A Low-Cost, Space-Efficient PDP-10

Technical Proposal

24 Jun 83

Pat Sullivan
Mike Uhler

Table of Contents

1.0 Introduction

Today's typical TOPS-10/TOPS-20 customer operates in a fairly centralized computing environment. This situation has been changing slowly as both TOPS-10 and TOPS-20 provide additional services to allow a user to move into a distributed computing environment. Complete integration will require significant additional software work to allow the customer to move gracefully into the distributed environment.

The corporate integration strategy encourages the customer to move into the distributed environment by providing hardware and software development on existing KL10 systems. The development emphasis is on products that make it easy for the customer to develop new applications on VAXes, PCs, and other components of the distributed environment.

A TOPS-10/TOPS-20 customer contemplating new applications development is then faced with two fundamental changes. First he must move from a centralized computing environment to a distributed environment. Second, he is faced with a change of architecture (e.g., PDP-10 to VAX). While there are ways to minimize the impact of either of these changes, it will still be a significant culture shock.

We can minimize the impact of the complete transition by allowing the customer to first move into the distributed environment with the PDP-10 architecture and then onto a new architecture. Although hardware and software upgrades to the KL10 may make it easier to move into the distributed architecture, it is likely to remain a centralized computer.

This proposal is for the development of a low-cost, attractively packaged PDP-10 processor that is designed to fit into the distributed computing environment. It is intended to be what has been called a "departmental machine" and it allows the current -10/-20 customer to move gracefully into the distributed environment on his existing architecture as the first step in full integration.

1.1 Goals

- Runs TOPS-10 and TOPS-20 with minimum software changes.
- Time-to-market of 18 months or less.
- Low cost.
- Attractive, space efficient package.

- Minimum engineering resources.
- Low-risk technology.
- High reliability.
- Use existing designs whenever possible.

1.2 Non-goals

- High performance CPU

2.0 Functional description

The system described by this proposal contains a single-CPU PDP-10 processor capable of supporting 20-32 users. Disk storage is provided by an RA60 disk drive with provision for 1 to 3 additional RA60/RA81 drives. Synchronous and asynchronous communication is initially provided by traditional line interfaces connected to a Unibus. This will ultimately be upgraded to the use the NI with additional hardware and software work.

The system is built around a PDP-10 processor with full 30-bit extended addressing support. The processor is an upgraded version of the KS10 (2020) design, which we call the KD10, and it uses as much of the existing KS10 design as possible. The processor fits on two extended hex modules and uses the AMD2901 family and Schottky TTL MSI parts.

The memory subsystem contains a memory controller module and one to four 1 Mword memory modules for a total memory capacity of 4 Mwords. The memory modules use 256K MOS RAM parts and include parity and ECC bits to allow single error correction and double error detection.

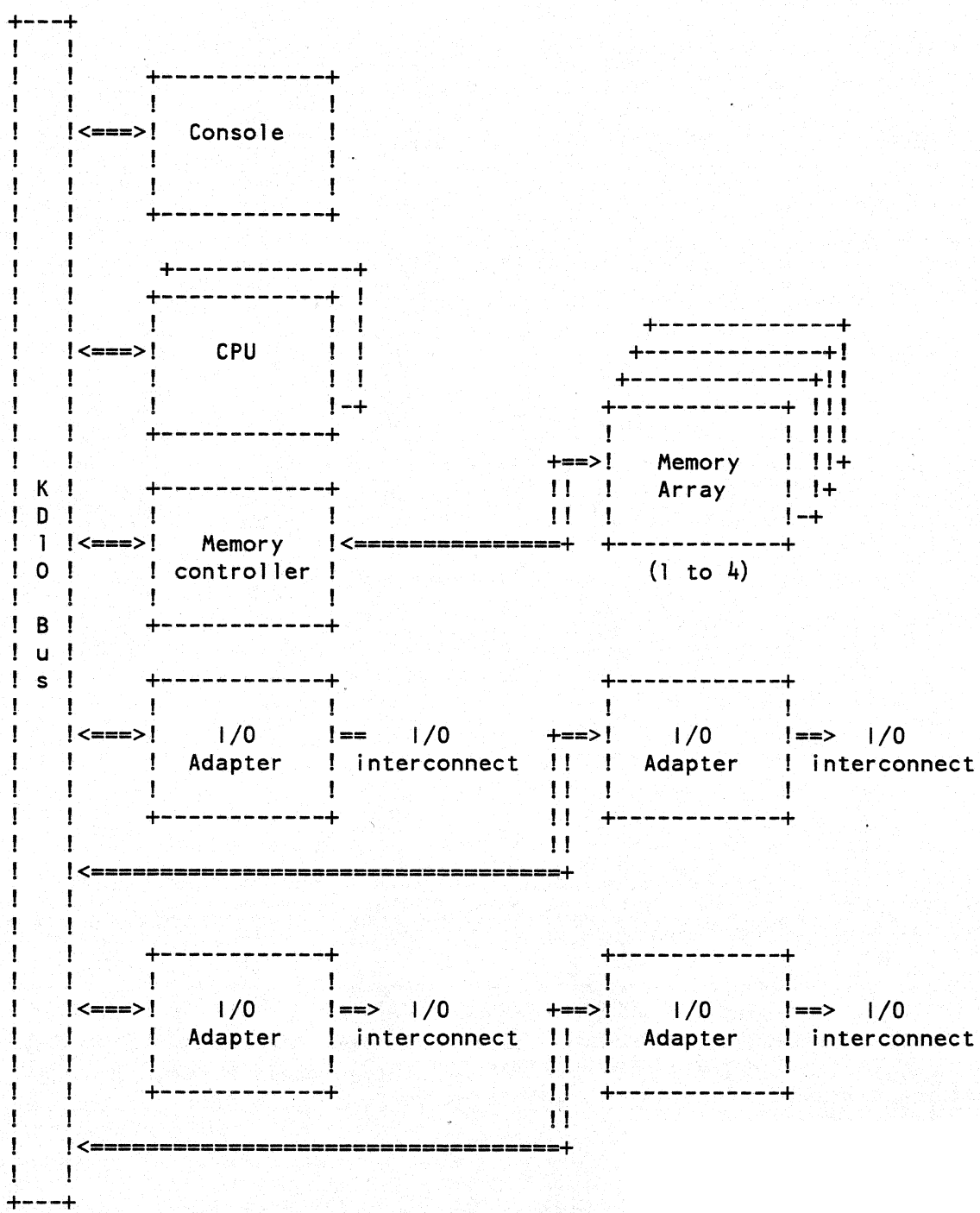
The I/O subsystem consists of 2 to 4 I/O adapters where, in the minimum configuration, one adapter is used for disks and the other is used for all other I/O functions.

The console is a single extended hex module containing an 8080 microprocessor with console code in PROM and RAM. It is an almost exact copy of the KS10 console module, modified only where absolutely necessary.

The main interconnect between the CPU, memory, and the I/O adapters is the KD10 bus which provides a control and data path between the system components. Bus operation is identical to that of the KS10 bus used in the KS10 processor.

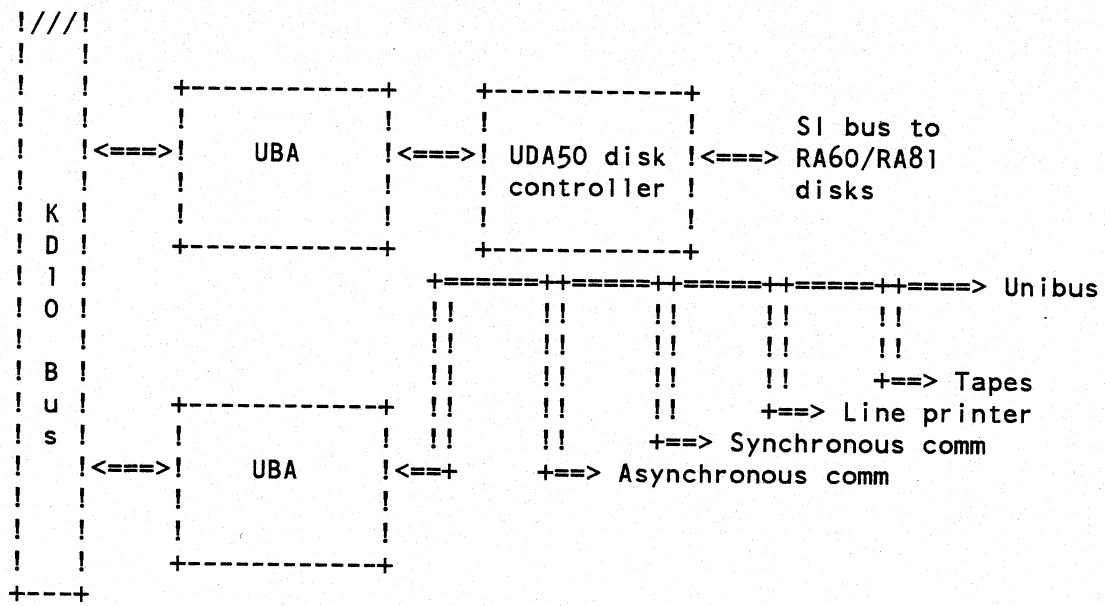
2.1 System interconnects

The following diagram shows the major interconnections between the components of the system:



2.2 I/O structure

Because time-to-market is important to the success of the product, the initial offering uses existing I/O interconnects to avoid the cost of engineering new ones. The first machines will use a Unibus Adapter (UBA), similar to the one used on the KS10, and a UDA50 disk controller to control the system disks. A second UBA will be used for all other I/O including asynchronous and synchronous communication, a line printer, tape drives, etc. Thus, the first machines will have the following I/O structure:

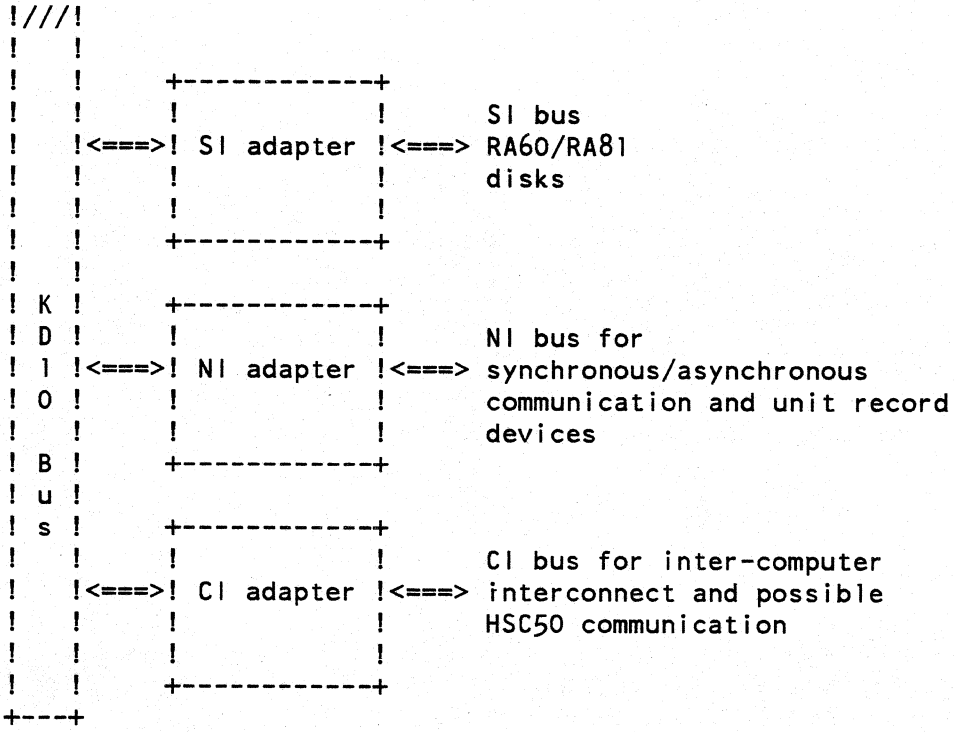


This structure is very similar to that used on the KS10, and allows us to take advantage of the software work done for the KS10. It also takes maximum advantage of existing corporate peripherals to minimize schedule.

Subsequent machines will be upgraded with CI/NI support at FCS plus 3 to 6 months. NI support requires release 6.1 of TOPS-20 which may not be available in time for FCS. Also, additional hardware work is necessary to develop CI and NI adapters and this work is not included in the 18 month time-to-market estimate.

Future investigation may also reveal the need for a more cost-effective or higher performance interconnect between the CPU and disks. If this is true, the UBA/UDA50 combination could be replaced by an SI adapter.

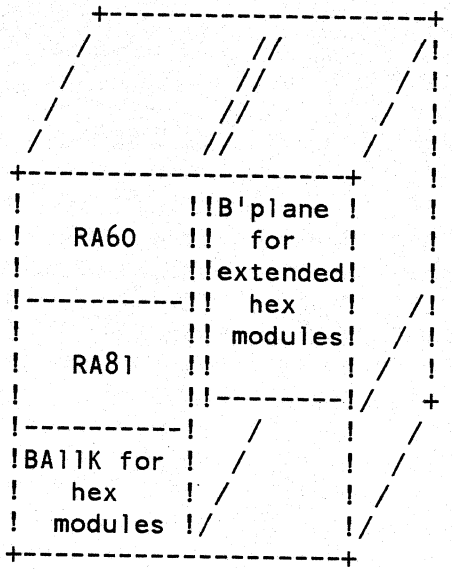
The ultimate I/O structure might look as follows:



3.0 Packaging

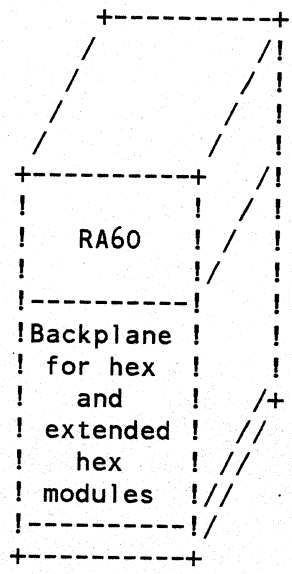
Space efficiency of a system has become increasingly important over the last few years. Because of the size of KL10 systems, customers have discovered that they are limited by the amount of floor space necessary to support their computing needs. In addition to price/performance ratios, today's customers are also using MIPS/square foot as a figure of merit in considering systems. We were very aware of this factor when considering proposed packages for the KD10.

We have considered two possible packaging schemes and others are also possible. The first package uses an 11/750-size cabinet and includes the processor, an RA60 removable media disk and an RA81 fixed media disk in the cabinet. This package might look like:



Possible package using
11/750-style cabinet

The second package uses an RA81 cabinet and includes the processor and an RA60 removable media disk in the cabinet. This package might look like:



Possible package using
RA81 cabinet

In both packaging schemes, additional disk storage would be obtained by adding an additional RA81 cabinet with up to 3 RA60 or RA81 disks.

4.0 Configurations

5.0 Manufacturing cost estimates

Our investigation into manufacturing costs is incomplete at this time, but our goal is an entry-level system in the $15,000 to $20,000 range.

6.0 Development time and resources required

Time-to-market considerations are of primary concern to the success of the project. As a result, we will make use of existing designs and the knowledge gained from previous designs whenever possible. For example, the KD10 processor is based on the existing KS10 design. Also, the I/O interconnect structure is based on existing corporate buses and VAX and 11-based hardware. We are designing new hardware only in areas that warrant that approach (e.g., the addition of extended addressing to the KS10 design).

Because of the limited availability of software and diagnostic resources, we are making the hardware/software interface identical to existing interfaces whenever possible. Doing so allows us to take advantage of existing software and minimizes the amount of new software required.

6.1 Hardware

The amount of hardware engineering work necessary is limited because we are making use of existing designs whenever possible. The effort is as follows:

1. Start with the existing KS10 processor design, add the logic necessary to support extended addressing, and take advantage of technological advances to reduce the size of the CPU from four modules to two.

2. Upgrade the existing memory controller module to support 4 MWords of physical memory addressing.

3. Upgrade the existing KS10 memory modules to use 256K MOS RAM parts and support 4 MWords of physical memory addressing.

4. Modify the clock logic on the console module to reduce the cycle time.

5. Modify the UBA design to support the data packing necessary to support the UDA50 disk controller.

6.2 Simulation

Simulation is an important part of producing a design that powers up and runs the first time. We are attempting to use the best parts of the Jupiter and Venus simulation strategies to minimize (and hopefully eliminate) the hardware debug stage of the machine. There are three parts to the KD10 simulation strategy: register transfer simulation, gate-level simulation, and gate level timing verification.

Due to the limited size of the design (estimated at 15,000 gates), we can afford to do rather extensive gate-level simulation. As a result, we will be doing register transfer simulation using the LISP simulator developed during the Jupiter project. This simulator has the advantage of requiring no additional resources outside the KD10 development group for support, generation of models, etc. This simulator will be used for initial functional debug and microcode development.

Gate-level simulation will probably be done using SAGE. The Venus project has thoroughly debugged the process and it is well understood. Because of the limited size of the machine, we should be able to run all functional tests through the gate-level simulator. In this manner, we will be able to insure equivalence between the register transfer model and the gate-level model.

Gate-level timing verification will be done with AUTODLY. This process is currently being debugged by the Venus project.

6.3 Microcode

Because we have limited time and resources, writing new microcode to support the PDP-10 architecture is out of the question. Because the processor design is structurally very similar to the KS10, our approach is to start with the existing KS10 microcode and make incremental changes. The required changes are as follows:

1. Start with the KS10 microcode and convert from MICRO (an unsupported microassembler) to MICRO2 format (the corporate microassembler).

2. Make incremental changes to convert from KS10 hardware/microcode interface to the KD10 hardware/microcode interface, making no optimizations. This step results in working microcode which supports an un-extended PDP-10 architecture.

3. Make incremental changes to add extended addressing functionality to the existing microcode.

4. Make incremental changes to add new instructions and functions that are supported by extended addressing but which are not implemented by the KS10 microcode.

5. Make incremental changes to take advantage of any additional hardware features that exist in the KD10 design.

6. Measure the performance of the microcode and optimize those functions that are performance critical.

6.4 Software

The KD10 hardware/software interface is similar to the KC10 interface for processor control and similar to the KS10 interface for I/O control. As a result, we can take advantage of existing monitor work and avoid commiting scarce software resources. Necessary software changes include the following:

1. Changes to the processor control code in those places where the KD10 hardware/monitor interfaced isn't exactly that of the KC10.

2. New MSCP disk driver similar to the existing PHYKLP monitor module.

3. Terminal line driver similar to the KS10 driver.

4. DECnet driver similar to the KS10 driver. This item may prove to be the largest part of the software effort.

5. Tape driver.

6. Line printer driver similar to the KS10 driver.

7. Modifications to DDT to reflect the new processor.

8. Modifications to BOOT to reflect the new processor and I/O structure.

6.5 Diagnostics

Because of the similarities with existing machines, the diagnostic effort is also reduced. Necessary diagnostic changes include the following:

1. Changes to the KC10 diagnostic in those places where the KD10 hardware/monitor interfaced isn't exactly that of the KC10.

2. Changes to the KC10 diagnostic monitor to include terminal, disk, and tape I/O similar to the KS10 diagnostic monitor.

3. Modifications to the KC10 functional diagnostics.

4. Modifications to the KS10 hardware diagnostics where necessary to reflect the new KD10 processor structure.

5. Changes to the KS10 microcode conversion/loading utilities to reflect the new KD10 processor structure.

6. Possible development of a new RA60/RA81 disk diagnostic.

6.6 Manpower estimates

Our initial estimates for the amount of work involved in hardware engineering, microcode, and simulation indicate that the project will require funding for four engineers for 12 to 15 months.

At present, we have no firm estimates for the amount of manpower necessary to do the software and diagnostic work.

CAD resources will be required during gate-level simulation and timing verification.

Additional assistance will be required during the layout and placement of the modules.

6.7 Computer resources

Computer resources will be necessary for design entry, microcode development, simulation, and module placement and layout. Our estimate is that the first three items listed will consume one-half to one KL10 for 10 to 12 months.

Layout and placement may require additional resources.

Gate-level timing verification will require the use of a VAX for one to two months.

7.0 Performance estimates

Accurate performance estimates are difficult without additional performance analysis. However, initial analysis indicates that we can halve the KS10 cycle time from 300 ns to 150 ns. To a first-order approximation, this should produce a machine that has twice the performance of a KS10. Beyond that, there are some known techniques based primarily on microcode changes that could

yield additional performance.

The KS10 performance is accepted as 0.2 x KL10. Halving the cycle time and making small optimizations should yield a machine in the 0.3-0.5 x KL10 range.

Mike Uhler

ù digital ù

INTEROFFICE MEMORANDUM

TO: Pat Sullivan

DATE: June 17, 1983
FROM: Bill Martel
DEPT: New Products
EXT: 231-6467
LOC/MAIL STOP: MR01-1/M31
REF.: 8.27

SUBJ: PRODUCT COST: PROJECT "MICRO20"

The enclosed is a first pass product cost indicator based upon composite Engineering/Manufacturing information for the subject project.

In assembling elements of the Project Costs, the following assumptions were made:

1. Cost Projections are for a steady state operations.

2. Should this project be realized, Prototype cost and Manufacturing Development Costs, will be generated.

3. Cost projections assumes a 10% yearly reduction in all areas.

4. Cost of modules assumes capabilities provided for Automated Methodologies in the FY84-85 time frame.

5. Module Cost, as generated, does not provide for ECO's (new components, etch cuts, artwork re-generation, etc.)

6. Costs are based upon the envisional CPU configuration as depicted herein.

7. This estimate will be updated and refined as new Engineering information becomes available.

8. Systems Test, FCC Test, etc. not included in the transfer cost as presented - only the Kernel level.

/ch
Encl.

PROJECT MICRO20

COST ELEMENTS

A - CPU ITEM	QTY	COST FY84	FY85	FY86
1 - CABINET (11750)	1	440	400	400
2 - CONSOLE (M8616)	1	656	623	592
3 - MEM CTRL (M8618)	1	383	364	346
4 - UBA (M8619)	2	928	882	838
5 - DPE (M8620)	1	697	662	629
6 - DPM (M8621) +2901	1	752	714	679
7 - BKPL (KS10)	1	345	328	312
8 - CARD CAGE (KS10)	1	300	300	300
9 - PWR SUP (KS10)	1	1100	990	890
10 - PWR CTRL (KS10)	1	300	300	300
11 - MEMORY (1 MW)	1	2400	2050	1507
12 - CABLES/HARNESS	-	400	400	400
13 - CONTROL PANEL	1	150	150	150
14 - MISC HARDWARE		200	200	200
TOTAL CPU MATERIAL		$9051	$8363	$7543
A - COMM				
1 - DWR BA11-KU	1	1064	958	862
2 - 8-ASYNCH (DMF32) 1-SYNCH-1-LP CTR	1	1157	1077	1044
3 - 24 ASYNCH (DMZ32)	1	2700	2525	2400
C - MASS STORAGE				
1 - RA60 DISK (205MB)		3863	3100	3000
2 - RA81 DISK (450MB)		6258	5632	5068
3 - UDA50 CTRL		1400	1100	990
4 - TU80-AA TAPE (CDC)		3900	3900	3900
5 - TU78-AB TAPE		13200	11880	10692

	QTY	FY84	FY85	FY86
D - PRINTER/TERMINAL				
1 - LA180-CA		$1000	$ 900	$ 810
2 - LP32-AA		3200	3200	3200
E - CPU KERNEL INTEGRATION				
KD10 OPTION LEVEL				
ASSEMBLY 9 HRS à $56.00		504	448	392
TEST 39 HRS à $56.00		2184	1960	1736
TOTAL LOH		$2688	$2408	$2128

Incremental costs given herein from A - thru E should be used in determining a typical systems transfer costs depending upon a configuration level.