+---------------+
! d i g i t a l ! I N T E R O F F I C E M E M O R A N D U M
+---------------+

TO: List

DATE: 28-Nov-78

FROM: T. Hess

DEPT: L.C.E.G

LOC: MR1-2/E47

EXT: 6448

DISTRIBUTED: 1-Dec-78

FILE: KXFIO.RNO

SUBJ: KXF10 Exec Mode

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

Copyright (C) 1978 by Digital Equipment Corporation

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-10
DECCOM	DECsystem-20	TYPESET-11

CHAPTER 1

INTRODUCTION

This document describes the operation of the Exec Mode and I/O instructions on the KXF10. All I/O instructions are ordinary instructions with the same format as normal instructions (opcode, AC and effective address). The opcodes are in the range 700 thru 777.

Any operation code in the range 700 thru 777, AC number, field or bit not described in this document should be considered reserved to DEC. The microcode will attempt to generate an illegal instruction trap whenever a reserved action is attempted.

Opcodes 700 thru 737 are for exec mode only (I/O legal) instructions. Opcodes 740 thru 777 are for exec and user instructions.

Included in this document is a description of the implementation of all 4096 sections of virtual address space, the EPT/UPT layout, and process context variables. An address break facility will be covered in a later document.

OPCODE Assignment Map

	0	1	2	3	4	5	6	7
700	APR0	APR1	APR2	-	PMOVE	PMOVEM	-	-
710	TIOE	TION	RDIO	WRIO	BSIO	BCIO	-	-
720	TIOEB	TIONB	RDIOB	WRIOB	BSIOB	BCIOB	-	-
730	TIOEW	TIONW	RDIOW	WRIOW	BSIOW	BCIOW	-	-
740	-	-	-	-	-	-	-	-
750	-	-	-	-	-	-	-	-
760	-	-	-	-	-	-	-	-
770	-	-	-	-	-	-	-	-

AC Field Assignments

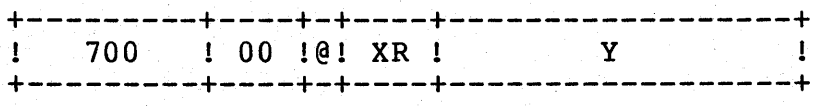
AC	700	701	702
00	APRID	-	RDSPB
01	-	RDUBR	RDCSB
02	-	CLRPT	RDPUR
03	-	WRUBR	RDCSTM
04	WRAPR	WREBR	RDTIM
05	RDAPR	RDEBR	RDINT
06	-	SWPIB	UPDTIM
07	-	SWPVB	-
10	-	SWPUB	WRSPB
11	-	SWPIA	WRCSB
12	-	SWPVA	WRPUR
13	-	SWPUA	WRCSTM
14	WRPI	-	WRTIM
15	RDPI	SWPIO	WRINT
16	-	SWPVO	-
17	-	SWPUO	-

CHAPTER 2

INTERNAL I/O INSTRUCTIONS

These instructions control the CPU. They transfer no data between the outside world and the CPU or memory.

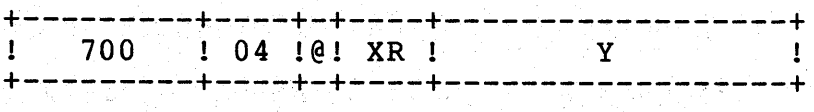
APRID



This instruction returns the microcode version number and the CPU serial number. The word is stored at E in the following format:

0-8	Reserved for microcode options.
9-17	Microcode version number
18-20	Hardware options (These are all zero at present)
21-35	Processor serial number

WRAPR



This immediate mode instruction decodes its effective address to control the processor. The effective address bits are used as follows:

19 — I/O reset. When this bit is set all internal devices are reset. In addition, an input/output init is generated.

20 — Enable conditions selected by bits 24 thru 31 to cause interrupts.

21 — Disable interrupts for conditions selected by bits 24 thru 31.

22 — Clear flags indicated by bits 24 thru 31.

23 — Set flags indicated by bits 24 thru 31.

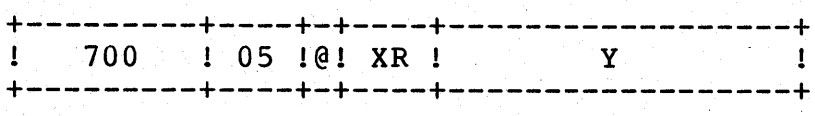
WARNING

The action of the processor is not defined when both bits 20 and 21 or 22 and 23 are set in the same instruction.

24-31 — To be described

33-35 — PIA

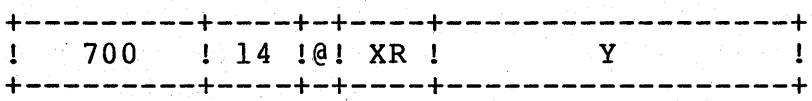
RDAPR



This instruction stores the APR status in the word addressed by E. The status is as follows:

08-13	Enables (to be described)
24-31	Flags (to be described)
32	Interrupt requested
33-35	PIA

WRPI



This immediate mode instruction decodes its effective address to control the priority interrupt system. The effective address bits are used as follows:

18-20 Cache fault insertion (to be defined).

22 Turn off program requests on selected levels (bits 39-35)

23 Clear PI system.

24 Initiate interrupts on selected levels (bits 29-35)

25 Turn on selected levels (bits 29-35)

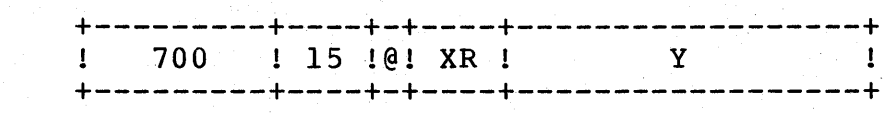
26 Turn off selected levels (bits 29-35)

WARNING

The action of the processor is not defined when both 25 and 26 or 22 and 24 are set in the same instruction.

29-35 Select levels for bits 22, 24, 25, and 26.

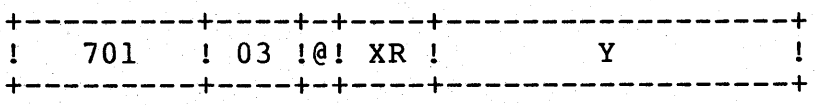
RDPI



This instruction store the PI status in the word addressed by E. The status is a follows:

11-17	Program requests on levels.
18-20	To be defined (set by WRPI).
21-27	Interrupt in progress on levels.
28	PI system on.
29-35	Levels on.

WRUBR



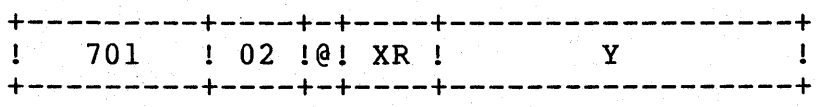
This instruction loads user process context with the words at E and E+1. The format of the first word (E) is:

0	Load AC block numbers
1	Load PCS
2	Load UBR
3	Do not update accounts
18-35	Physical page number of UPT.

The format of the second word (E+1) is:

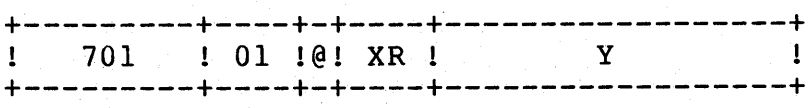
6-17	Previous Context Section
30-32	Current AC block
33-35	Previous AC block

CLRPT



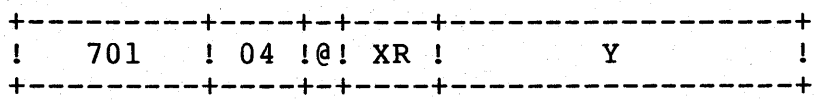
This instruction clears the hardware page table so that the next reference to the word at E will cause a refill cycle.

RDUBR



This instruction reads back the user process context and returns two words at E and E+1 in exactly the same format as used by WRUBR. In order to allow these words to be used directly in a WRUBR instruction, bits 0 thru 3 are set to 1 in the result.

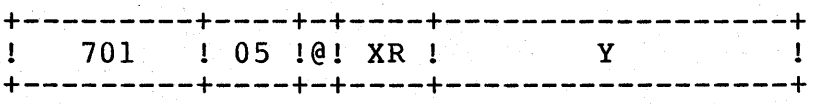
WREBR



This instruction loads the exec mode context from the word at E. The format of the word is:

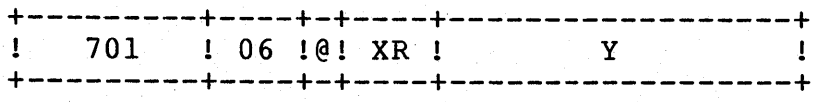
0-1	Cache strategy (Look and Load).
2	TOPS-20 style paging.
3	Trap and paging enable.
18-35	Physical page number of EPT.

RDEBR



This instruction returns the value given to WREBR and stores it in the word addressed by E.

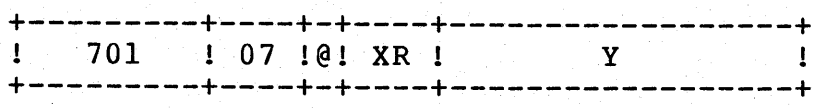
SWPIB



Sweep Cache, Invalidate Block (4 words)

Clear the valid and written state in all cache entries for the physical block specified by bits 9-33 in the location addressed by E. This instruction does not cause a cache sweep done interrupt. Bits 34-35 are ignored and assumed to be zero.

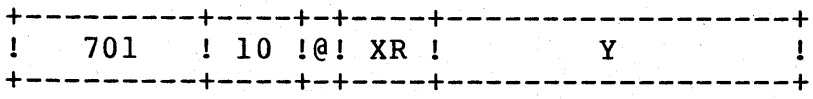
SWPVB



Sweep Cache, Validate Block (4 words)

Write into storage all cache entries that are in the written state for the physical block specified by bits 9-33 in the location addressed by E. Clear the written state associated with those words sent to storage, but do not change the validity of any entries. This instruction does not cause a cache sweep done interrupt. Bits 34-35 are ignored and are assumed to be zero.

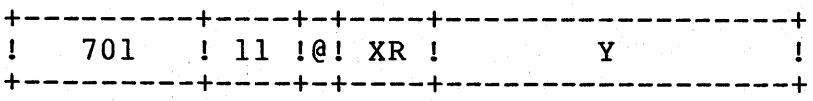
SWPUB



Sweep Cache, Unload Block (4 words)

Write into storage all cache entries that are in the written state for the physical block specified by bits 9-33 in the location addressed by E. Invalidate all entries for the specified block (i.e. clear valid and written state). This instruction does not cause a cache sweep done interrupt. Bits 34-35 are ignored and are assumed to be zero.

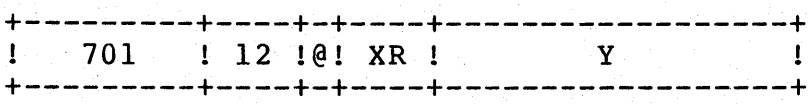
SWPIA



Sweep Cache, Invalidate All Pages

Set Sweep Busy and clear the valid and written state in all cache entries. At the completion of the sweep, clear Sweep Busy and set Sweep Done, requesting an interrupt on the level assigned to the processor.

SWPVA

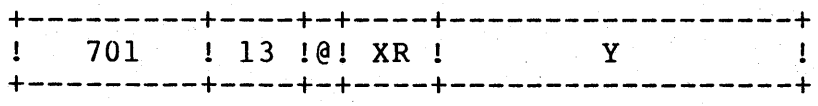


701	12	@	XR	Y

Sweep Cache, Validate All Pages

Set Sweep Busy and write into storage all cache entries that are in the written state. Clear the written state associated with those words sent to storage, but do not change the validity of any entries. At the completion of the sweep, clear Sweep Busy and set Sweep Done, requesting an interrupt on the level assigned to the processor.

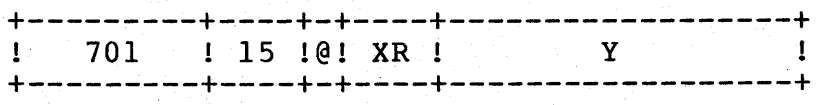
SWPUA



Sweep Cache, Unload All Pages

Set Sweep Busy and write into storage all cache entries that are in the written state. Invalidate all entries (i.e. clear valid and written state). At the completion of the sweep, clear Sweep Busy and set Sweep Done, requesting an interrupt on the level assigned to the processor.

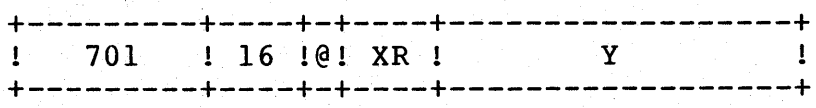
SWPIO



Sweep Cache, Invalidate One Page

Set Sweep Busy and clear the valid and written state in all cache entries for the physical page specified by bits 18-35 of E. At the completion of the sweep, clear Sweep Busy and set Sweep Done, requesting an interrupt on the level assigned to the processor.

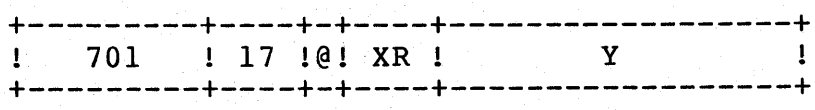
SWPVO



Sweep Cache, Validate One Page

Set Sweep Busy and write into storage all cache entries that are in the written state for the physical page specified by bits 18-35 of E. Clear the written state associated with those words sent to storage, but do not change the validity of any entries. At the completion of the sweep, clear Sweep Busy and set Sweep Done, requesting an interrupt on the level assigned to the processor.

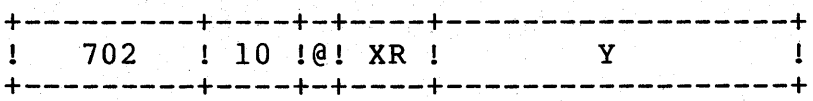
SWPUO



Sweep Cache, Unload One Page

Set Sweep Busy and write into storage all cache entries that are in the written state for the physical page specified by bits 18-35 of E. Invalidate all entries for the specified block (i.e. clear valid and written state). At the completion of the sweep, clear Sweep Busy and set Sweep Done, requesting an interrupt on the level assigned to the processor.

WRSPB



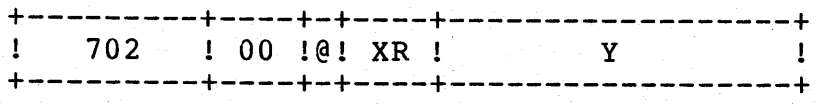
Write SPT Base Register

Load the word at E into the SPT base register.

BASE REGISTER FORMAT

All base registers are loaded with a physical word address. All high order bits must be zero. The address need not be on a page boundary and may be anyplace in physical memory. There is no range check on SPT or CST offsets. The monitor is assumed to always put correct data into page tables.

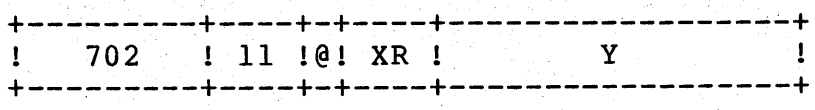
RDSPB



Read SPT Base Register

Store the SPT base register at E.

WRCSB



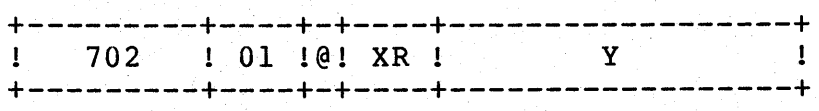
Write Core Status Table Base Register

Load the CST base register with the word at E.

BASE REGISTER FORMAT

All base registers are loaded with a physical word address. All high order bits must be zero. The address need not be on a page boundary and may be anyplace in physical memory. There is no range check on SPT or CST offsets. The monitor is assumed to always put correct data into page tables.

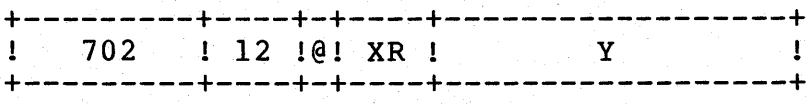
RDCSB



Read Core Status Table Base Register

Store the CST base register at E.

WRPUR



Write Process Use Register

Load the process use register from E. The process use register contains the AGER (age register) in the left few bits. The bits containing the AGER are cleared by anding the CST entry with the CST mask, then the entire PUR is ored with the CST entry.

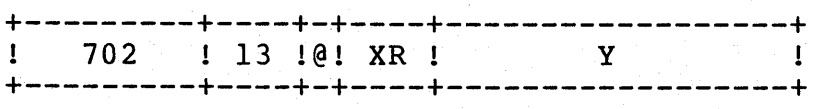
RDPUR

702	02	@	XR	Y	

Read Process Use Register

Store the PUR at E.

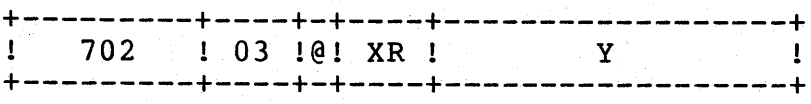
WRCSTM



Write CST Mask Register

Load the CST mask register from E. The CST mask register should contain a 0 for every bit in the AGER and a 1 in all other bit positions.

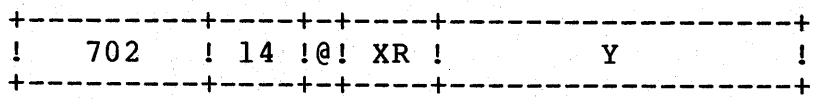
RDCSTM



Read CST Mask Register

Store the CST mask register at E.

WRTIM



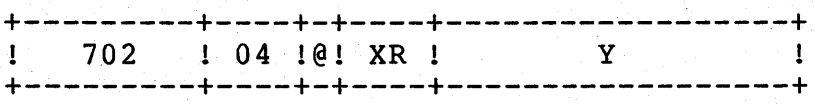
Write Time Base

This immediate instruction decodes its effective address to control various system time and accounting meters. For a complete explanation of the time and accounting meters see "Dolphin Clock Proposal" by Bill Bruckert, 24-Mar-78. The effective address bits are used as follows:

18	Set up accounts (bits 21-23).
21	Enable PI accounting.
22	Enable Kernel mode accounting.
23	Turn on accounting.
24	Turn off time base.
25	Turn on time base.
26	Clear time base.
33-35	PIA for interval timer.

Briefly, the time base consists of two counters that count at a 1 ms. and 10 us. rate. The millisecond counter consists of 1 word in the EPT at location 507. The 10 microsecond counter consists of 2 words in the EPT at locations 510 and 511. These two counters are driven from the same source and are updated in parallel by the microcode.

RDTIM

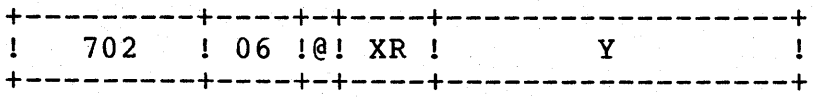


Read Time Base

Read the status of the accounting meters and time base, and the interrupt level assigned to the interval timer into the word addressed by E. The status is as follows:

21	PI accounting enabled.
22	Kernel mode accounting enabled.
23	Accounting on.
25	Time base on.
33-35	PIA for interval timer.

UPDTIM

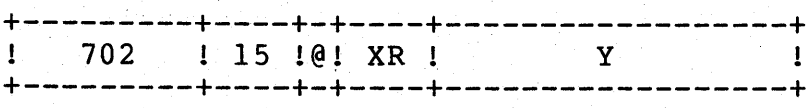


702	06	@	XR	Y

Update Time Base

Update the time base count from the hardware counter, and transfer the result from locations 507, 510 and 511 in the EPT to locations E, E+1, and E+2. Updating clears the hardware counter.

WRINT

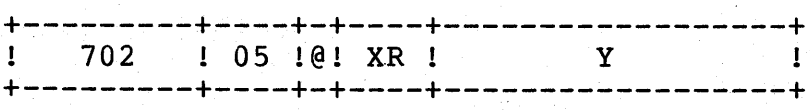


Write Interval Timer

This immediate mode instruction decodes its effective address to setup the interval timer. The effective address bits are used as follows:

18	Clear interval timer.
21	Turn interval timer on.
22	Clear interval flags.
24-35	Interval period.

RDINT



Read The Interval Register

Read the status of the interval timer into the word addressed by E. The status is as follows:

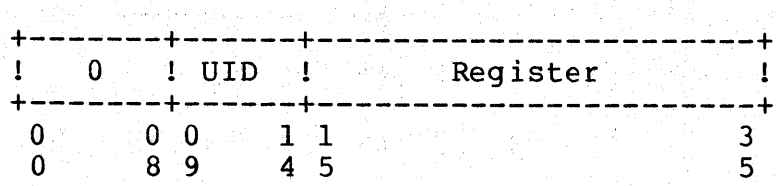
6-17	Interval count (current contents of the counter).
21	Interval timer on.
22	Interval timer done (causes interrupt).
23	Overflow (implies bit 22).
24-35	Interval period.

CHAPTER 3

EXTERNAL I/O

3.1 INTRODUCTION

The external I/O instructions on the KXF10 allow a program to read, write, modify and test registers in external devices. They do this by specifying a 27-bit address which is used as an I/O address and has the following format:



where UID is the Unit ID and Register is a register address within the unit.

The following address assignments for Unit ID currently exist (Ref. "Dolphin Bus Specification, Rev. 5"):

	CPU	Option Slot								Console	Memory
		01	02	03	04	05	06	07	10		
System 0	00	01	02	03	04	05	06	07	10	11	12
System 1	20	21	22	23	24	25	26	27	30	31	32
System 2	40	41	42	43	44	45	46	47	50	51	52
System 3	60	61	62	63	64	65	66	67	70	71	72
Expansion Bus 1		13	14	15	16	17	33	34	35	36	37
Expansion Bus 2		53	54	55	56	57	73	74	75	76	77

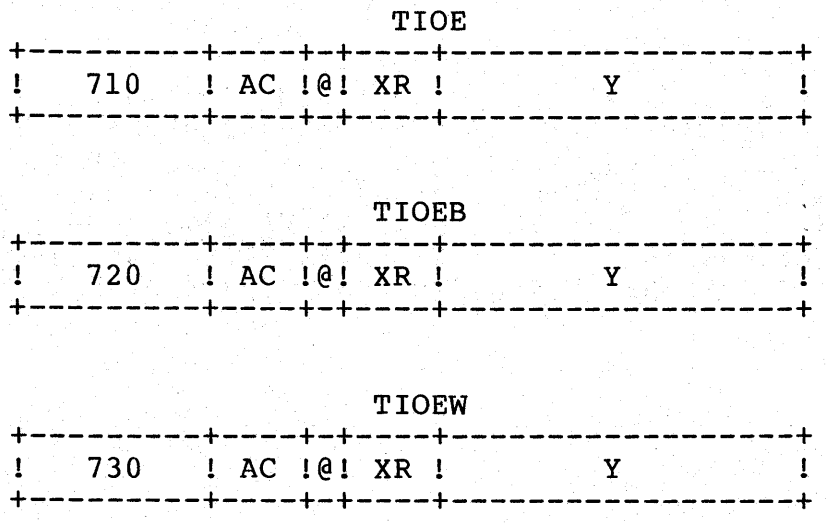
3.1.1 Specifying I/O Addresses

All external I/O instructions (i.e. those that specify an I/O address) will fetch the contents of their effective address, E, and use bits 9-35 as the I/O address.

3.1.2 UNIBUS Word And Byte Mode

The external I/O instructions come in three types, normal, UNIBUS word, and byte. The normal type transfers 36 bits of data and uses the full contents of AC. The UNIBUS word and byte types are for use only with UNIBUS devices. The UNIBUS word instructions transfer only 16 bits of data and only use AC bits 20-35. The byte instructions transfer only 8 bits of data and only use AC bits 28-35.

TIOE, TIOEW and TIOEB



Test Input/Output, no modification, skip equal

The effective address of this instruction is used as a I/O address. For TIOE, a word is fetched from the I/O register and and'ed with the contents of AC. For TIOEB, an 8-bit byte is fetched from the I/O register and and'ed with the contents of AC bits 28 thru 35. For TIOEW, a 16-bit byte is fetched from the I/O register and and'ed with the contents of AC bits 20-35. The instruction skips if the result of the and is zero. The contents of the AC are not modified.

TION, TIONW and TIONB

TION

711	AC	@	XR	Y

TIONB

721	AC	@	XR	Y

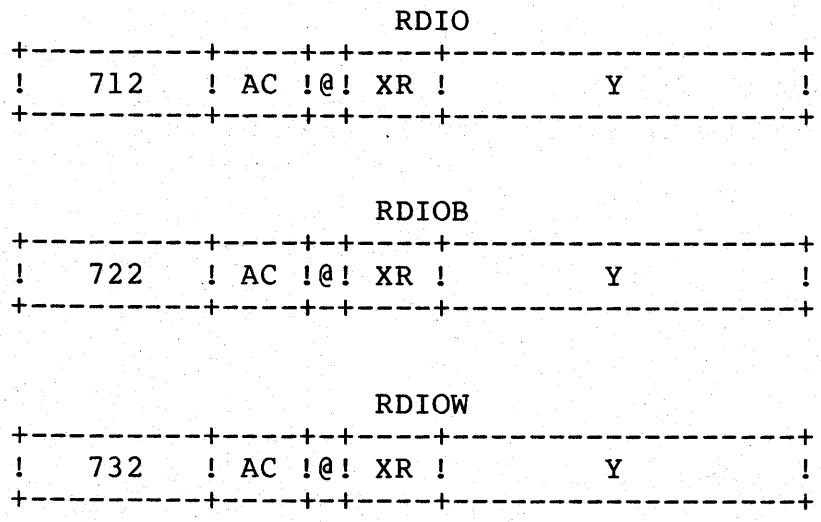
TIONW

731	AC	@	XR	Y

Test Input/Output, no modification, skip not equal

This instruction is the same as TIOE except the sense of the skip is inverted.

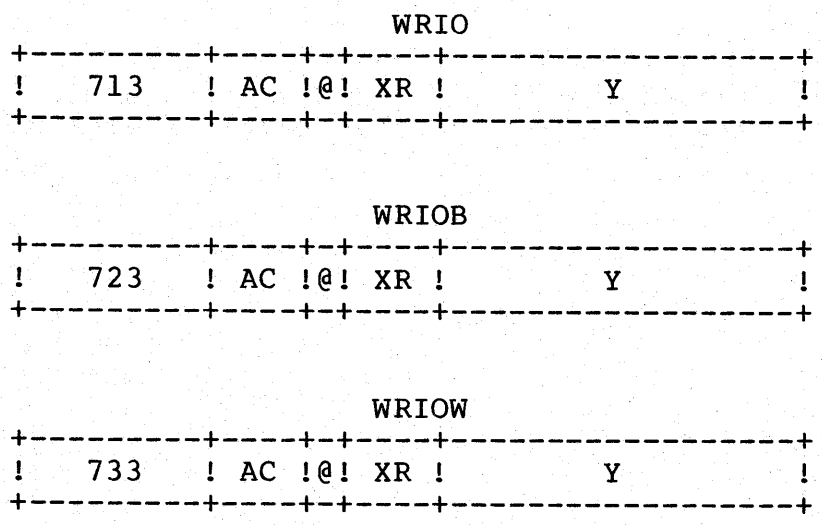
RDIO, RDIOW and RDIOB



Read Input/Output

This instruction uses the effective address to access an I/O register. One word, UNIBUS word or byte is fetched from the I/O register and stored right justified in AC.

WRIO, WRIOW and WRIOB

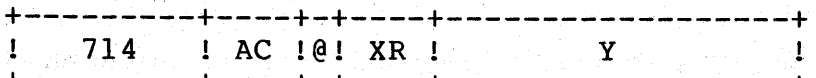


Write Input/Output

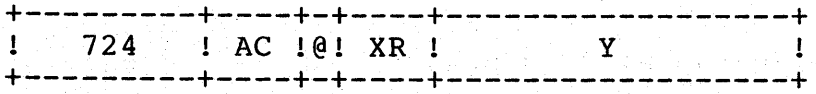
The effective address is used as an I/O address, and one word, UNIBUS word or byte is taken from AC and sent to the I/O register.

BSIO, BSIOW and BSIOB

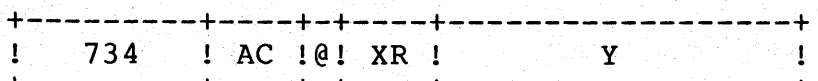
BSIO



BSIOB



BSIOW



Bit Set To Input/Output

The effective address is used to select an I/O address. One word, UNIBUS word or byte is fetched from the I/O register, or'ed with the contents of AC, and written back to the I/O register. This operation will use a read-modify-write cycle on the UNIBUS.

The contents of the ac are not changed by this operation.

The net effect of this operation is to set selected bits in I/O device registers.

BCIO, BCIOW and BCIOB

BCIO

715	AC	@	XR	Y

BCIOB

725	AC	@	XR	Y

BCIOW

735	AC	@	XR	Y

Bit Clear Input/Output

The effective address is used to select an I/O register. One word, UNIBUS word or byte is fetched from the I/O register, and'ed with the complement of the contents of AC, and written back to the I/O register. This operation will use a read-modify-write cycle on the UNIBUS.

The contents of the AC are not changed by this operation.

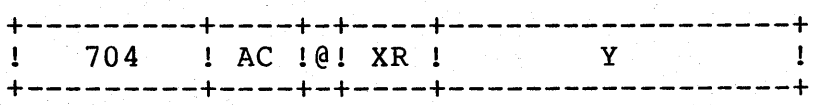
The net effect of this operation is to clear selected bits in I/O device registers.

CHAPTER 4

SPECIAL INSTRUCTIONS

The following instructions have been added to save time and space in the monitor.

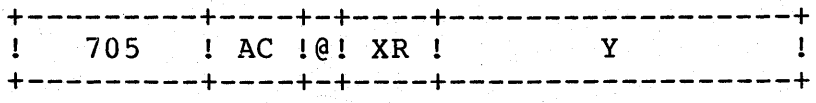
PMOVE



Physical Move from Memory

Move the physical memory location addressed by E into the AC. The effective address is calculated the same way as in the I/O instructions that reference external devices.

PMOVEM



Physical Move to Memory

Move AC into the physical memory location addressed by E. The effective address is calculated the same way as in the I/O instructions that reference external devices.

CHAPTER 5

VIRTUAL ADDRESSING

This section has been extracted from a memo titled "4096 Sections" by Len Bosack dated 12-Oct-78.

5.1 INTRODUCTION

The KL processor implemented 32 sections of virtual address and has a pager data structure that can accommodate at most 32 sections. The KXF10 will implement all 4096 sections of virtual address space forcing a change of the pager data structure to accommodate 4096 section pointers.

The following paging description is only valid when TOPS-20 style paging is enabled (see WREBR).

5.1.1 Pager Data Structure

An address would be converted to a physical page number (PPN) as follows:

VMA<6:8> is used to index into a "Super" Section Table (SST) in the EPT/UPT. One of 3 pointer types can occur here: No Access, Small User, or Shared (see below). The Shared Pointer (SPT) index yields the PPN of a Section Table page (ST) and address translation proceeds as on the KL10 after the section pointer fetch.

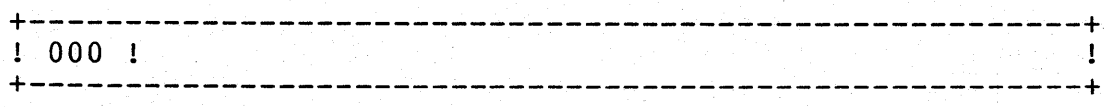
In an effort not to burden "small" (a few contiguous sections) processes with extra pager refill memory references and extra overhead storage, a Small User feature is provided. An address would then be converted as follows:

VMA<6:8> is used as index into the Super Section Table as before. A Small User pointer is found and points into the UPT/EPT (see below). VMA<9:17> must be in the range of the "Offset" and "Count" in the Small User pointer. The section pointer is then fetched from the EPT/UPT address plus the quantity derived from the Offset and Count specified in the pointer. Address translation proceeds as on the KL10 after the section pointer fetch.

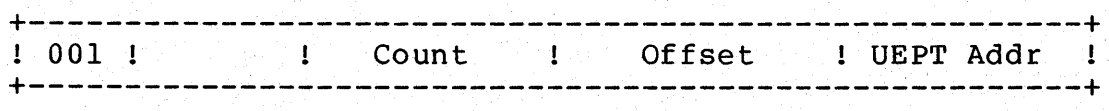
5.1.2 Super Section Pointers

The following pointer types will only be valid in the Super Section Table in the EPT/UPT locations 520-527. All other pointers will be interpreted as in the KL10.

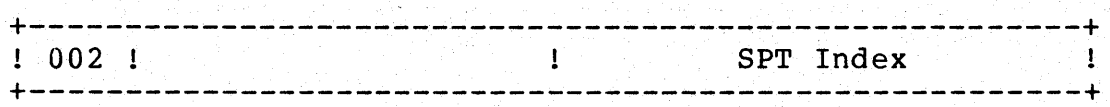
No Access:



Small User:



Shared:



CHAPTER 6

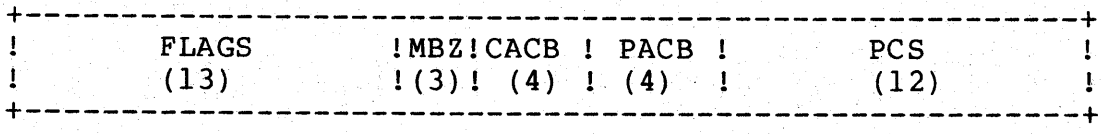
PROCESS CONTEXT VARIABLES

6.1 INTRODUCTION

This section contains a proposal for handling process context variables using an expanded PC flags word. The source for this proposal was a memo by Dan Murphy dated 15-Feb-77. Further discussion on a process context switch instruction is still needed. I hope that this section will stimulate some suggestions along these lines.

6.1.1 New PC Double Word

The format of the flags word shall be:



1. In kernel mode, or when stored on a page fail or MUUO, all of the above fields will be stored as defined. In kernel mode, an XJRSTF will restore all fields.

2. In user mode, PCS, PACB, and CACB will always be stored as 0. An XJRSTF in user mode will treat these fields as it does the user mode and user I/O flag now (i.e. ignore them).

6.1.2 Context Changing

Returning to a previous context may be done with an XJRSTF with restores the context variables stored in the previously saved PC word.

Entering a new context will be done as follows: All of the

"previous context" variables will be set to their corresponding values in the "current context", and the remainder will be set to pre-defined values taken from the new PC flag word. The following operations are defined as entering a new context:

1. Monitor call (MUUO).

2. Page fail trap.

3. Priority interrupt initiation.

Each of these operations will store a PC double-word containing the "current context" variables and then load a new PC double-word to set new values for those variables not set automatically. See next section for a description of the changes necessary to the EPT/UPT to implement this proposal.

CHAPTER 7

EPT / UPT DESCRIPTIONS

The following EPT/UPT layouts are proposed for the KXF10. Not included at this time is a description of a new mechanism for handling trap instructions under extended addressing.

NOTE

All areas that differ from the KL10 are marked with an asterisk (*).

Exec Page Table

*	0-77	Interrupt vectors for external devices.
*	100-137	Console commucations region.
*	140	Interrupt vector for APR (see WRAPR).
*	141	Interrupt vector for interval timer (see WRINT).
*	142-177	Reserved.
	200-377	TOPS-10 paging (Kernel 400-777).
	400-420	Reserved.
	421	Kernel Arithmetic Overflow trap instruction.
	422	Kernel Push-down List Overflow trap instruction.
	423	Kernel Trap Type 3 trap instruction.
	424-506	Reserved.
*	507	Time Base 1 (Millisecond timer).
	510-511	Time Base 2 (10 usec timer).
	512-517	Reserved.
*	520-527	"Super" section table (TOPS-20 paging).
	530-577	Reserved.
	600-757	TOPS-10 paging (Kernel 0-337).
	760-777	Reserved.

User Page Table

	0-377	TOPS-10 paging (User 0-777).
	400-417	TOPS-10 paging (Kernel 340-377).
	420	Address of LUUO block (TOPS-20).
	421	User Arithmetic Overflow trap instruction.
	422	User Push-down List Overflow trap instruction.
	423	User Trap Type 3 trap instruction.
*	424	MUUO old PC flags.
*	425	MUUO old PC.
*	426	MUUO Opcode and AC.
*	427	MUUO Effective Address.
*	430-431	Kernel no-trap MUUO new PC double word.
*	432-433	Kernel trap MUUO new PC double word.
*	430-431	User no-trap MUUO new PC double word.
*	432-433	User trap MUUO new PC double word.
*	440	Page fail code.
*	441	Page identifier (TOPS-20 paging).
*	442	Page fail virtual address.
*	443-444	Page fail old PC double word.
*	445-446	Page fail new PC double word.
*	447-503	Reserved.
*	504-505	User runtime (10 usec clock).
*	506-507	User accounting meter ("chargons").
	510-517	Reserved.
*	520-527	"Super" section table (TOPS-20 paging).
	530-777	Reserved.