DATE: November 5, 1981

TO: Prime Personnel

FROM: Ben Sprachman

SUBJECT: I/O at Prime Today

REFERENCE: None

KEYWORDS: I/O

ABSTRACT

This paper describes Prime's I/O interface from both a hardware and software point of view.

Table of Contents

1 INTRODUCTION

This report gives a brief overview of Prime's I/O as it is today. Both Hardware and Software are described from a cursory level. For a more detailed description refer to the reference documents or seed the applicable video tapes.

2 REFERENCES

PDR3060	System Architecture Reference Guide
PDR3110	Reference Guide, File Management System (FMS)
PDR3621	Primos Subroutines Reference Guide
PDR3044	DBMS Schema
PDR3061	Multiple Index Data Access System (MIDAS)
PDR3276	The DBMS Administrators Guide
PE-T-685	User's Guide to Magnetic Tapes
PE-T-42	Functional Specification - Prime Peripheral Input/Output Bus
PE-T-417	I/O Concurrency Present and Future Issues
PE-T-255	P400 CPU DMX and Interrupt Pri-Net

3 HARDWARE TODAY

3.1 CPU

On all of our current processors, I/O is controlled mostly by CPU microcode with some hardware. The simplest form of I/O supported is Programmed I/O (PIO) which transfers one word of data or control information across the I/O bus under direct control of a macro instruction. PIO is extremely slow and is used mainly to initialize controllers, talk to the VCP or control panel and to transfer data to and from some antiquated devices. For higher speed transfers, DMX is used.

Prime has basically four types of DMX, Direct Memory Access (DMA), Direct Memory Channel (DMC), Direct Memory Transfer (DMT) and Direct Memory Queue (DMQ). A fifth higher speed Burst Mode Direct Memory Access (BMDMA) uses the same format as DMA only it transfers four words per bus access instead of one. DMA uses register file entries as channel control words. To set up a controller for DMA, the CPU must write a negative word count into the first location of a DMA register file cell and then write a beginning address into the second half of the DMA cell. PIO is then used to tell the controller which DMA cell to use. The controller will then issue a DMX request to the CPU. When the request is acknowledged, the controller will start the DMX transfer. The controller presents the DMA cell address to the CPU and then the data which is being transferred is placed on the BPD bus. During DMXin, the CPU writes the contents of BPD to memory and during DMXout, the CPU reads memory and places the contents on the BPD bus. The CPU automatically increments the count and address in the DMA cell. The CPU generates an End-Of-Range signal once the DMA count is zero.

DMC is similar to DMA, only instead of using register file entries for DMA cells, consecutive pairs of memory locations within Segment 0 are reserved for DMC channels. Presently Primos uses 1024 DMC channels. Each channel holds the starting and ending address of the memory block in Segment 0 that is involved in the transfer. The CPU must write this memory locations's address to the controller using a PIO instruction, and then will respond to the controller's DMX request by incrementing the starting address in the DMC channel cell. An EOR is generated by the CPU when the starting and ending addresses are the same.

DMT is a high speed transfer method that requires the controller to furnish the CPU with the actual memory address in Segment 0 that is being written to or read from.

The maximum transfer rate for DMA is 2.5 MByte/sec, for DMC 1 MByte/sec, for DMT 2.5 and for DMQ .28 MByte/sec. Burst mode DMA has a maximum input transfer rate of 8.0 MByte/sec and an output rate of 5.6 MByte/sec.

DMQ is a special DMX mode used mostly by communications controllers.

When a controller requests DMQ, it must furnish the CPU with the address of a special Queue Control Block in Segment 0. The CPU must then read the control block to find out where to access the data. This added level of indirection and the need to update the queue pointers in the Queue Control Block make DMQ the slowest DMX mode. DMQ is widely used by communications controllers because it is the only DMX mode that allows data to be read or written anywhere in memory. The other DMX modes require that a Segment 0 window be used.

The actual peripheral address bus (BPA) is 18 bits wide. Presently only 16 bits are used, the two high order bits being zero. The first four segments in Primos's virtual memory space can be used for I/O mapping. Only Segment 0 is used today. The high order two bits of BPA are meant to select the segment to be used for I/O mapping. The virtual memory system is used to map a page from an I/O segment or window to a page in a users's memory space for I/O. STLB misses cannot be tolerated during I/O because most devices cannot suffer the latency required to process the STLB miss. For this reason a special STLB called the IOTLB is used for Segment 0. The IOTLB has an entry for each page in Segment 0. A routine called MAPIO is called to map a Segment 0 I/O window to a virtual page. This routine must wire, the virtual page to a physical page and modify the page table entry for the Segment 0 page so that it points to the desired target page.

For processors with a 1K word cache size like the P400, I/O operation is straightforward. The STLB entry for the particular virtual page that is being mapped to Segment 0 is invalidated by executing an ITLB instruction. The IOTLB is then loaded by simply accessing the particular page in Segment 0 with a LDA instruction, which causes an STLB miss. The STLB miss microcode must read the page table entry and load it into the IOTLB. References to that page in Segment 0 then correctly mapped to the desired page in virtual memory. This becomes more complicated on the P750 because the cache size no longer matches the page size. I/O pages no longer map directly into the cache and a problem of stale data in the cache can occur. If an I/O page maps into a different leaf in the cache than the previous virtual page did, the data transferred by DMX would not be placed in cache, and a future reference to the page transferred would result in a cache hit on the old data. To solve this problem a wider IOTLB must be used that contains extra bits for part of the tag field of the virtual target address. A special instruction called LIOT is used to load the expanded IOTLB. Extra logic was added to the CPU so that during DMX, a cache hit would result in the invalid bit for that particular cache cell being set.

3.2 I/O Bus

The standard Prime I/O Bus has basically remained unchanged from the days of the P300. It contains roughly 100 signals that are evenly divided between control and data. On the P850 the bus has been extended to cover about 75 inches. This is about the maximum physical length the I/O Bus will function reliably. The P850 required special

cables because of noise problems on the I/O Bus. Presently it is possible to plug in up to fourteen (thirteen on a P850) controllers into a system which is the upper limit for the present I/O Bus. The standard I/O interface present on all controller cards requires about 60 - 90 DIPS. The large number of gates required results mostly from the large number of signals on the bus and to lesser degree, from a complicated daisy chained priority network.

3.3 Current Controller Summary

There are quite a number of controllers in the Prime product line. In practice, only a handful offer significant volume. Many of others have been obsoleted or are useful in very limited markets.

Prime Controllers - Now

The most important current controllers follow. It should be noticed that almost all are being phased out by upcoming products. They include:

* BSMC/4005. This controller interfaces 40, 80, and 300 megabyte storage module (SMD's) as well as 32, 64 and 96 megabyte cartridge module drives (CMD's). Soon, it will also adapt 160 and 600 megabyte Winchester drives (FMD's).

* QAMLC/5104 (and variants). This controller provides character-based serial line multiplexing for up to 16 lines at once. Lines may be current loop or RS232C. They may attach directly to a terminal or rely on an intervening modem. They may also be used to attach to other serial devices such as some lineprinters (Centronix). This is the highest volume controller by far.

* URC/3156 (and variants). This controller (based on the MPC2) interfaces many record oriented I/O devices. These include a wide range of lineprinters (from 165 to 1220 lpm operation) and cardreaders.

* PNC/7041. This controller provides attachment to RINGNET.

* BMTC/2023 (and variants). This controller, in cooperation either a remote or integrated formatter, operates a wide range of magtape drives. They span the gamut from 555 bpi, 7 trk, 45 ips drives to 6250 bpi, 9 trk, 125 ips. ones. This controller is based on the MPC3 design.

* MDLC/5602 (and variants). This controller, with quite a number of options (reflective principally of microcode changes), provides the interface to the synchronous line protocols.

* VCP/2265. This pseudo-controller supports the virtual front panel operation. It also provides the RTC and a number of serial interfaces.

Prime Controllers - Predicted

There are quite a number of controllers underway. Some of these should supplant existing controllers. In fact, of the previous only the URC and BMTC controllers are not being phased out in some way. The new ones motivated by the RABBIT include:

* FERRET. This controller attaches directly to the micromachine, and via an AMLC port (it does not attach to BP). It will replace the VCP in all standard new systems before 1983.

* LCDTC. This controller provides a joint disk and magtape interface. The disk is a sealed PRIAM "Winchester" and the tape a 3M "streamer."

* LYNX. This controller is another hybrid offering both serial (8) and synchronous (1) lines.

Other new controllers supporting communications products include:

* HAWK. This controller, more frequently referred to as the packet-plexor, provides a moderate bandwidth (~1Mbit/sec.) contention bus at extremely low cost. There is a remotely attached version of this same controller denoted the FALCON which interfaces to 8 asynchronous serial lines.

* EAGLE. This controller has an affiliated cardcage which may accept interface cards to a wide range of line protocols. The controller itself is z8000 based and is truly intended to provide a communications front-end. There is a remote version of this unit as well known as the BEAVER (attaching via packet-plexor).

Prime Controllers - Special Interest

A number of Prime controllers serve very parochial market interests. They include:

* SOC/3003 (and variants). This controller (system option controller) has found most of its functions subsumed by the VCP/FERRET in current systems. Nonetheless, it remains the only manner of attaching a number of devices including VERSATEC printer/plotters as well as papertape readers/punches.

* GPIB/7000 (and variants). This controller provides a general purpose prototyping card with which to attach to the I/O bus. N.B. It is not the IEEE 488. Though serving a parochial interest, this

controller has moderate volume.

* DIS/6020 (and variants). This controller offers a general purpose parallel input mechanism (digital input system).

* DOS/6040 (and variants). This controller offers a general purpose parallel output mechanism (digital output system).

* AIS/6000 (and variants). This controller provides the opportunity for A/D conversion of input signals.

* AOS/6060 (and variants). This controller provides the opportunity for D/A conversion of output signals.

* FDC/4030. This controller, MPC2 based, interfaces the twin 8" floppy.

* MPC4. This controller is a basic building block. It is microcoded (and the control store contents may be downline loaded). Variants of it exist in "specials". The highest volume of these is the FORD/LUNDY controller which promises to sell as many as 600 in the near future. It's also used as the RAMTEK interface, the MEGATEK interface, and soon will be used for the HYPERCHANNEL interface.

Prime Controllers - Obsolete

A number of Prime controllers are no longer sensible to sell. There may be some in both the field service organization and customer sites. They include:

* SMC/4004 which was the non-burst mode variant of the 4005.

* UDC/4003 which was the predecessor of the 4004. It does interface a number of very low density disks, viz. FHD and MHD ones, that are no longer compatible with the 4005.

* AMLC with DMT based output rather than DMQ.

* MTC which was the non-burst mode predecessor to the BMTC. It is also an MPC2 based design.

* HSSMLC was the predecessor of the MDLC and is no longer needed. The HSSMLC offers certain functionality that the MDLC has chosen not to provide and so finds applications in specials.

* SMLC was the predecessor of the HSSMLC. It is completely supersetted.

* MACI which was intended to support multiple dial up lines.

* IPC which was intended to provide a high-bandwidth interprocessor link.

* ELF which was intended to provide a more general-purpose secondary I/O bus.

4 SOFTWARE TODAY

4.1 Language

Most of the 'I/O' software at the language level is used to do data conversion and formatting. This usually involves moving data to a language area buffer. e.g. A FORTRAN formatted READ calls the language routine F$RA to change the data to binary and set up a buffer to agree with the FORMAT Statement. On a write, the data is first formatted and then a lower level IOCS subroutine is called. On a read he IOCS routine is first called and then the data is formatted by the language routine.

4.2 IOCS (Input Output Control Systems)

The routines that make up IOCS fall into three classes: (1) Device-independent drivers (e.g., routines to read and write ASCII). (2) Device specific drivers that issue the correct format for a particular device, but allow the output to be read later by device independent drivers. (3) The lowest level IOCS functions - routines to perform raw data transfers (e.g., T$MT which moves raw data from magnetic tape to user address space).

IOCS also contains tables and routines for mapping logical device numbers to physical device numbers and vice versa. IOCS routines may run in the User's space or under PRIMOS depending on the type of I/O. Some of the I/O routines that are supported today are only used under DOS or by user under Primos when a device has been assigned. These include D$INIT, RRECL and WRECL. These routines initialize and read or write to a disk respectively. PRIMOS uses a completely different method to read and write to disk.

There are also a complete range of Device-Dependent drivers whose names begin with 'I$' for input and 'O$' for output. These routines are used for file system transfers, user terminals, paper-tapes, line printers, ·ag-tapes, plotters, etc.. There are a separate set of these routines .or each device, for binary and for ASCII data. There are also a set of Device-Independent routines which will read or write a buffer for a logical device. There are also many Class (3) routines for talking to various subsystems and devices such as card punches and readers.

4.3 File Management System

Prime's File Management System is based on logical disks. Disk partitions are built using the utility program MAKE. MAKE formats the disk and sets up the basic top level file system. For storage module disks the record size is the same as a sector, 1040 words. Each

physical disk partition contains a top level directory called the MFD. Contained in the MFD are a number of files which are created at MAKE time; they are MFD, BOOT, BADSPT and DSKRAT. The first physical record on the disk surface 0, head 0, track 0, record 0 contains the first entry of the MFD, the BOOT file. The DSKRAT file which is given the same name as the partition by MAKE contains all physical information about the partition. It contains the record size, number of records in the partition, the number of heads in the partition and then a bit mapped table indicating what records are available on the partition. The DSKRAT file always starts at record 2 on a partition.

Each storage module disk record contains a sixteen word header so that the 1040 word record size is composed of 16 words of header and 1024 words of data. The record header contains forward and backward pointers to the next and previous record in the file as well as the record type. The record type may be a SAM file,a DAM file, a SAM Segment Directory, a DAM Segment Directory or a UFD. There are also headers for UFDs, UFD entries, and Segment Directories. When a file is written onto a virgin disk, Primos will try to use every third record on the surface. This is done to optimize sequential file access. Each record that is read off of the disk requires some processing, and there is software overhead associated with reading each record. If a file was contiguously allocated on the disk, and a sequential access of two records was required, the head would pass by the second file before the CPU is ready to read it. In this case the CPU would have to wait for a complete rotation of the disk before the desired file is read. By spacing file entries three disk records apart, the CPU can read sequential files on the same rotational cycle.

Sequential access files (SAM files) must be accessed sequentially thus to access the tenth record in a file all ten records must be traversed to find the pointer to the correct record. Direct Access files (DAM files) can be accessed more directly because an index is contained that has entries pointing to all records in the file. The index can be multilevel which means that a DAM file access can take more than one file read.

Primos contains 64 associative buffers called Locate buffers that are used to cache most recently used disk information. These buffers are located in Segment 1. The virtual memory system is used to map these buffers to a user's address space. The last page in the users's stack segment, segment 6000 is mapped to point to a locate buffer by the routine Locate. Each buffer contains a copy of a 1K disk record. The sixteen word header for that record and other information for hashing to the correct buffer is stored in the procedure segment for Locate. The surface, head, track, sector number for the record is also contained in the Locate procedure segment. The Locate buffers speed up file system I/O by reducing the number of disk I/Os. On a write to the disk, a read-modify write may be avoided if the record header is contained in a Locate buffer. On a read, it may not be necessary to read from the disk at all if all the information is contained in a Locate buffer.

For language I/O, Primos uses a device independent driver such as WRASC

or WRBIN to format the data and copy it into a buffer. A call to PRWF$$ (Position Read Write File $$) is used to read, write or position the file pointer on one of the open Primos file units. PRWF$$ calls Locate to find the actual physical record. Locate hashes into the Locate Buffers looking for the desired record. If the record is not found in the Locate buffers then the disk must be read to find the record. Locate calls the routine WREC to write the request to the appropriate controller's request queue. WREC also implements a modified C-Scan algorithm by filling up the queue with requests in a way that minimizes seek time. The disk DIM (Device Interface Module) is a separate process that is running on the system or waiting on a semaphore at all times. It reads the disk request queue and sets up the controller for the required transfer by setting up the DMA channel, initializing the controller,creating the channel program for the controller to read/execute and causes the disk to do a preseek. From this point on the controller now handles the I/O transfer by requesting DMX to the CPU. The data is written directly into the appropriate Locate buffer by the DMA transfer.

In general many more reads of the disk than writes must occur. If the file is a DAM file, the directory must be read in, the header for the DAM files and the appropriate number of index levels before the actual record may even be obtained. If the file is a MIDAS file, even more records may have to be read to obtain the desired record. SAM files require that file be accessed from the beginning, no matter what record is desired. e.g. If the record '2000 of a SAM file is required, then it is necessary to read the '1777 records previous to the one required. This will never usually happen because SAM files are meant to be read sequentially. A simple file access requires only one call to PRWF$$, but PRWF$$ may call Locate several times to find the desired record.

For a simple file access many I/O requests may be made. For each file access, data must be moved at least two times. (1) The high level language routine does a copy when it formats the data. (2) PRWF$$ copies the data from the IOCS buffer to the Locate buffer.

4.4 Data Management Products

The two most important database products Prime has are DBMS (Data Base Management System) and MIDAS (Multiple Index Data Access System). MIDAS provides a series of programs and subroutines for the creation and maintenance of Keyed-Index Direct Access (KI/DA) or ISAM (Indexed Sequential Access Method files).

MIDAS is by far the most popular data base product. Over 80 per cent of Prime's customers have MIDAS. MIDAS files are built on top if the file system. They use a DAM Segment Directory to store the complete file. The Keys and pointers into a MIDAS file are stored in the first ten segments of the segment directory and the data starting at segment 185. The Keys are stored in a B-Tree structure, with usually less than three levels in the structure. The Keys have associated pointers which point to data records which are stored as DAM files.

DBMS is a much more extensive and complicated product. It implements the 1974 CODASYL standard for a data base system. It uses a DAM Segment Directory to store files. An arbitrary structure based on a user description of the data base called the Schema is used to index into data files. The Schema uses a B-Tree structure for speeding up key searches. DBMS files can extend across disk partitions, so a master file called the Schema Directory SCHDIR is maintained on the master DBMS partition. This file must be accessed to find the partition that a particular Schema resides on. A program called RAM (Random Access Manager) acts as an interface between DBMS and the File Management System itself. It is impossible to estimate how many file accesses take place to retrieve a data record in an average transaction, because the complexity and number of levels of indexing required is directly dependent on the Schema.

4.5 Communications

There are three basic communications products today, the AMLC for asynchronous lines such as terminals, the MDLC or SMLC for synchronous computer to computer communications and the PNC for RINGNET communications. The AMLC board is by far the most common hardware communications product. Software handles input and output of characters to an AMLC board in a completely different manner. When an AMLC board detects a character on one of its sixteen lines, it writes the character and a four bit line number into a tumble table contained in Segment 0. There must be a separate tumble table for each AMLC board. One of the AMLC lines is set to run at 110 baud on all systems and is used to interrupt the CPU every tenth of a second. The interrupt code notifies the AMLC DIM which causes the tumble table to be copied into a buffer called FRMLIOB (FRoM Logical I/O Buffer) which is wired in memory. The size of this buffer is set by the administrator at cold start and is usually about 140 words. Higher levels of software then poll the buffer, do the kill and erase character processing and set up lines terminated by carriage returns. The User program can then read the characters out of the buffer.

To write a character to an AMLC line, the User program calls a routine such as TNOUA which will write a character into the TOLIOB (TO Logical I/O Buffer). This buffer is also set up at cold start by the dministrator and is usually about 300 words deep. Every time the CPU is interrupted by the AMLC board the AMLC DIM transfers the contents of the TOLIOB into DMQ buffers. These buffers are usually about 10 words long and are setup at cold start time. Every character time a DMQ operation is started by the controller to empty out the DMQ buffers. For 9600 baud operation, a slight modification was made so that if no character was received in five seconds, the DMQ requests were terminated until another character was received. Terminal echoing is achieved by automatically copying characters received in FRMLIOB to TOLIOB.

DPTX products and Prime's Networks, other than Ringnet use synchronous controllers such as the MDLC/SMLC. These controllers use DMC for both

input and output. A process called NETMAN on Primos Rev 19 systems, controls the synchronous communications. It basically builds up message packets that are moved into buffers that are wired in memory at cold start. The SLC DIM handles the PIO and processes the interrupts required to transmit and receive the messages. All of Primes's emulator products used accept DPTX uses the Ring 3 routine T$SLC which handles the communications protocol at a low level and interfaces directly with the DIM.

The other important Communications controller is the Prime Node Controller (PNC) which is used on Ringnet. The software is very similar to the SMLC software, only fewer interrupts are used. The PNC is fully buffered so no over runs can occur. Presently 512 byte blocks are transfered across the network, but this will soon be increased to 2048 byte blocks. The routine MAPIO is called to dynamically allocate a Segment 0 I/O window for each block of data that is to be transfered. A separate DIM the PNC DIM is used to handle the interrupts, setup the DMC channels, and startup the controller with PIO instructions.

4.6 Magnetic Tapes

Presently magnetic tapes may only be read or written to by the Primos IOCS routine T$MT. Unlike the other controllers there is no separate process for the DIM running on the system at all times. T$MT is actually part of the tape DIM MTDIM which runs from the user's space. There are plans to make MTDIM a separate process in the near future. Presently tape I/O uses DMA or BMDMA, and has a DMA cells reserved for it use only. In the future DMA cells will be dynamically allocated. A maximum block size of 6K words may be read or written onto tape. The Primos Backup utility MAGSAV only writes 2K word blocks onto tape. Unlike the Disk Controllers, the Tape Controllers do not use channel programs, instead PIO is used to initialize the controller. The tape interface is much simpler than the communications or disk interface because there is no buffering of data.

5 APPENDIX

5.1 Table of Existing Controllers

Mnemonic:	Comments:	Desig:	Devices:	DMX Modes:	Bandwidth:
BSMC (2)		4005	40,80 MB 300 (SMD) 32,64 MB 96 (CMD) 160,600 MB (FMD)	BDMA DMT	1.2MB
SMC (2)	SPARES	4004	40,80 MB 300 (SMD) 32,64 MB 96 (CMD) 160,600 MB (FMD)	DMA DMT	1.2MB
UDC (4)	SPARES	4003	.5,1 MB (FHD) 6,12 MB (MHD)	DMA DMT	
FDC (1) (MPC2)		4030	8" FLOPPY		
LCDTC (2)			PRIAM DISK 3M TAPE	DMT DMA	1.2 MB
PNC (1)		7041		DMA	
VCP (1)		2265	TERMINET		

Mnemonic:	Comments:	Desig:	Devices:	DMX Modes:	Bandwidth:
QAMLC (8)		5104 5152/5154 5174/5175	TTY, VDU VDU-FORMS, VDU-GRAPH SERIAL LP: 125,140, 300 lpm FERRET MINIFLOPPY	DMQ (out) DMC (in)	184KB
AMLC (8)	SPARES		TTY, VDU VDU-FORMS, VDU-GRAPH SERIAL LP: 125,140, 300 lpm FERRET MINIFLOPPY	DMT (out) DMC (in)	
BMTC (2) (MPC3)	EXTERNAL OR INTEGRATED FORMATTER	2023	800 bpi @ 45 ips (7/9 trk) 1600 bpi @ 45/75 ips (9 trk) 6250 bpi @ 125 ips (9 trk)	BDMA	120/300KB
MTC (2) (MPC2)	SPARES EXTERNAL FORMATTER		800 bpi @ 45 ips (7/9 trk) 1600 bpi @ 45/75 ips (9 trk)	DMC	120/300KB

Mnemonic:	Comments:	Desig:	Devices:	DMX Modes:	Bandwidth:
MDLC (2)		5602/04 5622/24 5646-OPT 5650-OPT 5651-OPT	RS232C/ CCITT V.24 56 KBPS DDS BISYNC HDLC BISYNC 200UT/... HDLC 200UT/...	DMA DMC	56KB
SMLC (2)	SPARES			DMC	
HSSMLC (2)	SPARES SPECIALS			DMC	
URC (2) (MPC2)		3156-901 3156-902 3156-903 3156-905	LP/CR LP/CRP 1200LP/CR CR/P LP: 165, 200,300, 430,600, 750,905, 1000,1220, 1pm 64/96 char MATRIX LP: 300 1pm 96 char CR: 300 cpm	DMA DMC	< 2MB
GPIB (1)	SPARES SPECIALS (HI-VOL)	7000/7010			
AIS (1)		5000/6001			
AOS (1)		5060/6061			

Mnemonic:	Comments:	Desig:	Devices:	DMX Modes:	Bandwidth:
SOC (5)		3009	VERSATEC: 100,200 dots/inch		
- OPT A		3006 3003	PTR/P		
	SPECIALS	3007/3025	BPIOC		
LCLC (4)	LYNX				
(2)	HAWK				
CMLC (2)	EAGLE				
MACI (1)	SPECIALS	5402/ 5403/04	BELL 801 AUTO CALL		
DIS (2)		6020/5021			
DOS (2)		6040/5041			
IPC (1)	SPARES				
WCS (1)	IN-HOUSE	2076			
ELF (2)	SPECIALS	2254			
COMP. (1) PRODUCTS	SPECIALS	6080			
CAMAC (1)	SPECIALS	6090			
FORD/LUNDY (MPC4)	SPECIALS (HI-VOL)		HYPERGRAPH	BDMA	

Mnemonic:	Comments:	Desig:	Devices:	DMX Modes:	Bandwidth:
MEGATEK (MPC4)	SPECIALS			BDMA	
RAMTEK (MPC4)	SPECIALS (HI-VOL)			BDMA	
HYPER-CHANNEL (MPC4)	SPECIALS			BDMA	
IOBUS TESTER	IN-HOUSE	2024			

TECHNICAL DATA ON CPU'S & PERIPHEI

	SIZE			CLEARANCE			WEIGHT	HEAT OUTPUT	START CURRENT	RUN CURRENT	HUMIDITY NON CONDENSING	TEMP RANGE	DEV CAI LEN
	H	W	D	FRONT	SIDE	BACK	LBS	BTU'S/HR				oC	
CPU & CABINETS	63"	25"	33"	5'		2'	400	7000	20A	12A	95% MAX	0 - 50	
800 BPI MAG TAPE	63"	25"	33"	5'		2'	235	1025	10A	3A	40-80%	2 - 50	4-3
800/1600 BPI MAG TAPE	63"	25"	33"	5'		2'	300	2900	10A	5A	40-80%	2 - 50	4-3
75IPS HIGH PERF.MAG TAPE	63"	25"	33"	5'		2'	350	3754		5A	40-80%	15.5-32.2	4-3
300MB STORAGE MODULE	36"	23"	36"	2'	2'	2'	550	4550	39A	9.5A	20-80%	15.5-32.2	20'
80MB STORAGE MODULE	34"	19"	34"	2'	2'	2'	243	2390	22A	4.9A	20-80%	15.5-32.2	20'
10MB STORAGE MODULE	34"	19"	34"	2'	2'	2'	243	2390	22A	4.9A	20-80%	15.5-32.2	20'
12MB DISC	10.3"	19"	28.5"	5'			130	1700	7A	1.5A	20-80%	15.5-32.2	15'
MATRIX PRINTER	39.5"	28"	24.5"	36"	34"	36'	150	2700	10A	3.5A	5-95%	10 - 40.5	30'
CHAIN PRINTER 300/600	42"	36.5	32"	36"	36"	36'	570	2700	10A	3.5A	40-80%	4.4-35	40'
BAND PRINTER	46"	48.5	24.5"	36"	36"	36'	800	11150		15.0A	10-90%	10.37.7	40'
PRINTER PLOTTER	38"	28"	18"	36"	36"	36'	180	3010		6.5A	5-95%	0-50.0	50'
CARD READER	11'	19.2	14"	36"	24"	36'	60	2025	10A	2.5A	40-80%	10 - 37	15'
CARD READER/PUNCH	41"	44"	48"	36"	36"	36'	570	6400	15A	8A	30-90%	10 - 37	20'
TERMINET CONSOLE	34"	22"	20"	36"	24"	36'	98	600	10A	1.5A	10-95%	0 - 43	20'
PAPER TAPE READER/ PUNCH	10½"	19"	12"	36"			47.5	2400	10A	3A	10-95%	5 - 55	10'
DISKETTE	12.2"	19"	20"	24"			50	1600	7A	2A	20-80%	15.5-37.7	20'
96MB CMD	10.5"	19"	31.75	2'	2'	2'	175	2374	7.5	4.1	20-80%	10 - 35	20'
32MB CMD	10.5"	19"	31.75	2'	2'	2'	175	2374	7.5	4.1	20-80%	10 - 35	20'

NOTE 1: These figures are approximate

NOTE 2: Upto 8' can be lost in cabinet on interface cable length.