KI-10 IMPLEMENTATION

AND

POSSIBILITIES FOR A THIRD RING

Allen B. Goodrich

February 6, 1974

Technical Memo

TM.74-5

I. Introduction

The KI-10 is described as explained and implied by the System Reference section of the DEC System 10 Assembly Language Handbook. As yet, however, the TENEX implementation on this processor is still a mystery.

II. KI-10 Processor

Job Executing Environment

An executing job on a KI-10 uses two virtual address spaces: the User Virtual Address Space, and the Executive Virtual Address Space. The code in the Executive Virtual Address Space is called the Monitor in the DEC literature.

The User Virtual Address Space is defined in a page of data called the User Process Table. The User Process Table has similarities to the BCC 500 Context Block. Notably, each page of the User Virtual Address Space has a half word entry in the User Process Table, the software map.

The Executive Address Space is similarly defined by the Executive Process Table. However, there are two notable exceptions.

1) The lower 112K (224 pages) of the Virtual Executive Address Space is not paged. All references to these pages are absolute. For this reason the number of entries in the Executive Process Table for mapping is smaller than in the User Process Table.

2) 32 pages of the Executive Virtual Address Space are defined by entries in the User Process Table. Therefore, when a page fault happens to one of these 32 pages the hardware gets its map entry from the User Process Table, not the Executive Process Table providing a facility by which each process can have a unique area in the otherwise sharable Executive Address Space.

Associative Map

The KI-10 map is a real associative memory. The memory is 32 words long. Each word has two parts, the virtual and the physical. The virtual part has 9 bits for the page number and two control bits. One bit defines whether the entry is valid. The other defines User or Executive Virtual Address Space. The physical part has a 13 bit Physical Page Number and some control bits. The bit called P (Public) is used in User Mode to designate 'concealed' pages. In Executive Mode the Public bit defines the distinction between Kernal and Supervisor, which will be discussed later. The bit called W (Write) defines pages, in both modes, we used to call Read-Only. The bit called S (Software) is not used by the hardware. I believe there is also a bit X (eXtra) which DEC has room for in its Process Tables but warns not to use because DEC will use it later.

9 Bit Virtual Page Number	User Exec	Valid Empty	13 Bit Physical Page Number	P	W	S

Kernal/Supervisor

As stated above the bit called P in the map defines the distinction between Kernal and Supervisor Modes in the KI-10. The 112K of Executive Address Space which does not use the map is defined as Kernal only. Pages defined as Supervisor may not reference the unmapped Kernal space. Any mapped Kernal Page (a kernal page whose virtual address is larger than 112K) is, however, Read Only to the Supervisor. This is true in any case as it is a function of the implementation. Also, only Kernal Mode may execute 'privileged' operations. The supervisor has no more power than the user in that regard.

The mechanisms used to effect all of the above are unknown. The above specification might be a good start to a provable division between two rings but, as will be pointed out, not enough is known yet.

A question also exists about the Fast Registers for the Executive Address Space. The KI-10 intercepts memory references to Virtual Address 0-17 octal in all cases and uses a set of internal fast registers. The question is whether the Kernal and Supervisor must share the same registers. Multiple registers exist so that user and executive modes use different sets always.

User/Exec

A facility exists allowing a program running in the Executive Virtual Address Space to reference, and modify, words in the User Virtual Address Space. The feature is called the Executive XCT in the literature, an expansion of the User 'eXeCuTe' instruction. Depending upon many esoteric conditions an instruction executed with an XCT performed in the Executive operates normally, or the reference to the effective address is performed in the User Virtual Address Space. Provisions are made so the references can be made to fast register contents, even though the contents may not be in fast registers:

1) User Shadow Area - Real Core 0 - 17 octal -- the user's registers are stored here when the user is not current. It is not clear, however, whether this feature is hardware implemented or a feature of the DEC operating system.
2) Executive Stack Pointer -- a stack of executive copies. Making possible MUUOs called by the User able to call more MUUO's, recursive and nesting of MUUO's.

Provision is also made so that a User Public program may not cause the monitor to reference a User Concealed Page. Only MUUO's called from a concealed page may reference concealed pages.

TENEX

It is not clear how many of the above hardware features the implemented TENEX uses, i.e., it is rumored that supervisor mode in the Executive Virtual Address Space is not used at all. Also, since if we do decide to make hardware modifications, it may become significant to know how much of the resources, which DEC has defined as 'available to software', TENEX uses. For instance, if the Executive Process Table is full, then a decision to add enough entries to map the now unmapped address space would be impossible. Also, the map bit labelled S may have been used, obviously affecting any decisions concerning it.

III. Three Rings

The obvious thing to try is to prove the Kernal is secure from the Supervisor and therefore the two could be considered independent rings. Four things would seem to make the proof impossible, assuming no hardware changes.

1) Part of the Kernal is unmapped and therefore does not have the access controls which would otherwise be imposed by a core memory manager. Whether the memory manager be a part of the kernal or in a separate processor.
2) The rest of the kernal, the mapped part, is read only to the supervisor, always.
3) Apparently the Supervisor when making an MUUO to the kernal may give the kernal a parameter inside the kernal. However, this may be taken care of by the Disable Bypass in the PC word of the MUUO (too specific to explain, but it is what implements the protection to concealed pages in the User MUUO's mentioned earlier).
4) It is possible the two modes, kernal and supervisor, use the same fast register, however, it is equally likely that the fast registers go onto a stack every MUUO. This is not clear in the case of kernal/supervisor from the literature.*

Summary

Listing possible hardware changes now would be premature. However, the four points listed in the previous section are what need to be rectified or shown that they are already taken care of. They show that the kernal is

*Also it is not clear at this writing whether kernal UUO's are or can be defined different than Supervisor UUO's. Although I suspect the implied recursive nature of the MUUO's also implies that each MUUO may be defined in either ring with no ambiguity or security violation--more after talking to Rainer.

invisible 2), 4); that the kernal is unmodifiable 3), 4); and that it is controlled, which hopefully implies it is consistent 1).

IV. Questions

Two classes:

Hardware, Software, Existing System

1) Can virtual pages of supervisor (or user for that matter) be defined as absolute addresses below 112K (i.e., in the predefined kernal space)?

2) Does the kernal share fast registers with the supervisor? If not, is the Executive Shadow area used or a stack somewhere?

3) Exactly how does MUUO work? Expand on description on page 92 of DEC system 10 Assembly Language Handbook toward clearing up the confusion of the footnote in the section 'three rings'.

Theory, Philosophy, New System

4) Will kernal and supervisor code fit into 144K? Thus eliminating the need to increase the storage required, in the Executive Process Table, for map entries.

5) How small a part of the kernal must be fixed in core (not necessarily unmapped but immovable) for such cases as startup?

6) What exactly is the communication required between the Supervisor and the Kernal? I.e., Does the Supervisor need read access to Kernal pages above virtual 112K?

I also have a set of questions for Rainer, which I will not bother to write down. Answers to those and any more answered by any DEC hardware documentation if we get it will be reflected in the next memo on this subject.