PRIME
CENTRAL PROCESSOR SUMMARY

Central Processor Features

Central Processor	100		200		300		400		500	
Feature Availability	**Std.**	**Opt.**	**Std.**	**Opt.**	**Std.**	**Opt.**	**Std.**	**Opt.**	**Std.**	**Opt.**
8-channel programmable DMA system	√		√		√		√		√	
32-channel programmable DMA system					√		√		√	
Extended direct memory access (DMC, DMT)		√		√	√		√		√	
Direct memory queue (DMQ)							√		√	
Full control panel	√		√		√		√		√	
Unimplemented instruction trap	√		√		√		√		√	
High-speed register file (32 addressable 16-bit registers)	√		√		√		√		√	
Dual register sets (128 addressable 32-bit registers)							√		√	
8 general purpose registers									√	
Hardware multiply/divide and double precision arithmetic		√		√	√		√		√	
Automatic program loaders (standard devices)		√		√	√		√		√	
Microverification routines				√	√		√		√	
Processor byte parity			√		√		√		√	
Memory byte parity			√		√		√			
Error correcting memory								√	√	
Single- and double-precision floating-point arithmetic				√		√	√		√	
32-bit arithmetic logic unit							√		√	
32-bit integer arithmetic					√		√		√	
64-bit floating-point arithmetic							√		√	
Virtual memory capability (paging)					√		√		√	
Virtual memory capability (paging and segmentation)							√		√	
Stack processing instructions					√		√		√	
Writable control store						√		√		
2K-byte cache (80 nanosecond cycle time)							√		√	
Hardware process exchange							√		√	
Ring protection structure					√		√		√	
Business instructions									√	
Fast floating-point arithmetic									√	

Operational Characteristics

Central Processor	100	200	300	400	500
Word Size: Memory	16 bits	16 bits	16 bits	16 bits	16 bits
Internal	16 bits	16 bits	16 bits	32 bits	32 bits
Instruction size	basic format 16 bits; extended format, 32 bits				
Addressing	direct, indexed and indirect in sectored and relative modes; extended (double word format); and stack processing				
Minimum-maximum main memory (K bytes K = 1,024)	8–128K bytes	8–128K bytes	64–512K bytes	128K–8M bytes	256K–8M bytes
Memory access time	680 ns.	600 ns.	600 ns.	600 ns.	600 ns.

Operational Characteristics

Central Processor	100	200	300	400	500
Memory increment per board	8K, 16K, 32K bytes	8K, 16K, 64K bytes	64K bytes	64K, 256K bytes	256K bytes
Maximum program size	128K bytes per program			32M bytes	32M bytes
Maximum virtual memory space	–	–	128K bytes/user	32M bytes	32M bytes
I/O data path	16 bits	16 bits plus 2 parity bits	16 bits plus 2 parity bits	16 bits plus 2 parity bits	16 bits plus 2 parity bits
Maximum DMT I/O rate	1.3 Mb/sec.	2Mb/sec.	2.5Mb/sec.	2.5Mb/sec.	2.5Mb/sec.
Addressable registers in high-speed register file	32 (includes index register, accumulators, stack register, DMA addresses, etc.)			128	128
Standard instructions	112	117	145	318	545
Optional instructions	9	37	19	–	–
Instruction types	memory reference, input/output, generic, shift				
Typical instruction times					
Add to memory	2.44μs	1.96μs	1.56μs	0.56μs	0.56μs
Skip on condition	2.84–3.30μs	2.04–2.32μs	1.92μs	1.92μs	1.92μs
Hardware multiply	14μs	10.48μs	8.50μs	4.20μs	4.20μs
Hardware divide	18.2–19.6μs	13.68–14.72μs	13.50μs	4.76μs	4.76μs
Single Precision	–				
Floating-Point add		9.35+.48A+.8nμs	9.25μs	5.18μs	3.72μs
Floating-Point multiply		27.82μs	25.20μs	9.00μs	4.02μs
Floating-Point divide		39.46μs	37.90μs	11.92μs	6.04μs
Double Precision	–				
Floating-Point add				6.46μs	4.80μs
Floating-Point multiply				20.14μs	6.46μs
Floating-Point divide				24.04μs	8.68μs

Electromechanical Specifications

Chassis Capacity	5 boards	10 boards	17 boards
Chassis dimensions (W x H x D)	45.6 cm x 26.7 cm x 49.5 cm	45.6 cm x 40 cm x 49.5 cm	45.6 cm x 66.7 cm x 49.5 cm
Weight (including fans and power supply)	22.7 kg	24.9 kg	45.6 kg
Operating temp. range (°C)	0°–50°	0°–50°	0°–50°
Max. rel. humidity (no cond.)	95%	95%	95%
Mounting	table top or rack	rack	rack
Typical heat dissipation (BTU/hr.)	2,000	3,600	4,000
Voltage range (VAC)	190–250	190–250	190–250
Hz (single-phase)	47–63	47–63	47–63
Amps (typical)	5	9	10
Power supply	100 amp. main supply, chassis-mounted	100 amp. main supply, chassis-mounted	100 amp. main supply, chassis-mounted

PRIME

PRIME Computer Inc., 40 Walnut St., Wellesley Hills, Ma. 02181

PRIME PERIPHERALS SUMMARY

Storage Module Subsystem Specifications

	300Mb	80Mb	40Mb	12Mb Moving Head	Diskette
Bytes Per Disk	292.7M	77.0M	38.4M	11.6M	267.5K
Bytes Per Sector	2,080	2,080	2,080	896	880
Sectors Per Track	9	9	9	8	4
Tracks Per Drive	15,637	4,115	2,055	1,624	76
Cylinders Per Drive	823	823	411	406	76
Average Latency Time (MS)	8.3	8.3	8.3	12.5	83
Minimum Seek Time (MS)	6	6	6	8	20
Average Seek Time (MS)	30	30	30	38	390
Maximum Seek Time (MS)	55	55	55	70	770
Transfer Rate (Bytes/Sec.)	1.2M	1.2M	1.2M	325K	31.2K
Height	36 in. 91.4 cm	34 in. 86.3 cm	34 in. 86.3 cm	10.3 in. 26.2 cm	12.2 in. 30.9 cm
Width	23 in. 58.4 cm	19 in. 48.2 cm	19 in. 48.2 cm	19 in. 48.2 cm	19 in. 48.2 cm
Depth	36 in. 91.4 cm	34 in. 86.3 cm	34 in. 86.3 cm	28.5 in. 72.4 cm	20 in. 50.8 cm
Weight	550 lbs. 247.5 kg.	243 lbs. 109.3 kg.	243 lbs. 109.3 kg.	130 lbs. 58.5 kg.	50 lbs. 22.5 kg.
60Hz Start Current	39 Amp. 208 Vac.	30 Amp. 120 Vac.	30 Amp. 120 Vac.	12 Amp. 115 Vac.	14 Amp. 115 Vac.
60Hz Run Current	8.0 Amp. 208 Vac.	8.2 Amp. 120 Vac.	8.2 Amp. 120 Vac.	2.5 Amp. 115 Vac.	4 Amp. 115 Vac.
50Hz Start Current	39 Amp. 220 Vac.	22 Amp. 220 Vac.	22 Amp. 220 Vac.	7 Amp. 220 Vac.	7 Amp. 220 Vac.
50Hz Run Current	9.5 Amp. 220 Vac.	4.9 Amp. 220 Vac.	4.9 Amp. 220 Vac.	1.5 Amp. 220 Vac.	2 Amp. 220 Vac.
Operating Temperature	60–90°F 15.5–32.2°C	60–90°F 15.5–32.2°C	60–90°F 15.5–32.2°C	60–90°F 15.5–32.2°C	60–100°F 15.5–37.7°C
Operating Humidity (non-condensing)	20–80%	20–80%	20–80%	20–80%	20–80%
BTU/Hr.	4550	2390	2390	1700	1600
KVA Rating	1.23	0.70	0.70	.3	0.46

Magnetic Tape Subsystem Specifications

	Standard Performance Subsystem		High Performance Subsystem					
Drive Speeds Forward Rewind (nominal)	45 ips (114 cm/sec) 200 ips (508 cm/sec)		45 ips (114 cm/sec) 200 ips (508 cm/sec)			75 ips (190 cm/sec) 200 ips (508 cm/sec)		
Tape Handling	Auto-retracting buffer arms		Auto-retracting buffer arms			Vacuum Column		
Altitude	0–20,000 ft (0–6561 m)		0–20,000 ft (0–6561 m)			0–4000 ft (0–1219 m)		
Tape Reel Size	Up to 10½ inches (26 cm)		Up to 10½ inches (26 cm)			Up to 10½ inches (26 cm)		
Recording Format (tracks) Mode	7 NRZI	9 NRZI	7 NRZI	9 NRZI	9 PE	9 NRZI/ PE	9 NRZI	9 NRZI/ PE

Magnetic Tape Subsystem Specifications (Continued)

	Standard Performance Subsystem		High Performance Subsystem					
Density (bpi)	556/800	800	556/800	800	1600	800/1600	800	800/1600
Available as 1st drive	Yes	Yes	No	No	No	Yes	No	Yes
Available as 2nd,	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3rd or 4th drive								
Height	63.5 in. 161.2 cm		63.5 in. 161.2 cm					
Width	27.2 in. 69.0 cm		27.2 in. 69.0 cm					
Depth	42 in. 106.6 cm		42 in. 106.6 cm					
Weight	365 lbs. 164.2 kg.		435 lbs. 195.7 kg.					
Power 60Hz	120 Vac.		120 Vac.					
Power 50Hz	230 Vac.		230 Vac.					
Operating Temperature	60–90°F 15.5–32.2°C		60–90°F 15.5–32.2°C					
Operating Humidity (non-condensing)	30–80%		30–80%					
BTU/Hour	1025		2900					

Unit Record Device Specifications

	Matrix Printer	Chain Printer		Band Printer
Speed (lpm)				
64 Characters	200	300	600	1220
96 Characters	125	200	430	905
Print Positions	132	132	132	132
Characters per inch	10	10	10	10
Lines per inch	6	6	6	6 or 8
Paper Specifications				
Width (inches)	4 to 14.875	3.5 to 19.5	3.5 to 19.5	5 to 18.75
Number of copies	Original plus 5	Original plus 5	Original plus 5	Original plus 5
Vertical Forms Control	2 channel	8 channel	8 channel	12 channel
Static Eliminator	NA	Yes	Yes	Yes
Self Test	NA	Yes	Yes	Yes
Height	39.5 in. 100.3 cm	42.0 in. 106.0 cm		46.0 in. 116.8 cm
Width	28.0 in. 71.1 cm	36.5 in. 92.7 cm		48.5 in. 123.1 cm
Depth	24.5 in. 62.2 cm	32.0 in. 81.2 cm		24.5 in. 62.2 cm
Weight	150 lbs. 67.5 kg.	570 lbs. 256.5 kg.		800 lbs. 360 kg.
Power 60Hz	115 Vac.	115 Vac.		230 Vac.
Power 50Hz	230 Vac.	230 Vac.		230 Vac.
Operating Temperature	50–105°F 10–40.5°C	40–95°F 4.4–35°C		50–110°F 10–37.7°C
Operating Humidity (non-condensing)	5–95%	40–80%		10–90%
BTU/Hour	2700	2700		11150
KVA	0.80	0.80		3.30

Unit Record Device Specifications

	Card Reader	Card Reader/ Punch
Speed (cpm)		
Read	300	300
Punch	–	50–200
Capacity		
Hopper	550	1000
Stacker	550	850
Reject	–	100
Card Mechanism	Vacuum Picker	Vacuum Picker
Card Type	80 Column	80 Column
Print Station	–	Single line
Height	11.0 in. 27.9 cm	41 in. 104.1 cm
Width	19.2 in. 48.8 cm	44.0 in. 111.7 cm
Depth	14.0 in. 35.5 cm	28.0 in. 71.1 cm
Weight	60 lbs. 27 kg.	570 lbs. 256.5 kg.
Power 60Hz	115 Vac.	230 Vac.
Power 50Hz	230 Vac.	230 Vac.
Operating Temperature	50–100°F 10–37.7°C	50–100°F 10–37.7°C
Operating Humidity (non-condensing)	40–80%	30–90%
BTU/Hour	2025	6400
KVA Rating	0.60	1.90

Paper Tape Reader And Reader/Punch Specifications

	Reader	Reader/Punch
Peak Speed	200cps	200cps Read 75cps Punch
Reading Direction	Bidirectional, under program control	Bidirectional, under program control
Drive Mechanism	Stepper motor and sprocket wheel	Stepper motor and sprocket wheel
Reading Method	Photoelectric	Photoelectric
Height	7 in. 17.7 cm.	10.5 in. 26.6 cm.
Width	19 in. 48.2 cm.	19 in. 48.2 cm.
Depth	6 in. 15.2 cm.	12 in. 30.4 cm.
Weight	15 lbs. 6.7 kg.	47.5 lbs. 21.3 kg.
Power 60Hz	115 Vac.	115 Vac.
Power 50Hz	220 Vac.	220 Vac.
Operating Temperature	41–131 °F 5–55 °C	41–131 °F 5–55 °C
Operating Humidity (non-condensing)	10–90%	10–90%
BTU/Hour	390	800
KVA Rating	0.11	0.23
Operating Humidity (non-condensing)	10–95%	10–95%
BTU/Hour	2400	2400
KVA Rating	0.70	0.70

Electrostatic Printer/Plotter Operational Specifications

Typeface	Low Density Plot Gothic	High Density Plot courier
Resolution, Dots/Inch		
Vertical and		
Horizontal	100	200
Writing Head		
Configuration	dual array	dual array
Total Writing Nibs	1024	2112
Font, Dot Matrix	7 x 9	16 x 16
Characters Per Inch	12.5	12.5
Columns Per Line	132	132
Printed Lines Per Fan-fold Page (Min/factor Setting/max)	1/54/61	1/64/61
Printer Lines Per Inch (Factory Setting)	6.6	8.0
Plot Width, Inches	10.24	10.56
ASC11 Character Set	96	96
Speed Asynchronous	1000 lpm (printer) 1.2 ips (plotter)	1000 lpm (printer) 1.0 ips (plotter)
Simultaneous Print/Plot	standard	standard
Height	38 in. 96.5 cm	38 in. 96.5 cm
Width	19 in. 48.2 cm	19 in. 48.2 cm
Depth	18 in. 45.7 cm	18 in. 45.7 cm
Weight	160 lbs. 72 kg.	160 lbs. 72 kg.
Power 60Hz	115 Vac.	115 Vac.
Power 50Hz	230 Vac.	230 Vac.
Operating Temperature	32–105°F 0–40.5°C	32–105°F 0–40.5°C

PRIME

PRIME Computer Inc., 40 Walnut St., Wellesley Hills, Ma. 02181

PRIME Product Bulletin

PRIME 400

DESCRIPTION

The Prime 400 is a fast, large-capacity processor that offers an economical combination of high-speed computation and large virtual memory. Compatible with all other Prime processors, it runs software written for multi-user Prime 500, 350, and 300 systems and single-user Prime 100 and 200 systems. It runs programs up to 32 million bytes long. It supports up to 63 simultaneous users. It is logically compatible with all Prime peripherals, controllers, and I/O interfaces, so users can make plug-in upgrades from other Prime processors quickly and easily at a fraction of the total system cost. And its performance and capacity features give it up to a three-to-one speed increase over the Prime 300.

With features like cache memory, segmented and paged memory management and high-speed data transfer rates, the Prime 400 is a powerful and versatile processor for large-scale interactive data processing and computational timesharing systems. The Prime 400 is also ideally suited for distributed processing applications that off-load batch-oriented mainframes, and networks that use smaller Prime processors.

FEATURES

- Up to 32 million bytes of virtual address space per user
- Up to 63 simultaneous users
- Segmented and paged virtual memory management
- Optionally available error correcting MOS main memory expandable to 8 million bytes
- 2K-byte, 80 nanosecond access cache memory
- Embedded operating system for fast user access to all operating system resources
- Hardware-implemented rings of protection for system and user software security
- Automatic microprogrammed system integrity monitor
- Microcode word parity checking
- Hardware stack architecture to optimize shared procedures

PERFORMANCE FEATURES

Prime 400 users will be immediately aware of two fundamental characteristics: the processor is very fast, and the memory and peripheral configurations it supports are very large. The features of the processor's architecture responsible for the nearly three-to-one increase in speed compared to the Prime 300 are:

STACK ARCHITECTURE. Prime 400 programs operate in an environment consisting of a stack segment (containing all local variable values), an instruction or procedure segment, and a linkage segment

(containing statically allocated variables and linkages to common data). Especially efficient addressing modes are provided to access stack and linkage variables. Hardware implemented CALL and RETURN instructions eliminate software stack management routine overhead.

The Prime 400's stack structure optimizes the efficiency of such operations as parameter passing, subroutine and procedure calls, arithmetic expression evaluation and dynamic allocation of temporary storage and context switching.

CACHE MEMORY. A high-speed 80 nanosecond access 2K-byte, bipolar memory acts as a buffer between the central processor and main memory. Using a complex algorithm to determine what main memory information the central processor will most likely use, the cache memory significantly increases the apparent speed of the main memory. The cache algorithm assures a better than 85% "hit rate", reducing the effective main memory cycle time to approximately 400 nanoseconds. Memory mapping is completely overlapped with cache memory access, further reducing total instruction times.

HIGH-SPEED ARITHMETIC UNIT. The Prime 400 does all arithmetic and logical operations in its 32-bit-wide arithmetic unit. Processing data using a 32-bit format, rather than a 16-bit format, significantly improves the execution times of single- and double-precision integer and floating point arithmetic instructions. Also, the design of the arithmetic unit permits complex address formation, such as base plus displacement plus indexing calculations.

MICROPROGRAMMED CONTROL STORE. A comprehensive microcode structure reduces both the number of microcode steps and the time required for each step in an instruction execution cycle. For example, an add instruction is completed in two microcode steps, compared to five in the Prime 300.

INTERLEAVED MAIN MEMORY. The Prime 400 stores consecutive memory locations on separate memory boards, and uses two-way interleaving to speed up sequential memory accesses and maximize the cache hit rate. In effect, interleaving provides 32-bit transfers between the memory and CPU by allowing the processor to read or write two 16-bit words at a time.

DUAL REGISTER SETS. The Prime 400 processor has 128 32-bit hardware registers organized in two separate sets. These registers handle such functions as controlling the processor's 32 high-speed DMA channels and storing machine states during process exchange operations. The processor's process exchange mechanism dynamically and automatically assigns register sets to processes.

PROCESS EXCHANGE. A combination of hardware and firmware automatically allocates central processor resources to the highest priority process (a continuously executing sequence of machine code) in a queue of processes ready for execution. Process exchange handles the swapping of machine states necessary for coordinating between processes ready for execution and those waiting for a specific event to occur. Firmware within the process exchange mechanism automatically dispatches the next ready process of execution, without software intervention.

REAL & VIRTUAL MEMORIES

The Prime 400, like all other Prime central processors, uses MOS main memory. The processor can address up to 8 million bytes of main memory. Using 16K-bit memory chips, Prime offers up to 256K bytes of memory on a standard 16 x 18 inch (40.6 x 45.7cm) circuit board. The main memory's effective access time of 600 nanoseconds is nearly as fast as the central processor because of the Prime 400's 2K-byte cache memory. The cache is an integral part of the central processors, rather than being located on the main memory boards. This prevents memory bus delays from slowing down cache-to-processor transfers.

The Prime 400 uses virtual memory management facilities to provide multiple users with individual address spaces far in excess of the system's physical memory. These include both segmentation and paging, and provide all system users with individual virtual address spaces of 32 million bytes. Each user's address space consists of 128K-byte segments, half of which are available for user programs and half for PRIMOS operating system software. By embedding operating system functions in each user's virtual memory space, all operating system functions are immediately available as if they were an integral part of a user's program, reducing system overhead.

SYSTEM INTEGRITY FEATURES

The Prime 400 features powerful and flexible error detection capabilities. Parity checking is provided throughout the processor and main memory. Microverification routines, invoked either automatically or under program control, test the validity of the central processor's logic and use a diagnostic status word to indicate the cause of a malfunction. A machine-check mode of operation lets the user establish the remedial actions the system will take when it detects data errors or hardware fault conditions.

The processor is also equipped with a comprehensive, hardware-controlled memory protection system. A multi-ring protection hierarchy lets users assign programs to any of several security levels. Thus, multiple users can have open access to specified programs, while other programs and databases can be protected against unauthorized access, and operating system software can be guarded against accidental user intrusion.

INSTRUCTION SET

The Prime 400's instruction repertoire is a compatible superset of the machine instructions available with smaller Prime central processors. Addressing mode compatibility assures that user programs written for any other Prime processor can run on the Prime 400 without modification.

In addition to its compatibility with smaller Prime processors, the Prime 400 offers unique addressing modes and instructions that significantly expand its processing power. Over 80 instructions, including 32-bit arithmetic, NOTIFY and WAIT, and conditional store, provide better operating system communication, enhanced data handling capabilities and cooperating process communication. Also, enhanced address formation allows all instructions, both new and old, to use any combination of four user-accessible base address registers, two index registers and 32-bit indirect words. This permits all instructions to reference a virtual memory space of up to 32 million bytes, compared to the Prime 300's 128K-byte maximum.

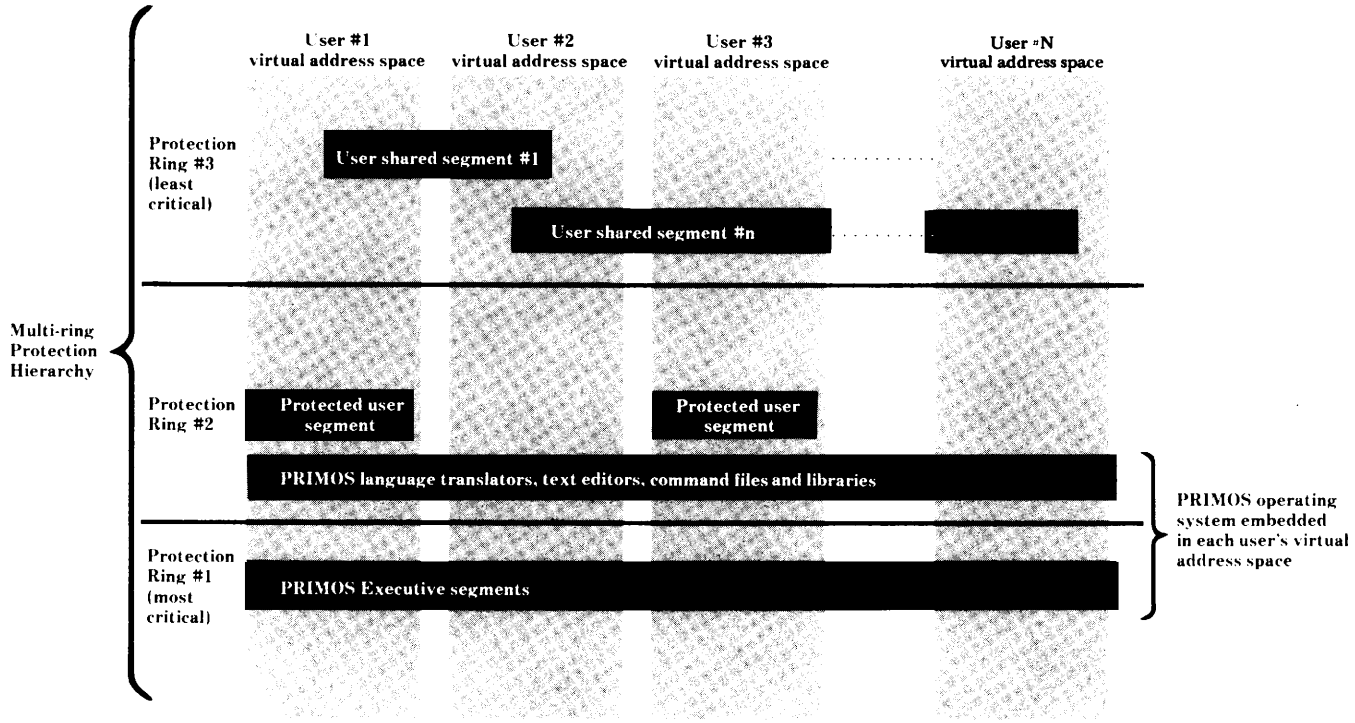
INPUT/OUTPUT

Direct-to-memory input/output operations are supported by three types of program-assignable I/O channels. Thirty-two program-assignable DMA channels, controlled by high-speed channel address registers, provide high throughput with a minimum of central processor control overhead. The channels have a maximum data rate of 2.5 million bytes per second. DMC channels, controlled by channel address words in the cache memory, offer an unlimited number of channels for medium-speed I/O transfers. The DMC channels have a maximum transfer rate of 960K bytes per second. DMT channels are provided for device controllers, such as the controllers for moving-head disks, that execute channel control programs. The maximum DMT throughput rate is 2.5 million bytes per second.

In addition to these direct-to-memory channels, a DMQ mode of operation provides a circular queue for handling communication devices. The queue reduces operating system overhead by eliminating interrupt handling on a character-by-character basis.

SYSTEM SOFTWARE

A single, multifunction operating system—PRIMOS—provides all control functions necessary to support multiterminal, batch, and multitask real-time operations. PRIMOS is embedded in each user's virtual memory space, assuring rapid access to operating system resources by user programs. The operating system supports shared, reentrant procedures, so multiple users can share a single copy of a software module, such as a FORTRAN compiler. In addition to FORTRAN IV, PRIMOS also supports BASIC, BASIC/VM, ANSI '74 COBOL, RPG II, and Macro Assembler languages. PRIMOS includes database-oriented file management resources that permit multiple keyed accesses to on-line data bases.



PRIMOS VIRTUAL MEMORY MANAGEMENT

PRIME 400 CPU

PRIME 400 CPU

PRIME

PRIME Computer, Inc., 40 Walnut Street, Wellesley Hills, MA 02181

PRIME Product Bulletin

PRIME 500

DESCRIPTION

The Prime 500 is the newest and most powerful member of Prime's family of hardware- and software-compatible central processors. It runs software previously written for single-user Prime 100 and 200 systems, as well as user programs written for multiuser Prime 300 and 400 systems, without modification. And since it is logically compatible with all Prime peripheral devices, controllers and I/O interfaces, plug-in upgrades from other Prime processors are easily accommodated at a small fraction of the total system cost. The Prime 500 also offers new performance enhancements for floating-point arithmetic, decimal arithmetic, character manipulation and editing operations, and can provide better than a three-to-one performance improvement, depending on the instruction mix, over the Prime 400.

With features such as cache memory, segmented memory management and new high-speed instructions, the Prime 500 is a powerful and versatile base for large-scale Prime systems providing interactive data processing and computational timesharing services. The Prime 500 is also ideally suited for distributed processing applications involving the off-loading of batch-oriented mainframes and networks including smaller Prime processors.

FEATURES

- 32M-byte virtual address space
- Up to 63 simultaneous users
- Segmented and paged virtual memory management
- Error correcting MOS main memory expandable to 8M bytes
- 2K-byte, 80 nsec access cache memory
- Business instructions for decimal arithmetic, character manipulation, and editing operations
- High-speed floating-point arithmetic unit
- 32-bit arithmetic unit
- Dual sets of 64, 32-bit registers for fast context switching via process exchange
- Eight, 32-bit general registers
- 32-bit internal decor
- Embedded operating system for fast user access to all operating system resources
- Hardware-implemented rings of protection for system and user software security
- Automatic microprogrammed system integrity monitor
- Microcode word parity checking
- Hardware stack architecture to optimize shared procedures

PERFORMANCE FEATURES

Users of the Prime 500 will be immediately aware of two fundamental characteristics: the processor is very fast, and the memory and peripheral configurations it can support are very large. The processor shares many of the high-performance features of the Prime 400 and adds to this performance base a new, high-speed floating-point arithmetic unit, direct hardware execution of business instructions, and an expanded instruction set that supports the processor's eight general registers. The most significant features of the Prime 500's architecture responsible for its high performance levels are described below.

BUSINESS INSTRUCTIONS. The Prime 500 provides high-level support for ANSI '74 COBOL and other business-oriented languages through comprehensive instructions designed for decimal arithmetic, character field manipulation, and editing operations.

Decimal arithmetic operations support packed or unpacked signed numbers of up to 18 digits. Operands differing in data type and/or scale factor are automatically handled during add, subtract, multiply, and divide, and comparison operations. Rounding may be specified on numeric operations and instructions are provided for binary/decimal and decimal/binary conversions.

Character operations can be performed on field sizes of virtually any length. Operations for moves, compares, translates, searches, etc. automatically handle justification, truncation, and padding. Numeric and character editing instructions are provided which easily produce fields in ANSI '74 COBOL-like picture formats.

Business-type operations are performed with a limited number of in-line instructions and, depending on the instruction mix, performance can be improved three or more times compared with the Prime 400.

FAST FLOATING-POINT ARITHMETIC. Users of Prime 500 systems can expect instruction execution times for single- and double-precision floating-point arithmetic to be comparable to those of the considerably more expensive IBM System 370 Model 158. Users upgrading from a Prime 400 can run all existing programs without any modification, and can expect an approximate three-to-one improvement in floating-point instruction performance, depending on the particular floating-point instructions involved.

One of the reasons for the Prime 500's fast floating-point speeds is the use of parallel logic in the floating-point arithmetic unit. Thus, binary multiplication is done four bits at a time, division is done three bits at a time and addition 48 bits at a time. This is significantly faster than the single bit algorithms traditionally used for multiply and divide. Separate logic handles exponents, control, and interim values, so that exponent and fraction calculations are done concurrently with arithmetic processing.

TYPICAL FLOATING-POINT TIMES CHART

Instruction	Prime 400	Prime 500	IBM 370/158
Single Precision			
Add	5.18 μsec.	3.72 μsec.	2.4 μsec.
Multiply	9.00	4.02	2.3
Divide	11.92	6.40	8.9
Double Precision			
Add	6.46	4.80	2.2
Multiply	20.14	6.46	3.6
Divide	24.04	8.68	23.2

STACK ARCHITECTURE. Prime 500 programs operate in an environment consisting of a stack segment (containing all local variable values), an instruction or procedure segment, and a linkage segment (containing statically allocated variables and linkages to common data). Highly efficient addressing modes are provided to access stack and linkage variables. Hardware implemented CALL and RETURN instructions eliminate the overhead of software stack management routines.

The Prime 500's stack structure has been designed to optimize the efficiency of such operations as parameter passing, subroutine and procedure calls, arithmetic expression evaluation and dynamic allocation of temporary storage and context switching.

CACHE MEMORY. A high-speed (80 nanosecond access) 2-K byte, bipolar memory acts as a buffer between the central processor and main memory. Using a complex algorithm to determine the main memory information that will most likely be used next by the central processor, the cache memory increases the apparent speed of the main memory to near that of the processor. The cache algorithm assures a better than 85% "hit rate". Memory mapping is completely overlapped with cache memory access for further reduction of total instruction execution times.

HIGH-SPEED INTEGER ARITHMETIC UNIT. All integer arithmetic and logical operations are performed in the processor's 32-bit wide arithmetic unit. Using a 32-bit format, rather than a 16-bit format, significantly improves the execution times of single- and double-precision integer arithmetic. Additionally, the design of the arithmetic unit permits complex address formation, such as base plus displacement and indexing, to be efficiently handled.

MICROPROGRAMMED CONTROL STORE. A highly efficient microcode structure, like that used on the Prime 400, assures high-speed instruction execution. The number of microcode steps as well as the time required for each step in an instruction execution cycle has been significantly reduced compared to smaller Prime processors, so that an add instruction, for example, is completed in two microcode steps versus five on a Prime 300.

INTERLEAVED MAIN MEMORY. Consecutive memory locations are on separate memory boards so that two-way interleaving can be used to speed up sequential memory accesses and maximize the cache hit rate. In effect, interleaving provides 32-bit transfers between memory and CPU by allowing the processor to read or write two 16-bit words at a time.

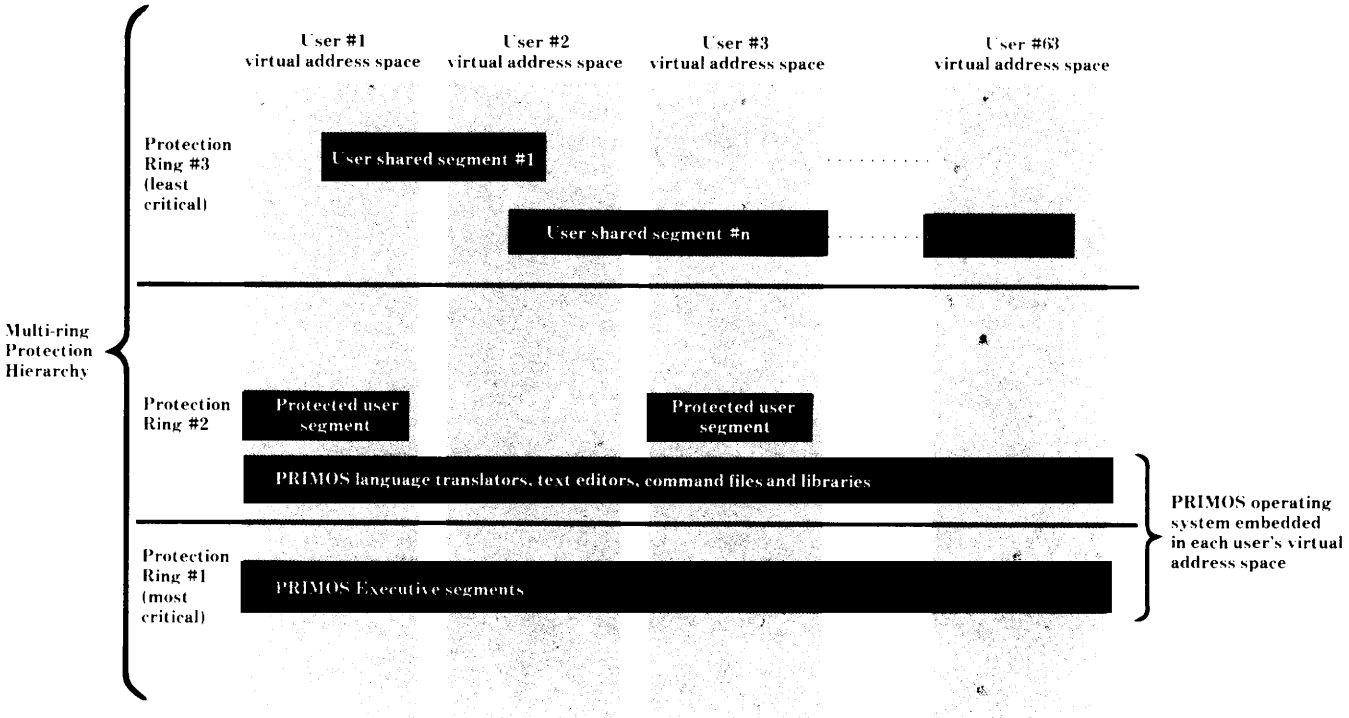
DUAL REGISTER SETS. The processor is equipped with 128, 32-bit hardware registers. These registers handle a variety of functions such as controlling the processor's 32 high-speed DMA channels and storing machine states in dual register sets during process exchange operations. Assignment of these register sets to processes is managed dynamically and automatically by the process exchange mechanism of the processor.

PROCESS EXCHANGE. A combination of hardware and firmware automatically controls the allocation of CPU resources to the highest priority process (a continuously executing sequence of machine code) in a queue of processes ready for execution. Process exchange handles the swapping of machine states necessary for coordinating between processes ready for execution and those waiting for a specific event to occur. Firmware within the process exchange mechanism automatically dispatches the next ready process for execution, without software intervention.

REAL & VIRTUAL MEMORIES

The Prime 500, like all other Prime CPU's, uses MOS main memory exclusively. The processor can address up to 8M bytes of main memory. The main memory's access time of 600 nanoseconds is reduced to near that of the central processor through the use of a 2K byte cache memory. The cache is an integral part of the CPU, rather than being located on the main memory boards, thereby preventing memory bus delays from slowing down cache-to-processor transfers.

To provide multiple users with individual address spaces far in excess of the physical memory available with a system, the Prime 500 CPU is equipped with virtual memory management facilities. These facilities include both segmentation and paging, and provide all system users with individual virtual address spaces. Each user's address space consists of 128K-byte segments, half of which are available for user programs and half for PRIMOS operating system software. By embedding operating system functions in each user's virtual memory space, all operating system functions are immediately available as if they were an integral part of a user's program, thereby reducing system overhead.



PRIMOS VIRTUAL MEMORY MANAGEMENT

PRIME 500 CPU

System Integrity Features

The Prime 500 shares the same powerful and flexible error detection features available on the Prime 400. Parity checking is provided throughout the processor and main memory. Microverification routines can be invoked, either automatically or under program control, to test the validity of the CPU's logic and indicate, via a diagnostic status word, the cause of a malfunction. A machine-check mode of operation allows the user to establish the remedial actions the system will take upon detection of data errors or hardware fault conditions. Additionally, the parity of each control word (microcode) is automatically checked.

The processor is also equipped with a comprehensive, hardware-controlled memory protection system. A multi-ring protection hierarchy allows programs to be assigned to any of several security levels. Thus, multiple users can have open access to specified programs, other programs and databases can be protected against unauthorized access, and operating system software can be guarded against accidental user intrusion.

Instruction Set

The Prime 500's instruction repertoire is a compatible superset of the machine instructions available with smaller Prime CPU's. Addressing mode compatibility is also provided so that user programs written for any single-user Prime 100 or 200 or any multiuser Prime 300 and 400 will run without modification on the Prime 500.

The Prime 500 features instructions that support the processor's eight general registers, and instructions for decimal arithmetic, character manipulation, and editing operations.

A set of approximately 80 instructions, including 32-bit arithmetic, NOTIFY and WAIT, and conditional store, provide enhanced operating system communication, data handling and cooperating process. Also, highly flexible address formation techniques in the Prime 400 and 500 allow all instructions to use any of four user-accessible base address registers, two index registers and 32-bit indirect words in any combination. This permits all instructions to reference a virtual memory space of 32M bytes, compared to a 128K-byte maximum for the Prime 300.

Input/Output

Direct-to-memory input/output operations are supported by three types of program-assignable I/O channels. Thirty-two, program-assignable DMA channels are controlled by high-speed channel address registers and provide high throughput with a minimum of CPU control overhead. The maximum data rate supported by these channels is 2.5 million bytes per second. DMC channels, controlled by channel address words in the first 8K bytes of main memory, offer an unlimited number of channels for medium-speed I/O transfers to a maximum transfer rate of 960K bytes per second. DMT channels are provided for device controllers, such as the controllers for moving-head disks, that execute channel control programs. The maximum DMT throughput rate is 2.5M bytes/second.

In addition to these direct-to-memory channels, a DMQ mode of operation provides a circular queue for handling communication devices. The queue reduces operating system overhead by eliminating interrupt handling on a character-by-character basis.

System Software

A single multi-function operating system—PRIMOS V—provides all control functions necessary to support multi-terminal, queued and multitask real-time operations. PRIMOS V is embedded in each user's virtual memory space to assure rapid access to operating system resources by user programs. The operating system supports shared, reentrant procedures, permitting a single copy of a software module such as a FORTRAN IV compiler to be shared by multiple users. PRIMOS V supports ANSI '74 COBOL, FORTRAN IV, BASIC, RPG II, and Macro Assembler Languages; both levels of Prime data management systems, DBMS, the CODASYL-compliant Database Management System and MIDAS, the Multiple Index Data Access System for multiple entry point access of keyed files; and FORMS, the Form Management System for manipulation of video and hardcopy terminals.

PRIME

PRIME Computer, Inc., 40 Walnut St., Wellesley Hills, Ma. 02181

PRIME Product Bulletin

PRIMOS

THE PRIME OPERATING SYSTEM

All Prime computer systems—from small, dedicated systems using the Prime 100 central processor, to large virtual memory Prime 500s that support dozens of concurrent real-time, timeshared and queued tasks—use a common, uniform operating system called PRIMOS. Since each central processor provides a different level of performance and functionality within the Prime processor family, PRIMOS is implemented in distinct but compatible levels to maximize the effectiveness of a processor's resources while minimizing operating system overhead.

As illustrated, PRIMOS is currently offered on four levels: PRIMOS II, III, IV, and V. PRIMOS II provides an interactive, single-user, disk operating system for PRIME 100 and 200 central processors.

PRIMOS III uses the paged memory management system of the Prime 300. It provides a virtual memory disk operating system that can support 31 simultaneous users.

PRIMOS IV optimizes the Prime 350 and Prime 400's high-speed computational ability and the Prime 400's exceptionally large memory capacity by integrating interactive, batch and real-time supervisory services in a single "embedded" operating system. The Prime 350 and Prime 400 feature segmented and paged virtual memory with two megabyte and 32 megabyte address spaces, respectively. Both support a 2K-byte bipolar cache memory, and a disk capacity that can exceed 1.4 billion bytes (350) and 2.4 billion bytes (400).

PRIMOS V, available with the top-of-the-line Prime 500, includes all the features and capabilities of PRIMOS IV on the Prime 400, plus assembly language support for the Prime 500's general register, 32-bit architecture. Under PRIMOS V, decimal arithmetic, character manipulation, and character editing instructions are directly executed with a combination of hardware and firmware. On the Prime 350 and 400, these instructions are automatically trapped to software subroutines at run-time for emulation.

Large Virtual Memory Ends Most Program Size Limitations

PRIMOS IV and V virtual memory management systems (a combination of segmentation and paging) make it possible for the Prime 350, 400, and 500 to support multiple concurrent processes each with a large private virtual memory space, and a large virtual memory space that is automatically shared with all other processes. PRIMOS IV and V support up to 63 concurrent interactive timeshared users on the Prime 400 and 500 at local or remote terminals. Furthermore, virtual memory resources are available on systems with

as few as 192 bytes of main memory. PRIMOS IV and V can automatically take advantage of additional increments of main memory which can be added (up to the maximum capacity of 8 million bytes on the 400 and 500) to minimize the paging demands of an expanded virtual memory. This feature lets PRIMOS maintain a consistently high level of system responsiveness when the number of processes running on the system increases.

PRIMOS TERMINOLOGY

FILE—a named collection of data and/or code that is created, manipulated, or deleted by PRIMOS File Management System.

PHANTOM USER— a process that is not connected to an interactive user.

PHYSICAL ADDRESS SPACE—an array of real or main memory space that can be a maximum 8 million bytes long.

PROCEDURE—a subroutine referenced by the Prime Procedure Call mechanism.

PROCESS—a separately scheduled entity. All programs of one user run under one process. Each process has a full virtual address space. Only one process at a time may actually be executing on the central processor.

REENTRANT PROCEDURE—a procedure that can be executed concurrently by several processes.

SEGMENT— a 128K-byte portion of virtual address space. Access rights are assigned on a per-segment basis. Calculations of segment number are usually made by PRIMOS IV and V.

SHARED PROCEDURE—a reentrant procedure that is present only once in memory regardless of how many processes are concurrently executing it.

TIMESHARED PROCESS—a process that gains use of the central processor for a fixed time period.

USER—a person associated with an interactive terminal.

VIRTUAL ADDRESS SPACE— the memory space available to each process to run programs. Half of the process' virtual address space is unique or private to that process, while the other half is common to or shared by all processes.

Embedded Design Gives 20 Times Faster Access

PRIMOS IV and V are exceptionally responsive because they depart from traditional designs in which the operating system is a separate, self-contained body of software operated by remote control through intricate and time-consuming sequences of interrupts and responses. Instead, PRIMOS IV and V provide direct and immediate control because they are embedded in the virtual address spaces of all processes. In fact, PRIMOS IV and V are an integral part of each process. The operating system responds immediately to all commands from users of terminals, program calls to the file system, library routines, and other shared procedures. Users are able to access any operating system resource in no more time than it takes for a user program to call a subroutine, which is about 1/20 the access time of conventional operating systems. Note that while operating system and processes share common address spaces, a comprehensive, multi-ring protection system completely protects the operating system against improper access or accidental modification.

Upward and Downward Compatibility Protects Your Software Investment

Programs and data files can be created on one Prime computer that can be used on any other larger Prime computer without modification. This compatibility among all systems holds true not only at the source language level but at the object code and memory-image level as well. Thus it is possible, for example, to create a memory-image file on a Prime 300 controlled by PRIMOS III or on a Prime 350 controlled by PRIMOS IV that executes properly on any larger Prime processor. Furthermore, FORTRAN, COBOL, and PMA programs that run on a Prime 300 can be recompiled to run on a Prime 350, 400, or 500 and utilize their expanded capabilities or simply run as is for a speed improvement. More significantly, the reverse is also possible if the program fits within the size limitation of the smaller system. Thus, a multi-user Prime 350, 400, or 500 can be used as an extremely efficient and economical software development system, creating programs and databases that can be transferred directly to Prime systems.

There are many direct benefits of such upward and downward compatibility. Obviously, a user's programming investment is preserved when upgrading to a larger system. The development of software and the communication among systems in a network of Prime computers is streamlined. Additionally, it encourages the establishment of programming and system design standards that are completely transferable among all Prime products.

Comprehensive System Integrity Features Offer Complete Protection

A combination of hardware, firmware (microprogrammed logic), and software components within PRIMOS IV and V monitors the complete hardware/software system to assure the user that the hardware is operating reliably and that processes being executed are secure. System integrity is constantly monitored by such features as main memory error correcting codes (an optional feature that automatically detects and corrects all single-bit memory errors) and microverification (microprogrammed routines that test the central processor's logic and help determine the cause of faulty operation). PRIMOS IV and V allow the system operator to "lock out" 2K-byte pages of main memory should errors be detected and reported within the pages. The operating system also includes file access integrity features such as forward and backward pointers and built-in utilities to repair damage or inconsistencies.

PRIMOS IV and V complement these integrity features with a multi-level security system for all users, programs, files, and the operating system itself. Certain security measures, such as the protection of the operating system from users and other processes, are automatic and unalterable by any user or program.

Prime's Philosophy Is: Software First

PRIMOS was designed before the computers it operates. This unique software first design philosophy is the key to Prime's ability to offer a range of processor performance and software functionality unmatched on systems costing ten times more. Software first explains why Prime alone offers upward and downward program compatibility among all central processors. Software first makes it possible to establish the uniform file system used by all levels of PRIMOS, all language translators, all utilities, and all libraries. Software first is why Prime systems are so efficient: hardware features have been designed specifically to optimize software performance. In short, software first is the difference between buying a computer with some software included, or buying a completely integrated software/hardware system—from Prime.

WHAT PRIMOS IV AND V DO ...Without Being Asked

One of the key factors in evaluating an operating system is not how it responds to user-initiated commands, but what it does for the user automatically, without being asked. These are the features that determine how flexible and efficient the system is in managing basic resources such as memory and disk file space, CPU time, I/O devices, data communication, etc. These are the implicit operating system functions that, if properly designed and implemented, are totally transparent to the user. Such transparency means major time savings during interactive program development and at job run time.

Implicit Functions

Time Scheduling. PRIMOS IV and V regulate the amount of central processor time used by processes by assigning time slices. The time slice, which is normally set to a 1/3 second duration, represents the maximum continuous time that the process may retain control of the central processor. Time slices are allocated on a priority basis with highly interactive processes receiving a higher priority and processor bound processes a lower priority.

Memory Management. Associated with each process is a large virtual address space. This address space is organized, or mapped, as multiple segments, each segment containing 64 2K-byte pages. The memory management system tracks the location of all pages and handles the physical movement, or swapping, of pages between disk and main memory. Thus, the memory management system automatically transfers the page from the disk to the least recently used area of main memory.

Procedure Data Sharing. In order to minimize paging and thereby maximize system response time, any number of processes can use the identical pages of a shared procedure or data segment. For example, in a multi-user environment, all users can share one copy of the Editor instead of each user having to page-in a separate copy. Shared procedures are reentrant so they remain unaltered by the processes that use them.

Security. A combination of PRIMOS software and Prime 350/400/500 hardware protects the operating system from the processes using its services, protects processes from each other, and enforces the access privileges established for all shared procedures and files. The security system automatically checks the validity of service requests such as read, write, and execute, and prevents such services from being performed if pre-established access rules have not been satisfied by these requests.

Disk Space Allocation. PRIMOS IV and V automatically assign logical files to physical disk records. This feature permits a user to create file structures without concern for the type of disk that stores the files or its operating format and physical characteristics. The operating system provides pointers on disk records to ensure file integrity and simplify access to the next record in a file.

I/O Handling. Built into PRIMOS IV and V are standardized resources for servicing interrupts, maintaining status information, and controlling data transfers for all Prime peripheral devices. These resources permit a user to communicate with any peripheral device using high-level call statements, after the user's process has either been attached to a device (e.g. magnetic tape or printer), or to the files contained on a disk device. A device such as a printer is shared among several users by submitting files to a spool manager that queues the files on a disk and schedules printing on a short/long basis.

Data Communication. PRIMOS IV and V handle all data communication between a Prime 350, 400, or 500 system, a wide variety of interactive terminals, other Prime central processors, and several EDP mainframes. PRIMOS IV and V communicate directly with most currently available asynchronous ASCII terminals operating at speeds up to 9,600 baud. Communication with other Prime central processors is handled via interprocessor controllers for locally connected processors, or via high-speed synchronous lines using a standard packet-switching host-access protocol (an initial implementation of the CCITT X.25 international standard). Remote Job Entry (RJE) communications are supported by IBM 2780, IBM HASP, and CDC 200-UT protocols.

WHAT PRIMOS IV AND V DO ...When Asked

The part of PRIMOS IV and V that is immediately visible to a user contains the resources that respond either to explicit user commands or procedure calls. These resources provide economical and easily accessible services to all users in the amount they need, without advance notification or the intervention of a central operator. Additionally, an extensive list of commands and calls can be used without a broad knowledge of the inner workings of the operating system or central processor. The major services provided by these resources are described below.

Job Control

PRIMOS IV and V support up to 63 processes including interactive users, phantom users, and RJE processes.

Interactive Users. As many as 63 interactive terminals can be on-line concurrently. The system handles any mix of direct-connected local terminals and modem-interfaced remote terminals. Individual users have complete freedom to use system resources as if they were sole users. The user issues commands from a terminal keyboard and gains immediate access to PRIMOS IV and V's resources, which include language translators, editors, debugging aids, and a variety of file structures. Immediate interaction between the user and the system is ideally suited for program development. The user controls and handles exception conditions as they occur instead of depending on a batch processor, thus completing the job in minutes instead of hours or even days. Interaction is what PRIMOS IV and V are all about.

Phantom Users. A phantom user is a process that, once initiated from a terminal, requires no further user interaction or terminal output until it is completed. The user can continue to initiate other phantom user processes from the same terminal, use the terminal interactively, or log it off the system. The operating system treats phantom and interactive users the same way, except that interactive users are given a higher priority for time slices.

Queued Jobs. Some jobs, like disk-to-tape media conversion, sorting, and report printing have turnaround requirements that are less stringent than those of interactive and phantom users. Such processes can be collected in a job queue and run on a first-in, first-out (FIFO) basis under control of a job queue manager. Executing the processes sequentially rather than concurrently conserves time slices so the system can maintain a high level of response to interactive processes.

Remote Job Entry. A Prime 300, 350, 400, or 500 can act as an RJE system by emulating the protocols used by the IBM 2780, IBM HASP, and CDC 200-UT. When any is used as an RJE system, card equipment can frequently be eliminated since programs and data files can be created interactively and then queued on a disk for direct transmission to a host mainframe.

Program Development

PRIMOS IV and V provide an extremely efficient environment for interactive program development. Not only do they assure fast response to all users, they also offer each user access to a complete set of sophisticated software development tools including state-of-the-art COBOL, BASIC, and FORTRAN compilers, RPG II, text editors, and utilities.

COBOL. Prime's COBOL is an implementation of the 1974 ANSI standard. By adhering to this standard, Prime provides an economical migration path for current program libraries to be transferred to the Prime 350, 400 or 500 from other systems with minimal conversion.

RPG II. Prime provides an RPG II translator that is functionally comparable to the one used with IBM's System 3, Model 10. However, when run under control of PRIMOS IV and V, RPG II applications can easily be expanded, using languages such as COBOL and FORTRAN IV and the common file system, to exploit PRIMOS IV and V's interactive and timeshared capabilities.

FORTRAN IV. Prime's FORTRAN is 1966 ANSI-compatible FORTRAN IV with extensions. It is processed by a very efficient one-pass compiler that produces highly optimized code which rivals the efficiency of hand-coded assembly language. As an indication of the flexibility offered by Prime FORTRAN IV, note that FORTRAN IV is the major systems programming language used by Prime. In fact, over 70% of PRIMOS IV and V is written in FORTRAN.

BASIC. Prime's BASIC/VM Compiler supports three modes of operation—conversational, queued, and immediate, to satisfy the computational needs of a wide range of Prime 350, 400, and 500 users. BASIC/VM shares the same file system used by all other Prime language processors. A BASIC interpreter that runs on all Prime processors is also available.

Macro Assembler. The Prime Macro Assembler (PMA) is a free-form, symbolic programming system providing extensive macro facilities that simplify the creation and use of application-oriented commands. PMA includes over 60 pseudo operations for such functions as assembly control, listing and loader control, variable definitions and storage allocation, program linking, and addressing mode control.

File Management

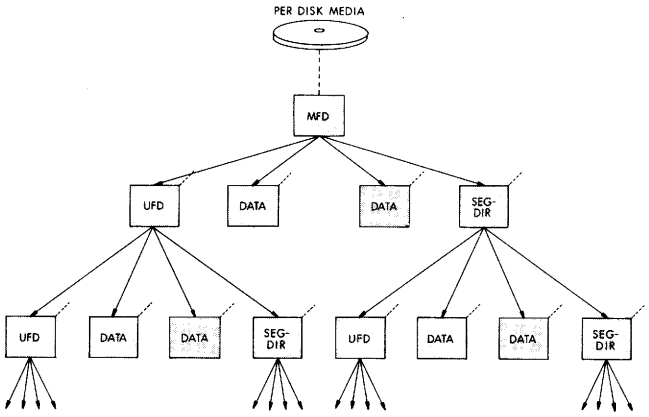
The Prime File Management System provides both implicit and explicit file management resources. The implicit, or automatic, resources manage the allocation of named files to physical address spaces in main and disk memories. The explicit resources provide the file access methods and interactive tools necessary to add, modify, and delete files. PRIMOS IV and V support the same file structure as PRIMOS II and III. As a result, program files and databases created under one level of PRIMOS are directly transferable to any other level, where they can be used without modification.

The full capabilities of the file system are available to COBOL, FORTRAN, and PMA and can be used to develop special file structures and access methods. However, most users simply refer to files by name which are then constructed and located by procedures such as the editor, language translators, and keyed index/direct access (KI/DA) method.

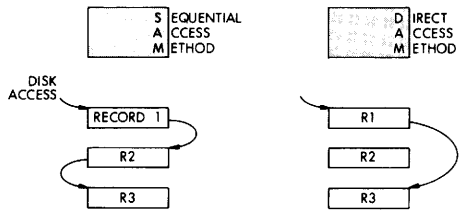
The file structure, as illustrated below, can be viewed as a heirarchy or tree-structure containing specialized file directories and data files.

Master File Directory. The Master File Directory (MFD) is the highest level in the file structure. The file system creates and maintains an MFD for a disk or a user-specified portion of a disk. The MFD contains the names and locations of user-file directories, segment directories, and data files.

PRIMOS IV FILE STRUCTURE



ACCESS TO RECORDS WITHIN A FILE



User File Directory. User File Directories (UFD's) are usually associated with individual users and user processes, and contain pointers to named data files and additional file directories.

Segment Directory. A Segment Directory (SEGDIR) is a file that contains pointers to subsidiary files. It permits rapid access to large collections of data that have an established order but variable size. Files are accessed by simply indexing by pointer position within a SEGDIR.

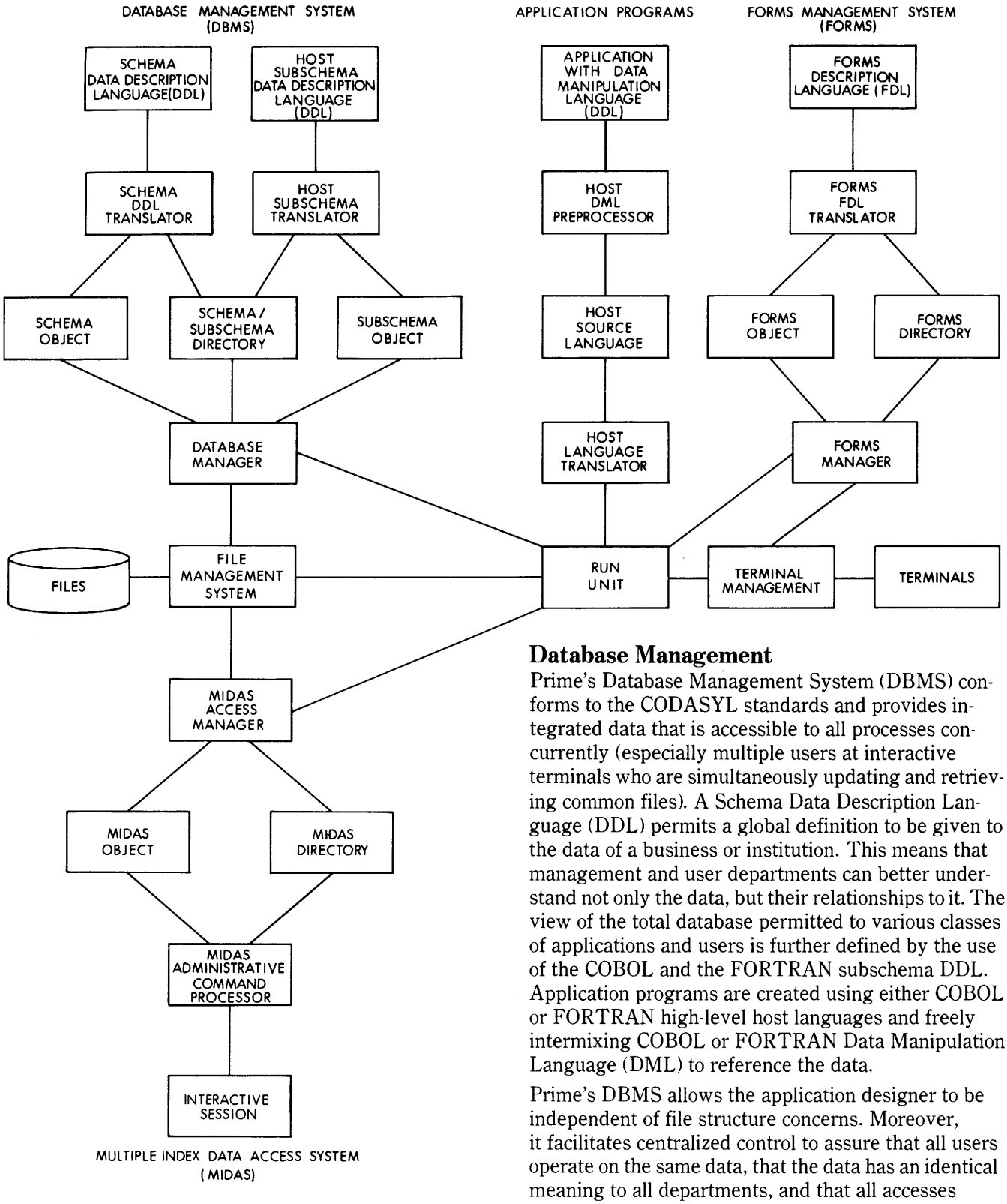
Sequential and Direct Access Methods. Files can be constructed to permit access to their contents using either a direct access method (DAM) or sequential access method (SAM). By convention, all file directories used by the operating system are structured as SAM files. In SAM files, each record contains a pointer to the next record in sequence, thereby reducing the number of directory accesses needed to search the file. DAM files, in contrast, store pointers to all records in the first record of a file. SAM or DAM files may be of any size that can be accommodated on the disk.

Keyed Index/Direct Access Method. KI/DA provides a fast and versatile method for locating, adding, deleting, and modifying items in any size data file. Up to 20 different key fields can be used to access a single item. KI/DA's shareable procedures use the SEGDIR and DAM capabilities of the Prime file management system to minimize the time required to locate an item.

Data Management

Prime's Multiple Index Data Access System (MIDAS) is a part of PRIMOS. It bridges the gap between the File Management System (FMS) and the Database Management System (DBMS). MIDAS files are directly supportable under DBMS; this permits a controlled migration from MIDAS to DBMS. MIDAS uses the resources of FMS and, in particular, relies on KI/DA as its access method. When combined with Prime's Forms Management System (FORMS), MIDAS becomes the basis for the development of transaction-oriented systems. MIDAS permits many users to access fixed- or variable-length records with locks specified at the data record to avoid concurrent usage conflicts. A single program can sequentially and randomly access a MIDAS file and do complete or partial file searches based on any combination of up to 20 keys with duplicates. Furthermore, each key can contain data in any format (e.g., single- or double-precision floating point, integer, ASCII, etc.) so conversion to a common format is unnecessary. MIDAS interacts with users and, by a series of questions and answers, describes, creates, maintains, and manipulates large, structured data files. MIDAS files are available via calls and standard READ/WRITE statements to application programs written in any of Prime's languages on 300, 350, 400, and 500 central processors with at least 128K bytes of main memory.

APPLICATION IMPLIMENTER'S VIEW OF THE INTERACTIVE COBOL DATA PROCESSING ENVIRONMENT



Database Management

Prime's Database Management System (DBMS) conforms to the CODASYL standards and provides integrated data that is accessible to all processes concurrently (especially multiple users at interactive terminals who are simultaneously updating and retrieving common files). A Schema Data Description Language (DDL) permits a global definition to be given to the data of a business or institution. This means that management and user departments can better understand not only the data, but their relationships to it. The view of the total database permitted to various classes of applications and users is further defined by the use of the COBOL and the FORTRAN subschema DDL. Application programs are created using either COBOL or FORTRAN high-level host languages and freely intermixing COBOL or FORTRAN Data Manipulation Language (DML) to reference the data.

Prime's DBMS allows the application designer to be independent of file structure concerns. Moreover, it facilitates centralized control to assure that all users operate on the same data, that the data has an identical meaning to all departments, and that all accesses are made securely. This is particularly helpful to businesses and institutions with complex data relationships and changing information requirements.

Prime's DBMS means reduced application programming expenses and shorter development times. Programmers concentrate on the logic of the application, not the details of data manipulation and file design, and sort/merge operations are significantly reduced.

Forms Management

Prime's Forms Management System (FORMS) is a set of software functions that are used to develop systems for interactive, multi-terminal, transaction processing. FORMS permits forms to be designed for a variety of CRT and hardcopy terminals using the Forms Description Language (FDL) with easy-to-use statements. Application programs are created using COBOL, FORTRAN, or PMA with standard READ/WRITE statements.

FORMS may be used separately or in conjunction with DBMS. The application program is unbound from the form's description, the type of terminal, and the database description until the program is run. This independence is similar in philosophy to the DBMS' facilities and results in similar savings in programming expense and time to develop new applications initially and in response to changing requirements.

Networking

The reach of the Prime File Management System can be extended to include files contained in other Prime 300, 350, 400, and 500 computers. This is accomplished automatically by PRIMENET, Prime's network software. PRIMENET makes it possible for a user or process on one Prime computer to access files on any other Prime computer in a network, without concern for any of the protocol details involved in managing the data transfer. Thus, regardless of where a file is actually located, PRIMENET makes it appear to the user that it is within the system to which he is connected. In addition, PRIMENET supports a wide range of network activities:

- A user can, with proper password identification, log in to any computer in a network from any terminal in the network.
- Users can run their programs on a remote system by logging into that system.
- Similarly, printed output can be spooled to a remote system.
- Processes running concurrently on different systems can communicate interactively with one another via special transmit and receive calls.
- An operator at a system terminal can send messages to a user on the local system or to all on-line network users.

HOW PRIMOS IV AND V WORK

PRIMOS, in conjunction with key hardware features in the Prime 350, 400, and 500 central processors, performs four major functions: activity scheduling, memory management, procedure sharing, and system protection.

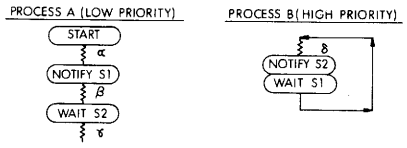
Activity Scheduling

PRIMOS IV and V automatically transfer the attention of the central processor from one activity (or process) to another with minimum overhead and complete protection. The key is a central processor feature called Process Exchange. As illustrated, this feature is a hardware dispatcher that manages the Ready List, a number of Wait Lists, Semaphores, and the Process Control Blocks containing detailed control and status information for each process.

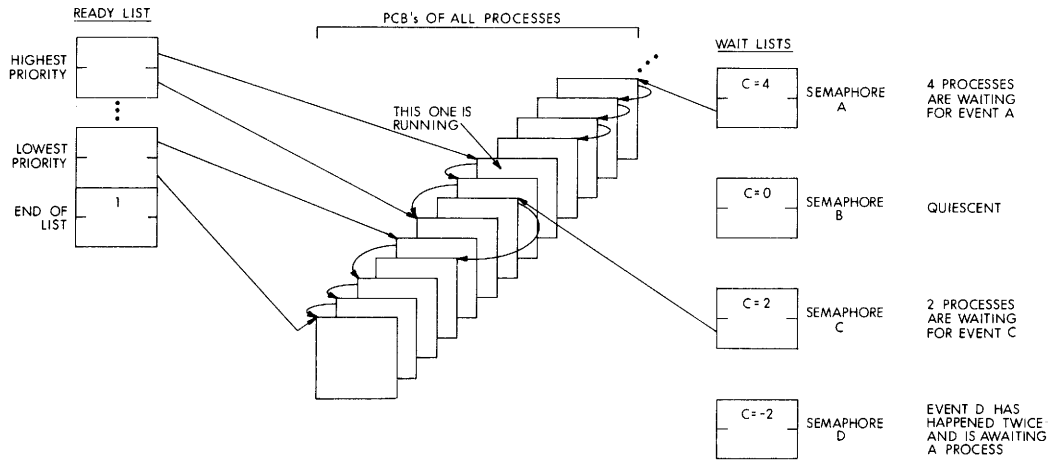
Process Exchange. Exchanges are caused asynchronously by hardware-generated interrupts, faults and checks, and synchronously by a process executing WAIT and NOTIFY instructions. These events activate the dispatcher to re-order the lists and get the highest priority process to run. This exchange takes 7 to 50 microseconds; without the hardware assist it would take 200 to 500 microseconds. The dispatcher also manages the processor's live registers, permitting sets of registers to be assigned to different processes. This reduces the need to save and restore the register's contents and speeds the process exchange.

Ready List. The Ready List identifies all processes that are ready to run. The list is ordered first by priorities and then chronologically. The highest priority process is the one running, the others are run in order of their priority as they appear on the Ready List.

PROCESS EXCHANGE EXAMPLE



EVENT	READY LIST	SEMAPHORE 1 (WORK FOR B TO DO) WAIT LIST	SEMAPHORE 1 (WORK FOR B TO DO) COUNT	SEMAPHORE 2 (B HAS DONE IT) WAIT LIST	SEMAPHORE 2 (B HAS DONE IT) COUNT
A STARTS	PCB A	PCB B	+1	–	0
(A RUNS α)					
A NOTIFIES S1	A PCB B	–	0	–	0
(B RUNS δ)					
B NOTIFIES S2	A PCB B	–	0	–	-1
B WAITS FOR S1	PCB A	PCB B	+1	–	-1
(A RUNS β)					
A WAITS FOR S2	PCB A	PCB B	+1	–	0
(A RUNS γ)					



Semaphores and Wait Lists. Each event that can cause an exchange is associated with a Semaphore (two words in memory) that keeps a count of the number of times the event has occurred without being serviced by a process, or a pointer to the processes awaiting the event (there is a Wait List for each Semaphore). The processes on a Semaphore's Wait List are ordered and serviced similar to the Ready List. Semaphores are associated with such events as: 'wait for a 1/3 second time slice,' 'disk read complete,' and 'character received from a terminal.'

To schedule a timeshared process, for example, an interrupt from the Real-Time Clock causes the hardware to notify the waiting time-keeping process, which awakens and exchanges run status with the previously running timeshared process. Once the time-keeping process begins running, it updates several counters including the 1/3 second counter. If the 1/3-second counter is full, indicating that 1/3 second has transpired since it was last reset, the timeshared process of next highest priority exchanges its ready status with the previous timeshared process through a series of WAIT and NOTIFY instructions. The time-keeping process now waits for the next Real-Time Clock interrupt and the highest priority process awakens and starts to run.

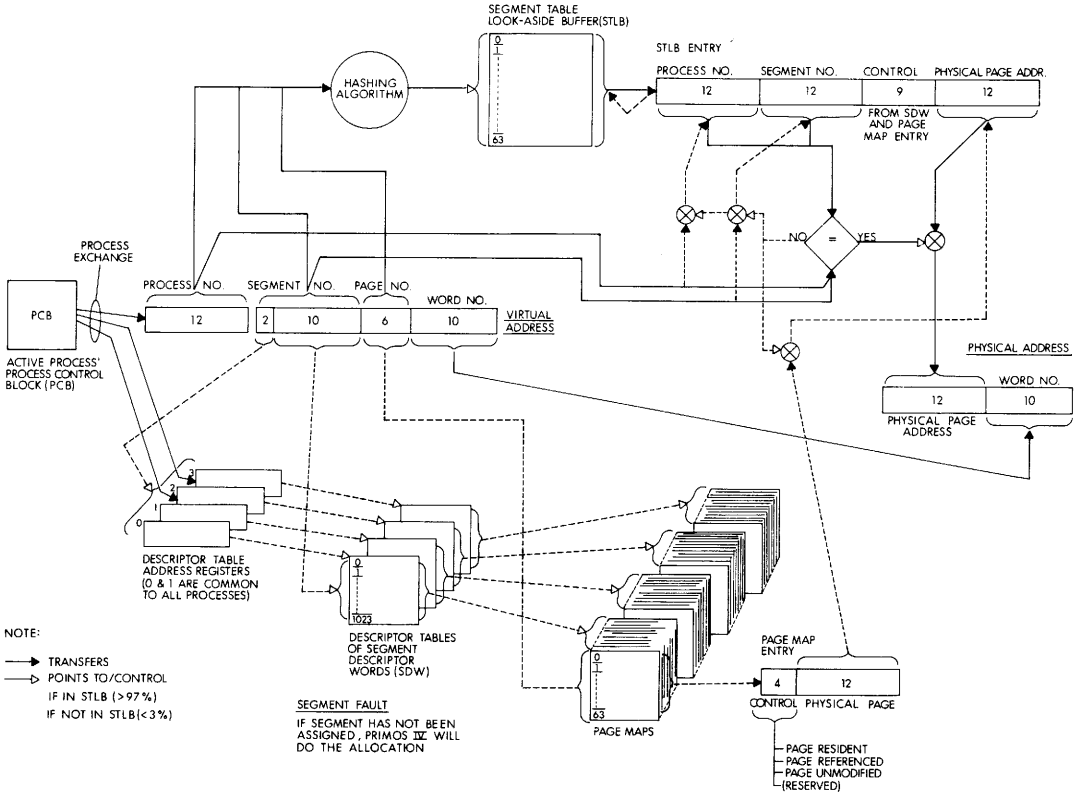
Memory Management

The user writes programs in any of Prime's languages, and is unaware of the memory management scheme as long as no single subroutine exceeds 128K bytes (one segment). If the entire program occupies one segment or less, the user may instruct the language translators to generate memory reference code that is compatible with a smaller Prime central processor. For larger programs, the user makes symbolic reference to subroutines and data arrays. The translators and loaders establish a correspondence between the symbolic reference and a virtual address.

Virtual-to-Physical Address Translation. The virtual address contains a segment number (one of 4,096), a page number (64 2K-byte pages/segment), and a word number (0 to 1,023; one 16 bit word = two bytes). The virtual address is translated into a physical address by a series of segment tables and page maps stored in main memory (see illustration). To speed translation, a Segment Table Lookaside Buffer (STLB) is used to hold physical page addresses for fast access by the processor. For a typical mix of operations, the required address information will be in the STLB more than 97% of the time and no overhead is added to the memory cycle time.

Demand Paging. When a process references a page that is not in main memory, a "missing page" fault occurs and is serviced by PRIMOS IV and V.

PRIMOS IV and V then replace the least-recently-used (LRU) page in memory with the referenced page. The next 'n' LRU pages are also freed in anticipation of subsequent demands. If the pages have been modified, they are first written back to disk before their memory space is overwritten. This demand paging algorithm has been designed to reduce the number of disk accesses.



Cache Memory. Two features of the central processor speed the effective memory cycle time: cache memory and interleaving. Cache memory is a high-speed bipolar 2 K-byte memory that retains the contents of the most recent memory references and is addressed on the basis of word number (the least significant 10 bits of a memory address). When referencing memory, the cache is interrogated first while the STLB is checked for correspondence. If correspondence exists, the word read from the cache is used; if not, the correspondence is established and the word is read from main memory and stored in the cache. When a word is written in main memory, it is also written in the cache. On average, more than 85% of memory references are found in the cache, reducing the effective memory access time. The access time is 80 nanoseconds from the cache and 600 nanoseconds from main memory.

Memory Interleaving. The processor interleaves memory references to pairs of sequentially addressed memory words. For example, if word 6 or 7 is read, both 6 and 7 are fetched simultaneously and the cache is updated. Interleaving speeds up sequential accesses and increases the cache hit rate.

Procedure Sharing

PRIMOS IV and V users can write procedures (or subroutines) that can be shared by other users. They can also use shareable procedures or databases that are part of PRIMOS or have been written by another user. The Prime 350, 400, and 500 hardware, together with PRIMOS servicing "fault" conditions, preserves the integrity of the shared resource and controls the access rights granted by its originator.

When a procedure is shared, it exists only once on disk and, when active, only once in main memory regardless of the number of processes using it. For example, if several users are concurrently writing source code with the Text Editor program, and one user references a part (or a page) of the Editor that has already been brought into memory for some other user, that user is automatically linked to that same copy rather than transferring another "private" copy from disk into main memory.

For commonly used procedures, sharing greatly improves memory utilization by reducing the number of disk transfers required to bring in the program or data referenced by a process but not found in main memory (missing page faults). Sharing procedures means PRIMOS is more responsive to its users.

Reentrance. For a procedure to be shared by concurrent processes (reentrant) and to be invoked by itself (recursive), its "pure" parts (those that do not change during execution) must be separated from those parts that do. This separation has already been done for PRIMOS IV and V's shareable resources and

is done automatically by the COBOL and FORTRAN translators when requested to do so by the user. Also the Prime Macro Assembler (PMA) makes it easy for the assembly language programmer to write and use shared procedures.

Procedure Call. References to a shared procedure are made either by user command (i.e., EDIT), by a language CALL statement in COBOL or FORTRAN, or a Procedure Call instruction in PMA. The language translators automatically generate code and variable areas that invoke and are compatible with the Prime 350, 400, and 500's Procedure Call (PCL) mechanism. PCL is a hardware function that does all the housekeeping required to transfer control from one procedure (or subroutine) to another.

PCL saves and restores the state of the calling procedure and initializes the state of the called procedure. It creates and maintains 'stack' areas for automatic data, and argument variables and linkage areas for static data. The PCL hardware performs these functions many times faster than would software equivalents. In addition, PCL guarantees that the access rights are followed, a function that cannot be performed by software. Thus, PCL and the concept of shared procedures makes PRIMOS IV and PRIMOS V responsive and secure computing utilities.

Security

A combination of hardware and software creates a secure timeshared computer utility. PRIMOS IV and V protect the utility, its files, and its shared procedures from unauthorized access. The Prime 350, 400, and 500 protect PRIMOS and users against unwanted intrusions or alterations by other users.

Log-in Procedure Protects Against Unauthorized Users. To gain access to the system, the user types a LOGIN command in which he identifies himself. This will activate a LOGIN program which cannot be defeated. The installation can optionally add to the LOGIN program whatever security locks are deemed necessary. The LOGIN program can activate accounting clocks that accumulate connect time, processor time, and disk transfer time until the user logs out (LOGOUT).

Passwords Protect User Files. The originator of a UFD (User File Directory) can define two passwords that must be satisfied when a user (or a process) tries to gain access to a file listed in the directory. (ATTACH UFDNAME PASSWORD.) One password is used by the owner, the other by a non-owner. For each file listed in the UFD, access rights can be defined by the owner; one set for the owner and one set for the non-owner. The rights granted depend on which password was used, and they control read (and execute), write, delete, and truncate accesses.

Segment Descriptor Words (SDW) Protect Shared Procedures. When a procedure is made ready for execution (loaded), links are established to bind that procedure to all other procedures referenced. To do this the user must be able to ATTACH to the UFD that lists the referenced procedures (as file names) and be granted read access rights to the named files. Although such rights may be denied one user by another, all users can gain read access to the system library (UFD LIB) and the command library (UFD CMDNCO).

When the link is first established, the procedure is assigned to a segment by PRIMOS. All subsequent references to the procedure are linked to the same segment. The segment is defined by a Segment Descriptor Word (SDW) listed in a Discriptor Table.

The SDW contains the access rights granted by the procedure's owner to other users: read, write, execute, and gate. PRIMOS creates the SDW and, because of its unique privilege level, is the only process that can modify or delete an SDW. These rights are automatically transferred as a code to the Segment Table Lookaside Buffer (STLB), where they are used to control every memory reference.

Central Processor Ring Structure Protects PRIMOS From User. The SDW defines access rights for each of three rings or levels of privilege: Ring 0 is the most privileged which, by design, has all access rights and the right to execute all instructions. Ring 3 is the least privileged and does not have the right to execute those instructions that can alter the system's mode of operation (such as HALT). Ring 1 has access privileges between those of Ring 0 and 3. PRIMOS IV and V enjoy Ring 0 privileges; timeshared user processes, Ring 3. The ring number is a part of each calculated memory address.

The security provided by a ring structure can best be explained when one considers that an address of an instruction to be fetched may be defined for a different ring than the address of the data (or arguments) referenced by the instruction. For example, if a user's process (Ring 3) calls a shared procedure that is part of PRIMOS (Ring 0), the procedure's instructions are associated with Ring 0 and the data, or arguments (in stacks and linkage areas), with Ring 3. When such a shared procedure is called (using Procedure Call), reference is made to its Entry Control Block (ECB) that contains information required to initialize the procedure and start it running. The ECB is located in a segment whose SDW defines the access rights for a Ring 3 caller. If gate rights have been granted, the Procedure Call mechanism will strengthen the active ring number from 3 to 0 and allow the Ring 0 procedure to execute. Otherwise, a fault condition occurs

and the user's process cannot execute the procedure. This would happen for those procedures defined by PRIMOS to be for its own use and not to be shared by a user's process

Weakening is an important function of the Procedure Call mechanism. The maximum of the called and call ing ring numbers is taken (max (0,3) = 3) and the resulting weakened ring number is inserted into each argument (or pointer) transferred by the Procedure Call to the called procedure. When the called procedure is running and makes a reference to a memory location pointed to by an argument, it is granted only the weakened Ring 3 privileges defined by the referenced SDW.

If a Ring 3 process references a location in the common address space, it would be granted the Ring 3 privileges previously defined by the 'owner' of the segment (usually PRIMOS) and would be denied those rights deemed by the owner to be unwise or harmful. Similarly, if the called Ring 0 procedure references a location in the private address space of the user's process, it would be granted only those privileges defined by the user of Ring 3.

The Prime 350, 400, and 500's segmentation and ring structure described above is the hardware that permits PRIMOS IV and V to offer a shared computing utility with complete and automatic access protection. The user need only define access rights and passwords, and then use passwords as required.

HOW PRIMOS IV AND V ARE ORDERED

PRIMOS IV and V are priced software products. Each may be ordered separately to run on suitable Prime 350, 400, or 500 configurations. Several of the software sub-systems supported by PRIMOS IV and V and described in this bulletin are priced separately and are also supported by PRIMOS III on Prime 300s with at least 128KB of main memory. Products that must be ordered individually are the COBOL compiler, BASIC/VM, RPG translator, each of the RJE emulators, and the PRIMENET network package. For detailed ordering information and availability schedules, please contact your local Prime Sales Representative or Prime's headquarters. Specifications may change as design improvements are introduced.

PRIME

PR1ME Computer, Inc. 40 Walnut St., Wellesley Hills, MA 02181