Single-chip microprocessors open up a new world of applications

Having freed system designers from the constraints of hard-wired logic, microprocessors are achieving new highs in performance and radically reducing the cost of computer control

by Laurence Altman, Solid State Editor
The first single-chip high-performance microprocessors are rolling off the production lines, splendid confirmation of the power of large-scale integration. These second-generation microprocessors, together with a handful of matched memory and logic circuits, provide equipment designers with the full power of computer hierarchy at unprecedentedly low cost—about the price of a dozen medium-scale TTL integrated circuits.

Inexpensive computer control—that is the miracle of the new microprocessor. All the other advantages inherent in its innovative organization are secondary, even the versatility that results from the ability to simulate logic with software and to reprogram corrections and system changes quickly with no changes in hardware.

The implications of such cheap, distributed one-quarter-horse computer power are only beginning to be understood. A Teletype computer terminal capable of transmitting 300 bits of data per second can now be implemented with 12 LSI packages costing less than $300. Traffic-light controllers can be built with 12 microprocessor family packages, where an equivalent TTL design requires 200. A simple gas-pump meter needs one microprocessor and only nine other packages, an electronic scale needs eight chips. A digital instrument panel displaying five functions can be built from a five-chip microprocessor family, plus display circuits. A microprocessor control system already is operating in experimental automobiles, monitoring dozens of operational parameters at a potential component cost of less than $200. No other electronic method could perform these functions at a practical cost.

Indeed, the microprocessor is really the first truly general-purpose LSI logic device—calculator chips are decidedly special-purpose. That's because the microprocessor, replacing hard-wired logic, offers the twin advantages of LSI circuitry and programability.

Software-implemented logic, few packages, low power-supply and cooling requirements, and few pc cards and connectors simplify system re-evaluation, re-design, and testing. They also reduce assembly costs and inventory requirements.

Using software programs to affect the behavior of the processor instead of hardware interconnections may be an unfamiliar technique to many circuit designers, but pays off by boosting system performance. A typical software program consists of a series of orders or commands to the processor stored in a companion read-only memory. Since ROMs are easy to program—and programable ROMs easy to reprogram—the microprocessor's behavior can be much more conveniently adapted to a changed application than if extensive, time-consuming changes in hardware were necessary. New designs, too, can be turned around faster because a standard microprocessor architecture can be used along with a different ROM program for each new application.

Designing systems with microprocessors is still largely
uncharted, but some rules of thumb are beginning to emerge. Since a complete microprocessor system requires from five to 50 ICs—including clocks, control logic and memory, and peripheral buffering—microprocessors should only be considered for a sequential digital design requiring more than 50 hard-wired logic ICs. Such a design will have more than a trivial number of steps to its logic flowchart and will also have some logic or arithmetic data-processing requirements. At the same time, the speed cannot be too fast, say, not more than 5 to 10 microseconds per instruction.

These system requirements, clearly less stringent than those handled by today's minicomputers, would cost $2,000 to $20,000 to build with hard-wired logic. But with microprocessors costing a 10th of that, their range is almost inexhaustible: point-of-sale terminals, electronic cash registers, inventory-control systems, credit-card verification systems, process, numeric, and machine control, automatic test systems, digital instruments, traffic controllers, communications systems, peripheral controllers, navigational systems, game machines. This is the domain of the microprocessor.

**How to choose a microprocessor**

Selecting a central processing unit (CPU), the major component in a microcomputer design, is a matter of deciding the best way to process the data. For example, data word length may be fixed by the processor design or it may be variable if the design allows multiple processor chips in parallel. A variable data word length is to be preferred when the needs of a variety of applications must be satisfied. For instance, a 16-bit CPU chip could be programed into 4-bit words for BCD display control, calculators, or cash registers, 8-bit words for CRT terminals or data concentrators, 12-bit words for handling the output of a-d converters, 16-bit words for general-purpose processing, or even 24- to 32-bit words for high-accuracy or high-throughput applications.

Instruction power is the next feature to watch out for in a CPU chip. Because the power of individual instructions and methods of counting may vary widely, the number of instructions in the set executed by a microprocessor is a poor index of its usefulness. The only realistic method of comparing instruction sets is to experiment with programs typically required for the intended applications and to compare the execution times and number of bits of storage they use.

Often overlooked in choosing a microprocessor is its interface structure—that part of the CPU which connects the arithmetic and logic unit and the control memory with the input/output peripheral circuitry. Clearly, this structure should adapt easily to a variety of system parameters without imposing a high overhead in hardware or software. The application may demand anything from a simple low-cost bus (either parallel or serial) having separate input, output, and address lines and heavily dependent for its control on the processor, to a sophisticated, high-speed, bidirectional bus with addresses and data multiplexed over the same lines. For maximum flexibility, look for provisions for input/output control, which allow convenient interfacing with peripheral components of varying response time. On the other hand, fixed I/O timing may provide higher I/O speed. In any case, the microprocessor I/O circuitry should directly interface with the 5-volt bipolar logic required to drive I/O lines; if not, buffers will be necessary, adding expense and needing more in the way of power requirements and board space.

Since the memory is often a major portion of the system cost, its selection is nearly as crucial as the CPU's. Read/write memories (random-access memories or RAMs) are best used for variable data storage and for program storage during program development. Programs for prototype or preproduction systems are often stored in a programmable ROM, while a ROM is used during high-volume production.

**What's available**

The rush with which microprocessor devices are appearing—and will continue to appear throughout the year, according to semiconductor manufacturers—is a tribute to the intensity of the demand for them. They fall into three classes.

The pioneering 4-bit microprocessor systems were

---

**What a microprocessor is . . .**

... but first, what it isn't. A microprocessor is not a computer but only part of one. To make a computer out of a microprocessor requires the addition of memory for its control program, plus input and output circuits to operate peripheral equipment. Also, the word is not short for microprogramable central processing unit. For, though some microprocessors are controlled by a microprogram, most are not.

What a microprocessor is, then, is the control and processing portion of a small computer or microcomputer. Moreover, it has come to mean the kind of processor that can be built with LSI MOS circuitry, usually on one chip. Like all computer processors, microprocessors can handle both arithmetic and logic data in a bit-parallel fashion under control of a program. But they are distinguished both from a minicomputer processor by their use of LSI with its lower power and costs, and from other LSI devices (except calculator chips) by their programmable behavior.

In short, if a minicomputer is a 1-horsepower unit, the microprocessor plus supporting circuitry is a 1/4-hp unit. But as LSI technology improves, it will become more powerful. Already single-chip bipolar and CMOS-on-sapphire processors are being developed that have almost the capability of the minicomputer.
built largely with p-channel MOS calculator technology, examples being Intel Corp.'s MCS-4, Rockwell International Corp.'s PPS-4 system, and Microsystem International Ltd.'s MC-1. Next came the 8- and 16-bit p-channel processor sets that are extensions of the early 4-bit units. These devices, while intended for applications up to the minicomputer level, generally require either multichip CPUs or considerable peripheral circuitry. Intel's MCS-8 and National Semiconductor Corp.'s IMP-16 are the most popular examples of each kind. Into the third and newest class, considered by many to be the second generation of microprocessors, fall the new n-channel 8-bit systems like Motorola's MC6800 and Intel's 8080 chips, which, together with matched memory and input/output circuit interfaces, form a completely self-contained large-capacity microcomputer family of chips (see pp. 88–100).

Introduced late in 1972, the 4-bit microprocessor units were the first to provide the microprogrammable parallel processing required in many keyboard and slow-throughput terminal and process-control applications. Indeed, using as few as two devices, like Intel's 4004 CPU chip plus a 1,024-bit ROM, a 4-bit microprogrammed dedicated computer could cost less than $50.

Intel's basic MCS-4 system, however, was designed for general applications. Its single-chip CPU performs all control and processing functions and interfaces directly with ROMs, which store microprograms and data tables, and RAMs, which store data and pseudo-instructions. This system communicates with input/output devices, found here on each ROM and RAM chip. In addition, a 10-bit parallel shift register is provided to expand the system's I/O capability. Thus just four chips are needed for complete microcomputer capability.

Even this relatively simple 18-pin 4-bit package provides up to 45 instructions, cycling in 10.8 microseconds with standard two-phase clock operation. The system can drive up to 4,096 8-bit ROM words (16 chips), 1,280 4-bit RAM characters (16 chips), 128 I/O lines (without a shift register), and unlimited I/O capability with shift registers. And, adding even more to the MCS-4's flexibility and further accelerating the design cycle, the CPU and RAMs may be interfaced with conventional electrically programmable and erasable ROMs, allowing fast program development and quick prototype realization.

Equally versatile is Rockwell's 4-bit PPS system, which comes complete with five compatible support circuits: a 256-by-4-bit RAM, a 1,024-by-8-bit ROM, a RAM-ROM combination chip containing a 704-by-8-bit ROM and 76-by-4-bit RAM, an I/O buffer, and a multiphase clock generator (Fig. 2).

The CPU in this system can directly address up to
4,096 8-bit ROM words and as many 4-bit RAM data words over its 12-bit parallel address bus. This large number of data words gives this family the capability of a 4-bit minicomputer. The basic instruction set contains 50 instructions, and instruction fetch and execution time is a speedy 5 microseconds.

Apart from power and clock-signal requirements, 21 multiplexed lines interconnect the CPU with ROM, RAM, and I/O circuits. These lines, as shown in Fig. 2, are functionally grouped into 12 parallel address lines, eight parallel data lines, and one write command and I/O enable line. The address lines originate at the CPU and are time-multiplexed within it to provide direct addressing capability for up to 4,096 locations on both ROM and RAM. In addition, the ROM has two chip-select inputs and the RAM has one chip-select input, which may be directly controlled by discrete outputs from the CPU or I/O circuits to expand on memory without the need for auxiliary circuitry.

The move to eight bits

For greater capacity, both Intel and Rockwell are extending their 4-bit p-channel systems to 8-bit capability. The Intel MCS-8 is an 8-bit fixed instruction set and consists of a single MOS chip in an 18-pin DIP. Also on the chip is an 8-bit data/address I/O bus that interfaces the processor with external memories. It contains a total of 14 instructions, which can control a lot of memory and I/O circuitry. It does, however, require substantial TTL circuitry to implement most 8-bit systems, a condition corrected by Intel's newer, very flexible 8080 chip.

Rockwell's soon-to-be-announced 8-bit system is a completely self-contained system. Prototypes will be operational in July, with deliveries to begin late in 1974. The PPS system consists of a CPU, RAMs, ROMs, clock generator, a direct-memory-access controller, and an assortment of general-purpose I/O devices, all of them accessible on the same bidirectional data/instruction bus (Fig. 3). This bus provides 8-bit parallel communication within the computer at a rate of 500 kilohertz—a most important factor for systems savings.

With this setup, more than 90 instructions can be executed in 4 microseconds each, which covers a ROM access for instruction fetch and a RAM access for data fetch, as well as the processing of the data. In addition, the system can be supplemented with special-purpose and custom I/O devices for specific applications. Examples are a 1,200-baud modem device and a keyboard/display controller with independent input and output buffers.

In Microsystem International's 4-bit system, the CPU contains two memory pointers—the program counter and a data pointer—which allow logical as well as physical separation of program and data. Both pointers are 12 bits long and can directly address 4,096 memory locations. Each memory location contains 4 bits of data. Up to 34 kilobytes may be addressed over field switching in the typical MC-1 microcomputer.

Even more powerful is National Semiconductor's multi-chip CFCP CPU, shown in Fig. 4, from which National's IMP-16 systems can be built. It can provide computing power that ranges from simple 4-bit keyboard address capability right up through full 16-bit minicomputer capability. In IMP-16 systems, processing is done by four 4-bit arithmetic logic units controlled by microprogrammable ROMs. With this arrangement, data exchange happens over a 16-bit-wide data bus, while I/O and control operations take place over a set of 16 general-purpose addressable registers (called FLAGS).

Consisting of a five chips, this CPU is contained on a board along with 256 words of random-access and 512 words of read-only on-card memory. Also available on the card are external interface circuits such as an address bus, data input and output buses, additional control flags, system timing lines, and an interrupt input.

In essence, the IMP-16, which is expandable to 32 bits, is composed of four 4-bit ALU slices, each with control registers, ALU logic, and I/O data lines. The control ROM contains all control logic and microinstruction storage necessary to control the ALU chips. A total of 43 instructions is available though, if one considers the multiplicity of branch conditions, I/O FLAGS, and general-purpose accumulators in the system, the effective instruction count comes to well over 150. (For smaller systems, National now also provides 4-bit and 8-bit versions of the IMP-16.)

The microprocessors now appearing from a good
many semiconductor manufacturers take full advantage of the knowledge gained in the past two years and incorporate those features that have proven most effective for the greatest variety of applications. What is obviously wanted is a single-chip 8-bit CPU device, offering 70 instructions or more, at speeds above 1 megahertz, with an extremely flexible input and output structure, and requiring only a few support memory and logic circuits to do most 8-bit jobs. Above all, these support circuits must be easy to use—that is, work directly with the CPU without requiring additional buffers and power supplies. Moreover, the CPU must be able to work directly with standard memory products. This adds up to a need for a self-contained 8-bit microcomputer set of chips—one CPU, and maybe five or six matched memory and logic hang-on packages.

The new generation

To build them, most manufacturers have settled on n-channel technology because it can pack many memory and logic structures onto one CPU chip, provides high capacity, and operates at high speed from 5-V (TTL) power supplies. Since the new n-channel memories would be directly compatible with such CPU chip, ease of use falls out automatically.

Besides the 8-bit n-channel microprocessors that have already arrived from Intel and Motorola, standard n-MOS products are being planned by Texas Instruments, American Microsystems, Signetics, National, Fairchild, General Instruments, and Western Digital. Rockwell and MOS Technology Inc. are staying with p-channel, and RCA has already announced a C-MOS microprocessor prototype.

The Signetics device, called PIP for programmable integrated processor, is a single-chip 8-bit unit in a 40-pin DIP. The customary address logic, control memory and ALU are organized around a bidirectional 8-bit data bus, and there are also 15 address lines for handling external memory and I/O circuitry. In the PIP device, the address logic handles all instructions. It also includes a return address stack that lets eight subroutine levels be stacked.

As for RCA’s C-MOS microprocessor, the two-chip 8-bit design has all the advantages that C-MOS circuits offer. (A single-chip version is under way.) It can operate off power supplies providing anywhere from 5 to 15 V, it

4. Sixteen-bit systems. The block diagram of National Semiconductor’s IMP-16 microprocessor system shows that the system is made up of four 4-bit ALU slices with control registers, ALU logic, and I/O data lines. Altogether 43 instructions are available.
has high noise immunity, and it dissipates power at the microwatt level. Needing so little power and being easy to use, the chip set will be particularly useful for low-cost high-volume applications, and the C-MOS process could make it especially attractive for use in cars.

The microprocessor will come in a 40-pin package and can be used with any mixture of RAM, ROM, and peripheral I/O circuits. It is capable of addressing up to 65,536 8-bit bytes, so that quite large and flexible processing systems could be implemented, even though it has only 25 instructions. What's more, when operating from a typical supply of 10-12 V, the machine cycle time is a respectable 3 microseconds. And using a standard 1-microsecond RAM, the chip set has a maximum 6-μs fetch-execute time for any instruction.

Another manufacturer favoring C-MOS for microprocessors is Intersil, which is developing a 12-bit single-chip CPU to work with its C-MOS and n-channel memories. Intersil chose a 12-bit structure so that designers could use software programs that already exist for PDP-8A systems—and in fact, when combined with appropriate memory and I/O hang-ons, the 12-bit unit can perform all the MSI functions of the PDP-8A mini-computer but needs only a fraction of the packages.

The implications of all this activity are tremendous. Indeed, many observers feel that the MOS microprocessor families now just emerging will have a bigger impact on the electronics industries than any other semiconductor device has had so far. Quite soon, too, improved LSI structures should result in single-chip microcomputers combining the CPU, I/O, and memory in one LSI device. Moreover, the same sort of excitement is being generated around the bipolar LSI processor work that's now a priority in many semiconductor laboratories; for that technology, too, points to full-instruction minicomputer capability on a few LSI chips, but at even faster speed.

The next two articles take a look at the first two n-channel microprocessors—Motorola's M6800 chip set and Intel's 8080 chip. Both are second-generation devices, offering high speed, ease of use, and flexible control capability for many new applications.

5. The rising generation. N-channel silicon-gate technology is bringing enlarged capacity, great versatility, and high speed to today's new microprocessor chips. This Intel 8080 CPU offers altogether 78 instructions in an 8-bit system.
N-channel MOS technology yields new generation of microprocessors

The latest microprocessor chips are faster than p-MOS devices and handle many more peripherals; often, too, as in Motorola's M6800 family, a CPU chip will come in a matched set of memory and input and output chips, simplifying system production.

by Link Young, Tom Bennett, and Jeff Lavell, Motorola Semiconductor Products Inc., Phoenix, Ariz.

The great promise that programable LSI circuits have for all kinds of control applications is fulfilled in the second generation of microprocessors, such as Motorola's and Intel's 8-bit devices. These new n-channel MOS chips have many more instructions and need much less in the way of costly systems circuit support than did the first wave of 4- and 8-bit p-channel systems. Their level of computing power is also high, and they are versatile and easy to use.

The n-channel metal-oxide-semiconductor microprocessors are completely self-contained. They are designed to work directly with a minimum number of memory and peripheral support chips, all of which are supplied in coordinated families to allow them to operate off the same voltage and power-supply conditions as the central processor chip.

A typical set contains the CPU chip, a random-access memory for fast scratch-pad logic control, a read-only memory for storing the system's program parameters, and a set of input and output chips. These input/output chips enable the CPU to control a large variety of industrial and communications equipment: process and manufacturing control systems, peripheral and terminal hardware, parameter-control systems of all types—from microcomputers in the automobile to control systems around the concept of the parallel data bus. Consequently, all memory and peripheral interface adapter (PIA) chips are simply designed to hang on its CPU's eight bidirectional data lines.

1. Eight-bit family. Motorola's M6800 family of components is organized around the concept of the parallel data bus. Consequently, all memory and peripheral interface adapter (PIA) chips are simply designed to hang on its CPU's eight bidirectional data lines.
2. Basic computer. One of the first microcomputer chip sets to have been fabricated comes from Intel. Called the MSC-4, it consists of four simple LSI blocks and provides 45 instructions with an instruction cycle time of 10.8 microseconds.

3. Heavy duty. The first 8-bit microcomputer system, Intel's MSC-B, can interface with over 16,000 8-bit words of read-only, random-access, or shift-register memory. Its drawback: substantial external circuitry is needed for most applications.

4. Sixteen bits on a board. National's IMP-16 architecture supplies a full 16-bit minicomputer capability on a single pc board. Microprogrammable ROMs control the four 4-bit arithmetic logic units that do the processing. Four- and 8-bit versions are also available.

5. Straightforward. The new n-channel microprocessor chips make designing microcomputers simply a matter of choosing a family of matched components. This Intel 8080 system, for example, requires only a half a dozen standard products.
for traffic lights—and anywhere else that random-logic computer control needs optimizing.

These second-generation n-channel units expand considerably on the system benefits offered by the first microprocessors. Over 70 instructions may be available, as against about 40 for the largest p-channel unit. As few as four packages are required to build a complete 8-bit microcomputer. Moreover, in the Motorola family, the M6800, TTL compatibility is achieved with only a single +5-volt power supply, instead of the usual three supplies. Therefore, board space, packaging count, and component costs are reduced, even while system capacity is increased.

**Other benefits to the system**

Consequently, as with all microprocessor system designs, board layouts are simplified. The complex interconnections required for large numbers of conventional ICS are replaced by ROMS. The only interconnect wiring on printed-circuit cards runs between the various address and data buses and input/output devices.

The cost savings are not limited to direct circuit component costs but they extend to other, related, system hardware costs. Connectors can be decreased in number, cabling can be simplified, the card cage can be reduced in size, and so on. Associated indirect costs also fall, of course, since assembly takes less time, documentation is simpler, and maintenance is easier.

Equally important to cost savings in hardware systems is the ability of system engineers to build a proposed design quickly. No hardwire logic need be simulated, optimized, or breadboarded. The logic design portion of the cycle now becomes the manipulation of functional building blocks, where the control sequence takes the form of writing a software program into an external ROM. Breadboarding consists of interconnecting a few LSI packages.

Design changes, too, are simply a case of modifying the control program, in contrast to designing and laying out the logic afresh. The various microprocessor manufacturers offer the use of simulators, so that most of the design can be verified even before it is committed to hardware. This all cuts at least 90% from the design time.

The numerous instructions and system versatility despite the very few packages stems directly from the organization of the new CPU chips. For example, the MC6800 chip is organized around the popular parallel data bus concept (Fig. 1), so that all the memory and peripheral interface chips simply hang on the MC6800's eight bidirectional data lines (16 address lines are provided). Up to 10 LSI chips can be directly attached to the bus for operation up to 1 megahertz. To drive still more peripherals, a bipolar extender can be added.

This direct access to a variety of interface and peripheral equipment, obtained with a minimum of packages (see "The M6800 microprocessing family," p. 92) is a tremendous advance on many of the early microprocessor chip families—even though the first single-chip microprocessor was introduced just two years ago.

Intel's MCS-4 and MCS-8 and Rockwell's PPS-4, which all used p-MOS silicon-gate technology, were excellent starting points, in that they were self-contained sets of circuits requiring no external logic. In the 4-bit MCS-4, for example, the CPU, random-access memory, and read-only memory interfaced optimally as a set (Fig. 2). However, these first microprocessors had major limitations. Selecting correct memory locations required complex address logic; 12-bit addresses needing three 4-bit words had to be multiplexed onto the CPU's input

---

6. *Selling well.* With microprocessor design techniques, systems such as this point-of-sale installation are capable of being implemented with only five or six circuit blocks, which are designed to work directly with the basic CPU family.
matrix. All instructions operated on complex 4-bit data signals, as did even simple word fetches for each instruction. Still more restrictive was the fact that input and output access was serial, not parallel, limiting the number of peripherals that could work with the CPU. Instruction speeds were also slow—it could take 80 microseconds to execute one—and power-supply requirements were complex and costly.

For larger systems, the 8-bit MSC-8 microprocessor chip set (Fig. 3) could be used, but it was not a self-contained system, requiring external TTL for any application. It was, however, quite powerful: the 8008 CPU of that system can interface with 16,384 8-bit words of read-only, random-access, or shift-register memory. It was also quite economical to build: all communication between functional units and the CPU is carried out over the single 8-bit data bus, a sync line, a ready line, an interrupt line, and just three status lines. Its low cost, together with its respectably fast instruction execution time of 12.5 microseconds, makes the 8008 microprocessor still very useful for moderate-performance systems in point-of-sale terminals, credit-card verifiers, calculators, and other keyboard-addressed applications.

It does, however, fall short of being a useful general-purpose microprocessor chip set, primarily because, unlike its 4-bit predecessor, the MCS-4, it is not a system of compatible parts. Indeed, it requires many small-scale packages to build even a moderately powerful system—a simple modem hook-up would need about 50 TTL packages, increasing circuit board area and systems costs.

Another problem is addressing it. True, its 18-pin package saves board space, but it must be multiplexed both for address and data on common input/output pins, which in the end lengthens excessively the time it takes to execute an instruction. Not only that, the need for seven control registers on the CPU chip makes it difficult to manage the logic cycles, limiting subroutines and creating problems in programing and interrupt handling. Finally, its outputs are compatible, not with standard TTL, but with rarely used low-power TTL, so that circuits are needed to boost voltage level in most applications.

Most of these problems were overcome with National Semiconductor’s IMP-16 (Fig. 4), a 16-bit microprocessor set that for the first time provided full mini-
The M6800 microprocessing family

The family comprises five chips: a single-chip central processor unit, a 128-by-8-bit static random-access memory, a 1,024-by-8-bit read-only memory, and one from each of two groups of input/output interface circuits—a peripheral interface chip designed to provide a buffer to terminal and peripheral systems, and a communications interface adapter circuit for interfacing communications hardware. They all operate from one 5-V power supply, and for many applications require far fewer interface packages than other microprocessor sets.

Basis of the M6800 family is the CPU chip (MC6800) packaged in a 40-pin DIP (see figure). Built with ion-implanted, n-channel silicon-gate technology, this chip contains all the functions required for multi-instruction processing: an arithmetic and logic unit, instruction decode and address registers, an instruction register, all of the clock and logic circuits required for timing, and a full complement of data-bus input and output matrices and address bus drivers.

The equivalent of about 120 MSI TTL packages, the chip provides 72 self-contained basic instructions that have decimal and binary arithmetic capability. The variable-length instructions include double-byte operations (such as increment or decrement, load, store and/or compare) and have tri-state outputs, two-accumulator capacity, and enough registers to provide seven addressing modes. A typical instruction time is under 5 microseconds, and there is direct memory access on the chip. Up to 64 bytes of memory can be addressed in any combination of RAM, ROM, or peripheral registers.

All other members of the set (see table) have been designed specifically to work directly with the CPU chip from the same 5-V power supply. The peripheral adapter (MC6820) is a bidirectional unit with two parallel 8-bit outputs that can either drive two peripherals or, if tied together, provide a higher throughput. The adapter can interface with Teletype and display terminals, with cassette and test equipment, with keyboards and control panels, and even with large computers for time-shared expansion of computer capability.

The communications interface adapter (MC6850), on the other hand, couples the processor to most standard modems for communications with other computer systems via telephone lines. For still more system flexibility, it's possible to use without adapters not only the static RAM and ROM in the table but other memories, too.
can work directly with the chip containing the CPU.

The smallness of the package count is dramatically illustrated by the comparison of the breadboard, engineering model, and final chip design of an MC6800-type CPU (see photograph on p. 82). The breadboard, a gate-to-gate implementation of the CPU employing basic gates and flip-flops, needs five 10-by-10-inch boards containing 451 packages. The engineering model is a functional implementation of the design and made extensive use of MSI logic packages and programmable ROMs to reduce package count to a mere 114, packed into a single 10-by-10-in. board by means of today's most effective hardwire logic techniques. Yet all this is replaced by the single 40-pin package containing the CPU chip. The example epitomizes the impact LSI chip design is having on the implementation of complex computer functions.

The new n-channel microprocessors go still further, by addressing themselves to other parts of the system as well. For the families of circuits are designed to minimize assembly costs by reducing the number of auxiliary parts necessary to realize a design.

Consider the block diagram of a typical small terminal, a generalized point-of-sale terminal (Fig. 6). Since every CPU needs several peripheral interfaces, one key to cost-effective designs with a microcomputer lies in the input/output interface.

Indeed, anything that has to interface with a microcomputer ought to be compatible with the data-bus arrangement and with the particular addressing scheme. Moreover, this bus-compatibility requirement holds good for not only in the input/output area but in the memory area as well. Consequently, since a microprocessor is a word-oriented system, more and more word-oriented memories are beginning to appear.

The M6800 family is directed at just these system needs. It includes flexible input/output adapters and word-oriented memories, in addition to the basic microprocessing unit, as indicated in the minimum system configuration of Fig. 7. This system can maintain its 1-MHz level of operation even when expanded to 10 modules (memories, input/output adapters and additional CPUs) on the principal data bus, with no external interface package.

In order to handle applications that require 1-MHz operation with more than 10 modules on the data bus, bus extenders are provided. For systems that do not require 1-MHz operation, up to 30 modules can be added to the data bus without requiring bus extenders—for example, more than 20 modules can be added to the data

---

8. **Communicating.** A boon for communications systems, this microprocessor setup can be implemented using standard communication interface adapters (CIAs). These adapters' function is to give the CPU system access to any standard modem.
One example of how few packages are necessary with microprocessors is given by a typical modem communications system (Fig. 8). Here the asynchronous communication interface adapter performs the basic serializing/deserializing function required to interface the modems with the CPU. It also provides such additional logic capability as start, stop, and parity compensation. It can be used with a line driver/receiver for high-speed data transmission up to 5,000 bits per second, or with standard modems like the single-chip Bell 100 Series low-speed modem. Significantly, the 116 TTL and modem packages formerly required by this system are here replaced by only seven packages. The assembly costs alone are reduced by as much as two thirds.

**Using the microprocessor**

Most microprocessor manufacturers supply the software required to program their devices in a form usable with readily available computer systems. The software programs for the MC6800, for instance, are currently available on the ubiquitous GE time-sharing network. A designer might use them in the sequence shown in Fig. 9. Working with the GE edit program, the designer enters his specific applications program, which is simulated on a cross assembler resident in the same host computer. The cross assembler checks for obvious errors and violations and indicates them to the designer. After the program has been assembled, the designer has two choices—go to hardware directly, or to simulate his system by making use of the large GE host computer containing all the parameters of the particular system. If he chooses to simulate and his program works, he can then go to the hardware stage. If his program does not run, the simulator will pinpoint his problem areas, and he can modify his program and go through the loop again. This process can be continued until the designer is completely satisfied with his program.

In addition, exercisers, hardware, and programs are provided by many manufacturers to verify breadboard operation. In the system that is described in Table 1, the designer chooses the cards required to breadboard his system, plugs them into the machine, cables the input/output cards to his various peripherals, reads his program in through the TTY, or equivalent, network that interfaces to the debugging card. His program is contained in the read/write memory until it is debugged.

Then, in the debugging stage, a panel switch enables the flexible RAM to look like the appropriate ROM. If his program does not run, the exerciser will help him find out why and enable him to modify the program. It's estimated that exercising aids like the off-the-shelf software and the Motorola Exorciser can save the designer from six to 12 man-months by providing him...
with a convenient method of communicating with the microcomputer.

Systems that are based on microprocessors are cheaper to manufacture, require shorter design cycles—and are also easier to modify or upgrade. The personality or function of the system, being determined by a master control program stored in a memory, is changed simply by modifying that program.

In the case of market testing, systems can be adapted in the customer's own environment to meet his needs better. For the first time a manufacturer has the capacity to make his product smarter and add features at any instant simply by expanding his master control pro-

gram. A whole range of products becomes potentially available by simply adding LSI modules with their associated features.

System flexibility mostly depends on the new type of memory used, and here the choice is rich. For example, an inexpensive, volatile read/write buffer memory could be used in conjunction with a cassette or a floppy disk for very low-cost systems requiring moderate speed. For faster systems, such as modem interfaces, ROMs, programable ROMs, or even dynamic RAMs with battery backup could be used. In this area, the emerging 4,096-bit RAMs appear to offer the best speed/cost tradeoff.

In switch to n-MOS microprocessor gets a 2-μs cycle time

Intel's 8-bit successor to its 4-bit p-MOS CPU chip has 30 extra instructions, is 10 times faster

by Masatoshi Shima and Federico Faggin,

Intel Corp., Santa Clara, Calif.

□ The first microprocessors borrowed many desirable architectural features from minicomputers—but not their speed. The 8-bit Intel 8080, however, achieves typical execution times of 2 microseconds, which are comparable to those of many of today's minis.

In so doing, it improves on the speed of its predecessor, the Intel 8008, by an order of magnitude or more, and, since it also adds 30 new instructions to the 48 shared with the 8008, it can be considered the start of a second, more powerful generation.

The key to both improvements is the shift from the 8008's p-channel MOS technology to n-channel MOS. Indeed, the decision to develop the 8080 was taken about 18 months ago, as soon as high-volume production of silicon-gate n-channel devices was feasible. The goal was a single-chip central processing unit (CPU) compatible with but markedly superior to the earlier 8008. The 8080's characteristics were to include:

■ A 10:1 speed improvement over the 8008.
■ None of the known limitations of the 8008 (such as interfacing problems and lack of multiple interrupts).
■ Improved functional capability plus retention of all
1. Comparison. Intel's earlier single-chip microprocessor, the 8008, has a separate scratch-pad memory and address stack (a). In the 8080, these have been combined into the six 16-bit registers (b). The accumulator has been moved into the arithmetic and logic unit, avoiding the use of the internal bus for data transfers between the scratch pad and the ALU during arithmetic and logic operations.
of the various features and instructions of the 8008.

- Economic feasibility—a small chip and conventional packaging.

If the higher mobility of electrons versus holes were the only difference between the n-channel and p-channel technologies, only a 2.4:1 improvement in speed could have been expected. But n-channel's lower threshold allows use of a 5-volt supply for internal logic, with a 4:1 improvement in speed-power product.

There are other contributions to higher speed. The higher substrate concentration of the n-channel starting material, combined with the lower supply voltage, allows channels to be shorter than with p-channel technology, so that input capacitance is lower and size smaller. Finally, lower junction capacitances and lower resistivities of diffusion and polysilicon areas, which result from the n-channel process and the use of substrate bias, reduce the interconnection time constants by a factor of four—and, in logic circuits of this type, one of the limiting speed constraint lies in the electrical properties of the interconnections.

The interfacing requirements were simplified because n-channel technology allows a big reduction of the power dissipation of individual output buffer circuits, so that the 8080 could be packaged in a 40-pin package to include 30 buffers as against the 12 of the 8008. In the 8008, each output buffer sinks two low-power TTL loads (440 microamperes) for a total dissipation of 250 milliwatts. Eight of the 8008 buffers are shared (time-multiplexed) for addresses and data outputs, reducing the number of package pins but increasing the complexity of the interface. The 8080's 30 output buffers, on the other hand, are six times faster, sink 1.9 milliamperes each, and dissipate a total of 150 mw. The 100 mw saved was used to improve the speed of the internal circuits (a 40-pin ceramic package allows a maximum dissipation of about 750 mw, so the power budget was limited).

The layout effort took 18 man-months because it required great care to minimize parasitics and to optimize signal flow for increased speed and smaller size. The result was a 165-by-191-mil chip that is smaller than

---

2. **8080 at work.** When connected in a microprocessor system, the 8080 requires only six external TTL packages, as against the 20 needed by the 8008. The address bus can access up to 64 kilobytes of memory and up to 256 input and 256 output ports.
The real improvement in execution speed and memory storage requirements for a same problems about eight times faster (and five times faster than the 8008-1, a high-speed version of the 8008). However, with the 30 new instructions, the 8080 offers speed improvements on the order of 10:1 to 20:1 with smaller storage requirements—from 95% to 70% for an equivalent program written for the 8080.

The internal organization of the 8080 is shown in Fig. 1b, while Fig. 1a shows the same detail for the 8080.

The most important change concerns the internal memory organization. The 8008 has separate memories: an address stack—eight 14-bit registers which comprise one program counter storing the current effective address and seven others that store the addresses of nested subroutines—and a scratch pad, which contains the 8-bit accumulator and six additional 8-bit registers used for memory addressing and temporary storage of operands. In the 8080, these memories have been combined into a single internal 16-bit-wide memory with paired 8-bit register organization. The 8080's program counter and stack pointer, also each 16 bits wide, replace the 8008's internal address stack.

The 8080 has an internal 3-bit stack pointer, which gives the user up to seven levels of nesting of subroutines. The 8080's 16-bit stack pointer can address up to 64 kilobytes of external stack memory, providing essentially as many nesting levels as needed.

The 8080's accumulator and its associated circuitry have been moved into the arithmetic logic unit (ALU) section, to speed up the operation of the processor (data transfers between memory and ALU on the internal data bus are therefore not required for arithmetic and logic operations). Notice that the 8080 memory is double-ended—information can be transferred from the internal bus 8 bits at a time, while 16-bit transfers can take place from the address register.

**Extra benefits**

This organization yields a number of other new features for the 8080. The most important are:

- New instructions allow the contents of any register pair (B-C, D-E, H-L, or ACCUMULATOR-FLAGS) to be quickly stored and retrieved by being "pushed into" or "popped from" the top of the external memory stack. This is a fast way to save the machine status (the contents of the registers) when an interrupt occurs and then restore the status after the interrupt has been serviced. The stack can also be used as an extension of the internal registers.

- Other new instructions allow easy manipulation of addresses and the memory stack, since the registers B-C, D-E and H-L, and STACK POINTER can be incremented and decremented with 16 bits in parallel.

- The temporary register pair W-Z can be used as a program counter to hold a direct address to quickly load or store H-L or ACCUMULATOR. Also possible are double precision additions between any register pair and H-L.

- Fast, parallel transfers of H-L to PROGRAM COUNTER or STACK POINTER are now possible with a minimum amount of internal control logic.

- The addition of decimal correction to the ALU section enables binary and BCD arithmetic to be performed at about equal speeds.

- The addition of many new, easy-to-use control and
The 8080's inputs and outputs

The Intel 8080 takes four control inputs and generates six control outputs:

- **SYNC**—output; a synchronizing signal that indicates the beginning of each memory cycle.
- **DATA BUS INPUT**—output; a signal that indicates when the data bus is in the receiving mode, i.e., when data is expected by the CPU.
- **READY**—input; a signal to the CPU that valid data is available. If not activated, the CPU enters a WAIT state.
- **WAIT**—output; a signal that acknowledges that the CPU is in the WAIT state.
- **WRITE**—output; a signal that tells the memory and output devices that valid data from the CPU is available on the data bus.
- **HOLD**—input; a signal used by an external device to request access to the CPU address and data bus. Request is granted upon completion of memory access and it is acknowledged on the HOLD ACKNOWLEDGE output pin. The CPU address and data buses become floating (in a high-impedance state), but internally, the CPU completes the execution of the current memory cycle. After that, the CPU idles for as long as HOLD is active. HOLD and HOLD ACKNOWLEDGE can be used for DMA (direct memory access) control and in multiprocessor applications.

**HOLD ACKNOWLEDGE**—output; signals acknowledgment of the HOLD state.

**INTERRUPT REQUEST**—input; the interrupt input is sampled at the end of the current instruction cycle and if the internal software control interrupt enable flip-flop is set, it initiates the interrupt servicing sequence.

**RESET**—input; a signal that clears the content of the program counter so that program execution will start from location zero in memory.

**INTERRUPT ENABLE**—output; a signal that displays the status of the interrupt enable flip-flop.

The CPU also provides eight status bits on the data bus at SYNC time:

- **HALT ACKNOWLEDGE**—a response to the HALT instruction.
- **INTERRUPT ACKNOWLEDGE**—follows the acceptance by the CPU of an interrupt request.
- **INPUT CYCLE**—indicates that the address bus holds the address of an input device.
- **OUTPUT CYCLE**—indicates that the address bus holds the address of an output device.
- **MEMORY READ**—indicates that the data bus will have data coming from memory.
- **M1**—indicates that the current cycle is for fetching the first byte of an instruction.
- **STACK**—indicates that the address bus holds the push-down stack address.
- **WRITE OUTPUT**—indicates that the data bus will have data for memory write or for an output operation.

---

**4. Timing.** The instruction LMB (load the content of register B into the location addressed by the contents of registers H and L) requires two machine cycles (M1 and M2). During M1, the address bus holds the program counter contents, and during M2 it holds the contents of the H-L register pair. DATA BUS INPUT shows when the CPU expects data from the data bus. WRITE shows when data from the CPU is available on the data bus. READY shows that valid data is available to the CPU. The bottom waveform shows the corresponding data-bus actions.
status signals simplifies interfacing, allows direct memory access, and helps in program debugging.

Figure 2 shows how the 8080 interfaces with outside chips to make a microprocessor system. An external crystal-controlled oscillator supplies two non-overlapping clocks, \( \phi_1 \) and \( \phi_2 \). Buffers interface to external address and data buses, and a gate and eight latches set up status bits during sync time. All inputs and outputs are TTL-compatible, with the exception of the two clocks, which require +12 V. A memory and the input-output devices complete the system.

The amount of external interface logic necessary to implement any system depends on that system’s complexity. The minimum requirement is six packages of conventional TTL (the 8080 needs at least 20).

External signals are organized on three buses. An address bus with 16 lines addresses up to 65 kilobytes of memory and up to 256 input and 256 output ports. A bidirectional eight-line data bus carries data to and from memory and I/O ports. A control bus synchronizes the CPU, external memory, and I/O devices, and also has the job of handling interrupts, direct-memory-access (DMA) controls, and CPU status information.

Instructions in the 8080, as in the 8008, use one, two, or three bytes of storage. Each instruction requires from one to five machine (or memory) cycles for fetching and execution. Machine cycles are called \( M_1, M_2, \ldots, M_5 \). Each machine cycle requires from three to five states—\( T_1, T_2, \ldots, T_5 \)—for its completion. Each state has the duration of one clock period (0.5 microsecond). There are three other states (WAIT, HOLD, and HALT) which last one to an indefinite number of clock periods, as controlled by external signals. Machine cycle \( M_1 \) is always the operation-code fetch cycle and lasts four or five clock periods. Machine cycles \( M_2, M_3, M_4, \) and \( M_5 \) normally last three clock periods each.

To understand the basic operation of the 8080, let’s refer to the simplified state diagram shown in Fig. 3, starting at cycle \( M_1 \) and state \( T_1 \).

During \( T_1 \) the content of the program counter is sent to the address bus, \( \text{SYNC} \) is true, and the data bus has status information pertaining to the cycle that is currently being initiated. \( T_1 \) is always followed by another state, \( T_2 \), during which the condition of the \( \text{READY} \) input is tested. If \( \text{READY} \) is true, \( T_3 \) is entered; otherwise, the CPU will go into the wait state (\( T_w \)) and stay there for as long as \( \text{READY} \) is false. \( \text{READY} \) thus allows the CPU to be synchronized to a memory with any access time and to any I/O device. Also, by controlling the \( \text{READY} \) line, the user can single-step through his program.

During \( T_3 \), the data coming from memory is available on the data bus and is transferred into the instruction register (during \( M_1 \) only). The instruction decoder and control sections then generate the basic signals to control the internal data transfers, the timing, and the machine-cycle requirements of the new instructions.

At the end of \( T_3 \), if the cycle is complete, or else at the end of \( T_5 \), the 8080 goes back to \( T_1 \) and enters machine cycle \( M_2 \), unless the instruction required only one machine cycle for its execution. In such cases, a new \( M_1 \) cycle is entered. The loop is repeated for as many cycles and states as required by the instruction.

It is only during the last state of the last machine cycle that the interrupt request line is tested and a special \( M_1 \) cycle is entered, during which no program-counter incrementing takes place and \( \text{INTERRUPT ACKNOWLEDGE} \) status is sent out. During this cycle, one of eight possible single-byte calls will be sent to the CPU by the interrupting device.

**Execution times**

Instruction state requirements range from a minimum of four states for non-memory referencing instructions, like register and accumulator arithmetic instructions, up to a maximum of 18 states for the most complex instructions—such as XTHL (exchange the contents of registers H and L with the content of the top two locations of the stack). At the maximum clock frequency of 2 megahertz, this means that assembly-language instructions can be executed in 2 to 9 \( \mu s \).

As an example of 8080 timing, Fig. 4 shows the timing diagram for the one-byte, two-cycle instruction LMB (load the content of register B into the memory location addressed by the contents of registers H and L). This example also illustrates the timing when a \( \text{WAIT} \) state is entered after the execution of LMB. Notice that seven states (a total of 3.5 \( \mu s \)) are required to fetch and execute the LMB instruction. The same instruction would require 28 \( \mu s \) by the 8008, 17.5 \( \mu s \) by the faster 8080-1.

Though this example demonstrates an 8:1 improvement in speed over the 8008, the real impact of the new 8080 will not be as a replacement for the 8008. The 8080 has, after all, adequate speed for a large number of applications. The 8080 will be used in new systems that were not feasible before because the first-generation microcomputers were not powerful enough.