X528

A new computer ideal for real-time applications...

SYSTEMS
ENGINEERING LABORATORIES

Introducing SYSTEMS 86 . . . Unmatched speed, economy, and versatility giving you unmatched price/performance. A totally integrated hardware-software computer system designed specifically for real-time applications.

SYSTEMS 86 satisfies the most demanding real-time requirements

SYSTEMS 86 will cost you less

With its vast expandable storage capacity, instant response, and ability to coordinate massive information flow in real time, SYSTEMS 86 gives you the highest performance return per dollar spent.

At a cost far less than you might expect, you can own a complete SYSTEMS 86 — including fully tested hardware and software supported by full service backup and training program.

SYSTEMS 86 will meet your needs today and tomorrow. It expands as your needs expand, but without the high costs of hardware and software modifications. You can practically forget about maintenance costs, because we've built SYSTEMS 86 using proven construction techniques, T^2L micrologic circuits, and reliable solid-state components.

SYSTEMS 86 is fast

It gives the instantaneous response you need for real-time results.

Its 600-nanosecond memory cycle time is the fastest in its class. SYSTEMS 86 has an unexcelled throughput rate, enhanced by nanosecond instruction execution speeds, rapid context switching, and an accelerated input/output rate of up to 1.6 million 32-bit words per second.

SYSTEMS 86 is versatile

Its hardware and functionally compatible software deal with a variety of general-purpose and real-time applications: dynamic simulation, data acquisition, communications, biomedical data monitoring, automatic systems checkout, geophysical monitoring, telemetry, and many others. For total applications flexibility, SYSTEMS 86 transacts concurrent foreground and background processing with an extensive assortment of real-time and batch processing job mixes.

SYSTEMS 86 performs mathematical operations, indexing, data transfers, and logical operations on bytes, halfwords, words, and doublewords. It has eight instructions just for bit manipulation, and it's the only 32-bit computer in its class capable of multiplying and dividing data in byte lengths. All alphanumeric data within SYSTEMS 86 are encoded in the American Standard Code for Information Interchange (ASCII) format, providing total language compatibility with communications equipment, peripheral devices, and other computers.

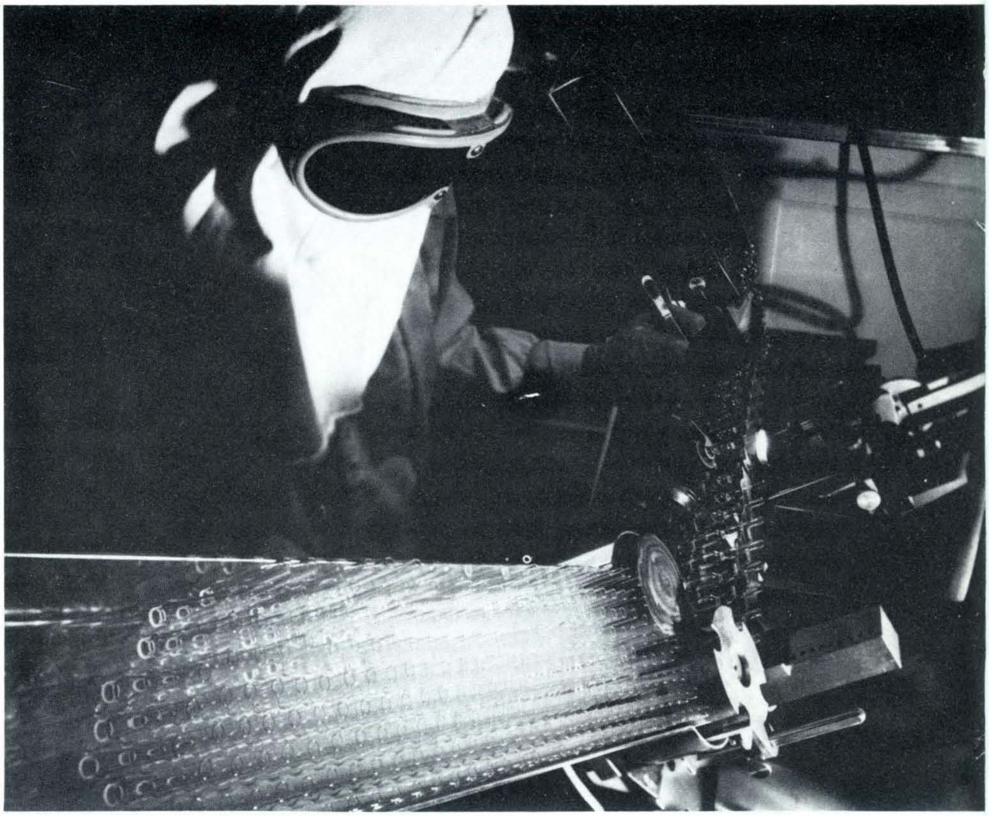
To meet the input, output, bulk storage, and communications needs of many different applications, we've given SYSTEMS 86 capabilities for interfacing with a broad spectrum of peripheral devices and control components. A special program transparency feature allows interchange of peripheral devices and other I/O components without making a single change to your software. This feature allows maximum use of your I/O equipment while retaining maximum CPU efficiency.

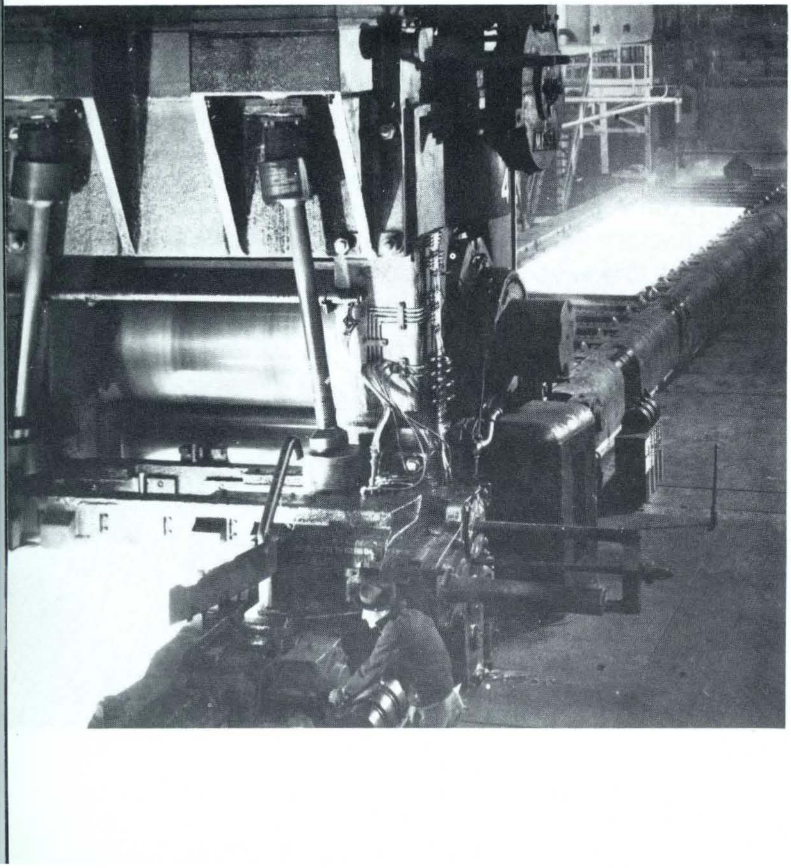
SYSTEMS 86 is easy to use

Hardware and software are blended into a totally integrated system that's designed to respond not only to its operating environment but also to the people who use it.

For easy, flexible development and control of your applications programs, SYSTEMS 86 provides a complete selection of modular software. The software handles routine programming tasks leaving users free to concentrate on the problem to be solved. For specific tasks, you can choose software for supporting multiprogramming in real-time environments, batch processing, man/machine interaction with remote terminals, system generation, program assembly, program compilation, and hardware diagnostics.

SYSTEMS 86 hardware has a multitude of user-oriented features, including automatic recovery after intermittent power loss, memory access protection, keyswitch control of access privileges, multi-purpose registers, and a large versatile instruction set. These are the kind of hardware features that work hand-in-hand with the software to provide unprecedented system usability with maximum efficiency.





SYSTEMS 86 is backed by our experience

At Systems Engineering Laboratories we take pride in our position as major innovators and producers of real-time computer systems. We've achieved an enviable reputation by providing superior price/performance, not only in our custom computer systems and standard computer products, but in our fast-growing, off-the-shelf line of 16-, 24- and 32-bit real-time computers.

We began in 1961 by developing custom data measurement systems for use with computers. In a short time we were experts in the field of low-level data acquisition, commanding the largest share of the market.

We added digital computers to our product line as a natural outgrowth of our experience. And combining our computers with our continually expanding line of measurement and control products gave us unmatched single-source capabilities. . . capabilities that enabled us to successfully penetrate the field of real-time computer applications.

Our real-time computer systems are at work in industry, government, education, medicine, and research. And they are dealing with a broad spectrum of applications. Here's just a sample:

Simulation — A large aircraft manufacturer uses our real-time computer systems for simulating the complete flight profile of multi-range jetliners. The computer systems test the strength of the aircraft under the most extreme flight conditions imaginable, as well as during such maneuvers as taxiing, take-off and climb, cruise, descent, and landing. Our systems also perform total simulation of helicopters and space capsules for pilot training in dynamic environments.

Test automation — Our real-time computer systems are being used for testing locomotives, tractor transmissions, diesel engines, automotive distributers, and the SATURN V launch vehicle.

Control — We produce real-time computer systems that supervise, monitor, and regulate continuous processes in a wide range of industries.

A large aluminum mill, for example, uses three of our real-time computer systems to regulate mill processes. Each system continuously monitors a wide range of variables, analyzes and processes data concerning these variables, and controls such critical functions as conveyor speed and roller heights.

At a nuclear power station, a Systems computer system provides complete, automatic closed-loop control of the nuclear reactor. In addition, the system handles power plant monitoring, on-line calculations, logging, and information display.

We also build computer systems for controlling a variety of other real-time operations. Typical examples include newspaper typesetting, automatic wire wrapping, large-scale circuit testing, jet engine testing, laboratory data analysis, and navigational data analysis for oil exploration.

SYSTEMS 86 has unprecedented hardware and software capabilities for dealing with applications such as these and more. Many more than we can cover here. The important thing is that SYSTEMS 86 is backed by the resources, knowledge, and technological experience that has made Systems Engineering Laboratories a leader in the field of real-time computer systems.

SYSTEMS 86 provides a faster memory, a more powerful CPU, and more versatile I/O

Here's why SYSTEMS 86 gives you better price/performance than any other medium-sized computer.

Central processor unit

Powerful priority interrupt system . . . containing up to 124 interrupt levels. Interrupt priority scheduling is fully automatic. Individual interrupt levels can also be enabled, disabled, requested, activated, and deactivated all under program control.

Real-time interrupt response . . . interrupt response time is less than 6.6 microseconds.

Versatile masking capabilities . . . masking can be performed as part of the execution of register-to-register, arithmetic and logical instructions. No additional instruction execution time is required for masking.

User-oriented programming systems . . . that exploit the SYSTEMS 86 nanosecond instruction execution speeds for foreground processing while giving the user unprecedented background processing capabilities with minimum system overhead.

Mode keyswitch . . . allows the user to command the degree of control that the currently active program has over system resources.

System protect hardware . . . guarantees instantaneous core storage of the current system status in the event of power failure, and automatic program recovery when power is restored.

System protect keyswitch . . . allows the user to selectively lock out intervention from the control panel, the system override interrupt, and auto-start interrupt. The user determines the degree of program interference that can be exerted by external sources.

Fast register imaging . . . execution of one instruction loads or stores the content of all eight general-purpose registers in only 5.4 microseconds. General-purpose registers assume a true multi-task role.

Rapid transfer of data between registers . . . execution of any register-to-register instruction takes only 600 nanoseconds.

Efficient memory utilization . . . over 30 percent of instructions are half-word instructions and can be packed two per memory location.

Halfword addressing . . . enables SYSTEMS 86 to access and operate on 16-bit operands packed two per memory location without any reformatting time penalties.

Variable precision arithmetic . . . fixed-point instruction set permits arithmetic operations to be performed on quantities of data ranging from a byte (8 bits) to a doubleword (64 bits). Add and subtract in byte, halfword, word, and doubleword data lengths. Multiply and divide in byte, halfword, and word data lengths.

Direct addressing of entire core memory . . . any bit, byte, halfword, word, or doubleword in up to 128K of memory is directly addressable.

Indexing . . . can be selected according to the exact needs of the problem. Positive and negative indexing quantities can range from a byte to the entire memory capacity in increments of bytes, halfwords, words or doublewords.

Pre and post indexing . . . at absolutely no "overhead" cost in time.

Multilevel indirect addressing . . . indirect addressing can go to any depth. Users can keep data sections of programs separate from procedures section for ease of use.

Real-time clock . . . provides an accurate reference for program sequencing . . . times real-time functions to occur at specific instants . . . measures elapsed time, time of day, etc.

Floating-point arithmetic hardware . . . floating-point arithmetic instruction set contains single-word formats for storage economy, and doubleword formats for increased resolution and fractional significance.

Floating-point guard digit . . . enables the least significant bit to be rounded automatically after all arithmetic operations.

Arithmetic exception interrupt . . . provides automatic hardware detection of overflow after every arithmetic operation . . . eliminates time consuming testing of overflow and underflow.

Memory

Fast memory cycle . . . full read/selective-replace/restore memory cycle time of only 600 nanoseconds — the fastest memory cycle offered by any comparable computer.

Memory byte parity . . . a parity bit stored with each 8-bit byte permits selective replacement of individual bytes with no added time penalty.

Word slicing . . . permits reading and writing of individual bytes or halfwords in any memory location.

Modularly-expandable core memory . . . standard 8K core memory modules make it easy and economical to expand memory from a basic 8K (8192 32-bit words) up to 128K (131,072 32-bit words.)

Total protection of vital information stored in memory . . . power fail-safe feature prevents modification of memory data, either in the event of power failure or during normal turn-on and turn-off.

Multiple memory ports . . . memory accesses for both input/output and program execution can be performed during the same memory cycle.

Direct memory access . . . provides an input/output path that completely by-passes the central processor . . . transfers data on a non-cycle stealing basis at maximum rates.

Memory page protect . . . gives the user program control over access to selectable pages of memory. Areas of memory can be kept private for individual user's programs. Foreground and background programs can be run concurrently. Foreground programs are protected from unchecked background programs.

Input/output

Fully automatic input/output system . . . requires minimum central processor involvement. Peripheral device commands from central processor are executed in only 600 nanoseconds . . . more central-processor time is available for program execution.

High-speed exchange of input/output data . . . data transfers occur at rates of up to 1,666,666 32-bit words per second.

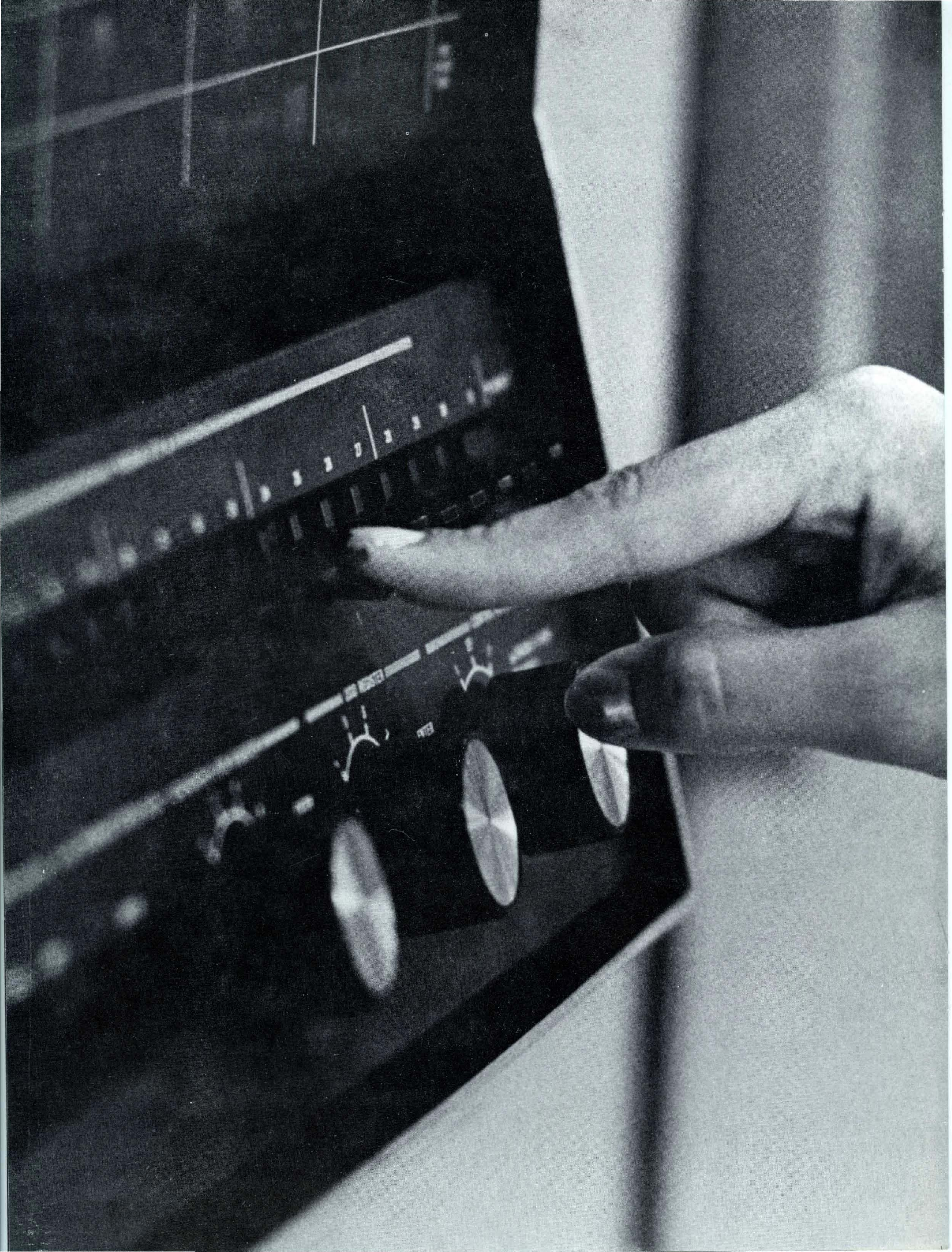
Concurrent data exchanges with peripherals . . . up to sixteen device controller channels perform concurrent fully buffered data transfers in either a single-word or a high-speed block-transfer mode. Each device controller channel interfaces with either a single peripheral device or a multi-peripheral device controller.

Automatic reinitialization of block transfers . . . enables continuous transfers from non-contiguous areas of memory. Each block transfer can be programmed to start automatically at the completion of the preceding block transfer.

Variable width data exchange paths . . . input/output data can be formatted in byte, halfword, or word increments.

Program transparent priority scheduling . . . of device controller channels allows the system to be reconfigured, expanded, and tuned to your exact needs without altering the software. Device controller channels can be interchanged without changing the virtual priority levels used in the interrupt control instructions because all parameters which effect programming are hardware implemented within the device controller channels.

High- and low-priority device controller channels . . . for maximum data handling efficiency when operating with both fast and slow responding peripheral devices.


27
REGISTER
ENTER

The hardware system designed for real-time response

SYSTEMS 86 provides the speed, storage, and input/output capabilities needed for real-time applications. It combines the reliability essential to effective use in a dynamic environment with the functional modularity necessary for economical growth and ease of maintenance.

The basic SYSTEMS 86 hardware elements include a highly flexible automatic input/output system capable of handling virtually any combination of high- and low-speed peripheral devices operating concurrently, a 600-nanosecond core memory that's modularly expandable, and a central processor with unsurpassed data manipulation and high-speed processing capabilities. A keyboard/printer is included as a standard input/output device.

SYSTEMS 86 is packaged to permit future expansion. All electronics, including up to 32K words of memory, are enclosed in an attractive cabinet requiring less than eleven square feet of floor space. A cabinet, identical in size to the computer cabinet, is provided for each additional 64K words of memory.

The control station for SYSTEMS 86 is available as either a compact control panel mounted on the computer cabinet or as an optional remote desk-type console that offers facilities for maximum work handling efficiency. Both configurations are designed with complete emphasis on human factors.

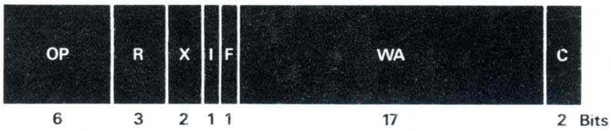
Easy-to-read, rear-illuminated indicators present total machine status at a glance. They display such functions as memory parity error, halt/run, active interrupt levels, and the content of all registers. Keyswitches, lever sense switches, and rotary selector switches provide finger-tip control of multiple register displays, transfer operations, register load and clear, program status, console interrupt, program start and stop addresses, privileged operation, and much more.

A 600-nanosecond memory that's expandable

Standard memory module size for SYSTEMS 86 is 8K words. The memory is readily expandable in 8K-word modular additions to a maximum size of 128K words. And memory is nonvolatile.

The entire memory system is addressable and alterable in bit, byte, halfword, word, and doubleword quantities. In just one 600-nanosecond memory cycle, a word can be read out of memory, partially replaced or modified, and then restored in memory. Individual memory locations can be addressed without the use of base registers, index registers, indirect addressing, sectorizing, mapping, or other address modifications. Memory reference instructions, however, have the capability of multilevel indirect addressing. Both preindexing and postindexing can be performed in conjunction with indirect memory addressing. Indexing of memory addresses can be by bytes, halfwords, words, and doublewords.

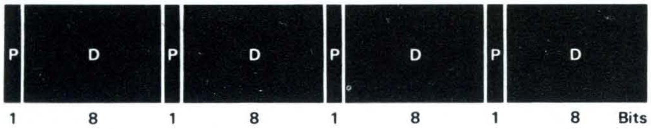
MEMORY REFERENCE INSTRUCTION FORMAT



OP	Operation code
R	General-purpose register address
X	Index register
I	Indirect addressing bit
F	Format bit
WA	Word address
C	Address code, including byte or halfword addresses

You can replace individual bytes in memory without forfeiting valuable time for parity generation. That's because each 8-bit byte in memory has its own parity bit. When a byte is written into memory, a parity bit is generated only for the new byte . . . not for the entire 32-bit word.

MEMORY WORD FORMAT



P	Parity bit
D	Data byte

The SYSTEMS 86 parity scheme guarantees protection of internal data. It prevents errors from damaging the content of memory, the content of registers, and machine status. Each time a word is read from memory, parity for all four memory-word bytes is checked automatically. If a parity error is detected during the process of fetching an instruction, fetching an operand, or making an indirect call while fetching an operand, the instruction is instantly aborted before any damage can occur. The parity error also causes the system to enter an error trap processing routine.

Errors occurring during memory read operations for I/O transfers do not jeopardize vital data and status information. Thus, parity errors detected during such operations merely establish testable conditions in the input/output system.

Memory modules for SYSTEMS 86 are available with either one or two access ports. One port always serves as the memory access for the central processor. In a double access port memory system, the second port is the link between memory and the direct memory access (DMA) channel.

The DMA channel permits input/output transfers of vast quantities of information without degrading central processor performance. The data transfer path goes directly to memory via a special 32-bit bus. Memory accesses for both program execution and input/output transfers are performed during the same 600-nanosecond memory cycle . . . no memory cycles are stolen.

The optional memory protect feature allows the user to dynamically control the degree of memory access provided by each port as well as each program's ability to read, write, and execute the content of selected 512-word pages within each memory module.

By exercising the memory protect feature you can exploit the full multiprogramming capabilities of SYSTEMS 86. It prevents core-resident programs from modifying one another or from branching into areas of memory occupied by data. Several foreground and background programs can operate concurrently with complete memory security.

The degree of access protection provided by the memory protect feature is determined by page and port protect codes stored in special protect registers. One protect register is provided for each memory module. The protect codes are dynamically assigned to each register under program control. Code assignment is fast, so overhead time is minimized when switching from one program to another. It takes just one 1.2-microsecond transfer instruction to replace the protect code for an

entire 8K-word memory module. Execution of another 1.2-microsecond transfer instruction preserves any protect code by storing it in a general-purpose register. The software for SYSTEMS 86 exploits the speed of the memory protect feature when allocating core in multiprogramming environments.

The console-mounted mode keyswitch works in conjunction with the memory protect feature. It has three positions: privileged, semiprivileged, and unprivileged. At each switch position the page protect codes exert different degrees of access control over memory — from permitting unlimited access privileges to restricting all write operations.

In addition to determining the effect of the page protect codes on memory access, the mode keyswitch regulates the currently active program's ability to execute certain privileged instructions. These include the transfer instructions that assign and store the protect codes, the interrupt control instructions, the device command instruction, and the halt instruction. Thus, the memory protect feature and the mode keyswitch not only protect memory, but control the execution of input/output transfers and modification of the priority interrupt status.

A central processor structured for rapid handling of byte, halfword, word, and doubleword data increments

The central processor performs arithmetic, logical, comparison, and data manipulation operations at real-time speeds and with maximum precision. Standard arithmetic hardware for SYSTEMS 86 performs fixed-point arithmetic operations on data in byte, halfword, word, and doubleword data lengths. SYSTEMS 86 can also be equipped with floating-point arithmetic hardware. Floating-point add, subtract, multiply, and divide can be performed in either word or doubleword formats.

Three central processor registers are set aside for special functions. The Instruction (I) Register stores the current fullword or two halfword instructions. The Transfer (T) Register holds operands acquired from memory during the execution of instructions. And the Program Status Word (PSW) Register preserves the program count and the current operating condition of all critical machine functions.

The eight general-purpose registers within the central processor are readily adaptable to arithmetical, logical, and shift operations. Each register stores a full 32-bit word and can be addressed and operated on by most instructions. Such versatility eliminates the need for a separate accumulator and other special registers reserved for logical and arithmetical operations. Instructions are available for direct addressing of one or more registers as well as for direct addressing of individual bits within registers. When switching from one operating environment to another, all general-purpose registers can be loaded or their content stored with just one instruction . . . in only 5.4 microseconds. Transferring data from one register to another takes only 600 nanoseconds. And complementing, masking, or both can be performed as part of any inter-register transfer without requiring additional time.

In addition to performing arithmetic and logical operations, five of the general-purpose registers can also perform special tasks. One serves as a link register, three double as index registers, and another is used during masking operations.

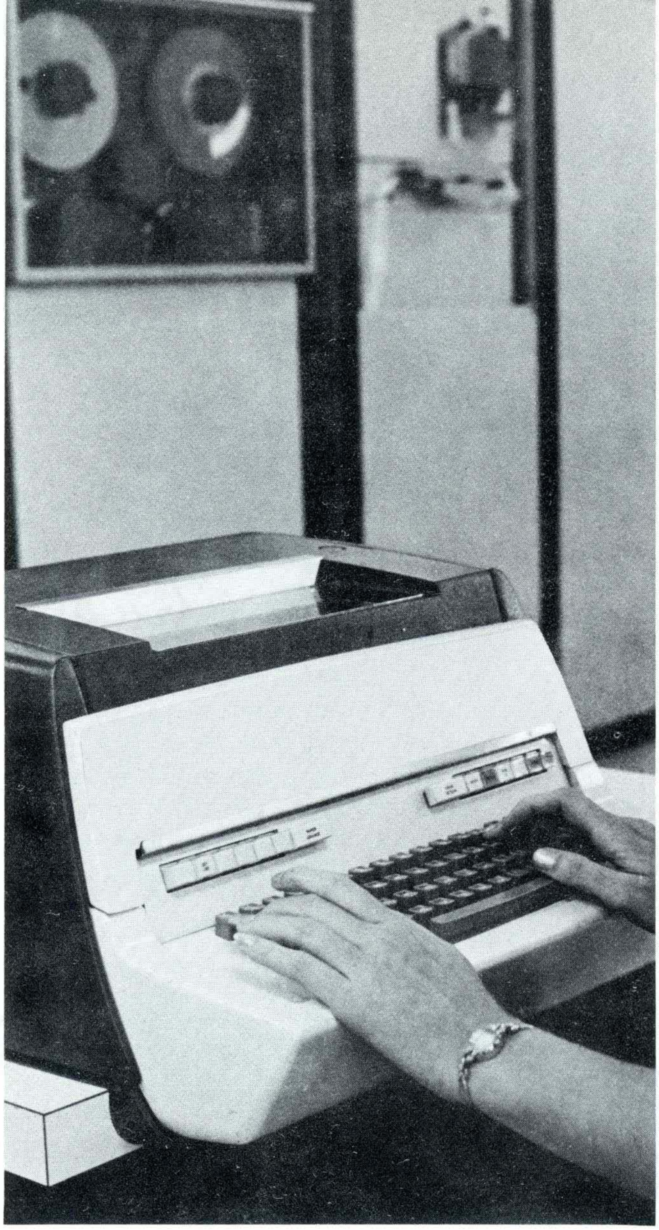
As part of the execution of the branch and link instructions, the link register automatically acquires the current program status word thereby freeing the PSW register for the branch routine's program status word. This linking simplifies the writing of reentrant programs.

The three index registers enable positive or negative indexing in quantities ranging from a byte to the total memory capacity. Thus, the indexing base can be selected according to the exact needs of the problem. Indirect addressing with indexing can go to any depth. Preindexing and postindexing require no additional time.

Using the mask register in conjunction with a wide variety of instructions results in outstanding data formatting and logical operating capabilities. The ability to mask is not limited to register-to-register instructions but can be performed as part of most arithmetical, logical, and transfer operations.

In the event of power failure or normal system turn-off, the central processor preserves the current machine operating status by storing the content of all registers in the nonvolatile memory. When power is restored the machine operating status is automatically recalled from memory. Programs resume without loss of information . . . and without operator intervention.

The central processor's flexible, fast-response interrupt system automatically schedules service routines for such functions as input/



SYSTEMS 86

output transfers, memory parity errors, non-present memory, and unimplemented instructions. Each service routine is assigned a unique interrupt priority level. The interrupt priority levels are structured so that the central processor always attends the most important task.

When an interrupt occurs, the response time is 6.6 microseconds or less. The interrupt priority logic automatically — and without programming — identifies each interrupt according to its priority. A program operating at one priority level can initiate interrupts at other levels and take advantage of the interrupt priority logic to handle the scheduling of all tasks, whether requested by the external or internal environment.

In addition to automatic hardware initiation of priority interrupts, five program instructions are available for dynamically controlling interrupts: Enable Interrupt, Disable Interrupt, Request Interrupt, Activate Interrupt, and Deactivate Interrupt. The high degree of operational flexibility offered by program control of the interrupts opens up a multitude of capabilities to SYSTEMS 86 users.

Input/output facilities automatically handle all data exchanges with minimum central processor involvement

Total utilization of SYSTEMS 86 resources is implemented by a unique Automatic Input/Output system (AIOS) linked to peripheral devices selected from a broad spectrum of equipments for gathering, digesting, converting, and communicating system inputs and results.

Each peripheral device is connected to the AIOS by a Device Controller Channel (DCC) containing all required data transfer logic, device electronics, and signal conditioning circuits. Data transfers between the DCC's and peripheral devices are performed independent of central processor control.

For maximum operating efficiency, high and low priority DCC's are provided. High priority DCC's exchange data at the high read/write rates of disc files, magnetic tape units, and other fast access devices. Low priority DCC's exchange data more efficiently at the transfer rates of slower responding devices, like card punch/readers, paper tape punch/readers, and line printers.

All parameters which effect input/output programming are implemented in the DCC's. This feature lets you tune the AIOS by adding or rearranging DCC's without changing the system software. When the position of a DCC is changed, the actual priority levels change. But the virtual priority levels used in the input/output interrupt control instructions remain the same.

The control techniques used by the AIOS minimize the part played by the central processor during input/output functions. You benefit because minimum time is taken from program execution. Internal central-processor operations continue while the AIOS maintains complete control of the transfer process. Transfers between memory and up to sixteen DCC's occur automatically in an interleaved manner. Data can be transferred in increments of single bytes, halfwords, words, or blocks at rates of up to 1,666,666 transfers per second. Each block transfer can be programmed to start automatically at the completion of the preceding block transfer.

The illustrations on these pages show two typical SYSTEMS 86 configurations.

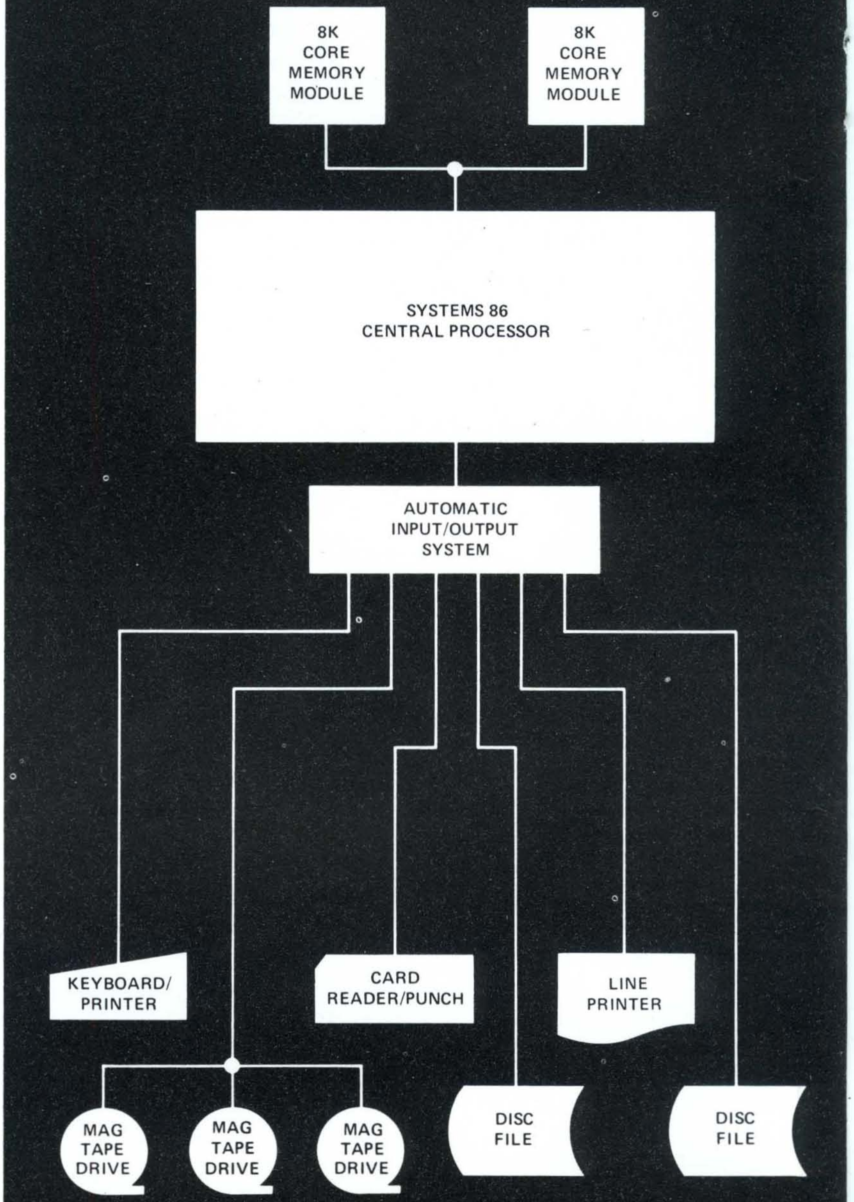
TYPICAL BATCH PROCESSING CONFIGURATION*—This configuration makes maximum use of the 600-nanosecond memory cycle time, the 32-bit word length, and the flexible I/O structure of SYSTEMS 86 to process large batches of data rapidly and accurately. Some of the other features that make SYSTEMS 86 especially well suited for batch processing include multilevel indirect addressing, variable length indexing units, fixed- and floating-point arithmetic hardware, and a powerful instruction set.*

When processing data in batches, most of the information in core is transient. Therefore, one or two 8K memory modules provide sufficient storage for the processor and utility programs, plus the largest element of data encountered with most applications.

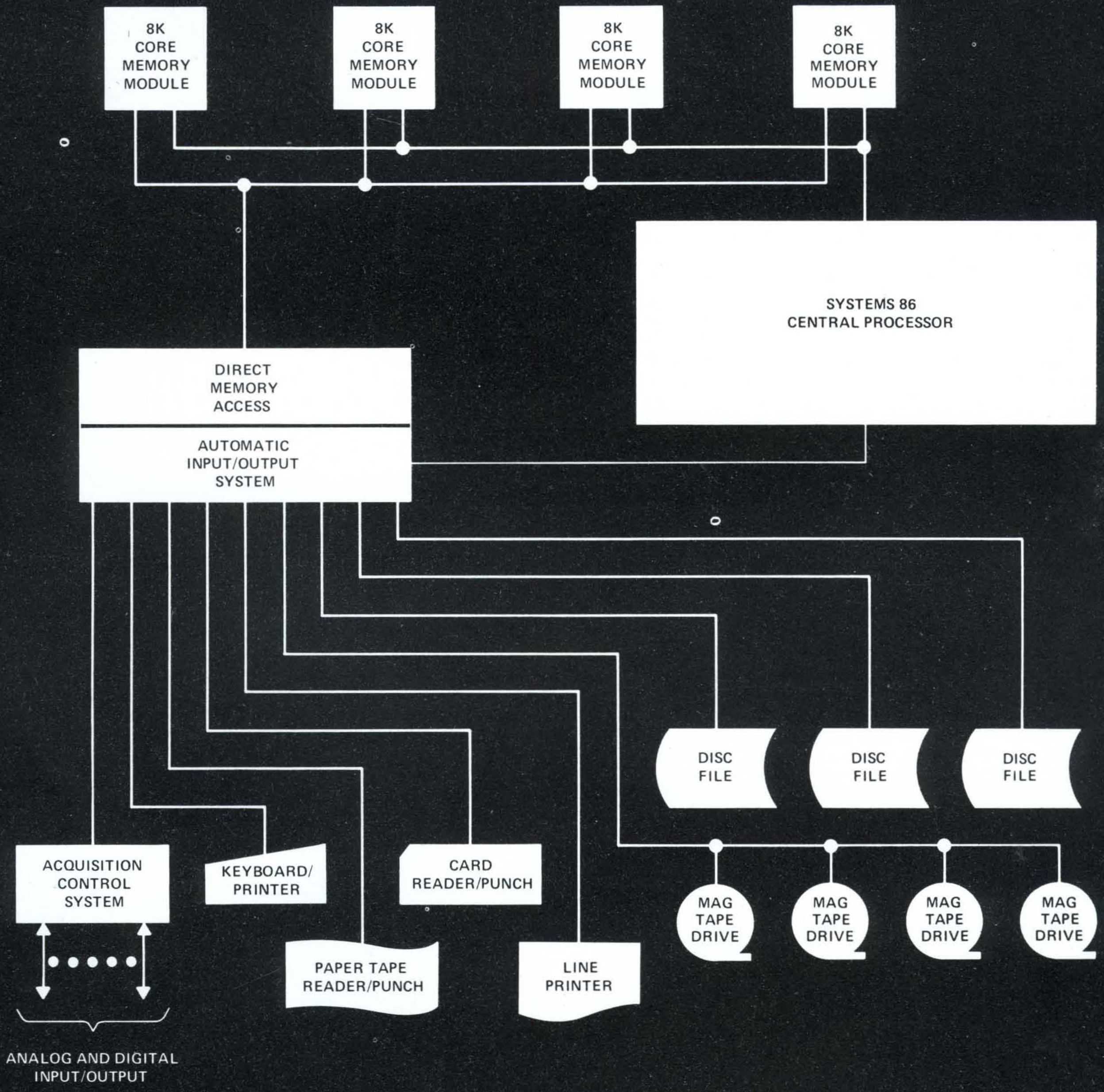
Peripherals can be selected to meet the data requirements of each batch processing application. An installation using FORTRAN IV, for example, may require a second high-speed line printer to increase system throughput. Additional disc files, magnetic tape transports, and paper tape equipment can also be included in the batch processing configuration.

REAL-TIME CONTROL/BATCH PROCESSING CONFIGURATION*—This configuration provides instantaneous response to external events, high-speed data handling, total I/O flexibility, and maximum throughput. It responds to real-time events while concurrently handling batch processing jobs on a time-available basis. The flexibility of the SYSTEMS 86 priority interrupt structure gives this configuration the ability to react to external events according to the needs of each problem. And because all I/O data is channelled through the DMA, the system can execute I/O transfers and simultaneously perform computations in the central processor.*

Selection of peripherals depends on each user's application. Fast-access bulk storage, such as fixed-head discs, are provided for storing high-use software modules. Remote terminals can be included for operation in an interactive conversational mode and for entering data into the batch job-stream. The acquisition control system interfaces SYSTEMS 86 to the external real-time process equipment.



TYPICAL SYSTEMS 86 BATCH PROCESSING CONFIGURATION



TYPICAL SYSTEMS 86 REAL-TIME CONTROL/BATCH PROCESSING CONFIGURATION

READ DATE
SAVE DATE

SYSTEMS 86 modular software matches your requirements

The computers you use in the 1970's must be designed and built to operate efficiently with the programming systems which will be used. No longer can users tolerate expensive overhead. No longer can core be wasted on programs which are not needed to solve specific problems.

SYSTEMS 86 has been designed from its inception with total integration of hardware and software. It has been designed to enable users to take full advantage of its exceptional real-time hardware capabilities. The monitors, for example, occupy minimum core, and contain only those segments needed for a particular installation. And because the monitors for SYSTEMS 86 handle all standard computer functions (i.e., input/output, bookkeeping, and scheduling), users will spend far less time and effort developing control programs.

SYSTEMS 86 comes complete with the following software:

Batch Processing System
Real-Time Monitor
Assembler
Macro Assembler
FORTRAN IV
BASIC
Utility Programs
Math Library
System Generation
Hardware Diagnostics

In addition, our experience in designing custom systems enables us to personalize software systems to meet your individual needs. You specify the requirements and we'll tailor a software package to do your job efficiently and economically.

Batch Processing System

The Batch Processing System is a complete software system which frees users from writing control functions and allows them to concentrate on resolving their specific problems. The system can be accommodated in 2.5K words of core while using tape or disc to hold the processor and utility programs. The system is designed for compile/load/go with no operator intervention. The entire system can work in only 8K words of core. But for complete efficiency, we recommend that 16K words of core be available.

The processors which will operate under the Batch Processing System include:

Assembler
Macro Assembler
FORTRAN IV
Utility Programs
Math Library
System Generation

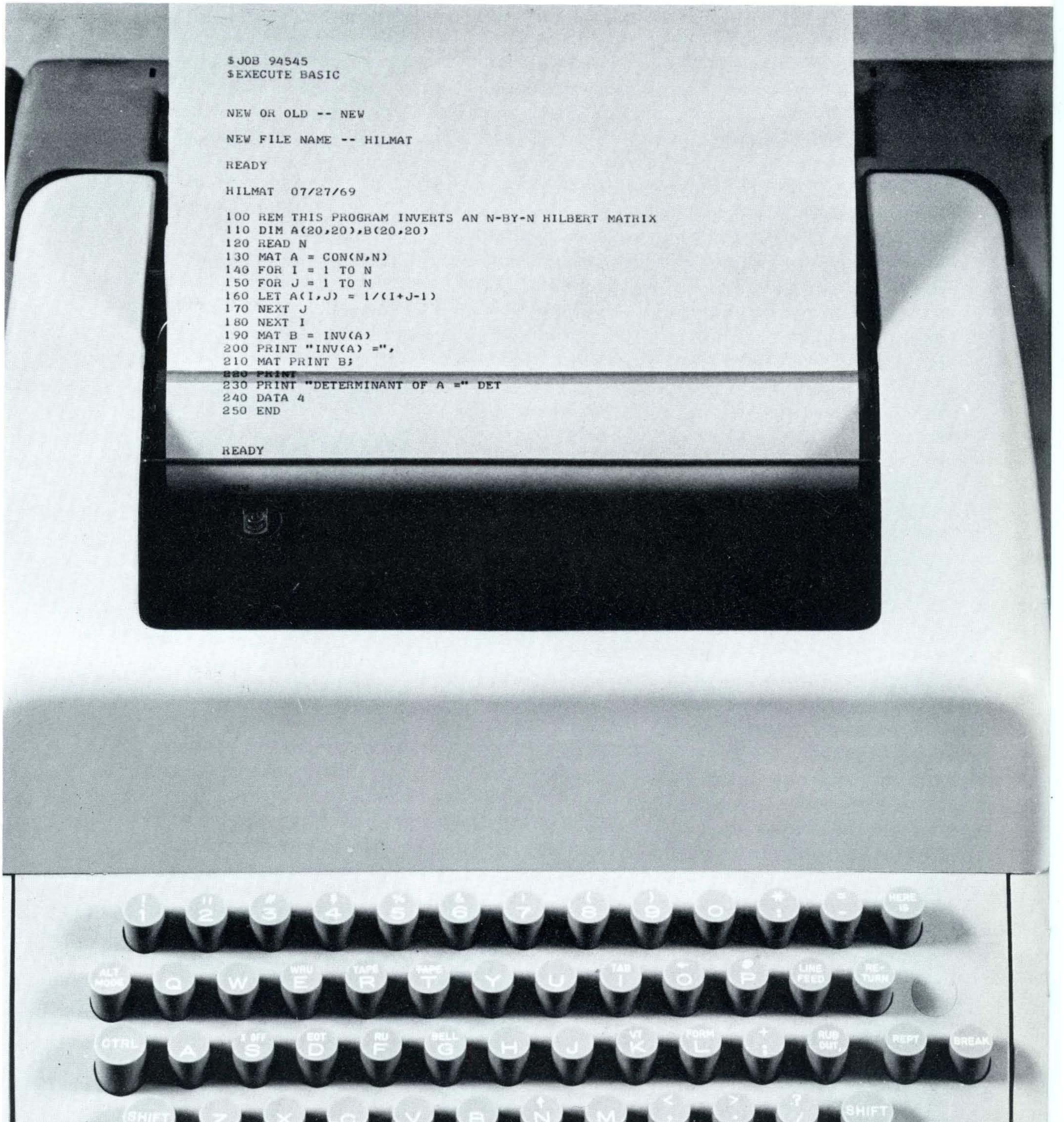
Real-Time Monitor

A Real-Time Monitor exploits the full capability of its computer by controlling multiple tasks, or programs, within the computer. Therefore, the quality of the Real-Time Monitor can either enhance or detract from the overall hardware capabilities of the system.

During the 1960's Real-Time Monitors established a concept of foreground and background processing. Some computers during that time were capable of handling only a single foreground program; while others that could handle multiple foreground tasks were limited to a single, non real-time, background program. The SYSTEMS 86 Real-Time Monitor is designed for user needs of the 1970's. It is a totally task-oriented system enabling you to control up to sixty-four tasks in either the foreground or background — and in any combination of real-time and non real-time programs.

This capability — coupled with asynchronous input/output while computing — will give you maximum system throughput regardless of your application requirements. The task orientation of the Real-Time Monitor makes SYSTEMS 86 equally suited for real-time applications, batch processing, and general-purpose scientific applications.

The Real-Time Monitor also gives SYSTEMS 86 strong remote terminal capabilities through either the conversational mode utilizing BASIC or the remote job entry mode which permits the standard entry of jobs into the batch stream from remote terminals.



```
$JOB 94545
$EXECUTE BASIC


NEW OR OLD -- NEW

NEW FILE NAME -- HILMAT

READY

HILMAT  07/27/69

100 REM THIS PROGRAM INVERTS AN N-BY-N HILBERT MATRIX
110 DIM A(20,20),B(20,20)
120 READ N
130 MAT A = CON(N,N)
140 FOR I = 1 TO N
150 FOR J = 1 TO N
160 LET A(I,J) = 1/(I+J-1)
170 NEXT J
180 NEXT I
190 MAT B = INV(A)
200 PRINT "INV(A) =",
210 MAT PRINT B;
220 PRINT
230 PRINT "DETERMINANT OF A =" DET
240 DATA 4
250 END


READY
```

Assemblers

SYSTEMS 86 offers both an Assembler and a Macro Assembler for fast, efficient coding of real-time application programs.

Both assemblers run under either the Batch Processing System or Real-Time Monitor and are truly I/O independent. This enables the user to assign the input or output to any media. One unique feature of both assemblers is their ability to define FORTRAN COMMON blocks. This makes it easy to transfer arguments from a FORTRAN program to a symbolically assembled subroutine.

The Macro Assembler is a powerful processor which permits users to nest macros, execute recursive macro calls, and pass parameters on to nested macros. Even though the Macro Assembler is a multi-pass processor, it maintains an extremely fast internal assembly speed of 6000 source statements per minute.

The Assembler is a subset of the Macro Assembler. It is designed specifically for users who have minimum core requirements and do not require macro capability. The internal assembly speed of the Assembler exceeds 10,000 source statements per minute.

FORTRAN IV

The FORTRAN IV Compiler is a three-pass compiler that not only meets all USASI specifications, but exceeds many of them. Several extensions in capability are included for the real-time user as well as for the scientifically oriented user.

The more powerful extensions to standard USASI FORTRAN IV which are provided include:

In-line symbolic coding . . . allows the user the flexibility to include the full set of assembly-language source statements within the main body of his program.

Mixed-mode arithmetic expressions . . . provide the user with the capability of mixing various types of variables within the same arithmetic expression.

Array extensions . . . provide the user with the freedom of using any arithmetic expression as an array subscript.

Real-time features . . . include facilities for interrupt utilization, as well as scheduling and testing input/output device assignments and status.

Asynchronous I/O . . . allows concurrent use of central-processor time while data is being transferred to and from memory.

Multiple entry and return . . . reduces core requirements by allowing the generation of one subroutine with various entry and return points.

ENCODE and DECODE features . . . provide a means of memory-to-memory data conversion in user-supplied internal buffer areas.

Bit and character types . . . provide for unique identification of bit and/or byte variables for maximum utilization of memory and the SYSTEMS 86 instruction set.

Optimization of the generated code is one of the major strong points of this FORTRAN compiler. The SYSTEMS 86 computer is fast, and the software takes full advantage of this speed.

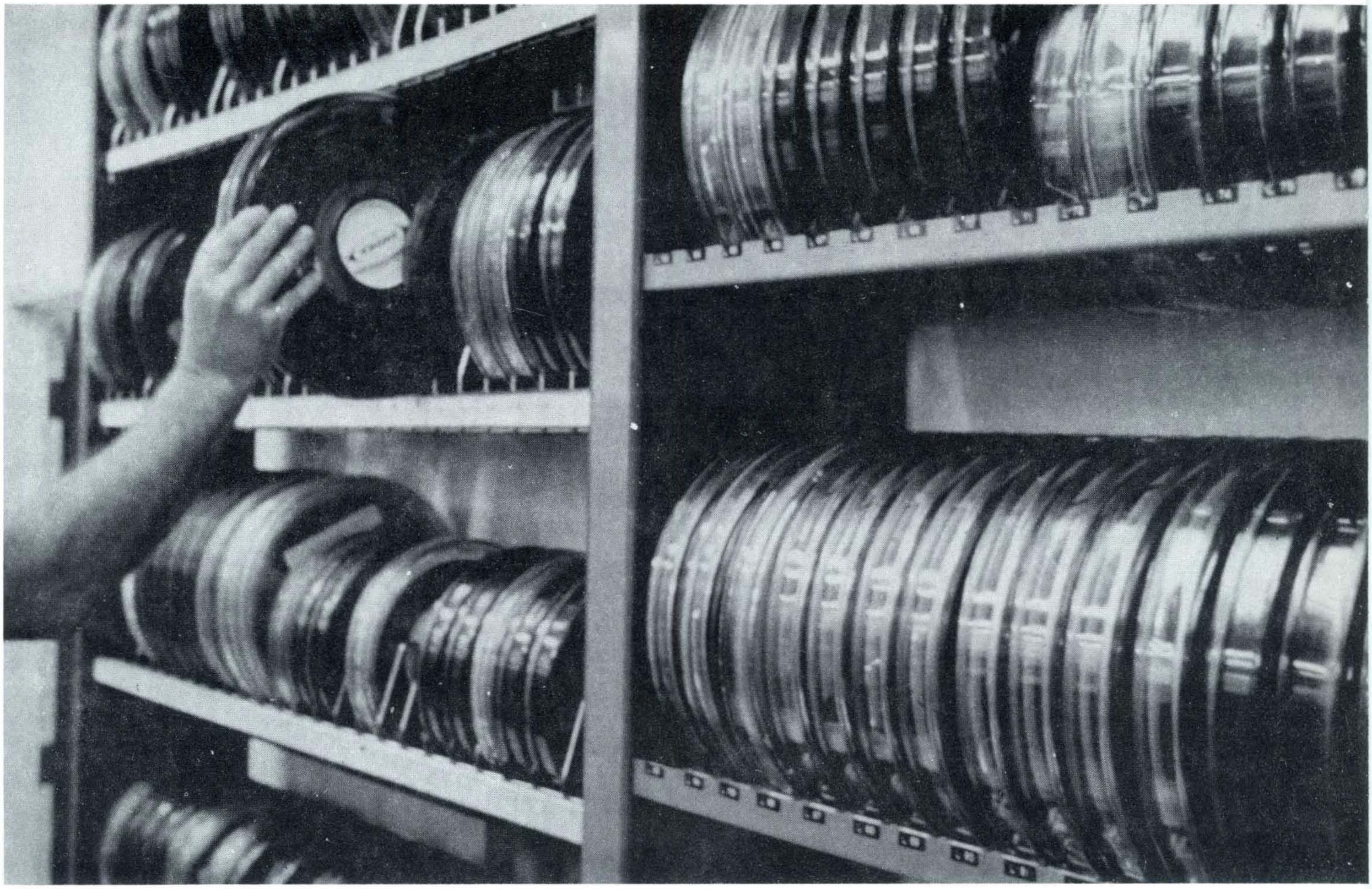
BASIC

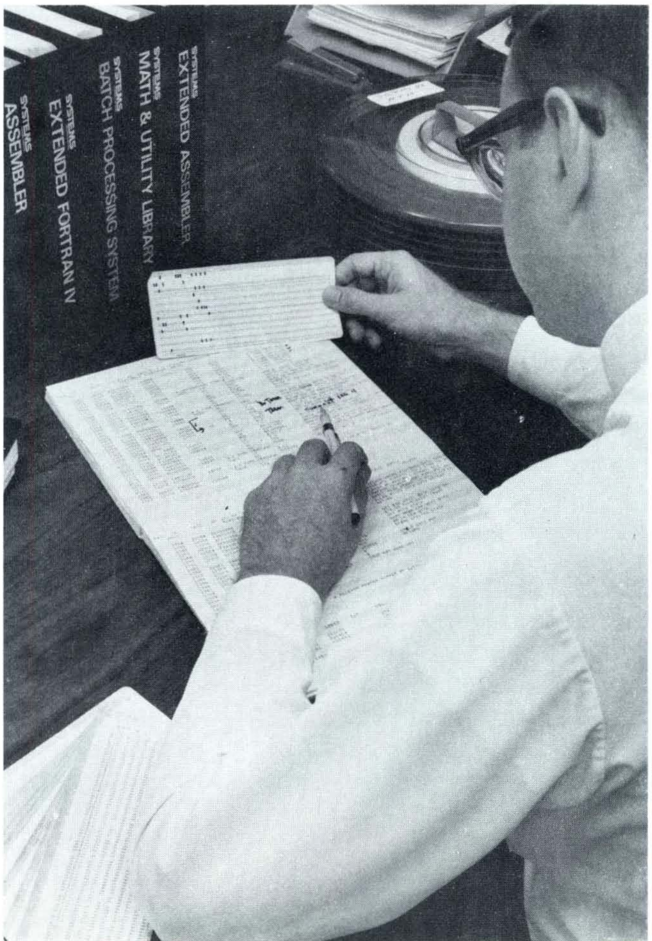
BASIC is a multi-user, terminal-oriented programming system that permits users to communicate with SYSTEMS 86 using simple, concise statements. Because of its simplicity it is intended primarily for use by non-programmers.

The user enters BASIC programs into SYSTEMS 86 simply by typing them at the teletypewriter or similar remote-terminal device. BASIC programs consist of a group of numbered statements that resemble ordinary mathematical notation. The computer responds in the same type of easy-to-understand statements. When BASIC programs are entered into SYSTEMS 86, it isn't necessary to type the numbered statements in the correct sequence. The BASIC system sequences all statements before compiling them into interpretive code.

BASIC flags syntax errors by checking statements as they are entered. If a syntax error is detected, the user receives an error message. He can then make the necessary corrections by retyping the statement in its correct form.

BASIC for SYSTEMS 86 also provides for command editing, program listing, compilation, execution, program storage, program retrieval, matrix computations, and string manipulation.




SYSTEMS ASSEMBLER
SYSTEMS EXTENDED FORTRAN IV
SYSTEMS BATCH PROCESSING SYSTEM
SYSTEMS MATH & UTILITY LIBRARY
SYSTEMS EXTENDED ASSEMBLER

Utility Programs

The utility programs offered with SYSTEMS 86 aid the programmer in debugging and updating his programs and converting any media, such as tape-to-printer or card-to-tape.

The utility program called "Debug" lets the user check out programs and get them in an operational condition in the shortest possible time. Program checkout can be performed at a teletypewriter. The user can stop at various points in his program, investigate program data, and output any portion of the program to either the teletypewriter or high-speed line printer.

Math Library

The extensive mathematical subroutine library greatly enhances the usability of SYSTEMS 86 software. It supports both FORTRAN IV and assembly language programs and is, itself, coded in assembly language to provide maximum efficiency and speed. The Math Library includes the full set of over 80 subroutines for single- and double-precision, fixed- and floating-point, as well as complex calculations. A special package for matrix manipulation is also provided.

System Generation

SYSTEMS 86 is capable of handling a wide assortment of tasks ranging from a small real-time job to the largest batch processing job. So, to enable SYSTEMS 86 to handle varied workloads with maximum efficiency, we provided the System Generation Program (SYSGEN).

SYSGEN tailors a software system to installation requirements. The software system tailored by SYSGEN operates at maximum efficiency and speed. It uses only those program elements needed, rather than all available elements. Under control of the operating system, SYSGEN develops a unique software system by processing a set of directives that specify complete hardware and software parameters.

Hardware Diagnostics

No computer system can operate efficiently unless its diagnostic programs allow the user to rapidly isolate malfunctions in all major hardware elements and peripheral devices. SYSTEMS 86 offers this capability as well as permitting the diagnostics to be run as tasks under the Real-Time Monitor while in a quiescent state.

COBOL

SYSTEMS 86 is just as adept at solving your business-oriented problems as it is at handling your simulation, test, and control tasks. We therefore plan to provide a COBOL compiler. Full specifications and operating information for SYSTEMS 86 COBOL will be presented in a brochure to be released in the future.

SYSTEMS 86 provides a complete selection of peripheral devices

Card Readers

Model 6211 — Reads 400 cards per minute
Model 6212 — Reads 1000 cards per minute

Card Reader/Punch

Model 6221 — Reads 500 cards per minute. Punches 100 to 460 cards per minute.

Paper Tape System

Model 6103 — Reads eight-level tapes at 600 characters per second. Punches eight-level tapes at 110 characters per second. Drives tape in forward or reverse directions at 62 inches per second.

Magnetic Tape Transports

Model 6511 — Channels: 7
Recording density:
Program-selectable at 556 or 800 characters per inch
Read/write speed:
75 inches per second

Model 6512 — Channels: 9
Recording density:
800 characters per inch
Read/write speed:
75 inches per second

Model 6521 — Channels: 7
Recording density:
Program-selectable at 556 or 800 characters per inch
Read/write speed:
150 inches per second

Model 6522 — Channels: 9
Recording density:
800 characters per inch
Read/write speed:
150 inches per second

Magnetic Tape Controllers

Model 6518 — Controls up to eight Model 6511 Tape Transports
Model 6519 — Controls up to eight Model 6512 Tape Transports
Model 6528 — Controls up to eight Model 6521 Tape Transports
Model 6529 — Controls up to eight Model 6522 Tape Transports

Movable Head Disc Files

Model 6411 — Capacity: 6 million bytes
Average access time:
62.5 milliseconds
Transfer rate:
156,500 bytes per second

Model 6412 — Capacity: 24 million bytes
Average access time:
62.5 milliseconds
Transfer rate:
313,000 bytes per second

Fixed Head Disc Files

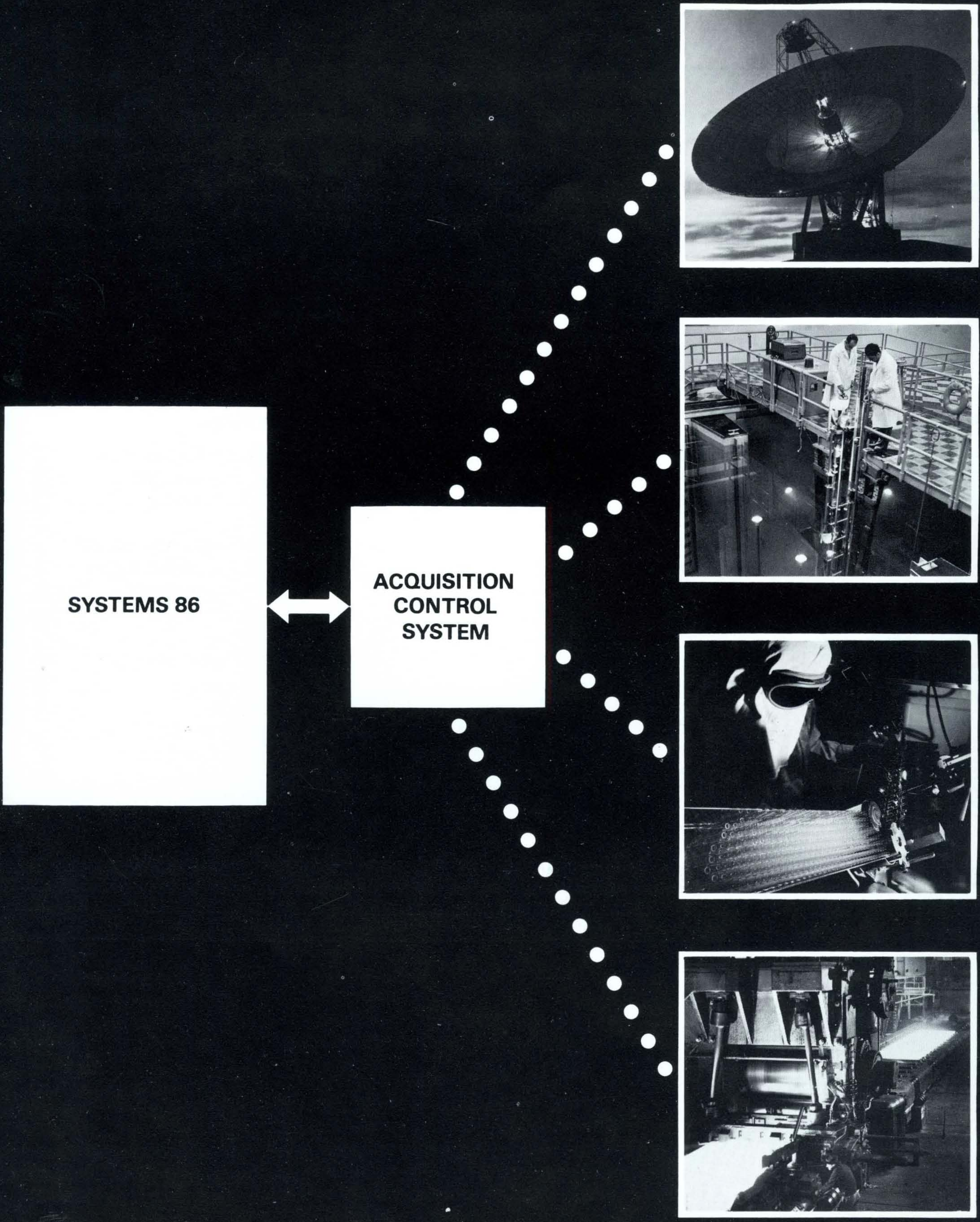
Model 6421 — Capacity: 500,000 bytes
Average access time:
8.6 milliseconds
Transfer rate:
260,000 bytes per second

Model 6422 — Capacity: 1 million bytes
Average access time:
8.6 milliseconds
Transfer rate:
260,000 bytes per second

Model 6423 — Capacity: 2 million bytes
Average access time:
8.6 milliseconds
Transfer rate:
260,000 bytes per second

Line Printers

Model 6361 — Prints 300 lines per minute
Model 6362 — Prints 600 lines per minute



SYSTEMS 86

ACQUISITION CONTROL SYSTEM

The Acquisition Control System interfaces SYSTEMS 86 to real-time applications

Applications such as simulation, industrial control, and test automation have several common requirements. Data must be gathered, formatted, processed, monitored, logged, and distributed. All this must be done in real time. The data may be in many forms. Received from many places. Distributed to many remote locations.

Meeting these requirements is an enormous task. You need a fast, versatile computer — one that can handle large volumes of data in real time. The computer must be equipped with an interface tailored to handle the types of signals and data used by a variety of equipment.

SYSTEMS 86 is the computer that will do the job for you . . . faster, more efficiently, and more economically than any other computer in its class.

We've developed a flexible Acquisition Control System (ACS) for SYSTEMS 86, so it can handle data exchanges with industrial machinery, data collection devices, measurement instruments, sensors, and just about any other conceivable type of analog or digital device. The ACS controls the data, channels it to intended destinations, schedules priority, performs data logging and interval timing, and much more.

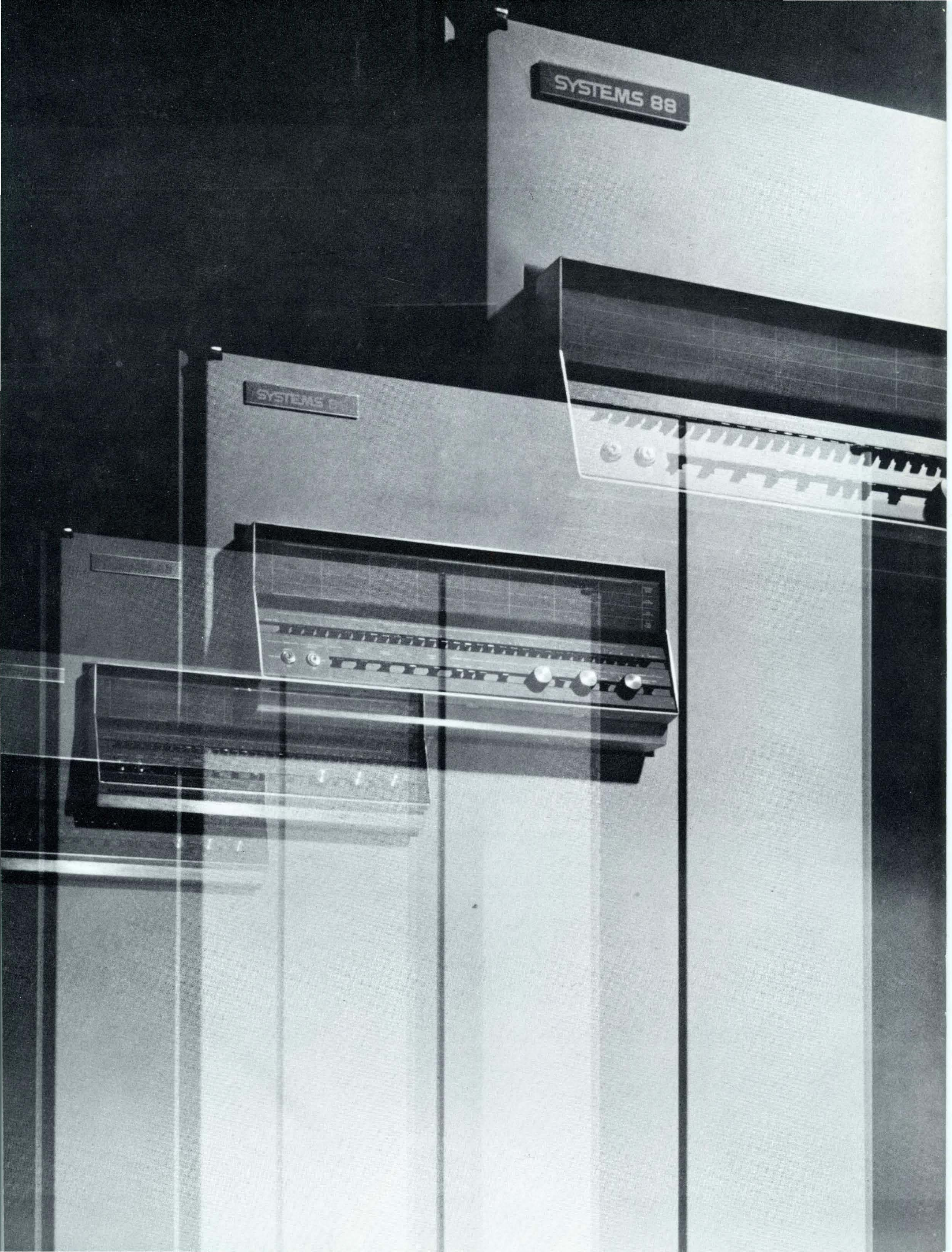
The ACS is completely modular. You can select the capabilities needed to match your applications requirements precisely, even if your application is unlike any other.

The ACS comes complete with the electronics needed for interfacing and data control, including input/output signal-conditioning circuits, low- and high-level multiplexers, relay multiplexers, digital-to-analog converters, analog-to-digital converters, logging printers, an interval timer, power supplies, and an applications-oriented operator's panel.

You can choose up to eight separate control subsections, each having a unique priority level. And you can assign each subsection the job of handling any one of the following:

Up to 1024 high-level analog inputs
Up to 768 low-level analog inputs
Up to 1024 wide-range relay analog inputs
Up to 64 voltage or current analog inputs
Up to 64 16-bit discrete word outputs
Up to 64 32-bit discrete word outputs
Up to 64 logging printers

The ACS fits right in with your SYSTEMS 86 configuration, because all electronics are packaged in standard SYSTEMS peripheral cabinets. And the entire ACS interfaces with just one device controller channel in your SYSTEMS 86.


SYSTEMS 88
SYSTEMS 88
SYSTEMS 88

SYSTEMS 88...the multiprocessor counterpart of SYSTEMS 86

We've given SYSTEMS 86 unmatched speed, economy, versatility, and usability to give you a single-processor computer with the best price/performance available. But a single-processor computer can't fill the bill for everyone. Some users have intricate real-time applications that can be handled faster and more efficiently by sharing the workload among several central processors.

For these users we offer the SYSTEMS 88 multiprocessor, which combines all the outstanding performance capabilities of SYSTEMS 86 with the advantages of greater throughput, higher efficiency, and increased workload capability.

SYSTEMS 88 consists of any combination of up to four Central Processor Units (CPU's) and Direct Memory Access units (DMA's). Each DMA provides a direct link between memory and an Automatic Input/Output System (AIOS). These units share up to 128K words of memory. Users get the advantages of centralized storage, shared facilities, and the flexibility, power and back-up of multiple CPU's. SYSTEMS 88 provides maximum throughput because its CPU's perform computations concurrently. It stores all programs and data in one accessible memory to avoid storage redundancy. The memory provides the link for inter-processor communications.

Each CPU and AIOS can access stored data and programs without conflict, because a unique priority level is assigned to each CPU and AIOS. SYSTEMS 88 comes complete with multiport/page memory protect hardware. The user can dynamically specify each CPU's and AIOS's ability to read, write, and execute the content of memory. The user can close memory ports, prevent execution of special instructions, and limit access to selected memory pages . . . all under program control.

SYSTEMS 88 provides the system security necessary in a multiprocessing environment. To guarantee this security, one CPU is assigned the job of managing memory access privileges. It's called the master CPU. Selection of the master CPU is made by plugging a special module into any CPU mainframe. The user can shift this module from one CPU to another. All other CPU's and AIOS's within SYSTEMS 88 are subordinate to the master CPU. Their ability to access memory and execute certain instructions depends on the access privileges granted by the master CPU.

SYSTEMS 88 offers an uncommon degree of flexibility. Flexibility to operate in a variety of multiprocessing environments. Flexibility to deal with changing requirements. And flexibility to grow.

Each SYSTEMS 88 CPU can be configured to handle a particular job in your multiprocessing environment by selecting the necessary options: floating-point hardware, additional device controller channels, additional interrupt levels, and peripherals.

You choose the number of CPU's, AIOS's, and memory modules that provide just the right multiprocessing capabilities to meet your present requirements. If you need more later, just add them on. Increasing the size and capabilities of SYSTEMS 88 is easy because hardware and software are completely modular.

Start with what you need to handle your present workload. As your workload expands, expand your hardware and software. In an orderly manner, and at the pace that suits you.

SYSTEMS 86

SYSTEMS 86 Specifications

Word size

32 bits (plus 4 parity bits)

Data size

Bit, byte, halfword, word, and doubleword

Memory size

8K (8, 192 words) expandable in 8K increments to 128K (131,072 words)

Memory cycle time

600 nanoseconds

Registers

Eight general-purpose
Instruction register (I)
Transfer register (T)
Program status word register (PSWR)

Standard machine traps

Unimplemented instruction
Non-present memory
Power fail-safe/auto start
System override
Privileged violation

Standard priority interrupt levels

Console keyboard/printer
Memory parity
Console
Real-time clock
Arithmetic exception
Call monitor

Fixed-point computation time

Add/subtract	1.2 μsec (single precision) 1.8 μsec (double precision)
Multiply	6.6 μsec (single precision)
Divide	10.8 μsec (single precision)

Floating-point computation time

Add/subtract	2.4 μsec (single precision) 3.0 μsec (double precision)
Multiply	6.6 μsec (single precision) 11.4 μsec (double precision)
Divide	11.4 μsec (single precision) 19.8 μsec (double precision)

Standard input/output device

KSR-33 keyboard/printer

Power requirements

220 volts, single phase, 60 Hz

Size (mainframe)

66 inches high, 51 inches wide, and 31 inches deep

Approximate weight (mainframe with 8K memory)

800 pounds

The SYSTEMS 86 high-speed instruction set lets users write programs that are shorter, faster, and occupy less storage space

MNEMONIC	OPERATION	EXECUTION SPEED (μsec)
LOAD/STORE		
LB	Load Byte	1.2
LH	Load Halfword	1.2
LW	Load Word	1.2
LD	Load Doubleword	1.8
LMB	Load Masked Byte	1.2
LMH	Load Masked Halfword	1.2
LMW	Load Masked Word	1.2
LMD	Load Masked Doubleword	1.8
LNB	Load Negative Byte	1.2
LNH	Load Negative Halfword	1.2
LNW	Load Negative Word	1.2
LND	Load Negative Doubleword	1.8
LI	Load Immediate	0.6
LEA	Load Effective Address	1.2
LCS	*Load Control Switches	0.6
LF	Load File	1.2 to 5.4
STB	Store Byte	1.2
STH	Store Halfword	1.2
STW	Store Word	1.2
STD	Store Doubleword	1.8
STMB	Store Masked Byte	1.2
STMH	Store Masked Halfword	1.2
STMW	Store Masked Word	1.2
STMD	Store Masked Doubleword	1.8
STF	Store File	1.2 to 5.4
ZMB	Zero Memory Byte	1.2
ZMH	Zero Memory Halfword	1.2
ZMW	Zero Memory Word	1.2
ZMD	Zero Memory Doubleword	1.8
ZR	*Zero Register	0.6
FIXED-POINT ARITHMETIC		
ADMB	Add Memory Byte	1.2
ADMH	Add Memory Halfword	1.2
ADMW	Add Memory Word	1.2
ADMD	Add Memory Doubleword	1.8
ADR	*Add Register to Register	0.6
ADRM	*Add Register to Register Masked	0.6
ARMB	Add Register to Memory Byte	1.8
ARMH	Add Register to Memory Halfword	1.8
ARMW	Add Register to Memory Word	1.8
ARMD	Add Register to Memory Doubleword	3.0
ADI	Add Immediate	0.6
SUMB	Subtract Memory Byte	1.2
SUMH	Subtract Memory Halfword	1.2
SUMW	Subtract Memory Word	1.2
SUMD	Subtract Memory Doubleword	1.8
SUR	*Subtract Register from Register	0.6
SURM	*Subtract Register from Register Masked	0.6
SUI	Subtract Immediate	0.6
MPMB	Multiply by Memory Byte	6.6
MPMH	Multiply by Memory Halfword	6.6
MPMW	Multiply by Memory Word	6.6
MPR	*Multiply Register by Register	6.6
MPI	Multiply Immediate	6.6
DVMB	Divide by Memory Byte	10.8
DVMH	Divide by Memory Halfword	10.8
DVMW	Divide by Memory Word	10.8
DVR	*Divide Register by Register	10.8
DVI	Divide Immediate	10.8
ES	*Extend Sign	0.6
RND	*Round Register	0.6
FLOATING-POINT ARITHMETIC		
ADFW	Add Floating-Point Word	2.4 to 3.6
ADFD	Add Floating-Point Doubleword	3.0 to 4.8
SUFW	Subtract Floating-Point Word	2.4 to 3.6
SUFD	Subtract Floating-Point Doubleword	3.0 to 4.8
MPFW	Multiply Floating-Point Word	6.6
MPFD	Multiply Floating-Point Doubleword	11.4
DVFW	Divide Floating-Point Word	11.4
DVFD	Divide Floating-Point Doubleword	19.8
ANMB	AND Memory Byte	1.2
ANMH	AND Memory Halfword	1.2
ANMW	AND Memory Word	1.2
ANMD	AND Memory Doubleword	1.8
ANR	*AND Register and Register	0.6
ORMB	OR Memory Byte	1.2
ORMH	OR Memory Halfword	1.2
ORMW	OR Memory Word	1.2
ORMD	OR Memory Doubleword	1.8
ORR	*OR Register and Register	0.6

MNEMONIC	OPERATION	EXECUTION SPEED (μsec)
ORRM	*OR Register and Register Masked	0.6
EOMB	Exclusive OR Memory Byte	1.2
EOMH	Exclusive OR Memory Halfword	1.2
EOMW	Exclusive OR Memory Word	1.2
EOMD	Exclusive OR Memory Doubleword	1.8
EOR	*Exclusive OR Register and Register	0.6
EORM	*Exclusive OR Register and Register Masked	0.6
BIT MANIPULATION		
SBM	Set Bit in Memory	1.8
SBR	*Set Bit in Register	0.6
ZBM	Zero Bit in Memory	1.8
ZBR	*Zero Bit in Register	0.6
ABM	Add Bit in Memory	1.8
ABR	*Add Bit in Register	0.6
TBM	Test Bit in Memory	1.2
TBR	*Test Bit in Register	0.6
COMPARE/BRANCH		
CAMB	Compare Arithmetic with Memory Byte	1.2
CAMH	Compare Arithmetic with Memory Halfword	1.2
CAMW	Compare Arithmetic with Memory Word	1.2
CAMD	Compare Arithmetic with Memory Doubleword	1.8
CAR	*Compare Arithmetic with Register	0.6
CI	Compare Immediate	0.6
CMMB	Compare Masked with Memory Byte	1.2
CMMH	Compare Masked with Memory Halfword	1.2
CMMW	Compare Masked with Memory Word	1.2
CMMD	Compare Masked with Memory Doubleword	1.8
CMR	*Compare Masked with Register	0.6
BU	Branch Unconditionally	0.6
BCT	Branch Condition True	0.6
BCF	Branch Condition False	0.6
BFT	Branch Function True	0.6
BL	Branch and Link	0.6
BIB	Branch after Incrementing Byte	0.6 to 1.2
BIH	Branch after Incrementing Halfword	0.6 to 1.2
BIW	Branch after Incrementing Word	0.6 to 1.2
BID	Branch after Incrementing Doubleword	0.6 to 1.2
BRI	Branch and Reset Interrupt	1.2
REGISTER TRANSFER		
TRR	*Transfer Register to Register	0.6
TRRM	*Transfer Register to Register Masked	0.6
TRP	*Transfer Register to Protect Register	1.2
TPR	*Transfer Protect Register to Register	1.2
TRN	*Transfer Register Negative	0.6
TRNM	*Transfer Register Negative Masked	0.6
TRC	*Transfer Register Complement	0.6
TRCM	*Transfer Register Complement Masked	0.6
XCR	*Exchange Registers	0.6
XCRM	*Exchange Registers Masked	0.6
TRSW	*Transfer Register to PSWR	0.6
SHIFT OPERATION		
NOR	*Normalize	1.2 to 6.0
NORD	*Normalize Double	1.8 to 6.6
SCZ	*Shift and Count Zeros	1.2 to 6.6
SLA	*Shift Left Arithmetic	1.2 to 6.6
SLL	*Shift Left Logical	1.2 to 6.6
SLC	*Shift Left Circular	1.2 to 6.6
SLAD	*Shift Left Arithmetic Double	1.2 to 6.6
SLLD	*Shift Left Logical Double	1.2 to 6.6
SRA	*Shift Right Arithmetic	1.2 to 6.6
SRL	*Shift Right Logical	1.2 to 6.6
SRC	*Shift Right Circular	1.2 to 6.6
SRAD	*Shift Right Arithmetic Double	1.2 to 6.6
SRLD	*Shift Right Logical Double	1.2 to 6.6
CONTROL		
EXR	Execute Register	0.3
EXRR	Execute Register Right	0.3
EXM	Execute Memory	0.6
HALT	*Halt	0.6
WAIT	*Wait	0.6
NOP	*No Operation	0.0 to 0.6
CALM	*Call Monitor	0.6
INTERRUPT CONTROL		
EI	Enable Interrupt	0.6
DI	Disable Interrupt	0.6
RI	Request Interrupt	0.6
AI	Activate Interrupt	0.6
DAI	Deactivate Interrupt	0.6
INPUT/OUTPUT		
CD	Command Device	0.6
TD	Test Device	1.2

*Indicates Halfword Instruction

Specifications subject to change without notice.

SYSTEMS
ENGINEERING LABORATORIES
6901 WEST SUNRISE BLVD., FT. LAUDERDALE, FLORIDA 33313

0869
Printed in U.S.A.