	THINK TIME	TYPING RATE	OUTPUT RATE
5	INSTRUCTIONAL	10 SEC.	3 CPS	1200 BAUD
2	ADMINISTRATIVE	5 SEC.	5 CPS	1200 BAUD
1	COMPUTATIONAL	10 SEC.	3 CPS	1200 BAUD

- o REPLICATION OF THIS GROUP OF 8 USERS PRODUCES HIGHER INTERACTIVE LOADS
- o SINGLE BATCH STREAM RUN WITH ALL TESTS

EDUCATIONAL TIMESHARING WORKLOAD

TEST #1



EDUCATIONAL TIMESHARING WORKLOAD

TEST #1 COMPARISON WITH VAX/VMS V1.5

CONFIGURATION	V1.5	V2.0
1 MEGABYTE/1 DISK	16 USERS	20 USERS
2 MEGABYTE/2 DISK	32 USERS	44 USERS

EDUCATIONAL TIMESHARING WORKLOAD

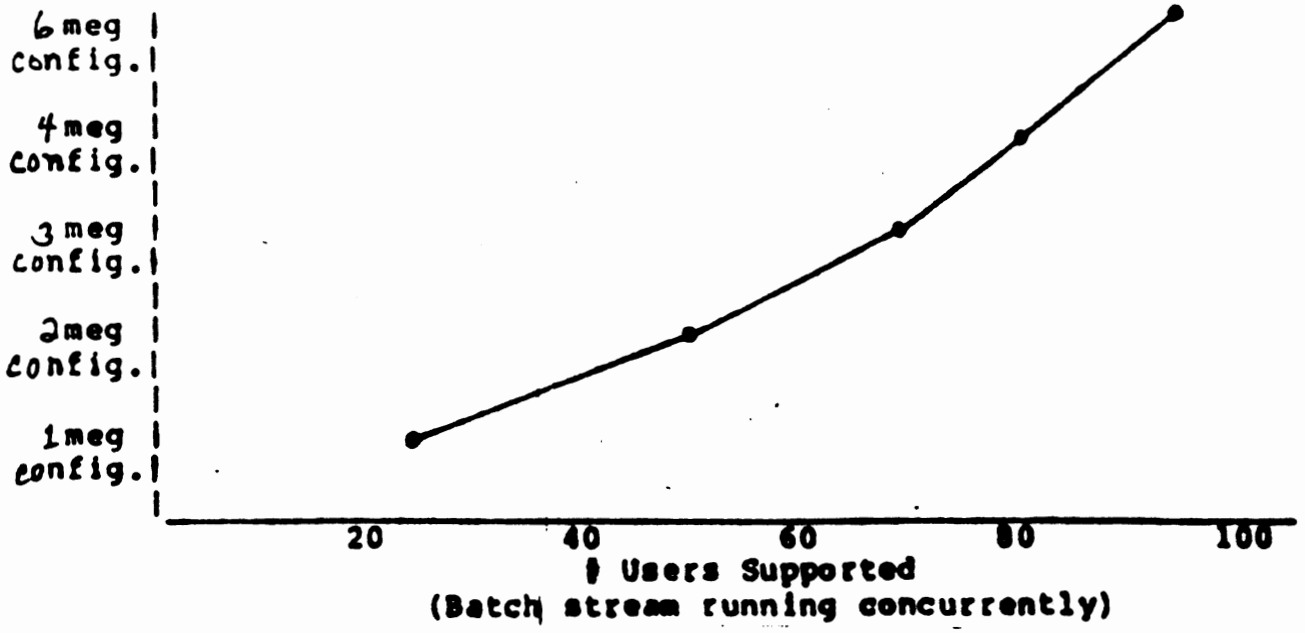
TEST #2

# USERS	TYPE	THINK TIME	TYPING RATE	OUTPUT RATE
5	INSTRUCTIONAL	16 SEC.	2 CPS	1200 BAUD
2	ADMINISTRATIVE	8 SEC.	3 CPS	1200 BAUD
1	COMPUTATIONAL	15 SEC.	2 CPS	1200 BAUD

- 8 USER GROUPS REPLICATED
- BATCH STREAM RUN CONCURRENTLY

EDUCATIONAL TIMESHARING WORKLOAD

TEST #2



ENGINEERING/SCIENTIFIC TIMESHARING WORKLOAD

# USERS	ACTIVITY
1	EDITING FORTRAN
1	EDITING TEXT FILE
2	FORTRAN COMPILES
2	CREATE DATA FILE; COMPILE, LINK AND RUN FORTRAN PROGRAM
1	CREATE, EDIT, COMPILE, LINK AND RUN (SEVERAL TIMES) FORTRAN LINEAR PROGRAMMING PROGRAM
2	CREATE, EDIT, COMPILE, LINK PROGRAMS (1 BASIC, 1 COBOL USER)

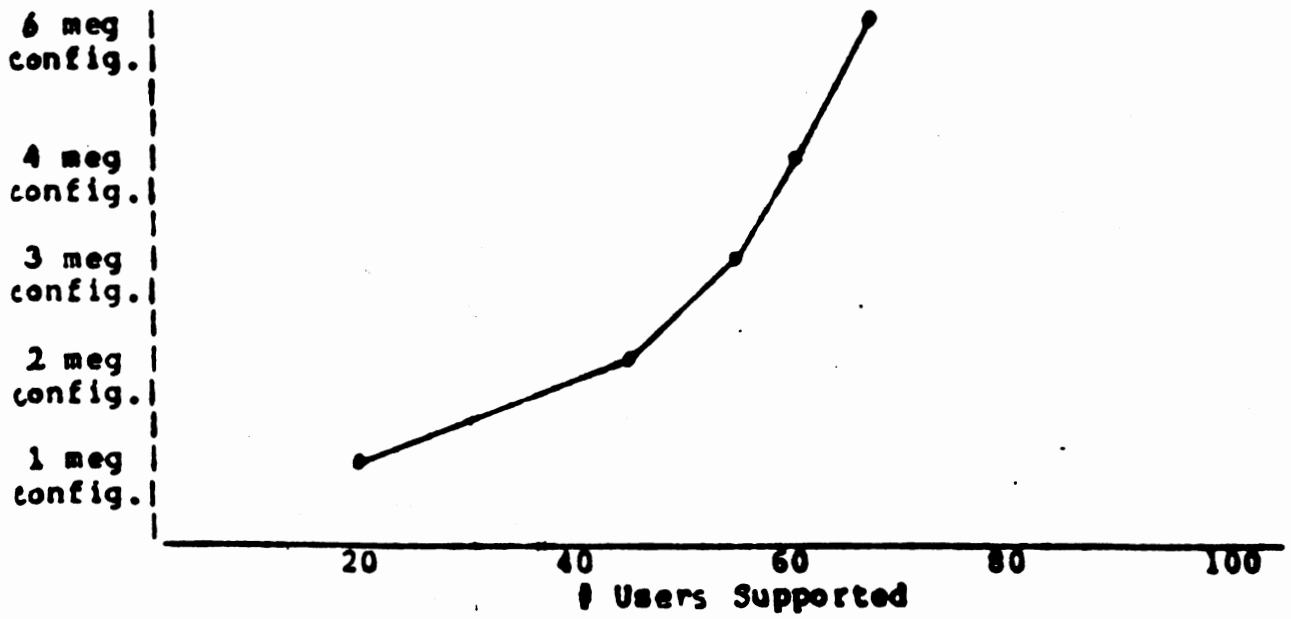
ENGINEERING/SCIENTIFIC TIMESHARING WORKLOAD

TYPING RATE = 5 CPS

THINK TIME = 10 SEC.

OUTPUT RATE = 2400 BAUD

ENGINEERING/SCIENTIFIC TIMESHARING WORKLOAD



CONFIGURATIONS COMMERCIAL ENVIRONMENT

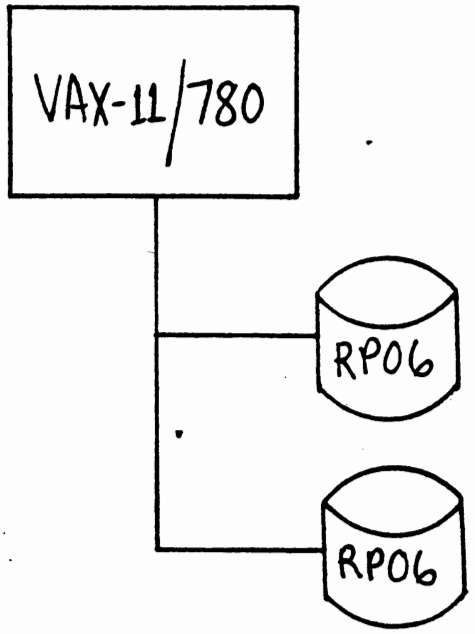
1/2 MEGABYTE

1 MEGABYTE

2 MEGABYTE

3 MEGABYTE

4 MEGABYTE



PURCHASE ORDER SYSTEM

- 4 COBOL APPLICATION PROGRAMS AND DATA BASE OF 5 FILES

- CONSISTS OF FILE MANAGEMENT PROCESSING, ARITHMETIC PROCESS, AND DATA MOVE

PURCHASE ORDER SYSTEM

# USERS	ACTIVITY	THINK TIME	TYPING RATE	OUTPUT
2	CUSTOMER INQUIRY/UPDATE	10 SEC.	4 CPS	105 CHAR
2	STOCK INQUIRY/UPDATE	10 SEC.	4 CPS	80 CHAR
1	ORDER ENTRY	10 SEC.	4 CPS	500 CHAR MAX.
2	DATA ENTRY	2 SEC.	5 CPS	N/A
1	FILE EDITING	10 SEC.	3 CPS	--

- RUN FOR MULTIPLES OF 8 USERS
- BATCH STREAM RUN CONCURRENTLY

digital