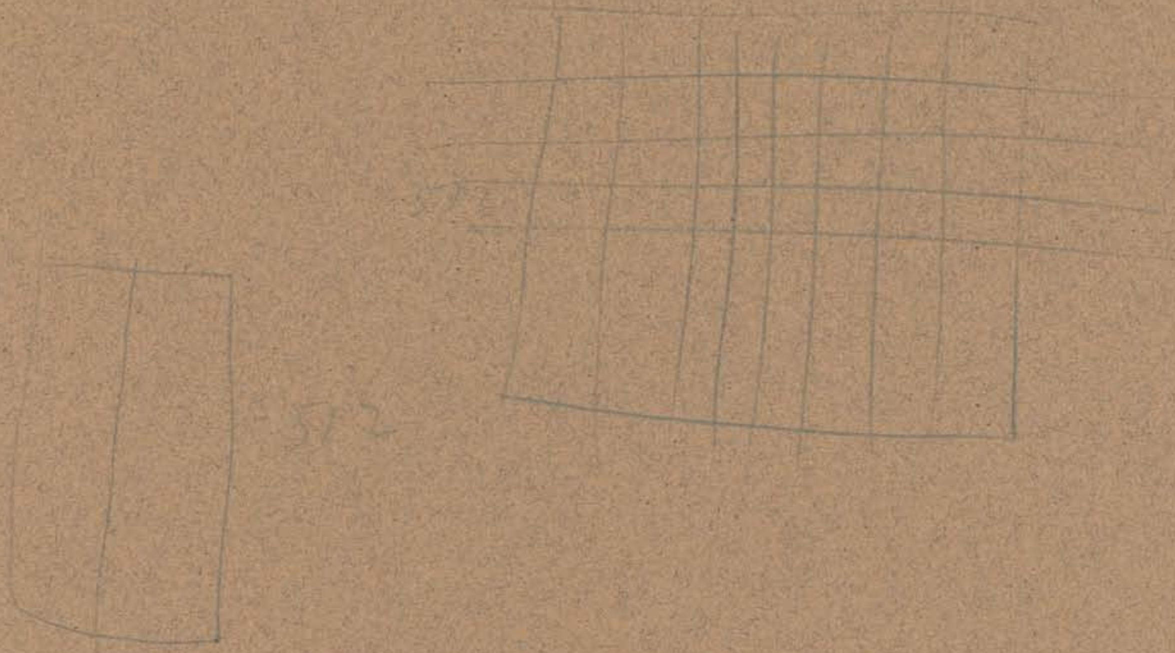
PRIME

Prime 200 small computer



Why Prime

There are some 60 small computers on the market – each claimed to be better and more wonderful than all the others. Unfortunately, most of the features that are supposed to demonstrate each system's unique superiority have little to do with what most people buy computers for, namely, getting a job done. Not the warm feeling of having a computer that goes 100 nanoseconds faster than any other. Not the sense of well-being that comes from knowing that your computer moves information around using only one bus, or two busses, or a trolley car.

Getting a job done faster, easier and more reliably is what the PRIME 200 is all about. To make that possible we came up with a design concept that you won't find on any of the other computers on the market today. We defined and perfected the software first, and then we developed the hardware that would maximize the software's performance. For the first time, hardware has been designed to support software. The result is a system that enables you to develop and execute software more efficiently and at less cost than ever before possible on a small computer. And if you agree that software is the key to getting things done, then you owe it to yourself to seriously evaluate the PRIME 200. The following pages will get you started:

pages 2 and 3 – design philosophy
pages 4, 5, 6 and 7 – software overview
pages 8, 9 and 10 – hardware
page 11 – performance examples
page 12 – product reliability

If you really don't want to read all the details and just want the specs, flip to the last page where you'll find them summarized on a handy tear-out card.

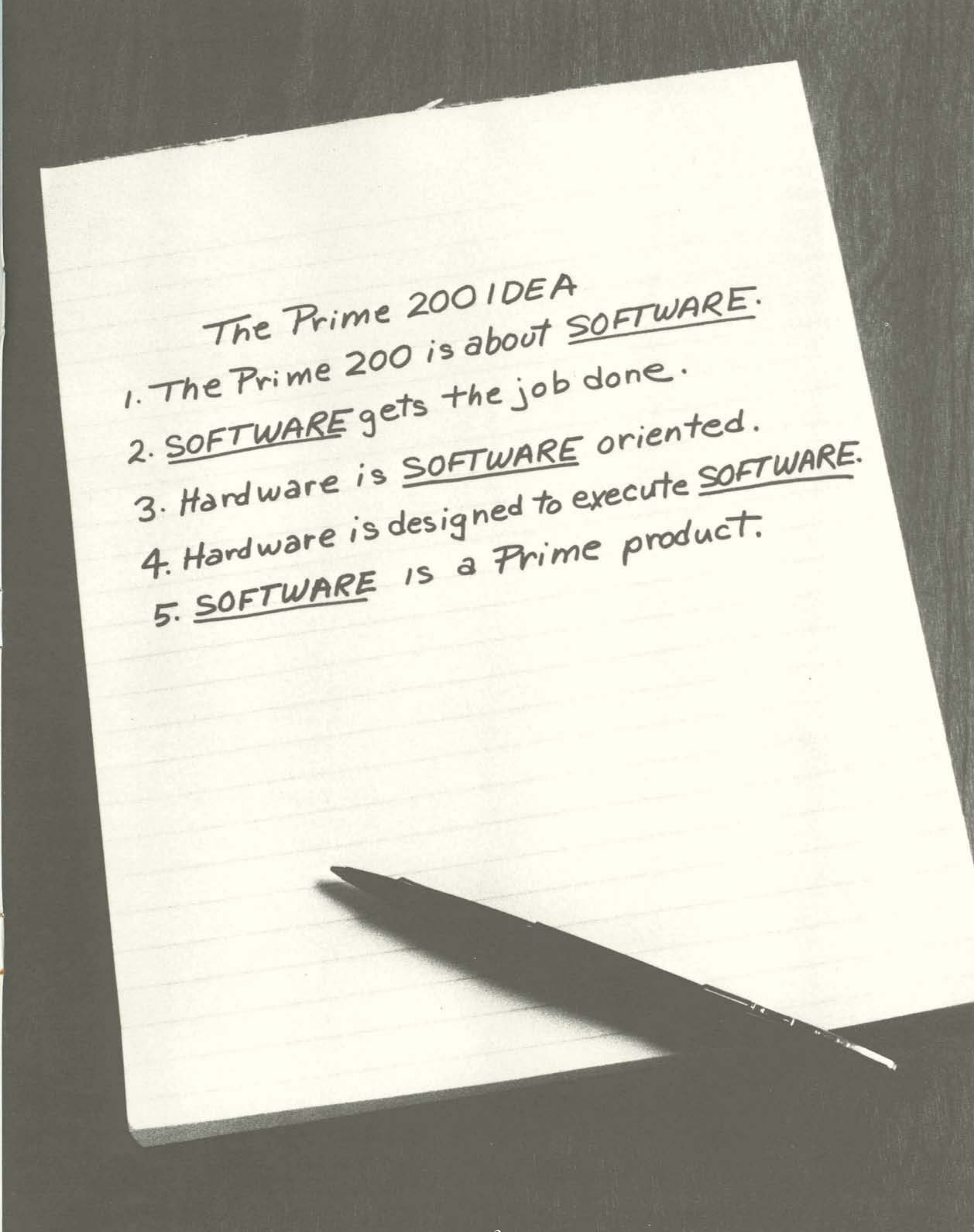
The Prime 200 Idea

Develop the software first, then build it a nice place to live. That, in a nutshell, is the basic design philosophy behind the PRIME 200 and what sets it apart from other small computers. Instead of the usual "technology in search of a purpose" approach to computer design, PRIME defined the purpose first and then developed the technology that would best achieve it. The purpose was to improve the one factor that has been wasting more money and limiting the performance of more systems than any other. Software.

The PRIME 200 was designed so users could develop and execute software with less blood, sweat and tears than ever before possible on a small computer. At a purchase price that gives the PRIME 200 a substantial price/performance edge over all major competitors. While other manufacturers belabor the pros and cons of various architectures, attempt to resolve the battle of the busses, and push logic a few nanoseconds closer to the speed of light, PRIME is devoting its efforts to providing a system that dramatically improves the economics of small computer usage.

In pursuing that objective, PRIME developed a system that offers the following major features:

- A FORTRAN compiler so efficient that it allows FORTRAN to be used as the PRIME systems programming language.
- An interactive Disk Operating System featuring sophisticated file handling techniques.
- A Real Time Operating System available in both memory/disk and memory only configurations.
- 100% microprogrammed logic, featuring a 64-bit wide microprogram word for optimum speed and efficiency.
- 750 nanosecond MOS memory, expandable to 32K words.
- Plug-in packaging of all mainframe components: processor, memory, interfaces, backplane, control panel, and power supply.
- The most comprehensive error detection system ever implemented in a small computer.



The Prime 200 IDEA

1. The Prime 200 is about SOFTWARE.
2. SOFTWARE gets the job done.
3. Hardware is SOFTWARE oriented.
4. Hardware is designed to execute SOFTWARE.
5. SOFTWARE is a Prime product.

First, The Software

PRIME programming systems offer users an operating efficiency and a level of functional capability never before available with a small computer – regardless of price. How this is possible can be better appreciated by understanding some of the basic design ground rules that molded the PRIME software system.

1. *One Software System For All Users.* All programs written for the PRIME 200 utilize a common set of system software elements. Programs developed with stand-alone system software are completely compatible with those developed in an operating system environment. Since there is no distinction between "basic" and "advanced" system software, there are no one-of-a-kind software packages to interfere with compatible growth into larger systems.

2. *File Compatibility Between Program Development and Real Time Environments.* A uniform and consistent file system supported by *all* PRIME software is the key to simple and manageable transition from program development to program execution. For example, it allows the user to generate real-time systems under Disk Operating System (DOS) control, load data bases under DOS control, and then execute programs under Real Time Operating System (RTOS) control, using exactly the same data bases. Conversely, the data collected under RTOS can be analyzed and maintained under DOS.

3. *High Level Language For System Software.* PRIME's FORTRAN compiler is extremely efficient. So efficient, in fact, that it is now practical to write systems software in FORTRAN. Experience has shown that a higher level language generally allows better algorithms to be developed to solve specific problems because the programmer can devote more attention to the problems and less to worrying about what the computer will do. As an indication of the efficiency of this approach, note that the entire Disk Operating System is written in FORTRAN.

4. *Software Should Take Maximum Advantage Of All Hardware Features.* The "software-first" design of the PRIME 200 means that not only are hardware features software-supported, but supported in the most efficient way possible. For example, when a hardware option is added to a users system, the FORTRAN compiler automatically implements it in executable code rather than wasting time by continuing to call a subroutine.

Operating Systems

Disk Operating System (DOS)

The Disk Operating System is an interactive controller and file handling system ideally suited to the single-user, program development and execution environment. It provides an extensive array of features to simplify program development and maintenance. For example:

- System file management features include multiple-level file directories and multiple volume control and file access methods. These features simplify the creation, deletion and updating of source, object, and data files.
- It functions as a batch processing system, providing automatic job and data stream routing by storing command sequences on a disk file.
- Numerous disk file access disciplines and built-in file integrity checks are provided. Files are constructed with forward and backward pointers so that accidental loss of one file will not snowball into a loss of multiple files. DOS quickly regenerates the record availability table in the event that a file is destroyed.
- Files are addressed by name rather than by absolute location.
- DOS supports compressed ASCII files. Typically this provides a 5:1 character reduction for assembly language files, and a 4:1 reduction for FORTRAN files.

Real Time Operating System (RTOS)

RTOS is a compact, multiprogramming operating system that schedules the computer's three basic resources – processing power, main memory, and mass memory – to control the execution of application programs in a real-time environment. It provides interrupt handling, multiprogram scheduling, simultaneous input/output and general supervisory functions.

- RTOS is available in two versions: a memory-only system for fast response applications and a memory/disk system for multitask applications involving resource sharing, memory mapping and large data bases.
- RTOS and DOS use the same file management system, allowing the user to maintain files off-line. The Disk Operating System can run under control of the RTOS, and, conversely, real-time systems can be developed under DOS control.
- RTOS is supported by a complete set of utility, support, debugging, and I/O device test programs.

Stand-Alone System

Stand-alone systems are provided with exactly the same translators, loaders and I/O controllers as larger operating system configurations. PRIME's Stand-Alone System includes the following major elements:

- FORTRAN IV Compiler
- Macro Assembler
- Desectorizing Link-Loader
- Support Library including FORTRAN functions, math library and I/O drivers
- Text Editor
- Debugging Package
- Input/Output Control System

Language Processors

FORTRAN IV

FORTRAN is the PRIME systems programming language. The compiler executes in one pass and produces highly optimized code. The source language is ANSI extended FORTRAN plus other extensions derived from the proposed Instrument Society of America FORTRAN for control applications. Significant extensions include: embedded logical functions such as AND, OR, NOT, and XOR; runtime trace; octal constants; and an extended FORTRAN library.

Macro Assembler

The PRIME Macro Assembler represents a major improvement in symbolic language processor design. In addition to the coding simplification features available with many present-day macro assemblers, it offers the user a simple, yet powerful, means of generating application-oriented macro statements that can significantly simplify application programming. It is now possible to develop unambiguous, easy-to-use macros that require no additional documentation and that can be easily modified by anyone familiar with the application.

In addition to its expanded macro capability, the Macro Assembler offers all of the features you would expect to find in a well designed symbolic assembler, including pseudo-ops for assembly, listing, and loader control; symbol and data definition; storage allocation; program linking; and conditional assembly.

Desectorizing Link-Loader

Relocatable or absolute program modules from any PRIME translator may be loaded, linked and bound with the Desectoring Link-Loader. Cross-sector references are automatically desectorized, and literals, links, and temporaries are optimized. A major feature of the Link-Loader is that the loader tables are preserved to provide for symbolic debugging. Furthermore, the Disk Operating System can be used to save and restore partial loads.

Edit and Debug Resources

The Text Editor is a full context editor providing such features as line-by-line and character-by-character editing, and automatic execution of string buffers for multiple changes of the same text throughout a program. A bulk media converter provides editing and transcription for large volumes of data.

Debugging aids include an interactive trace routine for examining. patching, tracing, etc. Symbolic references to memory are permitted through the loader symbol table.

Input Output Control System (IOCS)

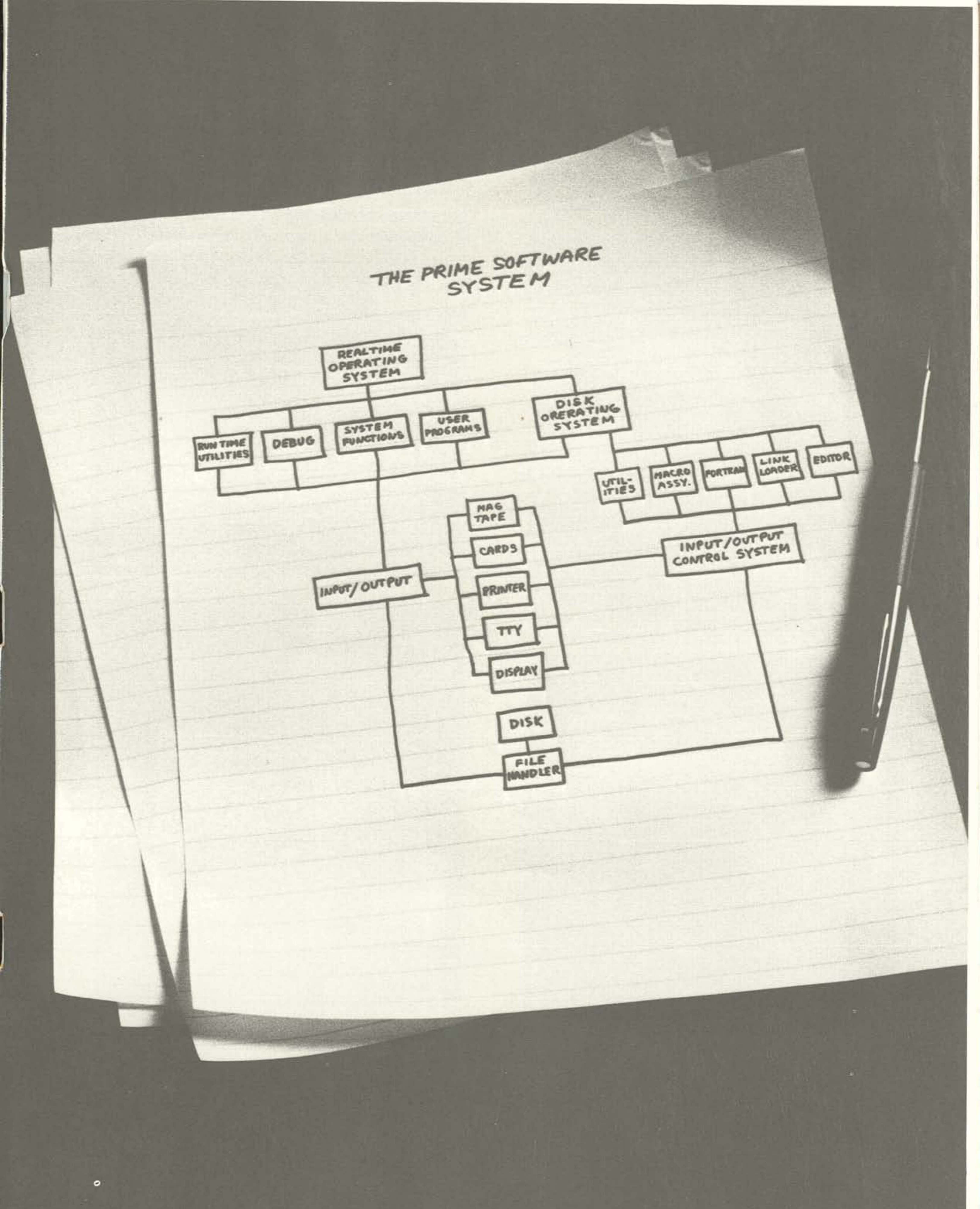
The IOCS is a system of stream control routines and device drivers which provide device independence for user programs and centralized I/O control for all system software. A unique IOCS feature is the provision for source file editing and merging.

Support Library

The PRIME support library includes I/O conversion and driver packages, and standard and extended math functions. The FORTRAN support package provides formatted I/O, encode and decode statement translators, and list processing functions. The extended I/O driver library handles interrupt driven and queued-request device handling.

Verification and Maintenance

Thorough verification routines are provided for the PRIME 200 and each of its peripheral devices. Each I/O controller can be operated in a diagnostic mode to permit programmed testing of I/O functions. Special processor features, including non-functional tests and microdiagnostics, greatly enhance the fault diagnosis capabilities of the various test and maintenance routines.


THE PRIME SOFTWARE SYSTEM
REALTIME OPERATING SYSTEM
RUN TIME UTILITIES
DEBUG
SYSTEM FUNCTIONS
USER PROGRAMS
DISK ORERATING SYSTEM
UTIL-ITIES
MACRO ASSY.
FORTRAN
LINK LOADER
EDITOR
MAG TAPE
CARDS
PRINTER
TTY
DISPLAY
INPUT/OUTPUT
INPUT/OUTPUT CONTROL SYSTEM
DISK
FILE HANDLER

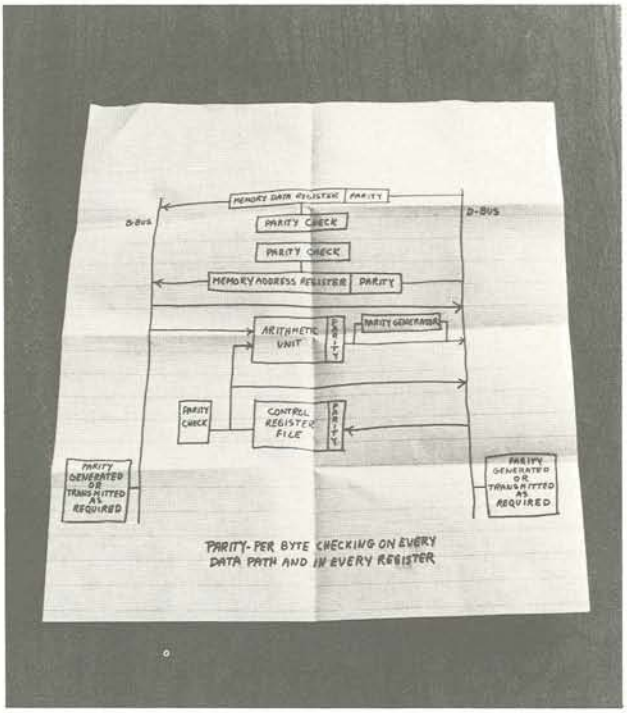
Then, The Hardware

The hardware in a PRIME 200 system has three goals in life: give the software a convenient place to do its exercises; interface simply and economically to a wide range of real-time environments; and do it all with unquestioned reliability.

Basic Specs

The PRIME 200 is a fully parallel, 100% microprogrammed, 16-bit computer. Its MOS memory has a 750 nanosecond cycle time and is expandable in 8K word increments to 32K words. The instruction complement includes 118 instructions, 15 of which are memory reference instructions for fast, efficient data handling without time-consuming register housekeeping (e.g., a memory-to-register add requires only a single word instruction).

The processor includes such standard features as direct, indexed, and multilevel indirect addressing in both sectored and relative modes; 64-level, vectored priority interrupt system; fully recursive, and push-pop stack processing capabilities; byte parity on all data paths; eight-channel DMA; asynchronous serial communication interface; and programmer's control panel.

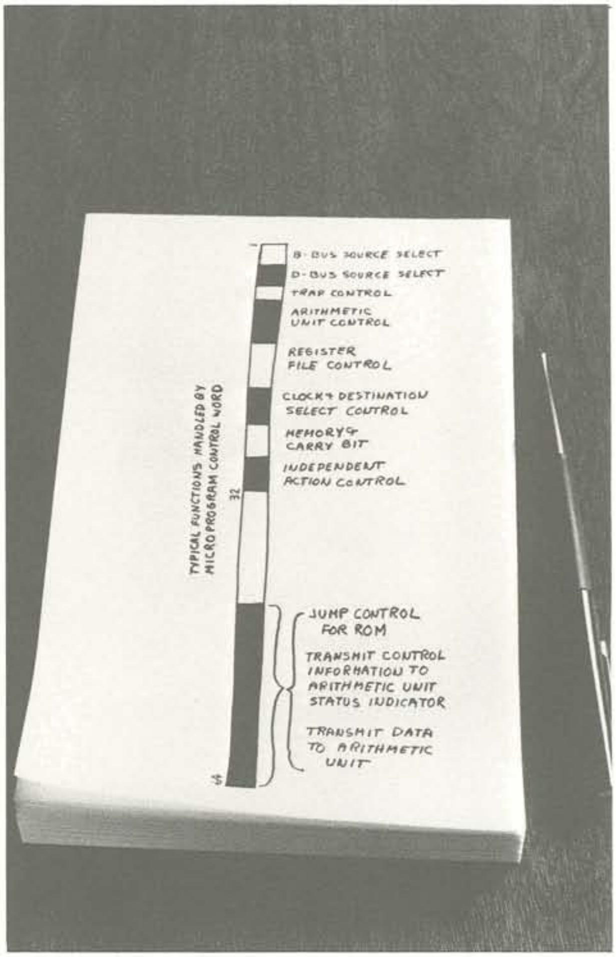


The computer is housed in a 10½ inch high chassis which provides ten slots for printed circuit boards. The entire central processor is fabricated on a single board and can be inserted in any slot position. Similarly, a complete 8K word memory module is contained on a single board. The remaining eight slots are used for additional memory, I/O controllers, user-designed interfaces, etc.

The Computer in more detail

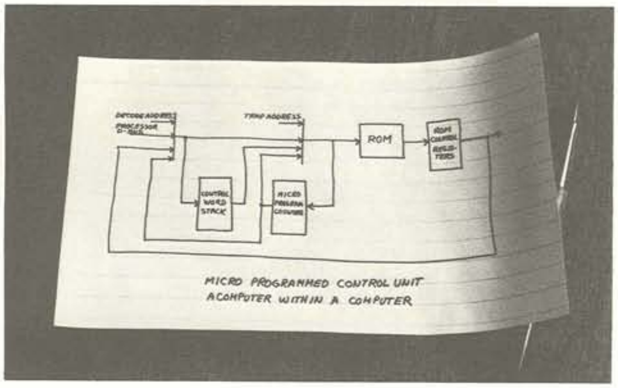
Microprogramming

The PRIME 200's instruction complement is implemented in a microprogrammed read-only memory. This control technique, which can be thought of as a special-purpose microprocessor within a general purpose computer, utilizes "horizontal" microprogramming for optimum speed and efficiency. A 64-bit microinstruction format allows a large number of functions to be controlled by a single microword. This minimizes the total number of microwords, and thus the total time needed to decode and execute the software instructions.


TYPICAL FUNCTIONS HANDLED BY MICROPROGRAM CONTROL WORD
1
32
64
B-BUS SOURCE SELECT
D-BUS SOURCE SELECT
TRAP CONTROL
ARITHMETIC UNIT CONTROL
REGISTER FILE CONTROL
CLOCK + DESTINATION SELECT CONTROL
MEMORY + CARRY BIT
INDEPENDENT ACTION CONTROL
JUMP CONTROL FOR ROM
TRANSMIT CONTROL INFORMATION TO ARITHMETIC UNIT STATUS INDICATOR
TRANSMIT DATA TO ARITHMETIC UNIT

Since the microprocessor's operation is overlapped with main memory, the PRIME 200 is ideally suited to executing memory reference instructions (which in typical applications occur 70 to 80% of the time).

In addition to optimizing the computer's performance, microprogramming also provides an open-ended design in which additional instructions, various diagnostic capabilities, and extensions to standard features can be easily implemented.



Error Detection

The reliability of data within the PRIME 200 is monitored by the most comprehensive parity checking scheme ever implemented in a small computer. Every byte of data on every data path in the processor, on the backplane, in every register, in every memory location, and in all I/O device controllers is checked for parity. In short, any time information moves anywhere in the computer, its parity is checked on a byte-by-byte basis to assure its reliability. As a result, the user is instantly aware of any condition that could cause erroneous processing of data and appropriate corrective action can be taken under software control.

In the event that a system failure has occurred, microdiagnostics are available to trace the fault to its functional level.

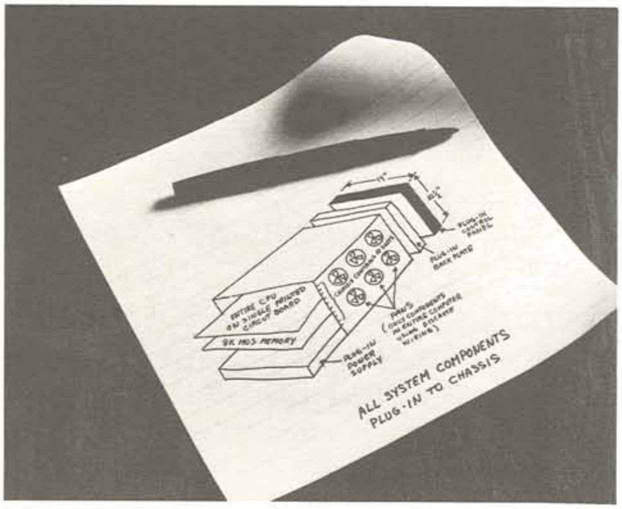
MOS Memory

Semiconductor memory technology offers important price/performance advantages that make it the logical choice for the PRIME 200. The conservatively rated MOS memory in the PRIME 200 gives the user a combination of high-speed (750 nanosecond cycle time), high density packaging (complete 8K memory on a single board), and low power consumption (full 32K memory operates with system's basic power supply).

Packaging

Unique physical packaging and mechanical design features contribute to the computer's outstanding reliability and ease of use. Basically, a PRIME 200 computer consists of just four elements: a chassis with 10 slots for mounting processor memory and various controller circuit boards, a control panel, a controlled impedance backplane for interconnecting the circuit boards, and a power supply. There are no reserved slots in the chassis; any circuit board can be inserted in any slot. To simplify servicing, the power supply, backplane and control panel are designed as plug-in modules that can be easily removed and replaced if necessary.

Printed circuit construction is used through the computer. In fact, with the exception of the wires feeding power to the cooling fans, there isn't an inch of wire in the entire computer.


PLUG-IN CONTROL PANEL
PLUG-IN BACK PLATE
ENTIRE CPU ON SINGLE PRINTED CIRCUIT BOARD
8K MOS MEMORY
FANS
PLUG-IN POWER SUPPLY
ALL SYSTEM COMPONENTS PLUG-IN TO CHASSIS

Interfacing

Simple, low-cost interfacing provided by PRIME's input/output, communications, and analog/digital interfaces is further enhanced by two general-purpose interface boards. One board combines standard PRIME bus interfacing logic with a wire-wrap pin section, giving the user a simple and low-cost means of fabricating specially designed controllers. The other is a compatible interface board which provides direct hardware connection between the PRIME 200 and any Series-16 controllers.

Input/Output

As a standard feature, the PRIME 200 contains a programmable, eight-channel, direct memory access (DMA) processor to facilitate block transfers of data directly between the memory and various device controllers. Any channel may be assigned to any device under program control; there are no hardwired channel assignments. Any number of device controllers can be attached to the PRIME 200 and any eight can operate in the direct memory access mode simultaneously.

In addition to DMA block transfers, the PRIME 200 also handles direct single-word transfers between device controllers and the main arithmetic register in the processor.

Interrupts

Interrupts can be handled via a standard party line interrupt technique which causes a program jump to a fixed location in memory, or they can be handled by a 64-level vectored priority interrupt system which allows interrupts to be vectored to any memory locations the user desires. All interrupts can be armed and disarmed, singularly or in groups, under program control.

Peripherals

From an ASR 33 Teletype to high-speed, fixed and moving head disks, PRIME supplies a comprehensive and expanding range of peripherals for data entry and storage. To support these devices (summarized in following table) PRIME has developed a series of powerful device controllers. For user convenience, the teletype, paper tape reader, and paper tape punch controllers as will as real-time clock are all packaged on a single board. A moving head disk controller supports up to four disks and a fixed-head controller supports two disks. Each disk controller occupies a single board and each provides programmable vectored interrupt capabilities, programmed I/O and DMA. Within the computer system, a TTY interface handles EIA compatible devices with speeds up to 9600 baud.

Fixed Head Storage File

Provides random access to 128Kor 256K words with an average access time of 8.7 milliseconds and a transfer rate of 4 microseconds per word.

Moving Head Disk Cartridge File

Handles standard 2315 cartridges and provides random access to 1.5M or 3.0M words with an average seek time of 70 milliseconds and transfer rate of 10 microseconds per word.

Paper Tape Reader and Punch

Reads 8 channel fanfold tape at 200 cps, and punches fanfold tape at 75 cps.

Teletypes

Model 33 ASR, Model 33 KSR, and Model 35 ASR

A closer look at performance

Software is the key to the PRIME 200's performance. Raw computing speed or isolated hardware instruction times tell only a part of the story. It's how efficiently the software uses these resources to get your job done that really counts. The following examples illustrate a few of the many factors that contribute to the PRIME 200's outstanding on-the-job performance.

Example 1: Extremely Flexible Macro Assembler

The advanced design of the PRIME Macro Assembler permits macro calls to be defined for optimum user convenience. For instance, any of the following calls can be used for the MOVE macro.

```
Macro Call
MOVE 1 WORD FROM A TO B
MOVE 1 WORD TO B FROM A
MOVE FROM A TO B, 1 WORD
MOVE INTO B, 1 WORD FROM A
```

```
Macro Definition
MOVE MAC WORD, WORDS, (FROM);
            = 2, (TO) = 3, (INTO) = 3
          IF (< 1 >.EQ.1) LDA< 2 >;
            STA <3 >: GO TO & X
          LDX< 1 >, 1
          LDA< 2> – 1, 1
          STA <3 >– 1, 1
          IRX
          JMP * – 5
&X        ENDM
```

Example 2: Single-word, memory reference instructions for greater efficiency and programming ease

Compare the number of instructions needed (and thus the total execution time) to ADD A, B, and C on the PRIME 200 vs. the number needed using general register logic common to several competitive systems.

```
PRIME 200              Competitive Systems
(using memory          (using general
reference logic)       register logic)
STA  TEMP 1            STA  1, TEMP 1
LDA A                  STA  2, TEMP 2
ADD B                  LDA 1, A
ADD C                  LDA 2, B
STA  RESULT            ADD 1, 2
LDA  TEMP 1            LDA 1, C
JMP  BACK              ADD 1, 2
                       STA  2, RESULT
                       LDA 1, TEMP 1
                       LDA 2, TEMP 2
                       JMP  BACK
```

Example 3: FORTRAN Compiler Generates Optimized Code

The following expanded output listing highlights the power and efficiency of PRIME's FORTRAN. Line 117 shows the direct use of index registers. FORTRAN statement 250 illustrates the efficiency of generated code for a typical DO loop. FORTRAN statement 260 shows the use of intrinsic logical functions.

```
250  DO 260 I = ONE, THREE
     LINK ♦250
112  LDA  ONE
113  STA  I
     CUFD(I) = XNAM(I)
114  LDA  I
115  ADD  XNAM
116  STA  000000
117  LDA  XNAM    , 1
120  LDX  I       , 1
121  STA  CUFD    , 1
     IF (TSET) HOMUFD(I) = CUFD(I)
122  LDA* TSET
123  SNZ
124  JMP  000000
125  LDA  CUFD    , 1
126  STA  HOMUFD , 1
     LINK 000124
260  CONTINUE
127  LDA  I
130  AOA
131  CAS  THREE
132  JMP  000135
133  JMP  000113
134  JMP  000113
270  DSKRAT(IW+1) = XOR(DSKRAT(IW+1), TUP(IB) )
135  LDA  IW
136  ADD = 'DAC  DSKRAT'
137  STA  T$1000
140  LDA  IB
141  ADD ='DAC  TUP–I'
142  STA  T$1001
143  LDA* T$1000
144  ERA* T$1001
145  STA* T$1000
     CHRAT =.TRUE.
146  LDA  ='000001
147  STA  CHRAT
RETURN
```

Reliability, Maintenance, Service

One of our goals in designing the PRIME 200 was to make it the most reliable computer available for a broad range of application environments. A "turn it on and forget it" kind of reliability that's absolutely essential in real-time applications.

The PRIME 200's reliability starts with the basic design of the computer itself. The system's architecture is based on proven design concepts which have demonstrated their reliability through millions of hours of usage. There are no hidden surprises and no unpredictable performance parameters for our customers to uncover after their machines are installed. Closely related to the basic design is its implementation. Reliability is built-in to the PRIME 200 through conservative, worst case component specification; rigid, incoming component quality control; big board construction; controlled impedance backplane and circuit boards; and a packaging system that eliminates all discrete wiring.

The next step is to insure the integrity of the computer's operation through a system of error and fault detection features. To this end the PRIME 200 provides such features as byte parity checking on every data path and memory location in the computer; completely interchangeable circuit boards, allowing boards to be swapped among slot locations to help localize hardware and software faults; and microdiagnostics for tracing hardware faults to their functional level.

Finally, once a faulty component has been identified, the user is given several options for repairing it. In those applications where even a few minutes of down time can be disastrous, the PRIME 200's modular packaging permits any component to be quickly replaced simply by unplugging it and plugging in a spare. A faulty processor, memory module or power supply can be replaced by the user in less than a minute.

In cases where time is less critical or it is not practical to stock spares, the user can have repairs made at one of PRIME's strategically located repair depots, or he can use our express service arrangement with major airlines for same-day delivery of parts to any major city in the country. And, of course, trained field service engineers are available for on-site maintenance.

Fully supported systems

For the end-user who wants a fully operational system installed, checked out and ready for application software development, PRIME offers a unique approach to packaged system support.

A series of fully supported systems provide all of the individual benefits of PRIME hardware and software plus expert installation, testing, documentation and demonstration of system operating parameters at the user's site by PRIME system support specialists. By purchasing a fully supported system, the user is free to concentrate on his particular application requirements while PRIME assumes full responsibility for system installation. And we guarantee the performance of the system, both hardware and software, for a full six months after the system is installed.

PRIME means first. First in quality. First in reliability. First in value. The name fits our objectives perfectly.

PRIME200 Hardware Specifications

The Central Processor

Parity per byte on all internal transfers

MSI/TTL logic throughout.

16-bit word length

118 instructions, including 15 memory reference instructions.

Integral Direct Memory Access processor,

Direct, indexed and indirect (in both relative and sectored modes) addressing.

Power monitor, power failure interrupt option.

Push-pop and recursive stack processing.

32 addressable hardware registers.

Double precision arithmetic.

Hardware multiply/divide.

Automatic program loading.

Real-time clock option.

100% microprogrammed logic.

64-level vectored priority interrupt system.

Unimplemented instruction vectored trap.

MOS Memory

750 nanosecond cycle time.

Parity per byte.

Modularly expandable to 32K words.

Battery standby power option.

Input/Output

Byte, word and multiword transfers.

I/O data rates up to 1 megaword.

I/O bus handles 64 devices

EIA compatible, serial interface handles devices with speeds up to 9600 baud

Byte parity on all I/O transfers.

Peripherals

Fixed-head storage file provides random access to 128K or 256K words with an average access time of 8.7 milliseconds and a transfer rate of 4 microseconds per word.

Moving-head disk cartridge file handles standard 2315 cartridges and provides random access to 1.5M or 3.0M words with an average seek time of 70 milliseconds and a transfer rate of 10 microseconds per word.

Paper tape reader and punch reads eight-channel, fan-fold tape at 200 cps and punches fan-fold tape at 75 cps.

Model 33 ASR, 33KSR and 35 ASR teletypes.

Packaging

Big-board packaging permits entire central processor to be fabricated on a single circuit board.

A complete 8K MOS memory module is contained on a single circuit board.

The PRIME 200 chassis provides 10 slots for mounting the processor, memory and controller circuit boards.

There are no reserved slots in the chassis; any board can be inserted in any slot.

Modular, plug-in packaging is used throughout the PRIME 200. Plug-in components include the processor, memory modules, interfaces, controlled impedance backplane, control panel, and power supply.

Physical and Electrical Requirements

THE PRIME 200 chassis measures 10½" H x 19" W x 20" D.

A chassis with power supply, processor, one memory and one I/O controller board weighs 70 pounds.

The system operates on 117 VAC, 50-60 cycle (230 VAC optional).

Prime Hardware Configurator.

Analog to Digital

Analog to digital differential input subsystem.

High-level, solid-state differential signed multiplexer.

Handles up to 64 channels.

11 bit plus sign digitization.

Programmable input range of 0 to ±1, ±2, ±5, ±10 volts full scale.

Operates in programmed I/O or DMA modes.

INSTRUCTION COMPLEMENT

Instruction Complement

Register Operate

CRA	Clear A
CRB	Clear B
CRL	Clear Long (A and B)
LDA	Load A
STA	Store A
LDX	Load Index
STX	Store Index
IMA	Interchange Memory and A
IAB	Interchange A and B
XCA	Transfer A to B and Clear A
XCB	Transfer B to A and Clear B
CEA	Compute Effective Address

Arithmetic

ADD	Add Memory to A
SUB	Subtract Memory from A
AOA	Add One to A
A2A	Add Two to A
SOA	Subtract One from A
S2A	Subtract Two from A
ACA	Add C-Bit to A
CSA	Copy Sign to C-Bit, set Sign Plus
SSP	Set Sign Plus
SSM	Set Sign Minus
CHS	Change Sign
TCA	Two's Complement A
PIM	Position for Integer Multiply
PID	Position for Integer Divide
MPY	Multiply
DIV	Divide
DLD	Double Precision Load
DST	Double Precision Store
DAD	Double Precision Add
DSB	Double Precision Subtract

Input/Output

OCP	Output Control Pulse
SKS	Skip if Set
INA	Input to A
OTA	Output from A
ISI	Input Serial Interface to A
OSI	Output Serial Interface from A
SMK	Set Interrupt Mask

Control

HLT	Halt
NOP	No Operation
SCB	Set C-Bit
RCB	Reset C-Bit
XFA	Index from A
XFB	Index from B
XFS	Index from S
XFX	Index from X
SGL	Enter Single Precision Mode
DBL	Enter Double Precision Mode
EMCM	Enter Machine Check Mode
LMCM	Leave Machine Check Mode
RMC	Reset Machine Check
E16S	Enter 16K Sector Addressing Mode
E32S	Enter 32K Sector Addressing Mode
E32R	Enter 32K Relative Addressing Mode
SVC	Supervisor Call
DIAG	Diagnose

Logical

ANA	And to A
ERA	Exclusive or to A
CMA	Complement A
LEQ	Convert A = O to True
LNE	Convert A ≠ O to True
LLE	Convert A ≤ O to True
LGE	Convert A ≥ O to True
LLT	Convert A < O to True
LGT	Convert A > O to True

Interrupt

ENB	Enable Interrupt
INH	Inhibit Interrupt
ESIM	Enter Standard Interrupt Mode
EVIM	Enter Vectored Interrupt Mode
CHI	Clear Highest Interrupt
INK	Transfer (Input) Status Keys to A
OTK	Transfer (Output) A to Status Keys

Shift

ALL	A Left Logical
ARL	A Right Logical
ALR	A Left Rotate
ARR	A Right Rotate
ALS	A Left Shift
ARS	A Right Shift
LLL	Long Left Logical
LRL	Long Right Logical
LLR	Long Left Rotate
LRR	Long Right Rotate
LLS	Long Left Shift
LRS	Long Right Shift

Shift

NRM	Normalize
SCA	Transfer shift Counter to A

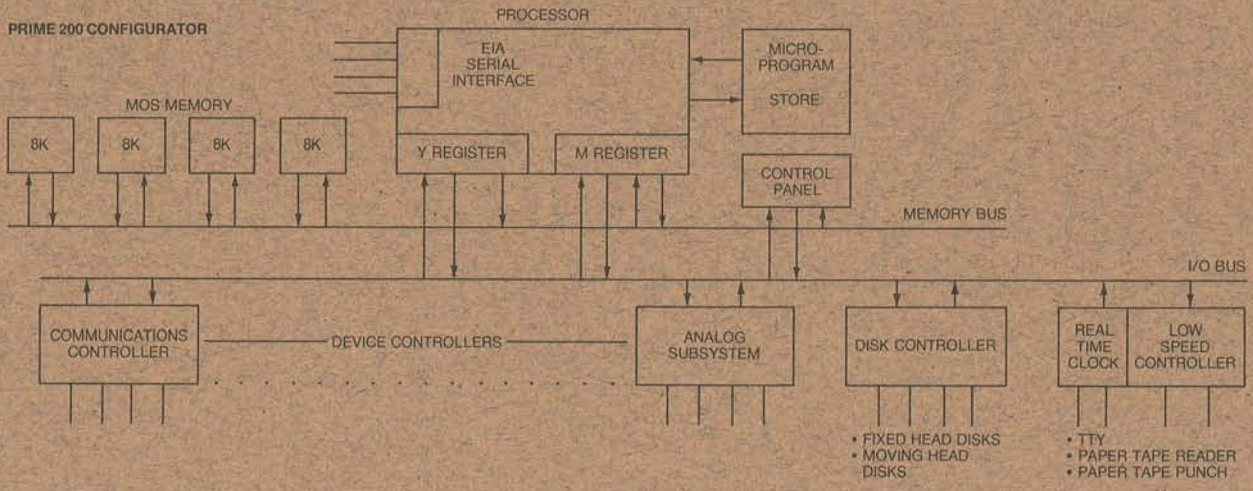
Byte Manipulation

ICA	Interchange Bytes of A
ICL	Interchange Bytes of A and Clear Left Byte
ICR	Interchange Bytes of A and Clear Right Byte
CAL	Clear Left Byte of A
CAR	Clear Right Byte of A

Transfer and Skip

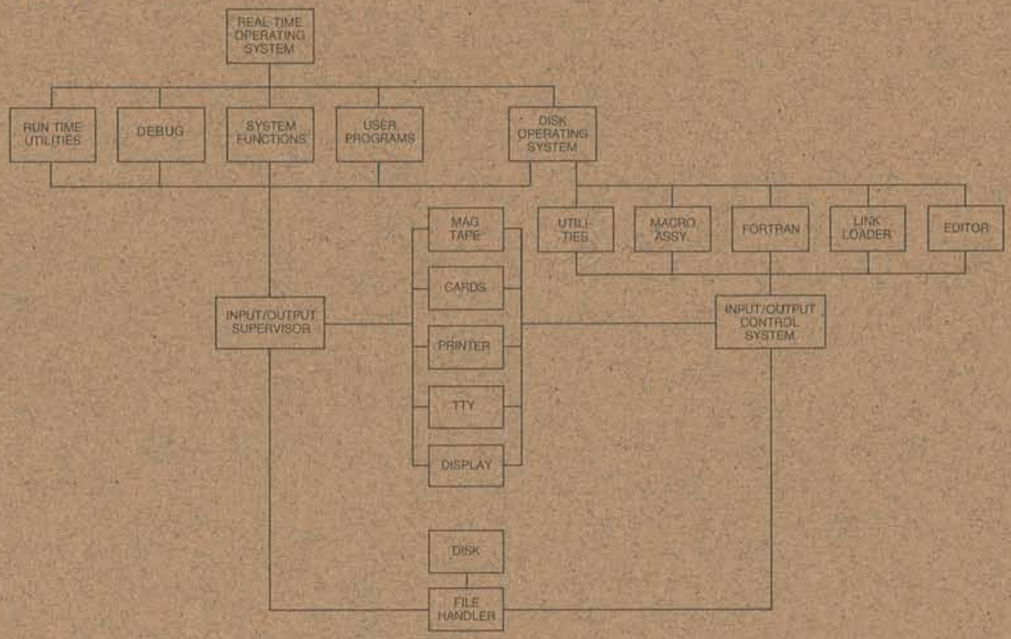
JMP	Unconditional Jump
JST	Jump and Store P
SKP	Unconditional Skip
IRS	Increment, Replace Memory and Skip
IRX	Increment, Replace Index and Skip
DRX	Decrement Replace Index and Skip
CAS	Compare A With Memory
CAZ	Compare A with Zero
SPL	Skip on A Plus
SMI	Skip on A Minus
SZE	Skip on A Zero
SNZ	Skip on A Not Zero
SGT	Skip on A Greater Than Zero
SLE	Skip on A Less than or Equal to Zero
SLZ	Skip on A Bit 16 Zero
SLN	Skip on A Bit 16 One
SASn	Skip on A Bit n Set
SARn	Skip on A Bit b Reset
SSC	Skip on C-Bit Set
SRC	Skip on C-Bit Reset
SMCS	Skip on Machine Check Set
SMCR	Skip on Machine Check Reset
SSS	Skip on Any of Sense Switches 1-4 Set
SSR	Skip on None of Sense Switches 1-4 Set
SSn	Skip on Sense Switch n Set
SRn	Skip on Sense Switch n Reset

PRIME 200 CONFIGURATOR



PRIME200 Software Summary

THE PRIME SOFTWARE SYSTEM



102646171

Disk Operating System
System file management features include multiple directories and multiple volume control and file access methods.

The entire Disk Operating System is written in FORTRAN.

DOS functions as a batch operating system, providing automatic job and data stream routing by storing command sequences on a disk.

Disk files are constructed with forward and backward pointers to prevent accidental loss of one file from snowballing into multiple losses.

Files are addressed by name rather than absolute location.

DOS supports compressed ASCII files.

Real Time Operating System
RTOS is a compact, multiprogramming operating system that schedules processor, main-memory and disk memory resources for optimum control of application programs executed in a real-time environment.

RTOS is available in memory-only and memory/disk configurations.

RTOS provides interrupt handling, multiprogram scheduling, simultaneous input/output and general supervisory functions.

DOS can run under control of RTOS, and conversely, real-time systems can be developed under DOS.

RTOS is supported by a complete set of utility, debugging, and I/O device test programs.

Stand-alone Systems
Stand-alone systems are supported with exactly the same language translators, I/O controllers, and loaders as larger operating system configurations.

FORTRAN IV
The compiler executes in one pass and produces highly optimized code.

The source language is ANSI extended FORTRAN plus other extensions derived from the proposed Instrument Society of America FORTRAN for control applications.

Source language extensions include embedded logical functions such as AND, OR, NOT and XOR; run-time trace; octal constants; and an extended support library.

Macro Assembler
A major improvement in symbolic language processor design, PRIME's macro assembler offers a simple, yet powerful means of generating application-oriented macro statements.

The assembler features pseudo-ops for assembly, listing, and loader control; symbol and data definition; storage allocation; program linking; and conditional assembly.

Desectorizing Link-loader
Relocatable or absolute program modules from a PRIME language translator may be loaded, linked and bound with the link-loader.

Cross-sector references are automatically desectorized and literals, links and temporaries are optimized.

The link-loader saves the loader table for fast symbolic debugging.

DOS can be used to save and restore partial program loads.

Edit and Debug Resources
A full-context editor provides line-by-line and character-by-character editing and automatic execution of string buffers for multiple changes of the same text throughout a program.

A bulk media converter provides editing and transcription resources for handling large volumes of data.

Debugging aids include an interactive trace routine for examining, patching, tracing, etc.

Input Output Control System
The IOCS is a system of stream control routines and device drivers which provide device independence for user programs and centralized I/O control for all systems software.

A unique IOCS feature is the provision for source file editing and merging.

Support Library
The support library includes I/O conversion and driver packages, and standard and extended math functions.

The FORTRAN support library provides formatted I/O, encode and decode statement translators, and list processing functions.

The extended I/O driver library provides both interrupt driven and queued request device handling.

Verification and Maintenance
Thorough verification routines are provided for the PRIME 200 and each of its peripheral devices. Special processor features, including non-functional tests and micro diagnostics greatly enhance the fault diagnosis capabilities of the various test and maintenance routines.

Prime Software Configurator.

Prime Computer Inc.
17 Strathmore Road, Natick, Mass. 01760
(617) 655-6999

Prime Sales Offices

7115 Leesburg Pike
Suite 111
Falls Church, Virginia 22042
(703) 533-9343

28 East Rahn Road
Kettering, Ohio 45429
(513) 435-1343

26211 Central Park Boulevard
Southfield, Michigan 48076
(313) 356-4840

800 East Enterprise Drive
Suite 115
Oak Brook, Illinois 60523
(312) 887-1845

6111 East Skelly Drive
Tulsa, Oklahoma 74135
(918) 663-0518

8 Valley Forge Executive Mall
650 East Swedesford Road
Wayne, Pennsylvania 19087
(215) 688-0396

1020 Corporation Way
Suite 206
Palo Alto, California 94303
(415) 968-6003

1851 Executive Center Drive
Jacksonville, Florida 32207
(904) 396-5253

1 Washington Street
Wellesley, Massachusetts 02181
(617) 237-4565

18321 Ventura Boulevard
Suite 1010
Tarzana, California 91356
(213) 881-8433

9745 Queens Boulevard
Forest Hills, New York 11374
(212) 896-6262

Prime Service Centers

17 Strathmore Rd.
Natick, Massachusetts 01760
(617) 655-6999

26211 Central Park Boulevard
Southfield, Michigan 48076
(313) 356-4840

8 Valley Forge Executive Mall
650 East Swedesford Road
Wayne, Pennsylvania 19087
(215) 688-0396

18321 Ventura Boulevard, Suite 1010
Tarzana, California 91356
(213) 881-8433

Authorized Representatives

Brennan Associates, Inc.
102 Carlson Bldg.
Bellevue, Washington 98004
(206) 454-9332

PLS Associates, Inc.
7335 East Quincy Ave., Suite 201
Denver, Colorado 80237
(303) 771-0140

4300 H. Silver Ave. S.E.
Albuquerque, New Mexico 87108
(505) 255-2330

3431 North 43 Street
Phoenix, Arizona 85018
(602) 957-9110