Option Description

KT11-B

PAGING OPTION

COMPUTER TYPE PDP-11/20	DRAWING SET NO. 76Ø5971 - KT11-B 76Ø5961 - KA11 Modification
PROGRAM NO. DECSPEC-11-Ø393 to DECSPEC-11-Ø398	PROJECT ENGINEER William Weiske, Jr.
PROGRAMMER J. Hittell	DOCUMENT NO. CSS-MO-F-9.2-2
APPROVED BY J. Holman	DATE April 1971

CONTENTS

KT11-B

PAGING OPTION

INTRODUCTION

1.1 General Description

The KT11-B Paging Option, designed and manufactured by Digital, provides, on a PDP-11, address mapping from a program's virtual address space to the physical bus address space by substitution and extension of the most significant bits of the virtual address as specified by entries in a "page map". This mapping permits user and monitor programs to be segmented and these segments to be allocated to available core on a space-available basis in a manner completely transparent to the user. The address mapping, along with the inherent memory and system protection features of the KT11-B, provide a multiuser, machine-language timesharing environment, in which the monitor and other users are protected from erroneous or malicious action by any user. The KT11-B facilitates the writing of reentrant code, shared by many users, thereby optimizing use of core and auxiliary storage.

The KT11-B is an interface between the processor and the Unibus, as shown in figure 1-1 below. In unpaged mode, bus operations are performed as in a normal PDP-11 system. With the KT11-B enabled, certain bus signals are modified to provide protection and relocation.

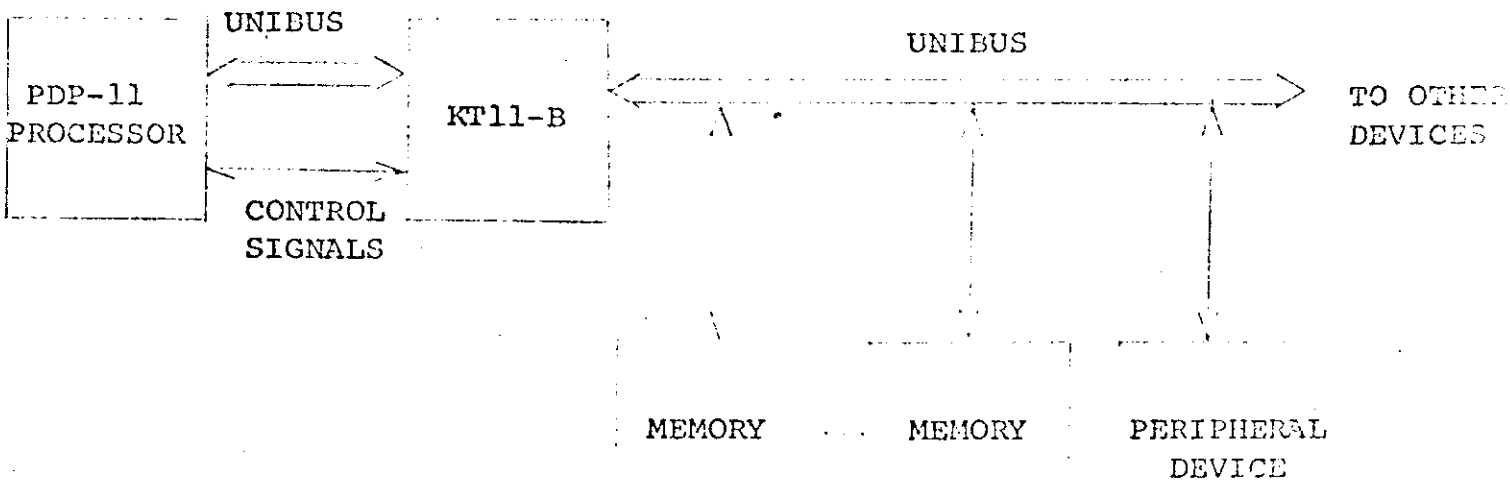


FIGURE 1-1

Figure 1-2 illustrates a user virtual program and a typical relocation to physical core. Note that the physical core available may be four times the size (262 K bytes) of the maximum size user program (64 K bytes).

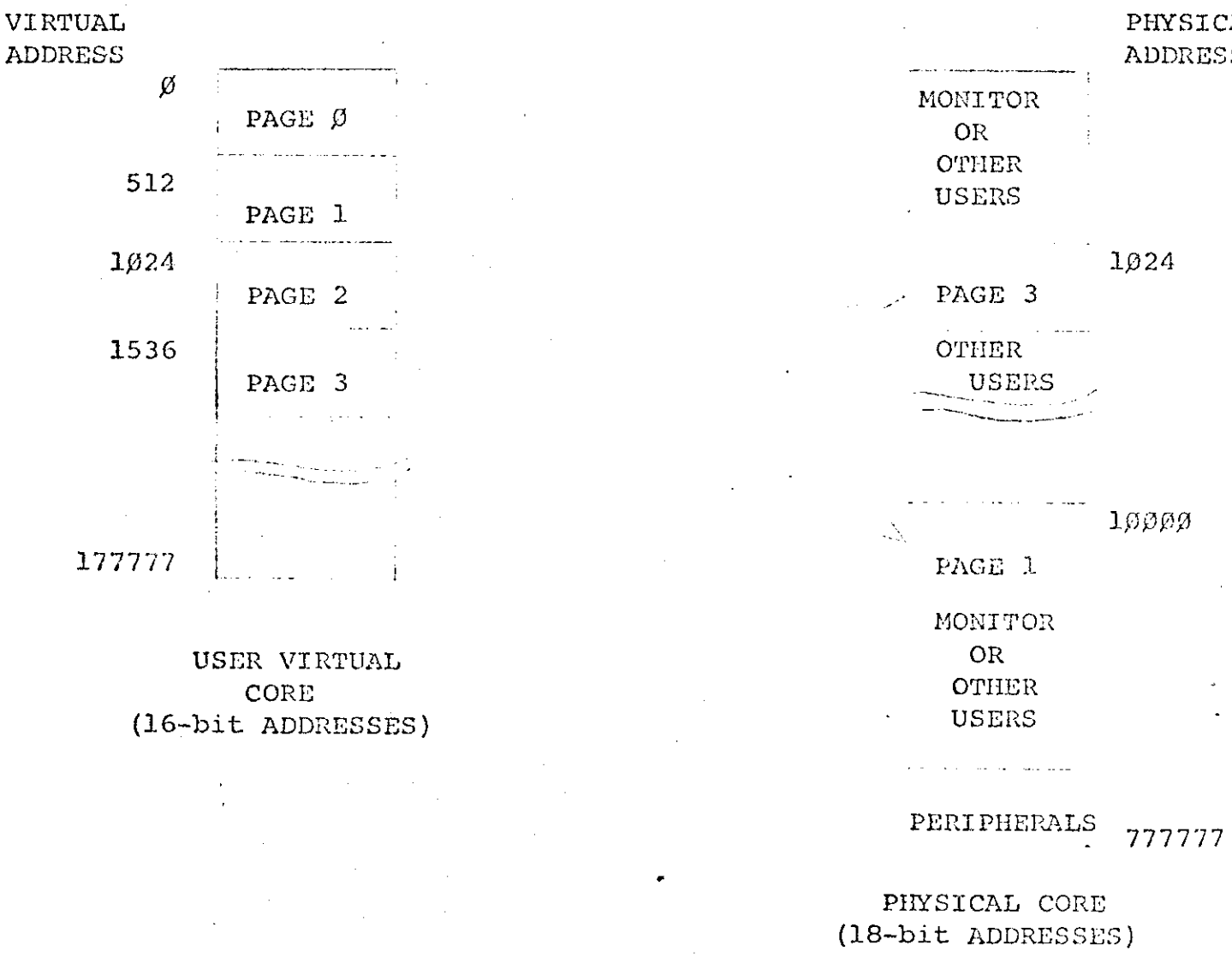


FIGURE 1-2

1.2 Specifications

a. Mechanical:

No. of Logic Panels	Two
Type of Logic Panels	H911
Dimensions	19 in. w, 20 in. d, 1Ø-3/8 in

b. Electrical:

Input Power	120/240 VAC ± 1Ø%, 47-63 Hz @
Power Supply	One H72Ø
Module Type	M-Series
Logic Levels	1 = +3V, Ø = ØV (TTL)

c. Operational:

Modes	Unpaged, Monitor paged, User Paged
Virtual Program Size	2^{16} bytes (max.)
Max. Physical Core	238 K bytes
Page Size	512 bytes

1.3 Availability

The KT11-B is a Computer Special Systems product and is available from Digital Equipment Corporation, 146 Main Street, Maynard, Massachusetts 01754 as a component in new installations.

SECTION 2

INSTALLATION

2.1 SITE CONSIDERATIONS

The KT11-B is built on two H911 logic panels. This assembly should be mounted in an H96Ø cabinet or equivalent, preferrably as close to the PDP-11 as possible. On the back door of this same cabinet should be mounted the H72Ø power supply; this power supply occupies an 11-½ " high mounting space. The cabinet is not supplied as part of the KT11-B.

Due to the action of the KT11-B, peripherals which are normally mounted with the processor (Teletype, reader, and punch controls and KW11-L real-time clock) must be mounted in a separate DD11 system unit (not supplied with the KT11-B). The KW11-L may also be mounted in slot BØ9 of the KT11-B.

2.2 CABLES

2.2.1 Power Cables

Line power is brought to the H72Ø power supply via a cable with a 15 amp 3-prong plug. This plug should be inserted into an outlet controlled by the PDP-11 console power switch, so that power is supplied to the KT11-B when the processor is turned on.

Logic power, including the ACLO and DCLO signals, is supplied via a cable assembly connected to a power end panel on one of the H911 panels.

2.2.2 Unibus Cables

The KT11-B must be immediately adjacent to the processor on the Unibus, as shown in Figure 1-1. BC11-A cable assemblies are used for all Unibus connections. The Unibus comes directly from processor slots A14 and B14 and enters the KT11-B in slots AØ1 and BØ1. The Unibus leaves the KT11-B from slots AØ3 and BØ3.

2.2.3 Control Cable

A Flexprint cable runs from slot BØ9 in the processor to slot D12 in the KT11-B. This cable carries non-Unibus signals necessary for the operation of KT11-B.

2.3 INITIAL OPERATION

2.3.1 Checkout

The KT11-B is checked out and demonstrated to be in working order by performing the following steps:

1) Turn on all devices and power in the system; no special test configuration is necessary for the system.
2) Verify that the examine and deposit console functions work and also that the MAINDEC processor and memory diagnostics run. (See Appendix A.)
3) Run the KT11-B diagnostic programs KT1, KT2, KT3, and KT6, allowing each to make at least twenty passes (the teleprinter bell will ring at the end of each pass).
4) Run the MAINDEC processor diagnostics in user mode using the KTU program.
5) Run the KTES system exerciser program in both exec and user modes with all available devices implemented in the program (DECtape, RP11 disc, KW11-L clock, high-speed reader and punch, and Teletype reader and printer).

2.3.2 Acceptance

The KT11-B is accepted in-house prior to shipment and is considered to be meeting its specifications by performing the steps described in 2.3.1 above. In addition, the system should be exercised with the KTES program for at least eight hours running all available devices except the Teletype and high-speed reader and punch (these are omitted since they may be subject to excessive wear if operated continually over long periods of time).

SECTION 3

OPERATION AND PROGRAMMING

3.1 CONTROLS AND INDICATORS

There are no operator controls or indicators on the KT11-B. The KT11-B registers may be examined manually by using the processor console functions.

3.2 DIAGNOSTIC SOFTWARE

Appendix A contains the descriptions of the diagnostic and exercise programs for the KT11-B. Note that certain MAINDEC standard diagnostic programs are modified to run with the KT11-B.

3.3 INPUT/OUTPUT CODING

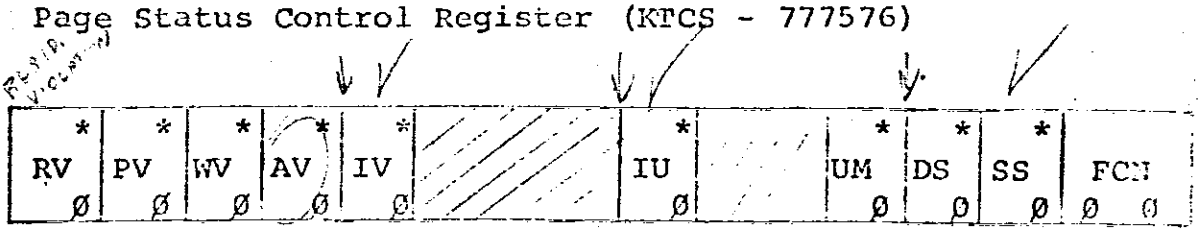
3.3.1 General

The KT11-B may be considered as an extension of the central processor. It's purpose is to provide system protection and program relocation such that many users may use the system on a timeshared basis. The individual users (the programmers writing programs to be run in user mode) need not be concerned about the KT11-B; in fact, each user writes programs as if he had the entire PDP-11 and 32K words of core memory to himself. Certain restrictions to users are imposed by the KT11-B, but otherwise, it is entirely transparent. It is the systems programmer, the person who writes the monitor, who needs to be concerned about the KT11-B. Therefore, the systems programmer should be intimately familiar with the operation of the KT11-B before attempting to write a timesharing monitor making use of the KT11-B. To this end, Section 4 of this Option Description, describing the principles of operation of the KT11-B, should be thoroughly understood before continuing reading in this section.

3.3.2 KT11-B Registers

Eleven bus addresses are assigned to the operation registers of the KT11-B. There are 8 additional addresses which reference the words of the scratch-pad holding the 8 most recently accessed Page Map Entries.

3.3.2.1 Page Status Control Register (KTCS - 777576)



* = READ ONLY

Ø = Cleared to Ø by the Unibus INIT signal

FCN -Determines KT11-B operating mode

Ø -Unpaged operation - the bus signal INIT clears FCN.

1 -Destination only paging - paging only operational while processor is in "DESTINATION" or "EXECUTE" state. This is intended as a maintenance mode only. Note that traps do bus operations in the "SERVICE" state of the processor, and therefore, will be unpaged.

2 -Normal paging - this is the standard operating mode.

3 -Start user - this mode is entered by the Exec in order to start or restart a User. The next bus operation made by the processor while in any state except "FETCH" that is the next processor bus operation not a fetch causes the KT11-B to load the D-lines of the bus with the contents of the KTPC (the paging program counter) - the address and control (A and C) lines on the bus are ignored. The following bus operations cause the KT11-B to load the D-lines with the contents of KTPS (the paging processor status). User paging is then invoked and the bit Ø of KTCS is cleared (normal paging is resumed). Note that if the instruction executed immediately after setting the FCN bits to 3 is an RTI, the processor program counter and status register will be loaded from KTPC and KTPS respectively.

SS -SOURCE STATE - This bit is affected only if all the "ERROR" flags, bits 15 through 11 of KTCS are clear. Under this condition, it is set whenever the processor does:

1. Any bus operation while in the "SOURCE" state during normal paging.

2. A DATO operation while executing a TRAP or JSR during normal paging.

3. A DATO while executing a JSR during destination only paging.

4. Any bus operation while executing a RTS or RTI and paging during normal or destination - only paging.

It is cleared whenever the processor does any bus operation while in the "FETCH" state during normal paging if all the error flags are clear.

DS -DESTINATION STATE - Bit is affected only if all the error flags are clear. Under this condition, it is set whenever the processor does:

1. Any bus operation while in the "DESTINATION" state during normal or destination - only paging.

2. The second DATO operation while executing a TRAP during normal paging.

3. The second bus operation while not in the fetch state and executing an RTI during normal or destination only paging.

It is cleared whenever the source state bit is cleared.

UM -USER MODE - This bit governs whether USER (set) or EXEC (cleared) paging is used to generate physical address. It also governs use of privileged instructions, the ability to modify processor priority, and system reaction to interrupts and traps. This bit is set as the final act of the START USER SEQUENCE. It is cleared by interrupts, paging violations (anything that sets an ERROR flag) and EXEC traps, as well as the bus signal INIT.

IU -INTERRUPTED USER - This bit is loaded from the USER MODE bit whenever an INTR signal is received on the bus or a paging violation occurs. It is cleared any time a DATO is made to KTCS or an INIT occurs.

IV -INSTRUCTION VIOLATION - This bit is set whenever a USER attempts to execute one of the privileged instructions (HALT, WAIT, or RESET) or when an attempt is made to transfer program execution from a public to a private page at any other place but at an entry point instruction, "BR .+2." The instruction is not executed and a trap is made to the exec (see 4.2.3). This bit is cleared by INIT or a DATO to KTCS.

AV -ADDRESS VIOLATION - This bit is set whenever an odd address error (attempt by the processor to do a word bus operation to an odd address) occurs. It is cleared by INIT or a DATO to KTCS.

WV -WRITE VIOLATION - This bit is set whenever an attempt is made by the processor to do a DATIP, DATO, or DATOB to a page for which the protection key is not "WRITE ENABLED". It is cleared by INIT or a DATO to KTCS.

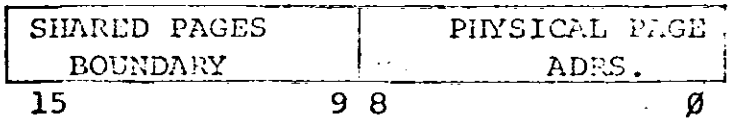
PV -PRIVACY VIOLATION - This bit is set whenever the processor attempts a bus operation to a page which is not keyed "PUBLIC" when the operation does not occur during the "FETCH" state and the last instruction fetched during normal paging which was not from the scratchpad and did not of itself constitute an instruction violation from a page keyed as PUBLIC. This bit is cleared by INIT or a DATA to KTCS.

RV -RESIDENCY VIOLATION - This bit is set whenever the processor attempts to reference a page which is not keyed as "RESIDENT". It is cleared by INIT or a DATO to KTCS.

3.3.2.2 USER and EXEC Map Control Registers (KTUM - 777614 and KTEM - 777616)

THESE REGISTERS SPECIFY THE LOCATION AND CHARACTER OF THE USER AND EXEC PAGE MAP PAGES (see 4.2.2).

Both registers are 16 bits wide:



The boundary between simple user paging map entries and shared page map entries is determined by the high order 7 bits of the User Map Control Register. If these bits equal N, the first N entries in the page map are simple entries (if these 7 bits are all clear, then 2^7, or all of the entries in the page map are simple entries).

The high order 9 bits of the physical address of the User Page Map Page are in the low order 9 bits of the User Map Control Register.

The extent of simple exec paging is determined by the high order 7 bits of the Exec Map Control Register (KTEM). If these bits equal N, there are N page map entries in the Exec Page Map Page. (If these 7 bits are all clear, then 2^7, 128 entries, are in the page map page.) The remaining entries (128 - N) are in the User Page Map as "Exec-per-process" entries (see 4.2.2.2).

Loading (DATO or DATOB) either of these registers causes the associative memory to be cleared (see 4.1.2) and the replacement register to be set to 10_8.

3.3.2.3 PAGE INSTRUCTION COUNTER (KTIC - 7776ØØ)

This register is loaded with the low order 16 bits of the Address Lines on the Unibus whenever normal paging is operational, no error flags are set, and the processor does a bus operation while in the "FETCH" state.

3.3.2.4 PAGE STACK POINTER (KTSP - 7776Ø2), PAGE PROGRAM COUNTER (KTPC - 7776Ø4), AND PAGE PROGRAM STATUS (KTPS - 7776Ø6)

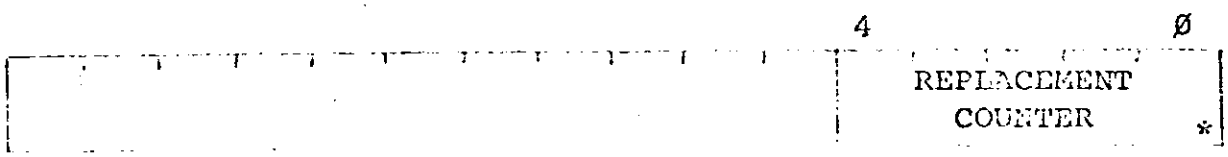
These registers are loaded during the "MODIFIED SEQUENCE" discussed in 4.2.3. The first DATO of such a sequence has the processor Status Register on the D-lines of the bus. This is loaded into KTPS. The second DATO of the sequence holds the processor stack pointer on the A-lines and the Processor Program Counter on the D-lines. These are loaded respectively into KTSP and KTPC. Note that the modified sequence may be entered due to a paging violation in Exec Mode. If recovery is to be possible after such an event, KTSP, KTPC, and KTPS must be saved after User program interruption and restored when that User is to be restarted.

3.3.2.5 PAGE DATA REGISTER (KTDT - 777612) and PAGE ADDRESS REGISTER (KTAD - 77761Ø)

These registers are loaded from the D and A lines respectively whenever any of the error flags become set or a time-out occurs.

3.3.2.6 PAGE MAINTENANCE REGISTER (KTMR - 777575)

This register holds the replacement counter and 5 other flags which have meaning only when observed with an oscilloscope during certain KT11-B internal operations. The Replacement Register is incremented through the range $1Ø_8$ to 57_8 whenever a look up sequence occurs. It is set to $1Ø_8$ whenever KTUM or KTEM are loaded.



3.3.2.7 PAGE WINDOW (KTWN - 777572)

Bus operations to this address (after paging) are done to the USER virtual address specified by the contents of KTAD. Page violations can occur both at the paging to KTWN and from the contents of KTAD. This permits the Exec to reference User paged addresses conveniently.

WINDOW on a Window causes a time-out trap.

SECTION 4

PRINCIPLES OF OPERATION

4.1 OVERALL DESCRIPTION

The KT11-B provides address mapping from the program's virtual address space to the physical bus address space by substitution and extension of the most significant bits of the virtual address as specified by entries in a "page map". This mapping permits access to the entire physical bus address space which is 4 times larger than any program's virtual address space.

The operating system in a multiuser (machine language timesharing) environment is protected against the users.

4.1.1 Goals

The KT11-B accomplishes mapping and system protection with minimum interaction with the processor outside the UNIBUS. System protection is achieved by:

1. Keying pages as "public" or "private" and "writeable" or "write protected".

2. Providing an entry point instruction which must mark an entry from public to private code execution to permit use of private code.

3. Disallowing execution of privileged instructions in user mode (e.g. Halt, Reset).

4. Disabling modifications to processor priority while in user mode.

5. Processing odd address violations and disabling processor handling of such violations while in user mode.

6. Breaking up "trap loops" (TRAP service routines calling traps without an intervening instruction) while in user mode.

7. Entering exec mode to service interrupts and exec traps with push operations of the hardware interrupt service sequence directed to special registers.

8. Trapping to exec mode when power fail occurs.

9. Disabling processor handling of stack overflow violations.

It is possible to implement a "demand paging" scheme for user program memory allocation since:

1. Pages may be keyed as "resident" or "non-resident".

2. Record is kept of the first location of each instruction executed.

3. Record is kept of the progress of each instruction through processor states.

Provision is made for user-exec communication through exec traps in user mode and user paged read-writes in exec mode (the "page window").

The KT11-B is power cleared to a non-paging mode in which KA11 operation is identical to that of a KA11 without the KT11-B except for some time penalties on bus operations, and loss of stack overflow trapping.

4.1.2 Major Functional Components

Refer to the KT11-B drawing set for the logic diagrams.

4.1.2.1 Associative memory -

An 8 x 8 associative memory permits simultaneous comparison of the 8 8-bit words it contains against the seek data. The word that matches the seek data shows a hi match output while the match outputs for all other words are low. (Optionally, a 24-word associative memory is available, allowing fewer core look ups.)

Each word of the associative memory contains the page number (7 bits) and User tag (1 bit) the protection key and relocation information held in the corresponding word of the Scratchpad. The User tag is 1 if the information is for a page of a program running in User mode and Ø for a page of a program running in Exec mode.

4.1.2.2 Scratchpad -

8 words of 16 x 16 IC memory contain the Page Map Entries (protection keys and relocations) for the 8 pages identified by the associative memory as just described. The remaining 8 words contain some of the KT11 operational registers.

One 16 bit word at any time may be made available at the output of the scratchpad or written from the scratchpad input. (With a 24-word associative memory, 16 more words of scratchpad are added.)

4.1.2.3 Replacement Counter -

When a page is referenced whose page number has no match in the associative memory, the user's or exec's core page map is accessed for the appropriate Page Map Entry. The page number and the current state of the User Mode bit are then loaded into the associative memory and the page map entry found in the core page map is loaded into the corresponding word of the scratchpad. The address in the associative memory and in the scratchpad which is to be so loaded is specified by the replacement counter. The replacement counter is then incremented to point to the next associative memory address. This sequence of operations is referred to as a "Lookup Sequence".

4.1.2.4 Unibus Receivers and Drivers -

Unibus signals from the processor may not in some cases be passed on to UNIBUS devices (including memory) until the relocation specified for the referenced page has been performed and the protection key checked against the requested bus operation. If a Page Map Entry must be fetched from core, the bus signals D 15:00 from the processor must be blocked so that the lookup operation may be made on the bus.

4.2 GENERAL SPECIFICATIONS

4.2.1 Exec and User Programs

A distinction is made in this specification between programs which have privileged access to system resources and those which do not. If a program is given such privileges, the KT11-B cannot protect the system against abuse and failure caused by that program. Such a program (the "Exec") is generally responsible for resource allocation among unprivileged programs ("User" Programs). These resources include

1. CPU Time.
2. Process Priority.
3. Core Storage.
4. Sharable Code.
5. Sharable Devices (Disks, Drums, etc.).
6. Non-sharable devices (DECtapes, Paper tape, etc.).

The Exec is responsible for efficient management of these resources and thus assumes control whenever a user attempts to:

1. Halt the processor.
2. Execute a trap loop.
3. Access core or sharable code in a manner not permitted him.
4. Use a part of his address space for which no physical core is then allocated.

Processor priority cannot be changed by a User. The user may request system resources through Exec traps (instructions which cause traps to the Exec). The Exec receives all interrupt requests as well as maintains processor priority and system activity to satisfy user resource requests in the most effective manner. The Exec is also responsible for orderly system shut down and restart on power transients, and therefore, receives power fail and recovery traps as well.

The system starts in Exec mode without paging. The Exec establishes paging parameters and enters paged operation in its start sequence.

4.2.2 Page Maps and Keying

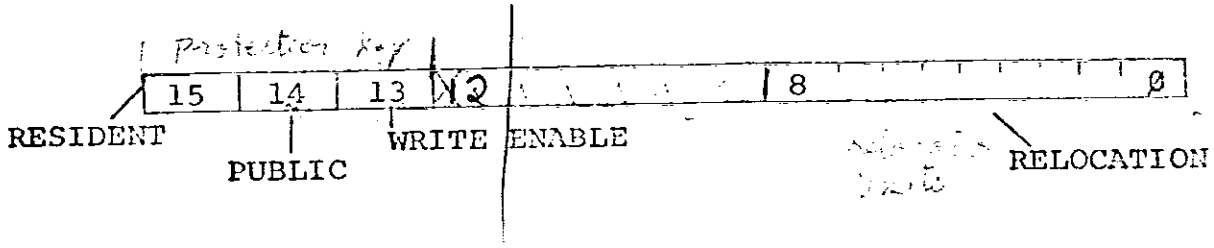
Two of the parameters of the paging operation are the address of the User and Exec Page Maps in core. When the processor is executing a User mode program, references to pages whose page number and mode do not appear in the associative memory cause fetches ("page map entry lookups") to be made in memory to a table of User Page Map Entries, the User Page Map. Such references when referring to User page Ø cause a fetch to be made of the zeroth sequential entry in the User Page Map. Referrals to User Page 1 (if the entry for User Page 1 does not appear in the associative memory) cause a fetch to be made from the first sequential entry in the User Page Map, and so on. Each entry consists of one 16 bit word and contains the protection key and relocation for the corresponding page.

Registers in the KT11-B (the User and Exec Map Control registers) specify the physical core addresses for the User and Exec Page Maps. Paging may be switched for various users by modifying the User Map Control register to point to different page maps.

Keying for a page indicates to the KT11-B whether or not that page is "resident", "public", and "write-enabled". Three bits of each Page Map Entry word hold the protection key for the page each controls.

The (virtual) address space for any program in a PDP-11 is 2^{16} bytes (2^{15} words). If each page is made to contain 2^{9} bytes (256 words), then each User will have 2^{7} virtual pages. Since each page map entry requires 1 word, this implies that each User will require a page map consisting of 128 words (half a page). System information for that User could occupy the remainder of that page. Choice of a page size of 2^{10} bytes implies a 64 word page map (one eighth of a 512 word page) which represents unacceptable core inefficiency for the Exec - 32Ø words lost per core - resident User. (User management is greatly eased if all the information pertaining to a user and no information not pertaining to that User is contained in one page which the Exec can allocate and may be swapped in and out of core with that User.)

There are 2^{18} bytes in a physical address space of the Unibus. If each page is 2^9 bytes, there are 2^9 physical pages. The physical page address is thus specified by 9 bits selecting bytes within a page and 9 bits specifying the page address within the physical address space. The 9 bits specifying the byte address within a page are taken from the 9 low order bits of the address lines of the processor bus (see 0.2.4)-A <Ø8:ØØ> . The 9 bits specifying the page address are taken from the 9 relocation bits of the appropriate page map entry word. Each page map entry word contains 3 bits for the protection key, 9 bits for the relocation, and 4 bits unused.



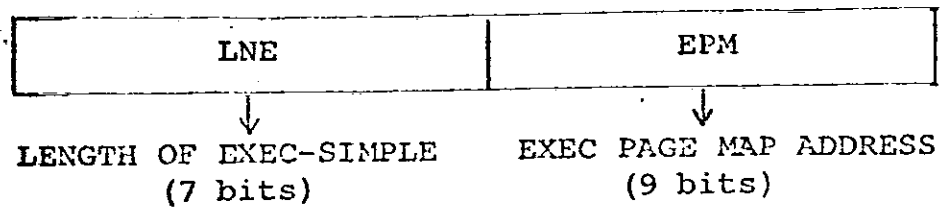
4.2.2.1 The Exec Page Map Page (EPMP)

The Exec Page Map Page is considered to be in page zero of the Exec's virtual address space. For reasons discussed in 4.2.6, exec page zero is not paged through the associative memory. Instead it is considered to be the page specified by the low order 9 bits of the Exec Map Control Register. The PDP-11 Interrupt and trap vectors are in addresses Ø - 377 of this page. The Exec Page Map Entries are in addresses 4ØØ and up. (Note that Execs running in separate processors, sharing some common core and peripherals but governing private busses with private peripherals as well, can maintain distinct interrupt vectors).

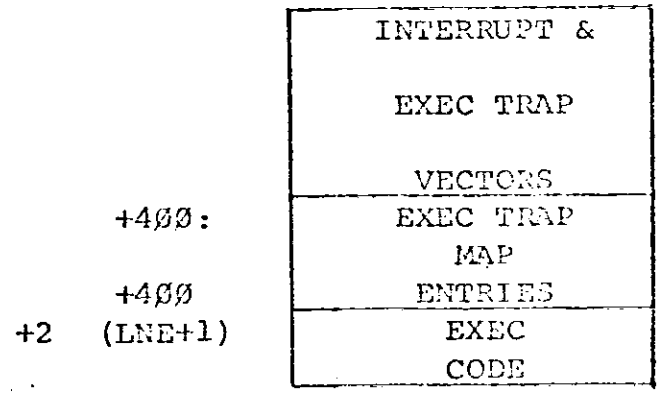
There is for the Exec, a set of parameters including Dectape Directories User I/O buffers, User Page Maps, User Context etc. that is unique to each User. This set can be accessed by dedicating some Exec address space to each core-resident user. Tables are then required to couple each User with the appropriate space. Considerable use of indexing is required in this scheme.

In order to ease the pressure on exec address space, reduce the instruction time when referring to these "per-process" variables, and to decrease monitor complexity, "exec-per-process" paging is provided. Pages containing such variables are associated with "Exec-Per-Process Entries" in the lo address half of the User Page Map Page. References to these pages of the Exec are thus paged according to the currently active user (whose Page Map Page is referenced by the User Map Control Register).

The boundary between Exec simple space and Exec-per-process space is determined by the high order 7 bits of the Exec Map Control Register. The Exec Map Control Register and Page Map Page are thus configured as:



PHYS ADRS EXEC ADRS EXEC PAGE MAP PAGE
1000_8*EPM Ø



Exec address Ø through 1000_8* LNE +777 are in Exec simple space and have page map entries in exec addresses 4ØØ through 4ØØ + 2* LNE. Exec addresses above 1000_8 * LNE + 777 are in Exec-per-process space and have page map entries in addresses 2*x (LNE + 1) through 376 of the User Page Map Page.

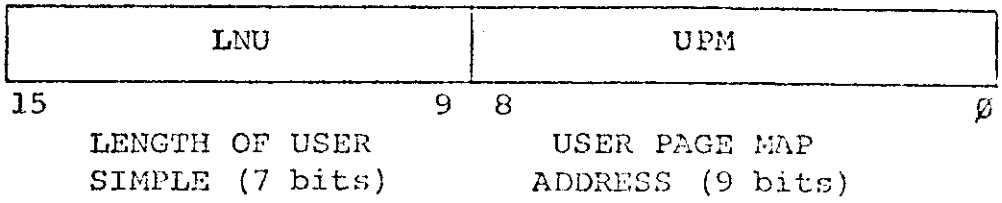
4.2.2.2 The User Page Map Page (UPMP)

The UPMP holds the USER PAGE MAP ENTRIES and the Exec-per-Process Entries. It has an address in Exec-Per-Process space but (generally) not in the user space. We will establish the convention that the UPMP appears in the first page of exec-per-process space - namely 1000_8* (LNE + 1). The physical address of this page is determined by the low order 9 bits of the User Map Control Register. (By our convention, the low order 9 bits of address 2* (LNE + 1) of the user page map page are equal to the low order 9 bits of the User Map Control Register.)

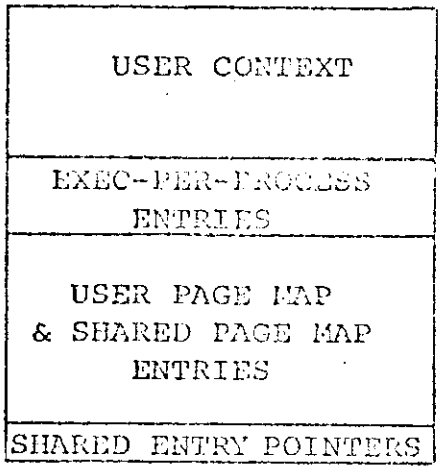
To provide for shared paging in which several users share use of code, movable in core, the User address space is broken into two parts: simple user space and shared user space. The high order 7 bits of the User Map Control Register determine the boundary between simple and shared user space. The User Page Map Page does not hold the page map

entries for shared pages but holds "shared entry pointers" instead. The low order 9 bits of the Shared Entry pointer contain the physical page and the high order 7 bits contain the entry number in which the Shared Page Map Entry is to be found. (Shared Page Map Entries may be in addresses 4ØØ-777 only of any page.)

A User Page Map Page and the User Map Control Register are arranged as follows:



PHYS ADRS EXEC ADRS
UPM $1ØØØ_8$*(LNE + 1)



+2*(LNE+1)

$+4ØØ_8$

$+4ØØ_8$+2*(LNU+1)

$+777_8$

(Some other Page Map may have shared entry pointers which point to entries Ø through LNU of this Page Map.)

4.2.3 Exec Trap and Interrupts

A variety of conditions will cause an operating user program to be stopped and control transferred to the Exec. In each case, this is done via a slight modification of the standard trap interrupt sequence: the DATO's of the modified sequence cause no modification of core and are directed instead to special registers (the Page Program Status Register and the Page Program Counter) within the KT11-B Control. The address to which the Process Program Counter

was directed is also saved. In this modified sequence a flag bit, PGIU, the "interrupted user", bit is set or cleared to distinguish between termination of User and Exec Programs.

Exec Trap 4 - This vector handles Exec Bus Errors (odd address errors and time outs) and Exec or User Illegal Instructions, and Paging Violations. All EXEC traps except time out and paging errors will run through the standard sequence. The Stack Overflow trap is disabled to protect the Exec from stack pointers set by the User below the stack boundary: interrupts even through the modified sequence would cause the stack overflow flag to be set if the stack is below the stack boundary. Exec Bus Errors and Exec or User Paging Violations proceed according to the modified sequence. As a consequence, if Exec Paging Violations and Bus Errors occur (time-out, etc) status will be lost. It will be impossible to restart that User. If the memory bank in which the vector is located is dead, the processor will halt with address 4 in the address indicators. If the memory bank in which the trap service routine is located is dead, the processor will repeatedly execute the trap vector - only the address 4 will be seen in the address indicators but the processor will run until stopped from the console. The assumption will be made in the description to follow that the first few instructions of the Exec 4 service routine will be successfully executed. These instructions save the special registers, setup the stack pointer, and clear the PGIU and other error indication bits in the Page Control Register.

Exec Bus Errors, Exec and User Paging Violations, and User Instruction Violations are processed by the modified sequence. Exec Instruction Violations are processed by the standard sequence. Thus the first action of the Exec 4 trap service routine should be to test the Page Control Register. Bits will be set indicating whether a Bus Error, a Page Violation, or a User Instruction Violation has occurred and in which mode. If any bits are set, the special registers should be saved, the stack pointer adjusted, software flags set, and all error bits in the Page Control Register cleared - an uninterruptable sequence of perhaps 3Ø microseconds. If no bits are set, an Exec Illegal Instruction caused the trap.

Exec Trap 1Ø - Reserved instructions from both User and Exec routines come through this trap. This can be used to cause emulation of unimplemented instructions. If the reserved instruction came from a User, the PGIU bit is set. Otherwise, it is unaffected. Reserved instructions cause operation of the "modified sequence" and transfer to Exec mode if executed in User Mode. If in Exec Mode, the standard sequence is exercised. The PGIU bit should be cleared early in the service routine for this trap (and before interrupts are enabled).

Exec Trap 14 - This trap is taken for execution of op code ØØØØØ3 or operation of the trace flag in either User or Exec Mode through the standard sequence if from the Exec Mode or through the modified sequence (with a set of PGIU), if from User Mode. Through use of this trap users can share DDT without reserving any address space for its use. The Exec can also use DDT as part of its address space.

Exec Trap 2Ø - IOT instructions cause execution of this trap (as in Exec 1Ø and Exec 14).

Exec Trap 24 - Power fail and restart from both User and Exec Modes come through this trap (standard sequence from Exec, modified sequence from User). The PGIU bit can be tested to determine whether this is from the Exec or User. This bit should be cleared within 2 milliseconds of the trap to provide for another power fail. When power is coming up, all devices on the UNIBUS are initialized. This puts the KT11-B in unpaged operation in which physical (rather than Exec Virtual) address 24 is used.

Exec Trap 3Ø - EMT instructions (as in Exec 1Ø and 2Ø).

Exec Trap 34 - Trap instructions in the KA11 receive higher priority than bus requests. The potential exists for a user program to postpone interrupt servicing indefinitely by successive trap instructions. The KT11-B detects attempts to execute two successive TRAP instruction in User mode and directs the second to the Exec through the Modified Sequence. This vector is also used for execution of the TRAP instruction in Exec Mode.

Interrupts in User Mode

The Exec needs to service interrupt requests since Users are not generally empowered to run at any other priority level than level Ø; Users are generally not given access to I/O registers; and the Exec needs system activity information in order to manage system resources. If an interrupt occurs while a user is running, it is necessary to save the user program counter and status register and transfer control to the Exec at the location and status specified by the interrupt vector for the interrupting device. No reliance may be made on the contents of the User stack pointer. The "modified sequence" is therefore, put in operation if a User program has been interrupted. The interrupt vector is taken from Exec paged core. The PGIU bit is set in the page control register. This bit is cleared when an interrupt is serviced while an Exec program is running.

4.2.4 Paging Violations

Residency Violations - As system requirements for core grow, the Exec may begin to swap users in and out of core from a swapping store. To cut down on the swapping overhead, the Exec may choose to swap out only the portions of user programs not currently or about to the needed. Thus users may continue to run with only portions of programs "Resident" in core. Those pages of running user's programs which the exec swaps out it marks as "Non-Resident" in their page maps. (The residency key of the appropriate page map entry is made clear.) When the user tries to reference such a page, a "Residency Violation" occurs and a Page Violation Trap occurs. The Exec can then bring in the required page, set the residency key bit in the page map entry, and restart the user.

There are difficulties, however, in restarting the user:

1. The program counter is advanced some undetermined number of addresses as the execution proceeds through multiple word instructions.

2. Registers used in auto decrement and auto increment addressing are modified by the operation of the instruction. Unless the point to which the instruction proceeded before causing the residency violation is known, the modifications made to the registers cannot be determined.

3. Implicit stack operations occur in some instructions (JSR, RTS, RTI, and TRAP). Which modifications to the processor stack pointer were made before the instruction was stopped must be known.

Therefore, while in user mode, the address from which the instruction currently being executed was fetched, and the processor states that were entered during its execution are saved in KT11-B registers.

Privacy Violations - To provide for controlled use of leased programs and sub-programs, pages may be marked as "Private". Private pages may only be accessed by instructions fetched from private pages. An attempt to access private pages from public pages result in a "Privacy Violation" and "Page Violation Trap". Execution may be transferred to private pages from public pages only at Entry Points. The first instruction fetched when execution is transferred from instructions in public to those in private pages must be the Entry Point Instruction "BR .+2" (000400). Other instructions will cause an Instruction Violation and a Page Violation Trap. This permits leased sub-programs to be shared between users with some protection against user errors.

Write Violations - One bit of the protection key in each page map entry word governs whether the bus operations DATa In-Pause, DATa Out, and DATa Out Byte are permitted for each page. This bit "Write Enable" must be set to 1 to enable these operations (DATa In is not governed by this bit). Attempts to do a DATIP, DATO, or DATOB to a page operation is not performed and a Page Violation Trap occurs.

Leaving this bit clear for a page ensures that no modifications may be made to that page. (Note that if a page is not write enabled, there is no need to write back on the disk each time it is swapped out. It can't have been modified.)

Instruction Violations - Attempted execution of certain privileged instructions have a different effect when fetched in User mode than when fetched in Exec. Execution of these instructions in Exec Mode is as in a normal KA11 (e.g. HALT stops processor operation). Attempts to execute these instructions in User Mode cause "Instruction Violations". A table of privileged instructions with their suggested effects (a function of Exec coding) is presented below:

Mnemonic	Code	Effect
HALT	000000	Monitor stops user job. Types message on job TTY
WAIT	000001	Monitor pauses user job, restarts when I/O for user done.
RESET	000005	Monitor resets user I/O tasks.

Address Violations - In the KA11, when an attempt is made to reference a word at an odd address, the reference is not made and a bus error trap occurs. If the processor stack pointer has been set to an odd value or set to point to non-existent memory, then another bus error will occur during the error trap. Such bus errors on bus errors cause the KA11 to halt. The KT11-B modifies this operation to cause transfer to be made to Exec mode through the Page Violation Trap when an odd address error condition occurs. The odd address error condition is disabled when generated by a User program. Thus in User mode, the processor will not halt even if the processor stack pointer has been set to an odd value. The Exec should set the processor stack pointer to some appropriate value before causing any operation making use of the processor stack or permitting interrupts. Odd address errors in User Mode cause "Address Violations".

Time Out Violation - Normally users will not be enabled for memory addresses that do not exist. But should a memory or device go bad that the user has been enabled for, it is more an Exec than a User function to work around its loss. Therefore, if a time out should occur on what should have been a legitimate bus operation, transfer shall be made to EXEC mode through the Paging Violation Trap. This condition is a Time-Out Violation.

Paging Error Sequence - Transfer is made to the EXEC whenever any of these violations are caused by the user. When these violations occur, the status of the address (A <15:00>) and Data (D <15:00>) lines are saved in registers of the KT11-B and the processor is forced through a fake TIME OUT ERROR to do a Page Violation Trap. The entry in the associative memory for the page causing the violation is cleared. This prevents conflicts between an old page map entry in the associative memory and a new page map entry modified by the Exec (to permit writing, etc) and put in the core page map page.

4.2.5 User Traps

User TRAP Instructions (if not trap violations) are serviced by the routine selected by the trap vector at location 34 in user page zero. The trap proceeds normally through User paging except that the processor priority and state of the T-bit are not effected.

4.2.6 Starting a User

When a User is to be started or restarted by the Exec, that User's context must be loaded into the pertinent machine registers (general registers, program status, program counter, User Map Control register, system software flags and pointers). System state words relevant to the old User must be saved or discarded.

The contents of the associative memory represent state words that need not be saved (they are only copies of the page map page words) but must be cleared from the system before starting a new user. To accomplish this, paging of Exec Page Ø, (whose entry in the associative memory would be 8 bits of Ø) is treated as a special case and paged to that physical page specified by the Exec Map Control Register. A Ø-word in the associative memory thus will not match. All bits in the 8 x 8 associative memory are cleared (loaded with "null") upon any modification (DATO & DATOB) of the User Map Control Register. Thus the old state is cleared.

A control sequence is provided in the KT11-B to permit loading of program counter and program status and setting of the User Mode

bit in one uninterruptible machine cycle. When a certain bit in the Page Control Register is set, the next bus operation originated by the CP and not an instruction fetch causes the KT11-B to enter the Start User Sequence. The first two bus operations encountered while in this sequence cause the KT11-B to ignore the address specified on the bus and respond with the contents of the two special registers (the Page Program Status and the Page Program Counter). An RTI is the only instruction that will restore the User PC & PS properly. User mode is entered when the KT11-B sees a load of the Processor Status. The program sequence to return to a User should be:

```
MOV #340, PS        ;make sequence non-interruptible.

MOV KTSP, R6        ;restore User Stack Pointer.

INC KTCS            ;set "start user" bit.

RTI                 ;cause new PC and PS to be loaded
                     from KTPC and KTPS and User Mode
                     to be entered.
```

4.2.7 Exec-User Communication

The User may request services of the Exec through Exec Trap Instructions (see 4.2.3). The Exec may examine and deposit in User locations through operation of the "Page Window". An address on the bus (after paging) is identified as the Page Window. Bus operations to the Page Window cause the data on the D lines to be written into or read from the User location specified by the contents of a register in the KT11- the Page Address register. A lookup is made for the User Page Map Entry associated with the address so specified. A Shared Entry Pointer is processed if required and finally the User Paged address is referenced and the bus operation specified for the Page Window is performed on the User Paged address generated in this manner.

4.3. INTERNAL OPERATION

The various KT11-B operations are under the control of two shift registers: a two-bit X shift register, and a four-bit Y shift register (located on the M826 clock module). The state of the KT11-B at any time is given as a combination of the contents of the two registers; it is written in the form X.Y. When the KT11-B is idle (not currently responding to any Unibus operation), both registers are in the zero state, and the state of the KT11-B is given as Ø.Ø. When the KT11-B receives an MSYN signal from the processor, the M826 clock is started and a series of states are entered, depending on the function to be performed. Figure 4-1 is a state diagram of the KT11-B. Table 4-1 is a state table, giving the conditions of transition from one state to the next and noting some of the signals asserted in a particular state.

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
STATE 60							
.MOK & EXZE] .GE*-INTRAP:	1.Ø	CLKA XSHL	-	-	AMXAØ,B1 AMXB18←1	DVD CPSSYN	PAGE MAP ENTRY WINF←Ø; INTR: PGIU← PGUM(1)
MOK & EXZE] GE*-INTRAP:	-	-	SPOR2 ◁3:1▷ ← 6 SPOR2 ◁Ø▷ ←PGUM(Ø)	SPMXØ	-	-	PAGE MAP ENTRY OR EXEC ZERO
K & EXZE] E*-INTRAP:	1.Ø	CLKA XSHL	-	-	AMXAØ,B2	DVD CPSSYN	NORMAL PAGING WINF ← Ø; INTR: PGIU ← PGUM(1)
-EXZE *-INTRAP:	-	-	SPORØ	SPMXØ	-	-	NORMAL PAGING & NOT EXEC ZERO
INTRAP:	Ø.8	CLKA YSHR	-	SPMXØ	-	DVD CPSSYN	INTR OR EXEC TRAP WINF ← Ø; INTR: PGIU ← PGUM(1)
	1.Ø	CLKA XSHL	-	-	AMXA2,B2	DVD CPSSYN	UNPAGED OPERATIO

TABLE 4-1 (Cont.)

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
STATE Ø.1							
WINF(1):	1.1	CLKA XSHL	SPOR2←-6	-	AMXAØ LATCH B (7.1)	-	WINDOW, GET US PAGE MAP ENTRY WELF-- 1; LONGF
WINF(Ø)*-PGC<1:Ø> (3):	Ø.Ø	CLKC YSHR	SPOR2←-1 PT:WRITE	-T:SPMXØ LATCH SPMX	-	T:SSYN	INTREX SEQ, WRITE PG6
WINF(Ø)*PGC<1:Ø> (3):	Ø.Ø	CLKC YSHR	SPOR2←-3	-	-	T:SSYN D	SUSR SEQ; READ PGS
STATE Ø.3							
WINF(1):	Ø.1	CLKA YSHR	SPOR2←6	LATCH SPMX	AMXAØ,B1 AMXB18←1 LATCH B	-	WINDOW - FORM PAGE MAP ENTRY #
WINF(Ø)*-PGC<1:Ø> (3):	Ø.1	CLKA YSHR	SPOR2←1	SPMXØ	-	-	INTREX, SETUP
WINF(Ø)*PGC<1:Ø> (3):	Ø.1	CLKA YSHR	SPOR2←3	-	-	D	SUSR, SETUP PG
STATE Ø.7							
WINF(1):	Ø.3	CLKA YSHR	SPOR2←4	SPMX2 T:LATCH SPMX	AMXAØ,B1 AMXB18←1	-	WINDOW, GET AD OF PAGE MAP ENTR

TABLE 4.1 (CONT.)

CONDITION	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
WINF(Ø)*-PGC<1:Ø>(3):	Ø.3	CLKA YSHR	SPOR2←2 PT:WRITE	SPMX1	-	CPD	INTREX, WRITE PG
WINF(Ø)*PGC<1:Ø>(3):	Ø.3	CLKA YSHR	SPOR2←3	-	-	-	SUSR, SETUP PG8
STATE Ø.15							
-PGC<1:Ø>(3):	Ø.7	CLKA YSHR	SPOR2←3	SPMX1	-	CPD	INTREX, SETUP PG PGUM←Ø
PGC<1:Ø>(3):	Ø.7	CLKA YSHR	SPOR2←2	-	-	D T:SSYN	SUSR, READ PG7
STATE Ø.14							
-PGC<1:Ø>(3):	Ø.15	CLKA YSHR	SPOR 2←3 PT:WRITE	-T:SPMX1 LATCH SPMX	-	T:SSYN CPD	INTREX, WRITE PG PGIU←PGUM(1)
PGC<1:Ø>(3):	Ø.15	CLKA YSHR	SPOR2←2	-	-	D	SUSR, SETUP PG7
STATE Ø.12							
-PGC<1:Ø>(3):	Ø.14	CLKA YSHR	SPOR2←3	SPMX1	-	CPD	INTREX, SETUP PG
PGC<1:Ø>(3):	Ø.14	CLKA YSHR	SPOR2←2	-	-	-	SUSR, SETUP PG7

TABLE 4.7 (CONT.)

CONDITION	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
STATE Ø.8							
-	Ø.12	CLKA YSHR	-	-	-	-	
STATE 1.Ø							
- [AMOK+EXZE] *PAGE*WINF(Ø):	1.1	ACLK YSHL	-	SPMXØ	AMXAØ AMXB18←Ø T:LATCH B	-	LOOKUP, TRANSLF A<15:Ø9> TO <7 (ENTRY #) LONGF←Ø; TIME<Ø
[-AMOK+EXZE] *PAGE*WINF(Ø):	-	-	SPOR2<3:1> ←7 SPOR2<Ø> ←PGUM(Ø)	-	-	-	LOOKUP & EXEC ZERO NEED EITHER PGU OR E
[AMOK+EXZE] *PAGE*WINF(Ø):	1.1	ACLK YSHL	-	SPMX3 KEY 2	AMXAØ,B2 T:LATCH B (7.1)	-	NORMAL PAGING A<8:Ø> AS IS LONGF←Ø; TIME← WELF←Ø
AMOK*-EXZE *PAGE*WINF(Ø):	-	-	SPORØ	-	-	-	NORMAL PAGING
-PAGE:	3.Ø	CLKC XSH	-	-	AMXA2,B2 T:LATCH A,B	C;CPD	UNPAGED OPERATION

TABLE 4.1 (Cont.)

CONDITION	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
PAGE*WINF(1):	1.1	ACLK YSHL	SPOR2←7	LATCH SPMX LATCH KEY	AMXA1,BØ T:LATCH A		WINDOW, BUS OF ON PAGED (PGA) LONGF←Ø; TIME← WELF←Ø
STATE 1.1							
[-AMOK+EXZE] *WINF(Ø):	1.3	ACLK YSHL	SPOR2 ◁3:1▷ 7 SPOR2 Ø ←[PGUM(1) +LONGF(1)]	SPMX2 T:LATCH SPMX	AMXAØ LATCH B T:LATCH A AMXB18 ←LONGF(Ø)		LOOKUP OR EXEC ZERO SPMX2 MUST BE DRIVEN DIRECTLY FROM CI A PASS←LONGF(1)
[AMOK+EXZE] *WINF(Ø):	1.3	ACLK YSHL	-	SPMX2 T:LATCH	AMXAØ LATCH B	C;CPD	NORMAL PAGING C◁1:Ø▷(Ø) *PGUM(1 TRIN←TRAP -[FETCH+SERVICE] PGUM(Ø): TRIN← FETCH*PGC◁15:12 (Ø): PGSS←Ø, PGDS← SRC: PGSS←1 DST: PGDS←1 PASS LONGF(1)
AMOK*-EXZE*WINF(Ø):	-	-	SPORØ	-	-	-	NORMAL PAGING & NOT EXEC ZER

CONDITION	NEXT STATE	CLOCK	SCRATCHPAD	I/O OUT LATCH	ADDRESS LATCH	BUS	OTHERS
WINF(1)*WELF(1):	1.3	CLKA YSHL	SPOR2←5	SPMX2 T:LATCH SPMX	AMXAØ LATCH B T:LATCH A	-	WINDOW, LOOKUP FOR (PGA) SEE ABOVE FOR SPMX2 PASS←LONGF(1)
WINF(1)*WELF(Ø)	1.3	CLKA YSHL	SPOR2←4	LATCH SPMX LATCH KEY	AMXBØ LATCH A T:LATCH B	C;CPD	WINDOW, BUS OPR PASS←LONGF(1)
STATE 1.3							
-[AMOK+EXZE] + WINF(1)*WELF(1)	3.3	ACLK XSHL	-	LATCH SPMX	LATCH A,B	-	LOOKUP PAGE MAP
[AMOK+EXZE] *WIND *WINF(Ø)*-PKER	1.7	CLKA YSHL	SPOR2←4	LATCH SPMX	LATCH A,B	-	WIND*WINDF: PGAV WINDOW DETECTE SETUP PGA
[AMOK+EXZE] * PKER *-[WINF(1)*WELF(1)] *-[PGC<15:12>(Ø)*FETCH]	1.7	CLKA YSHL	SPOR2←4	SPMXØ LATCH KEY	LATCH A,B	-	PROTECTION ERR
[AMOK+EXZE] * -WIND *-[WINF(1)*WELF(1)] *[-PKER+PKER *FETCH*PGC 15:12>(Ø)]	3.3	CLKA XSHL	SPOR2←Ø	SPMXØ LATCH KEY	LATCH A,B	C;CPD	NORMAL PAGING

TABLE 4 (CONT.)

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
STATE 1.7							
-PKER:	Ø.7	CLKA XSHR	SPOR2←4	LATCH KEY SPMX2	-	-	WINDOW OPR. WINF←1
PKER:	1.15	CLKA XSHL	SPOR2←4 PT:WRITE	SPMXØ LATCH KEY	-	-	PROTECTION ERR KEY 2 : PGRV←1 KEY Ø*-C 1/ ODD ADRS ERR: PGRV←1
STATE 1.15	1.14	CLKA YSHL	SPOR2←6	SPMX1	-	CPD	PROTECTION ERI
STATE 1.14	1.12	CLKC YSHL	SPOR2←6 PT:WRITE	SPMX1	-	CPD	PROTECTION ERI AMXØ; PT: AMCLK PAGE TRAP
STATE 1.12	Ø.12	CLKA XSHR	-	-	-	-	
STATE 1.8							
WINF(1):	1.Ø	CLKA YSHL	SPOR2←4	LATCH SPMX LATCH KEY	AMXA1,BØ	-	WINDOW OPR LOOKUP FOR (PGA OK
WINF(Ø):	1.Ø	ACLK YSHL	SPOR0	LATCH SPMX		-	SIMPLE LOOKUP COMPLETED RCIX

TABLE 4.1 (Cont.)

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
STATE 3.Ø							
-[AMOK + [illegible] +WINF(1)*WELF(1)] +[LONGF(1)*PASS(Ø)]] *PAGE:	3.8	ACLK YSHR	SPOR1	SPMX1 T:LATCH SPMX KEY 1 T:LATCH KEY	LATCH A,B	MSYN	PAGE MAP LOOKUPS
[[AMOK+EXZE] *-[WINF(1)*WELF(1)] *[-PAGE*-PADSEL]:	2.Ø	CLKC XSHL	SPOR1	SPMX1 T:LATCH SPMX LATCH KEY	LATCH A,B	MSYN C;CPD T*-PKIV *-FETCH *-PAGE: WSSYN	NORMAL PAGING & UNPAGED OPR BUS OPS DONE HERE FOR UNPAGED OPR T*PKIV: PGIV←1
PAD SEL:	3.1	CLKA YSHL	SPOR3	SPMX1	LATCH A,B	C;CPD	SCRATCHPAD ACCESS IN UNPAGED OPR.
STATE 3.1							
-AMOK+EXZE] *[LONGF(Ø)+PGUM(1) +PASS(1)] +[WINF(1)*WELF(1)]] *TIME(Ø):	3.Ø	CLKA YSHR	SPOR1	SPMX1	LATCH A,B	MSYN	PAGE MAP LOOKUP (IN USER PAGE MAP PAGE IF EXEC-PER-PROCE
-[AMOK+EXZE] *LONGF(1)*PGUM(Ø) *PASS(Ø)*WINF(Ø):	1.1	CLKA XSHR	SPOR2←7	SPMXØ	AMXAØ,B1 AMXB18←Ø LONGF(Ø) LATCH B<7:Ø>	-	GET ADDRESS OF PAGE MAP ENTRY IN USER PAGE MAP PAGE-EXEC-PER-PROCESS

CONDITIONS	LOOP STOP	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
[AMOK+EXZE]*TIME(Ø) *-[WINF(1)*WELF(1)] *-PAD SEL:	3.Ø	CLKA+C YSHR	-	SPMX LATCH KEY	LATCH A,B	MSYN C;CPD T*-FETCH: DYSSYN	NORMAL PAGING
PAD SEL:	2.1	CLKA XSHL	SPOR3	SPMX1	LATCH A,B	C;CPD	A<5:1>(6): AMCLI RCLR
TIME(1):	2.1	CLKA XSHL	SPOR2←4	SPMXØ	LATCH A,B	C;CPD	TIME OUT DURI PG.
STATE 3.3							
[-[AMOK+EXZE] *[-LONG+PGUM(1) +PASS(1)] +[WINF(1)*WELF(1)]]	3.1	CLKB YSHR	-	LATCH SPMX	LATCH A,B	T:MSYN	PAGE MAP LOOK FROM CORE T*PASS(Ø): LONGF←LONG
-[AMOK+EXZE] *LONG*WINF(Ø) *PGUM(Ø)*PASS(Ø):	3.1	CLKA YSHR	SPOR2←6	LATCH SPMX	LATCH A,B	-	EXEC PER PROCE GET USER MAP CONTROL T*PASS(Ø): LONGF←LONG
[AMOK+EXZE]*-PVER *-[WINF(1)*WELF(1)]:	3.1	CLKB YSHR	SPOR[illegible] PT*WREN: WRITE	SPMXØ LATCH KEY	LATCH A,B	MSYN C;CPD	NORMAL PAGING BUS OPS OCCUR THIS STATE

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
[AMOK+EXZE]*PVER *-[WINF(1)*WELF(1)]:	3.7	CLKA YSHL	SPOR2←Ø PT*WREN: WRITE	SPMXØ LATCH KEY	LATCH A,B	-	PROTECTION ERROR
PAD SEL:	-	CLKA	-	-	-	-	PAD SELECTED TIME OUT NOT POSS.
TIME(1):	-	CLKC	-	-	-	-	TIME OUT
STATE 3.7							
[AMOK+EXZE] *-[WINF(1)*WELF(1)]:	1.7	CLKA XSHR	SPOR2←4	SPMXØ LATCH KEY	-	-	PROTECTION ERROR DURING FETCH
-[AMOK+EXZE]: +[WINF(1)*WELF(1)]:	3.3	CLKA YSHR	-	-	LATCH A,B	-	GET SHARED PAGE MAP ENTRY
STATE 3.15	3.7	CLKA YSHR	-	-	LATCH A,B	-	FORM ADDRESS SHARED PAGE ENTRY
STATE 3.14	3.15	CLKA YSHR	-	LATCH SPMX	AMXA1,B1 AMXC18←1 T:LATCH A,B	-	PASS←1

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
STATE 3.12							
	3.14	CLKA YSHR	-	LATCH SPMX	AMXA1,B1 AMXC18←1	-	SHARED PAGE ENTRY (POINTER IS IN INPUT LATCHES)
STATE 3.8							
-[PASS(Ø)*LONGF(1)] *WINF(1):	1.8	CLKA XSHR	SPOR2←4	SPMX1 SPMX LATCH KEY	LATCH A,B	-	WINDOW OPR LOOKUP FOR (PC COMPLETED REL. ADRS. IN S.P.
-[PASS(Ø)*LONGF(1)] *WINF(Ø):	1.8	ACLK XSHR	SPOR1 PT:WRITE	LATCH SPMX	LATCH A,B	-	SIMPLE LOOKUP DONE AMMX1; AMXM T: AM CLK
PASS(Ø)*LONGF(1):	3.12	CLKA YSHR	-	LATCH SPMX	LATCH A,B	-	SHARED ENTRY POINTER IN LATCHES FORM ADDRESS OF SHARED PAGE ENTRY
STATE 2.Ø							
-[PKIV*PAGE] *-PAD SEL:	Ø.Ø	CLKC XSHL	-	LATCH KEY	LATCH A,B	C;CPD DVSSYN	BUS REFERENC DONE SHUT DOWN FETCH*PAGE: PBIN←KEY 1 (

CONDITIONS	NEXT STATE	CLOCK	SCRATCHPAD	INPUT LATCH	ADDRESS LATCH	BUS	OTHERS
PAD SEL:	Ø.Ø	CLKC XSHL	SPOR3	LATCH KEY	LATCH A,B	C;CPD	SCRATCHPAD REFERENCE DONE
[PKIV*PAGE] *-PAD SEL:	Ø.Ø	CLKA XSHL	SPOR2←4	LATCH KEY	LATCH A,B	C;CPD	INSTRUCTION VIOLATION OR TIME OUT
STATE 2.1							
PAD SEL:	2.Ø	CLKC YSHR	SPOR3 -T*C<17>(1): WRITE	LATCH KEY SPMX1	LATCH A,B	C;CPD;D T:SSYN	SCRATCHPAD REFERENCE
-PAD SEL:	2.Ø	CLKC YSHR	SPOR2←4	SPMXØ	LATCH A,B	C;CPD	INSTRUCTION VIOLATION OR TIME OUT AMMXØ;PT:AMCLK

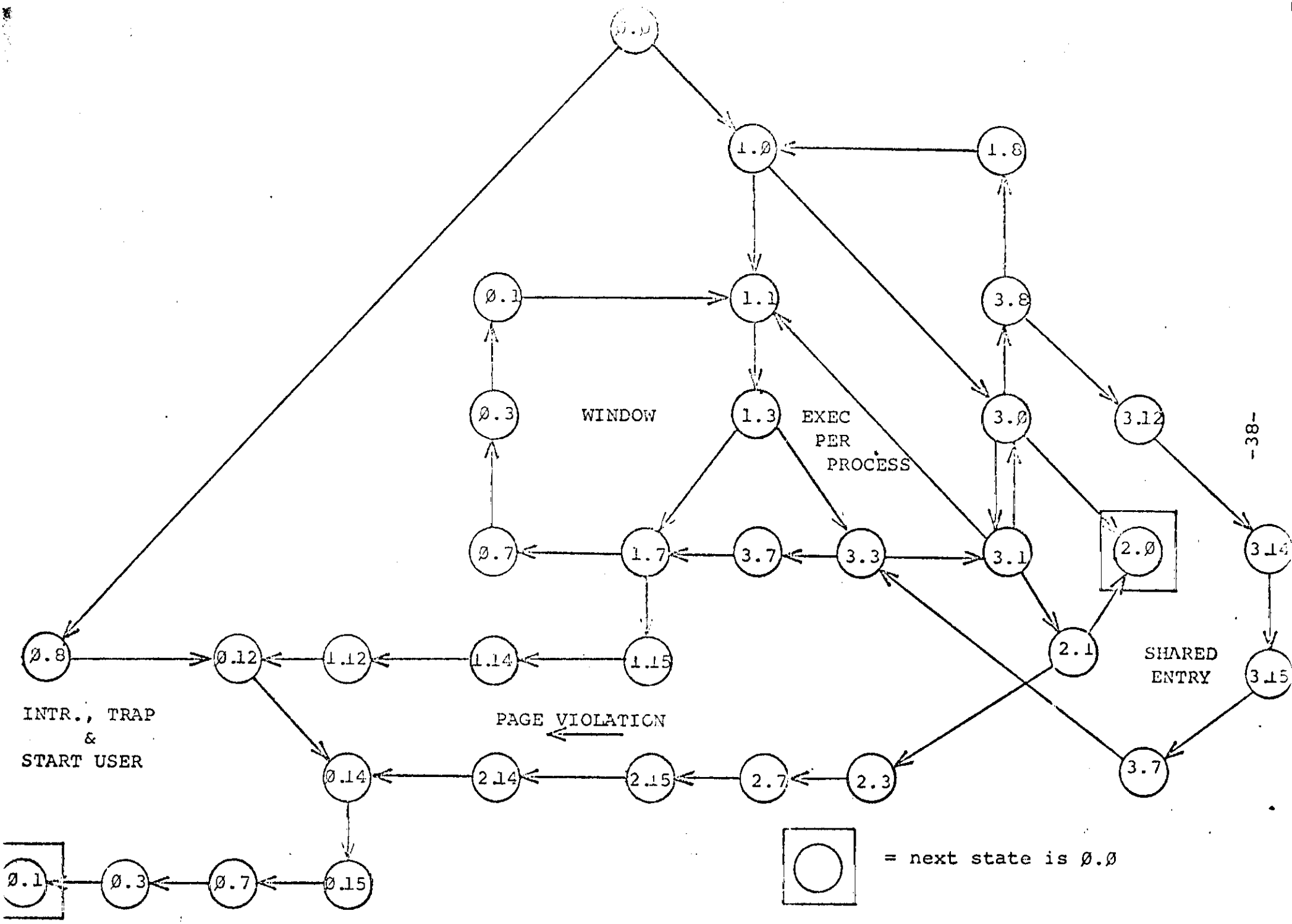


Figure 4-1 KT11-B State Diagram

SECTION 5

MAINTENANCE

5.1 SPECIAL TEST EQUIPMENT

The W13Ø/W131 maintenance board used for troubleshooting the PDP-11 processor may also be used for troubleshooting the KT11-B. When it is to be used, it should be inserted in slot A14 of the KT11-B logic. The functions of the switches and lights are thus redefined according to Figure 5-1. The lights display the KT11-B state, certain enabling signals, and certain status bits. Two switches are used: one enables the maintenance clock and the other provides the clock pulse used for toggling from state to state. No other special maintenance equipment exists. The W13Ø/W131 is not included as part of the basic KT11-B.

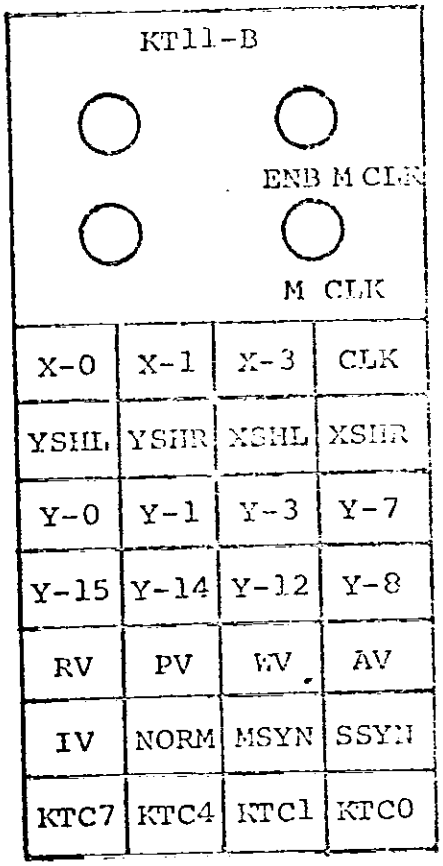


Figure 5-1

5.2 MAINTENANCE TECHNIQUES

The KT11-B requires no periodic maintenance or calibration other than an occasional check of the M826 clock frequency. The clock pulses may be seen with an oscilloscope on pin A13A1 in the KT11-B logic panel when the processor is performing bus cycles. They are most easily seen while executing a "BR ." instruction. The period from rising edge of two consecutive pulses should be 190 to 200 nanoseconds.

In the event of suspected KT11-B failure, the following steps should be taken to isolate the cause and return the system to service. Refer to the KT11-B Block Schematic, drawing set #7605971, and the KA11 processor modifications for the KT11-B, drawing set #7605961, and also to the state diagrams and table of section 4.

a. Check all power supplies in the system to insure all devices are receiving proper power.

b. Check all Unibus cables and connections.

c. Make sure that the KT11-B can cycle in unpaged mode by depositing and examining into core memory from the console.

   If examine or deposit do not seem to work for any core bank, use the W130/W131 maintenance board to toggle through the states 0.0; 1.0; 3.0; 2.0; and then back to 0.0. If the KT11-B state does not begin at 0.0, check the distribution of INIT in the logic and it's effect on the X and Y shift registers; also verify that all bits in the PGC register are cleared (drawing D15). Make sure the address, data, control, and MSYN signals are getting to the memory, then verify that SSYN is returned to the processor. If failure occurs on only one memory bank, verify operation of that bank.

d. If unpaged operation seems to work, run the MAINDEC diagnostic programs for the processor and memory to verify their operation (see Appendix A for certain MAINDEC modifications).

e. Run the KT11-B diagnostic programs KT1, KT2, KT3, and KT6. These should be able to catch any basic faults (see Appendix A).

The following steps briefly describe methods of using the console and maintenance board to manually sequence the KT11-B in several modes of operation, allowing observation of internal signals.

f. Exec Simple:
   1) Set up a page map in page 1:

```
14ØØ/   16ØØØØ
14Ø2/   16ØØØ1
14Ø4/   16ØØØ2
  :       :
1776/   16Ø777
```

2) Deposit an exec map pointer of 177ØØ1 in the KTEM register.

3) Start paging by depositing a 2 in KTC.

4) Use Load Address - Examine and the maintenance board to toggle through the states for pages 1, 2, and 3, and the last page to examine KTM.

g. Exec Per Process

1) Set up Exec map pointer in KTEM: ØØØØØ1
2) Set up User map pointer in KTUM: ØØØØØ1
3) Set up user page map in page 1:

```
1ØØØ/       Ø
        16ØØØØ
        16ØØØ1
  :       :
1376/   16Ø777
14ØØ/   16ØØØØ    same as step f. above.
```

4) Start Paging.
5) Toggle through the states.

h. Shared User:

1) User map pointer: ØØØØØ5
2) User map at page 5:

```
54ØØ/   16ØØØ1
  :       1ØØ1
  :       2ØØ1
          3ØØ1
            :
        177ØØ1
```

3) Use the same exec map and pointer as in step g. above.
4) Move 3 into KTC to start user; do two examines and watch the states carefully - you should be in the start user sequence. User light should come on after the two examines.
5) Examine page zero; you should be in user simple mode.
6) Examine page one; you should be in shared user mode.

i. Start User:
   1) Use the same map and pointer as above.
   2) Move 3 to KTC.
   3) Do two examines, stepping through the states carefully at the end, you should be in user mode.
   4) Examine user page zero.
   5) Press START to initialize to unpaged mode. Change user map pointer (KTUM) to 177701. Repeat steps 1-4 for pages 1, 2, 3 and last page (ie. examine KTM).

In all of the above procedures, deposit meaningful information into the core locations you will be examining before starting the procedure.

SECTION 6

MODULE LIST

6.1 Modules

Table 6-1 is a list of modules used in the KT11-B.

DEC TYPE	FUNCTION	QUANTITY
M111	Inverter	1Ø
M112	2-input NOR	5
M116	4-input NOR	1Ø
M133	2-input NAND	24
M135	3-input NAND	6
M139	8-input NAND	1
M167	Comparator	1
M2Ø3	R/S Flip-Flop	1
M2Ø6	D Flip-Flop	1
M2Ø7	J-K Flip-Flop	1
M225	Scratchpad & Registers	1
M24Ø	R/S Flip-Flop	1
M244	Multiplexer Latch	6
M259	Associative Memory	4
M6Ø2	Pulse Amplifier	1
M611	Power Inverter	3
M627	NAND Amplifier	7
M721	Unibus Receiver-Driver	4
M783	Unibus Drivers	6
M784	Unibus Receivers	2
M826	Clock & Shift Register	1
M93Ø	Unibus Terminator	2
BCØ8-7	Control Cable	1
BC11A-8	Unibus Cable	2

Table 6-1 Module List