PDP-K TECHNICAL MEMORANDUM # 6

TITLE: Why an 18-Bit PDP-K?

AUTHOR(s): Ad van de Goor

INDEX KEYS: Computer Family
Instruction Set
Addressing Capabilities
Data Types

DISTRIBUTION KEYS:

REVISION: None

OBSOLETE: None

DATE: March 13, 1970

0.0 Abstract

A family of program - and peripheral compatible computers is proposed. The PDP-K, as described in Technical Memorandum #4, is considered the only choice. The PDP-11 is not considered because of its shortcomings as described in Technical Memorandums 2 and 3.

The PDP-K is designed so that implementation as an 18- or a 36-bit machine can be invisible to the programmer. Its instruction set includes very powerful instructions and the number of instructions is four times that of the PDP-11.

1.0 Introduction

The success of the IBM 360 family of computers has made it obvious that a series of program compatible computers has advantages over several independent non-program compatible computers. It is this concept that underlies the architecture proposed here.

Several things are required for a family of computers in order to be successful. First, the programs must be upward compatible. Programs written for small versions of the family must run efficiently on larger versions. Second, each member of the family must be capable of efficiently operating in its intended market. That is, it must offer the computing power required at an attractive price. Third, important system programs should not have to be re-written (for efficiency reasons) for each member of the family.

The decision to market a family of computers will put "all the eggs" into ONE basket. DEC will be stuck with the decisions and their consequences for probably two or three generations of computers. It is a commitment to a hardware and software effort that requires big resources. The motivation for creating such a family seems to be twofold. First, for internal efficiency through sharing hardware, software, and peripheral cost among members of the family, and certain economies in sales and service support. Secondly, for providing a captive market of customers who outgrow their present computer by making it possible for them to run their old programs on a more powerful processor.

Clearly, this decision should not happen by default, but should be undertaken with the full awareness of its importance to DEC.

2.0 Market for the PDP-K

The PDP-K is intended to fill the gap between the PDP-15 and the PDP-10. Typical machines the PDP-K has to compete with are the XDS Sigma 5 and the SEL Systems 86. The latter machines are well designed, have built-in floating point arithmetic, and rather elaborate instruction sets.

3.0 Why not extend current PDP-11?

The KA-11 processor currently "exists," and is capable of a certain amount of expansion. However, the history of the KA-11 development indicates that certain compromises were made in favor of a low production cost and a short development time.

PDP-K Technical Memoranda #2 and #3 discuss ways to "get around" the address- and op code space problems. All solutions, however, are cludgey and an obvious add-on and/or inconsistent with the current PDP-11 instruction set. As competition for Sigma 5 class machines, these solutions will be barely acceptably or not acceptable at all.

Below the major aspects of a computer are discussed with reference to an extended PDP-11, the PDP-K and "the competition." Each of the points discussed might, in itself, only have a small influence on the overall performance. Together, however, they represent a reasonable performance measure and an extremely powerful sales tool or handicap.

3.1 Addressing

3.1.1 Maximum Addressable Amount of Core Memory

The PDP-K can address up to 128K[1] 36-bit words compared with the Sigma 5's, 128K of 32-bit words. The PDP-11, however, can only address up to 16K of 32-bit words. This makes the extended PDP-11 unattractive. PDP-K Technical Memorandum #3 discusses ways to extend the PDP-11 addressing capabilities, no solution looked very promising, however.

[1] K= 1024

3.1.2 Indexing

3.1.2.1 Level of Indexing

The PDP-11 has only 1 level of indexing. Many competitors allow for multiple levels. The PDP-K allows triple indexing with the SMOV[1] and MMOV[1] instructions.

3.1.2.2 Pre/Post Indexing

The PDP-11 has only one method of indexing: pre-indexing. Unfortunately, this is not the most popular method. The PDP-K allows for pre, post, or pre and post indexing combined when the SMOV and MMOV instructions are used.

3.1.3 Levels of Indirection

The PDP-K allows for 1 more level of indirection than the PDP-11 when the SMOV and MMOV instructions are used.

3.2 Instruction Set

First the data types on which instructions operate will be examined; after that the classes of instructions are discussed.

3.2.1 Data Types

3.2.1.1 Bits

The PDP-11, unlike the PDP-K and the Systems 86, does not have bit addressing or bit operations. In any system, bit diddling is important, however.

3.2.1.2 Bytes

The PDP-11 byte handling is fine. Three byte instructions, however, are very marginal: BISB, BICB, and BITB. Most of the time, they are used to set, clear, or test a single bit requiring a 16-bit mask.

[1] See PDP-K Technical Memorandum # 4.

3.2.1.2 (cont.) The suggested bit diddling instructions of the PDP-K are, therefore, a considerable improvement.

The PDP-K has PDP-10 type byte handling which, although slower, allows for the handling of arbitrary size bytes. This minimizes core memory useage and does not cost memory address space (like used for byte addressing).

3.2.1.3 Words

The PDP-11 and PDP-K both have words which are considered their basic data type. Most competitive machines have word sizes twice that of the PDP-11 or PDP-K.

3.2.1.4 Double Length and Bigger Words

As PDP-K Technical Memorandum #2 shows, these data types can only be implemented in a way which is non-consistent with the other data types of the PDP-11. The PDP-K incorporates these data types in a coherent manner

3.2.2 Classes of Instructions

3.2.2.1 Boolean Instructions

These are defined as logical operators on a single bit to allow for efficient bit diddling. The PDP-11 has no such instructions, the PDP-K has a rather complete set. Some competitive machines have instructions of this class.

3.2.2.2 Logical Instructions

These are defined as Boolean operators operating on more than 1 bit (typically a word) in parallel. The PDP-11 has only 2 instructions in this class; the PDP-K has 6 which account for the majority of uses. All competitive machines have more than 2 logical instructions.

3.2.2.3 Integer Arithmetic

The PDP-11 has a bare minimum of this class of instructions. Missing instructions like IMUL, etc., could be implemented, but only in a awkward way. Like most competitors, the PDP-K has a more complete set.

3.2.2.4 Floating Point Arithmetic

The PDP-K has a set of single and double precision floating point instructions implemented in a way consistent with other instructions. The PDP-11's floating point instructions have to be implemented in a non-consistent way.

3.2.2.5 Test Instructions

The PDP-K, unlike the PDP-11, has test instructions on all data types.

3.2.2.6 Control Instructions

In addition to a conditional branch, the PDP-K has also a conditional jump. The PDP-K has, also, more conditions.

3.2.2.7 Arithmetic Test Instructions

These are the class of AOS, SOS, AOJ, etc. instructions in the PDP-10. They have a high frequency of use. The PDP-K has a complete set; the PDP-11 has none.

3.3 Other Instructions

There is a whole group of instructions (like shift, rotate, repeat, etc.) which the PDP-K does have and are implemented in a general way. Some of these instructions are extremely powerful and are unique in the sense that "the competition" does not have them.

4.0 Conclusion

From the discussion before, it is quite clear that the PDP-11 cannot meet the requirements of the market place. The PDP-K is a superior machine and, as hopefully coding examples will show, will outperform "the competition."

The PDP-K is designed in such a way that it can be implemented, in a program compatible way, as an 18- or 36- bit machine.

The PDP-11 is a nice little machine, It does not take a genius, however, to see that an 18-bit version, as proposed, is an extremely useful medium scale computer. If DEC fails to recognize this, other companies will. So, while we are ahead, let us stay ahead!