+---------------------------+
! ! ! ! ! ! ! ! interoffice
! d ! i ! g ! i ! t ! a ! l !
! ! ! ! ! ! ! ! memorandum
+---------------------------+

To: Ulf Fagerquist
Vern Poulter

Date: 07 Apr 83
From: Mike Uhler
Dept: L.S.E.G.
DTN: (8-)231-6448
Loc/Mail stop: MR01-2/E85
Net mail: UHLER at 10

Subject: PDP-10 Product Options

1.0 Introduction

During the replanning stages of the Jupiter II project, questions have been raised concerning the decision to build a CPU whose nominal performance is four times a KL10. Why, it has been asked, should we not build something whose performance is less than 4x a KL10 but which we could ship sooner? This memo analyzes the possible alternatives and presents arguments that support our decision.

2.0 What are the options?

In looking at the available options, it seems clear that the number is limited. We must either do some sort of KL10 conversion or build a variant of the Jupiter architectural design. To do anything else would require a major development effort with a corresponding increase in schedule.

We have invested over three years effort working the bugs out of the architecture with the result being the Jupiter architectural design. This design includes the following significant enhancements of previous designs:

- Full virtual address space implementation. The Jupiter design includes support for the full 30-bit virtual address space as defined by the architecture. The KL10 implements a 23-bit subset of the virtual address space.

- Correct implementation of the extended addressing architecture. The Jupiter design correctly implements the full architecture.

- Enhancements to the hardware monitor interface. The Jupiter design includes enhancements to supply more information to the monitor on interrupts, page faults, and MUUOs. This information reduces the number of times that the monitor has to "guess" what happened and increases the efficiency and performance of the interface.

- Enhancements to improve timesharing efficiency. The Jupiter design includes enhancements to decrease the cost of context switching and other multi-programming effects.

- Increased functionality. The Jupiter design includes increased functionality in areas which are weak in previous implementations of the architecture (e.g., address break).

- Designed-in interface to corporate buses. The Jupiter design includes integral interfaces to the CI and NI buses. These interfaces were part of the original architectural design and do not exhibit the problems of "added on later" interfaces.

The Jupiter architectural design takes advantage of an eight year learning curve with the KL10 and KS10 designs.

3.0 A possible KL10 conversion

When considering producing a PDP-10 that is a conversion of the existing KL10 design, one must consider several things.

The existing PDP-10 design team has little experience with the KL10 design. Most of the original KL10 designers have moved to other groups and would not be available for such a conversion. The ramp-up time to learn the KL10 design in the detail necessary to do a low-risk conversion could be quite high.

The KL10 design exhibits a large number of bugs, mostly relating to extended addressing and PXCT. At present, these bugs are being circumvented in software, but it's only a matter of time before we find a bug that can't be fixed in this manner. If the conversion is to be a design that will last, these bugs must be fixed as part of the conversion. This makes the process less of a conversion and more of a design effort.

A KL10 conversion into gate arrays is a non-trivial task. Because gate arrays cannot be ECOed, extensive design verification techniques must be used at the expense of schedule. Our estimates of the cost of doing the Jupiter design in gate arrays indicate that there is an added cost of 9 months because of the verification process that is necessary. An equivalent schedule penalty is probably applicable to a KL10 conversion to gate arrays.

Due to bus width limits, the KL10 physical memory addressing capability is limited to 4 Mwords. Analysis indicates that this

is probably insufficient to support a machine whose speed if much greater than a KL10. Any KL10 conversion would have to include an expansion of the physical address space.

Any KL10 conversion would have to include the design of more optimal I/O interfaces to the CI and NI buses. While there exist CI and NI interfaces today, the requirement that they obey the RH20/DTE20 protocol inhibits the efficiency of the interfaces. They are in essence, bolted-on interfaces that are limited by the design of previous interfaces.

Finally, converting the KL10 to 10KH or MCA technology appears to be limited to a performance increase of 1.5-2.0.

4.0 A Jupiter, but at what speed?

Even if the Jupiter architectural design is used to build a PDP-10, there is the question of the optimal goal for the machine performance. Selecting such a goal is a function not only of the technical difficulties in achieving the goal, but also time-to-market questions. The optimal product is obviously one which has an infinite KC/KL ratio and which would be available next week. In the absence of such a product, what is the optimal tradeoff between performance and time-to-market?

First, let's consider the technical difficulties involved in implementing a Jupiter design at various KC/KL ratios. Based on previous performance analysis, we believe that a machine whose nominal performance is 2 to 3 times a KL10 is relatively easy to do. A machine whose performance is 4x a KL10 is within reach, but some careful design must be done to use the available gate resources in an optimal way. Finally, a machine whose nominal (as opposed to peak) performance is 5x a KL10 appears to be quite difficult to do.

Therefore, from a technical viewpoint, the choice seems to be between a machine in the 2-3x a KL10 and one which is 4x a KL10. Let us consider possible schedules for both machines.

The primary difference in the design of these two machines is the requirement that additional design work must be done during the architectural and register transfer design stages of the 4x machine. Some additional simulation may also be necessary to to confirm that the performance goals are met. Our estimates are that this additional work amounts to no more than 3 to 6 months in additional schedule.

It is also worthwhile to compare the schedule for a possible KL10 conversion to that for a Jupiter design that is 4x a KL10. Because any KL10 conversion must include design time to correct the existing KL10 bugs, the process is not a simple conversion. An MCAed KL10 also involves increased risk and schedule because of the additional simulation time to make sure that the design is

correct. Our initial comparisons of these two alternatives indicates that a 4x KL10 Jupiter design would require no more than 6 to 9 months additional schedule beyond even a simple KL10 conversion.

5.0 Conclusions

The viable options for building a PDP-10 seem to be limited to a KL10 conversion and a variant of the Jupiter architectural design. The Jupiter architectural design takes advantage of the eight year learning curve with the KL10 and solves many of the existing KL10 architectural problems.

Selecting the performance goal for a Jupiter architectural design involves tradeoffs with time-to-market. A machine whose nominal performance is 4x a KL10 seems to be a nearly optimal choice between performance and time-to-market considerations. In addition, such a machine does not have a schedule which is significantly longer than that for a KL10 conversion.

Continuing with a PDP-10 design based on the Jupiter architectural design which has a nominal performance that is 4x a KL10 seems to be the best tradeoff between performance and time-to-market.