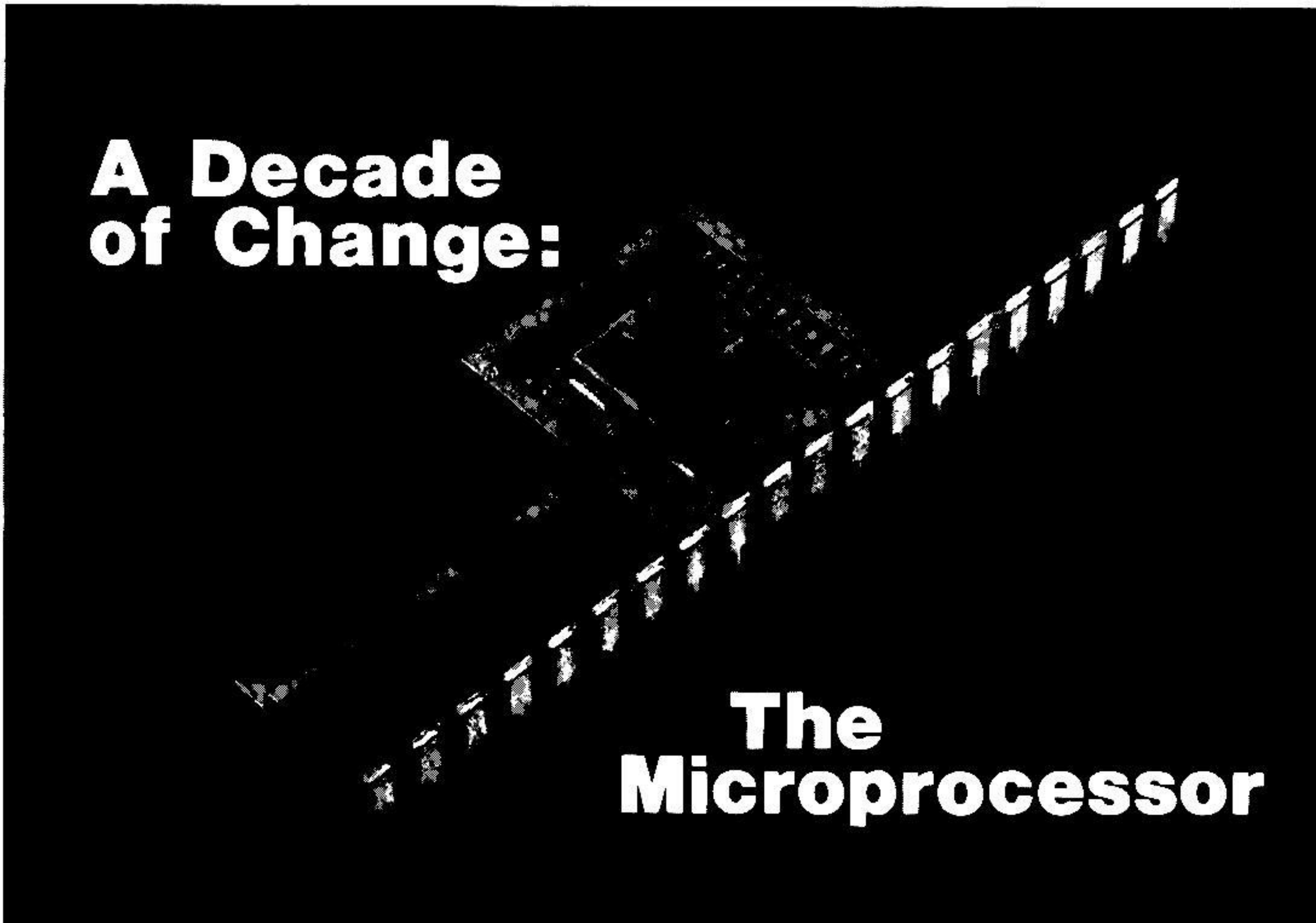
A Decade of Change: The Microprocessor



Today, microprocessors are used in everything from microwave ovens to automobiles to home computers. They've come a long way in a few short years.

ROBERT GROSSBLATT, CIRCUITS EDITOR

NAMES LIKE UNIVAC AND ENIAC HAVE MYstical connotations for some of us. Visions of huge rooms filled with glowing tubes and punched cards dance before our eyes, and our ears fill with the all-pervading murmur of massive air-conditioning systems. We remember staring awestruck as those giant computers flexed their electronic muscles to produce printed portraits of George Washington. (Wow!—*Editor*)

Those machines aren't really part of microcomputer history—they're part of its pre-history. Microcomputer history really began one day in the early 1960's when Jack Kilby of Texas Instruments made the first integrated circuit. After it was shown that an entire circuit could be fabricated from a single solid chunk of semi-conductor material, the stage was set for the development of the microprocessor. But things didn't really take off until the early 1970's.

Several important events occurred in 1970:

- The development of *L*arge *S*cale *I*ntegration (LSI) techniques made it possible to fabricate a complete circuit on a single piece of silicon.
- Memory technology matured to the point that standard products became widely available.
- A huge market opened up for hand-held calculators.

A year later Intel introduced a general-purpose IC that did just about everything necessary needed to support calculator-type operations. Intel called that IC the 4004; it had a four-bit data path, and it was the first microprocessor.

First generation microprocessors were designed for specific applications. The 4004 was designed to handle calculator logic, and the eight-bit 8008, introduced a year later, was designed to control an intelligent terminal.

Intel's IC's were designed at a time when memory was very expensive. Consequently, from the beginning, their microprocessors relied heavily on internal registers to store and transfer data. As shown in Fig. 1, the 8008's six storage registers are eight bits wide, and the program counter is fourteen bits wide. That limited the 8008 to 16K of directly addressable memory. Further, the internal stack is only seven words deep. That's fine for a controller, but it really limits use of the 8008 as a computer.

The microcomputer industry really came into its own with the second generation of microprocessors: Intel's 8080 and Motorola's 6800. Although both are microprocessors, their design philosophies are radically different.

Intel increased power in the 8080 by moving the stack itself to external memory, and by giving it a sixteen-bit stack pointer. As shown in Fig. 2, the 8080 also

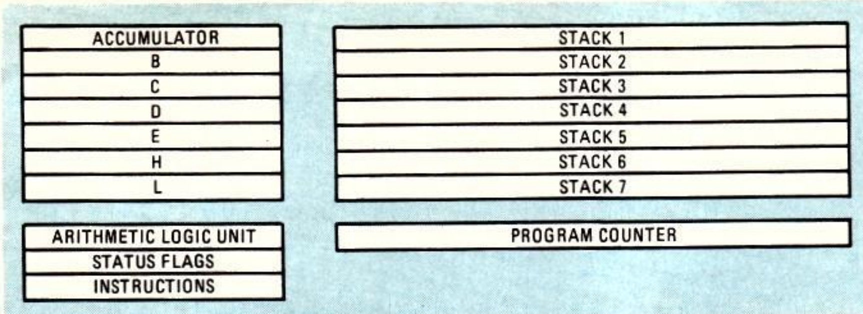


FIG. 1—THE FIRST 8-BIT MICROPROCESSOR, Intel's 8008, had an internal seven-byte stack, and it was limited to 16K bytes of memory.

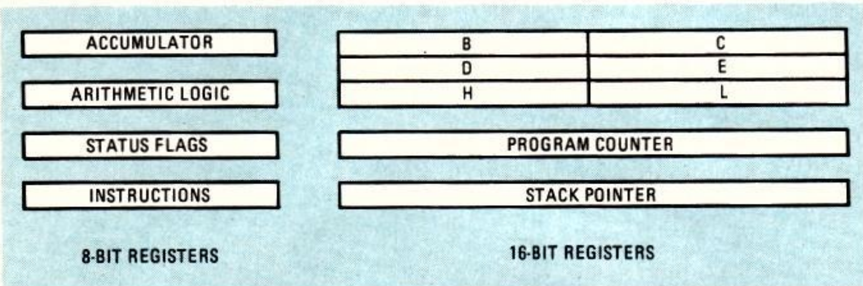


FIG. 2—THE FIRST GENERAL-PURPOSE MICROPROCESSOR, Intel's 8080, features an eight-bit accumulator and three general-purpose sixteen-bit registers.

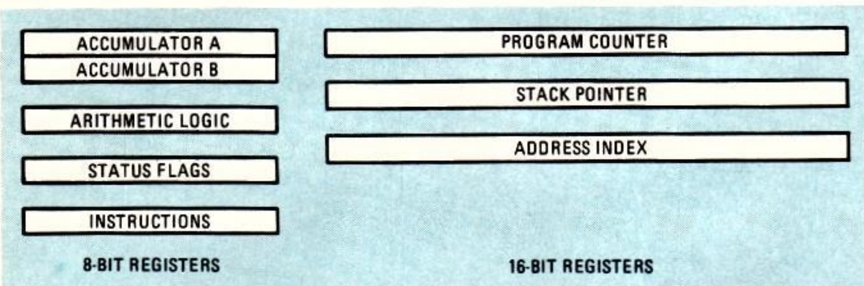


FIG. 3—MOTOROLA'S 6800 emphasizes quick memory access at the expense of few internal registers.

has three sixteen-bit registers in addition to the program counter, which was itself increased to sixteen bits. That gave the 8080 the capability of addressing the now-familiar 64K of external memory.

Motorola's 6800 was designed along different lines altogether. Instead of emphasizing internal registers, temporary data storage occurs in external memory, and the microprocessor has only enough registers to keep track of where things are. Other than the primary and secondary accumulators, the only "data" register in the 6800 is an index register that is used as a counter, and to do indexed memory addressing. The 6800's internal structure is shown in Fig. 3.

There are advantages and disadvantages to both design philosophies. For example, the 6800 must access memory much more often than the 8080, but it can do that much more rapidly than the 8080 can. But by far the most important consequence of the difference in those IC's is in overall memory organization.

To the 8080, all addresses are the same—it takes the same amount of time to access any location in its 64K range. Also, the processor goes to location ØØØØ for its first instruction after power up and after reset.

The 6800, on the other hand, has a special set of two-byte "zero page" addressing instructions that allow page zero—the first 256 bytes of memory—to be accessed rapidly. All other address instructions are three bytes long. That makes zero-page real estate valuable—the microprocessor can address page zero locations in two-thirds the time it can address locations in other pages. Since zero-page addresses are, in a sense, the 6800's substitute for registers, the operating system is put at the top of memory. When a 6800 is first powered up, it goes to address FFFE for its first instruction.

The Motorola and Intel design philosophies have shaped the architectures of just about all microprocessors that have followed, down to the latest sixteen- and thirty-two bit microprocessors.

CP/M

In 1973 the microcomputer industry was rather limited. First-generation micros like the Altair and the Imsai stored data on paper tape or cassettes. Floppy disk drives had just begun to appear when Gary Kildall, a consultant working for Intel, developed CP/M, a *D*isk *O*perating *S*ystem (DOS) for Intel's 8080. When Imsai licensed CP/M and began distributing it, the home computer market exploded. Since CP/M was just about the only DOS available, the 8080 became the microprocessor of choice for most home computers.

Because of the popularity of CP/M, any heir to the 8080's throne would have to have an instruction set compatible with that microprocessor. Since most of the software available for home computers was coded in the 8080's native tongue, microprocessor manufacturers realized that it was a matter of simple economic survival to make sure that such software would run on any new microprocessor.

By the time that the second generation of microprocessors began winding down, the home-computer market was an economic reality. CP/M gave birth to a variety of sophisticated systems-programming languages, and user-oriented application programs began to fill the market. That caused more CP/M computers to be sold and more programs to be written for them. Then, when major manufacturers like IBM and Xerox began adopting CP/M for use in their products, the stage was set for the birth of the third generation of microprocessors.

Intel introduced the 8085, an improved version of its 8080, and then upgraded that with the 8085A. The old 8080 required *three* voltages to operate: +12, +5 and −5. The 8085A required only +5 volts, and a much simpler clock circuit now sufficed. In addition, a series of new interrupt and I/O signals were tied to various pins on the IC—in an attempt, perhaps, to simplify the complex I/O set-up of the 8080.

Unfortunately, Intel paid a high price for that increased flexibility. Because they dedicated so many pins to I/O, and because they remained locked to a standard forty-pin package, the data bus had to be multiplexed with the lower eight bits of the address bus. And that meant that the 8085 would always need external latching. A big gripe with the 8080 was that it needed extra support IC's (an 8224 Clock Generator and an 8228 System Controller); and since the 8085 also needed at least an eight-bit latch and some decoding, the 8085 never became as popular as Intel had hoped. However, a CMOS version became available relatively early; Tandy used it in their portable computers, and that prevented it from becoming totally obscure. But perhaps the real reason for its lukewarm reception was that it was introduced three years after the Z80.

Zilog's Z80

From the moment it appeared, the Z80

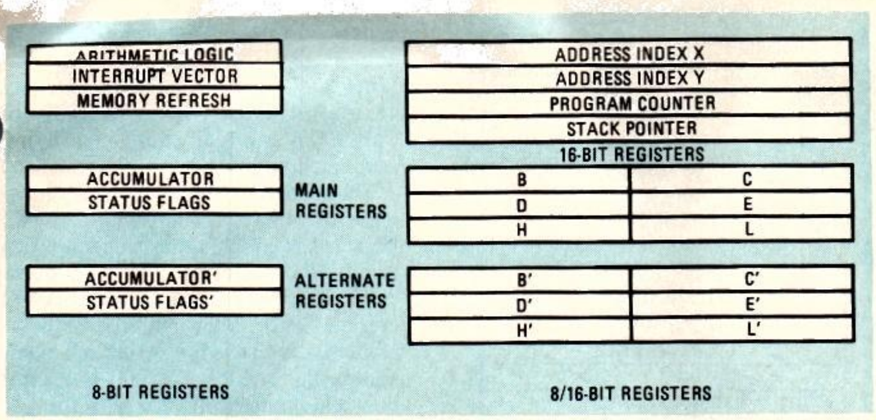


FIG. 4—ZILOG'S Z80 has the same registers as the 8080, but it provides a duplicate register set for quick response to interrupts.

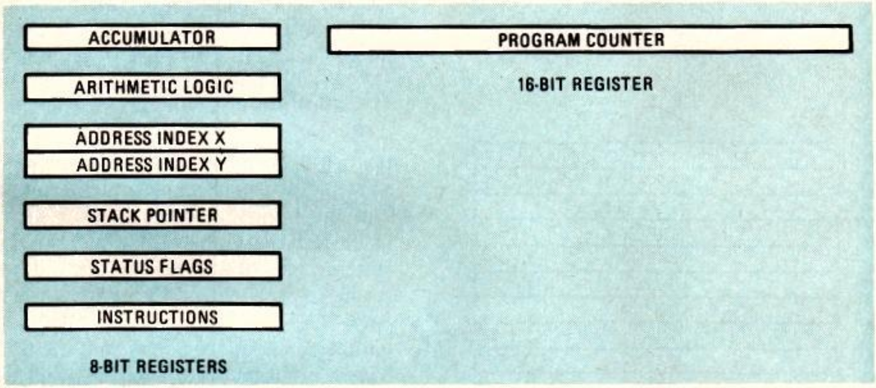


FIG. 5—MOS TECHNOLOGY'S 6502 has no general-purpose 16-bit registers, but it has the most flexible addressing schemes of all the early eight-bit processors.

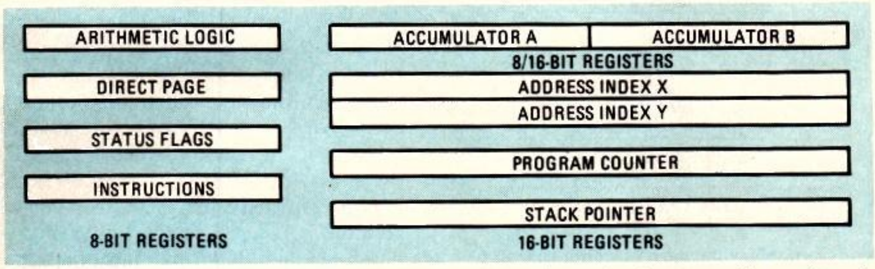


FIG. 6—MOTOROLA'S 6809 was intended to bridge the 8/16-bit gap; but it never really caught on, in spite of its having some really powerful features.

Microprocessor Speed

How fast a program runs on your computer depends on more than just its clock speed. The clock frequency determines microprocessor speed, but the instruction set, as well as the way a program is written,determine how fast a program runs.

Most of a microprocessor's time is spent accessing locations in your computer's memory, and the instruction that tells the microprocessor to do that can be two, four or even more bytes long. The overall speed of your computer is a function of how many cycles of the system clock it takes to complete a particular instruction.

A computer running at 8 Mhz will complete a four-byte instruction in exactly the same length of time that it takes a 2-MHz computer to complete a one-byte instruction. Overall program execution speed, therefore, depends on how much the microprocessor has to do and how quickly it can do it. The same program running on two computers with different microprocessors will undoubtedly run at two different speeds. Which runs faster depends on what the program is asking the microprocessor to do.

Some microprocessors have instruction sets that are quick at number crunching, while others are better at memory access. In general, microprocessors with internal registers, like the Z80, are better at number crunching because they have one-byte commands to manipulate register data. Microprocessors like the 6502, on the other hand, which do most things in memory, access that memory quicker than the register-oriented type of microprocessor. **R-E**

was a winner. It incorporated all the 8080 instructions as a subset of its own greatly-expanded instruction set. And that allowed it to run all the existing 8080-based software. In terms of hardware, the Z80 was designed according to the Intel philosophy. Zilog maintained the register structure of the 8080, but also added a set of alternate registers that duplicated the main set, as shown in Fig. 4. The alternate registers allow an increase in processing speed because, during an interrupt, the main registers can be swapped with the alternates by using a fast one-byte instruction.

Combined with the other advantages it had over the 8080—a single supply-voltage and a simple clock—the Z80 became *the* upgrade for the 8080. It had built-in features that required extra IC's in an 8080-based system. The Z80, for example, can refresh dynamic RAM IC's automatically.

One interesting point about the Z80 is that, when it was first introduced, Zilog believed that the huge new instruction set would be responsible for most sales of the IC. As it turned out, 8080 code had become *the* standard, and the Z80's extra instructions were, for the most part, ignored. Intel, on the other hand, showed awareness of that need for compatibility, because the 8085 added only two instructions to the 8080's repertoire.

The 6502

In 1975 MOS Technology introduced the 6502. Just as the Z80 is a high-performance version of the 8080, the 6502 is a souped up 6800—but with differences. You can run 8080 code on a Z80, but 6800 code is gibberish to a 6502. What the 6502 got from the 6800 was its design philosophy. The 6502 is a top-down processor; its strength is in powerful addressing modes, rather than a lot of internal registers. Among its advantages over the 6800 were: on-IC clock generator, an improved instruction set, new addressing modes, faster access to the stack, and built-in BCD arithmetic.

At first glance, the 6502's architecture might look like a step backward. As shown in Fig. 5, the sixteen-bit index register of the 6800 was split into two eight-bit registers, and the stack pointer is only eight bits wide. That limits the stack to 256 bytes, but the designers of the 6502 decided that a 256-byte stack is more than adequate for most applications. By limiting its length, it could be placed in a dedicated area of memory (1ØØ to 1FF). And that allowed the stack to be accessed very quickly, which increased overall execution speed. Splitting the index register allows both halves to be used independently for some very powerful indexed addressing modes.

What saved the 6502 from obscurity was being chosen as the microprocessor for the Apple, Atari, and Pet computers. The huge successes enjoyed by those ma-

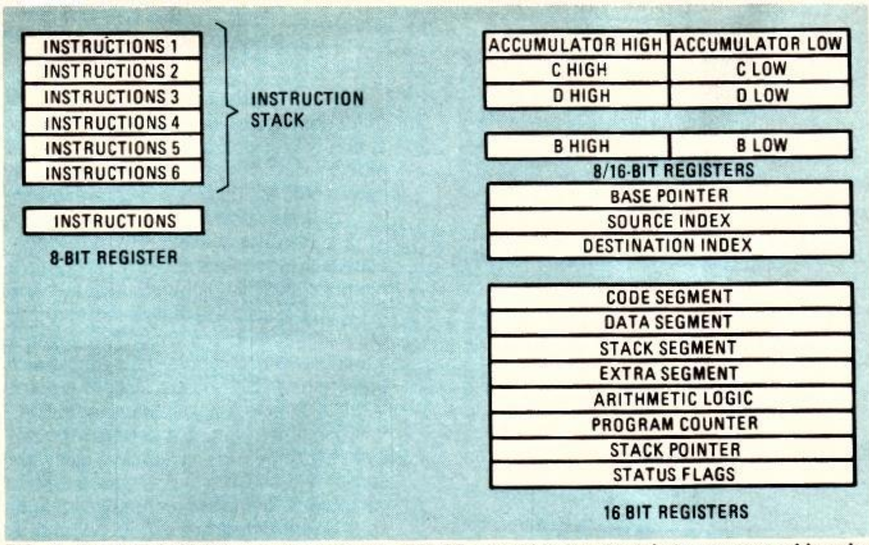


FIG. 7—INTEL'S 8088/86, workhorse of the IBM PC, provides one-megabyte memory addressing through the use of segment registers.

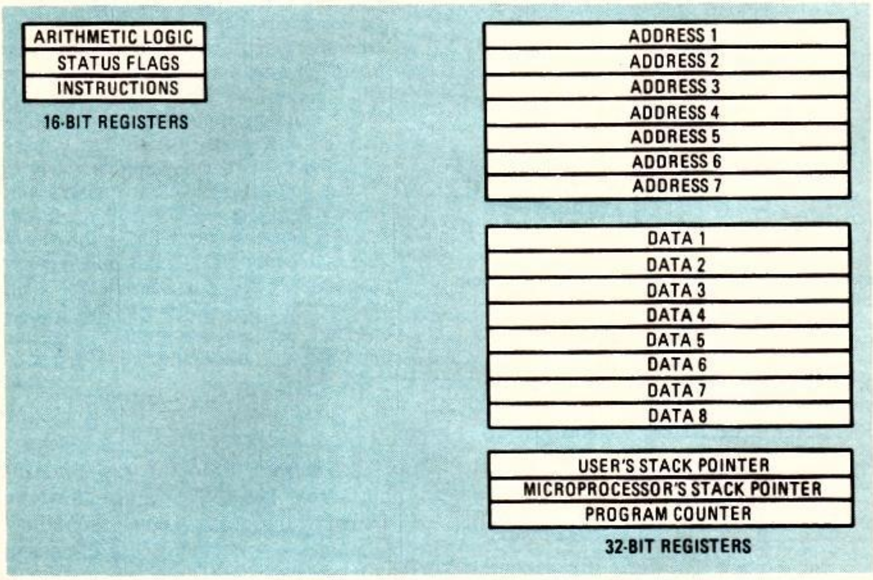


FIG. 8—MOTOROLA'S 68000, heart of Apple's Macintosh, is probably the most powerful sixteen-bit microprocessor on the market.

chines transformed the 6502 into one of the few silicon superstars.

The fourth generation of microprocessors began to emerge around 1980. The home-computer market was a mammoth economic reality by that time, and IC manufacturers were selling microprocessors as fast as they could make them. The third generation's last gasp was Motorola's 6809. It included advanced addressing and a multiply instruction. That instruction was innovative, unique, and it allowed a substantial increase in program execution speed. The 6809's structure is shown in Fig. 6.

Unfortunately for Motorola, the 6809 came late in the history of microcomputing, so few machines were designed around that IC. (Radio Shack's *Color Computer* is a notable exception.—*Editor*) One interesting footnote to computer history is that Apple was so impressed with the 6809 that they used it in early *MacIntosh* designs.

Fourth-generation microprocessors came about through refinements in IC technology. The advent of *V*ery *L*arge *S*cale *I*ntegration (VLSI) vastly increased on-IC component density. Results include sixteen-bit microprocessors, 64K (and larger) dynamic RAM's, as well as performance upgrades for the previous generation of microprocessors.

Sixteen bit microprocessors weren't really new. For example, TI's TMS9900, was a second-generation, top-down, sixteen-bit microprocessor with two sixteen-bit registers: a stack pointer and a program counter. All other data storage occurred in external memory. The TMS9900 has separate address and data buses because it comes in a 64-pin package. The TMS9900 is a powerful microprocessor, and the way it was marketed is a perfect example of how *not* to do it. TI used the IC in its home computer, the *TI99*, but TI did nothing to support outside developers. When the computer failed to catch on, Texas Instruments let it die, and the TMS9900, essentially an IC ahead of its time, died with it.

The second-generation design philosophies of the 8080 family and the 6800 family showed up in third generation sixteen-bit IC's. Intel released the 8086 and 8088; they're direct descendents of the original 8080, so they emphasize the use of internal registers for storing and manipulating data. Zilog's Z8000 is a more-powerful, sixteen-bit version of the well-established Z80. Motorola's 68000, a muscular, sixteen-bit version of the neglected 6809, is more powerful than either Intel's or Zilog's microprocessors. We'll look at each in turn.

The 8086 family

As we mentioned earlier, it takes more than good design to make a silicon superstar. Intel struck it rich when IBM jumped into the home-computer market with a machine based on the 8088. Intel was a bit surprised, as well. The 8088 is a watered-down version of Intel's more powerful 8086. As shown in Fig. 7, the 8086 has a number of 16-bit registers, but the 8088 has an eight-bit-wide data bus, and that means that sixteen-bit data must be loaded into the microprocessor eight bits at a time. That's why it's called an 8/16 bit IC. The 8086, on the other hand, is a true sixteen-bit microprocessor.

IBM's choice of the 8088 was a serious miscalculation of both their marketing ability, and the viability of the market. One of the forgotten oddities of the IBM PC is that it originally showed up in the market with only 16K of RAM. The incredible popularity of the PC led to the development of huge amounts of applications software, and much mainstream eight-bit software was rewritten to operate on the 8088. Once again, Intel had produced the microprocessor that became the industry standard.

Several new features showed up in the new sixteen-bit series that had no counterpart in their eight-bit predecessors. A multiply instruction was added (as Motorola had done with the 6809), but it could be used effectively only in the 8086. The 8088 had a tough time handling thirty-two bit results, so most IBM-PC programmers do multiplication with the traditional shift and add approach, and that slows the PC down drastically.

Although the 8086 and 8088 are stand-alone microprocessors, Intel has some support IC's that can speed things up. The 8087 Math Co-Processor, for example, does high-speed number crunching, and it substantially increases the computing speed of the 8086. Note that the eight-bit data bus of the IBM PC restricts the

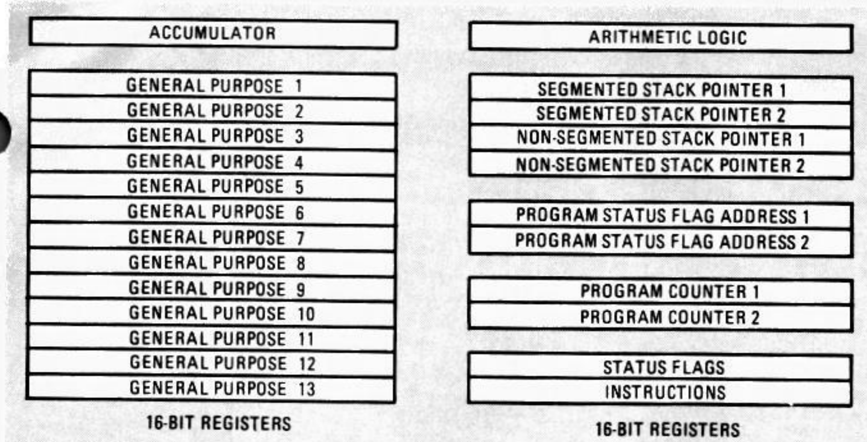


FIG. 9—ZILOG'S Z8000 has not been very popular, in spite of powerful register and memory-addressing structures.

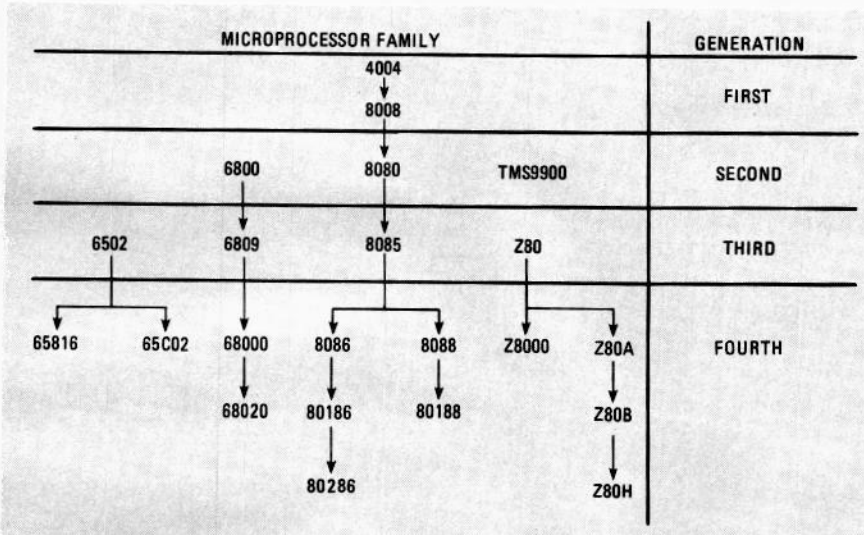


FIG. 10—THE GENEALOGY OF THE FIRST FOUR GENERATIONS OF MICROPROCESSORS is indicated here. What will be next? Stay tuned . . .

usefulness of the 8087 in that computer.

The 80186 and 80188 were introduced by Intel in 1983 as upgrades of the 8086 and the 8088, respectively. Those two IC's took advantage of new VLSI techniques, but didn't really represent any advance in performance. However, they include a lot of the support circuitry for timing and bus control that had to be done externally with the earlier IC's.

Intel's 80286, which IBM uses in its *PC/AT*, is a major enhancement of the 8086. Besides having an on-IC memory-management system, it has a "virtual memory" mode in which the internal registers can generate 24-bit addresses. In practical terms, this means that the 80286 can directly address as much as sixteen megabytes of memory. By contrast, the 8086 can generate only 20-bit addresses, and that allows it to address "only" one megabyte of memory directly.

The 68000

Motorola's 68000 is probably the most-powerful sixteen-bit microprocessor to show up on the market; Apple chose to use it for the ill-fated Lisa and for the MacIntosh. Because its design is similar to that of the 6800, it has a variety of powerful addressing modes. Motorola learned a few things from Intel, apparently, because they added sixteen thirty-two-bit registers, as well as two stack pointers—one for the user and one for internal housekeeping. Unlike most other microprocessors, the 68000 doesn't have a dedicated accumulator. Rather, any of the data registers can be used as an accumulator. The 68000's organization is shown in Fig. 8.

Motorola designed the instruction set so that there could be as many as 64,000 instructions. The Intel IC's, by way of comparison, continued the 8080's practice of limiting the IC to a maximum of 256 instructions. In fact, the 68000 has only 56 basic instructions, but the IC's addressing flexibility makes it easy for programmers to access the full power of the IC without having to remember separate instructions for each special case.

One strength of the 68000 is the width of its registers. If the 8088 is an 8/16 bit microprocessor, then the 68000 should be called a 16/32 bit microprocessor. It has, like the 80286, a "virtual memory" scheme, but its wider registers allow some versions of the 68000 to manage more than 4 *gigabytes* of memory.

The Z8000

The Z8000 from Zilog is a sixteen-bit version of the Z80. It is a well-designed, true sixteen-bit microprocessor with thirteen general-purpose sixteen-bit registers, several stack pointers, and the ability to address as much as sixteen megabytes of memory. The Z8000's organization is shown in Fig. 9.

In spite of its potential, however, the Z8000 has not enjoyed the popularity of its eight-bit predecessor. When the Z80 hit the market, it was successful because it was compatible with the 8080. Unfortunately, the 8086 and the Z8000 are totally incompatible.

Other improvements

The last major advance of the fourth generation had to do with IC technology. Faster versions of popular eight- and sixteen-bit microprocessors began to appear. The Z80, whose original operating speed was a mere 2.5 MHz, became available in 4- and 6-MHz versions (the Z80A, and Z80B, respectively). The original 6502 ran at 1 MHz; its upgrade, the 6502C, runs at 4 MHz. Rockwell, NCR, and GTE each came out with CMOS versions of the 6502, the 65C02, that uses less power, runs faster, and has a larger instruction set than the original.

The 65816

New microprocessors like Western Design Center's 65816 aim at higher speeds, better memory handling, and increased compatibility. The 65816 has a software-selectable mode in which it can emulate a 6502, and it's the first sixteen-bit microprocessor to use the mainframe technique of "cache memory," which is similar to the virtual memory modes we've already mentioned. In a cache system, intermediate data and program information are stored in high-speed memory, and less-important data is stored in a slower memory system such as a disk.

Conclusions

The overall genealogy of the important microprocessor families is outlined in Fig. 10. As you can see, the trend in new products is toward wider buses, higher speeds, and more flexibility. True 32-bit microprocessors are now showing up.

When the 8080 first appeared, it cost over $250. Today you can buy one for less than three dollars. And when you hold one of those twelve-year-old IC's in your hand, keep in mind that you're looking at more computing power and speed than was ever dreamed of in the pre-historic days of the Univac—all of twenty-five years ago. **R-E**