MICROPROCESSORS

In 1972, Intel Corporation sparked an industrial revolution with the world's first microprocessor, the 4004. The 4004 replaced the logic of a numeric calculator with a general-purpose computer, implemented in a single silicon chip. The 4004 is shown in Fig. 1. The 4004 integrated 2300 transistors and ran at a clock rate of 108 kHz (108,000 clock cycles per second). In 1997, the 4004's most recent successor is the Pentium II processor, running at 300 MHz (300 million clock cycles per second) and incorporating nearly 8 million transistors. The Pentium II processor is shown in Fig. 2.

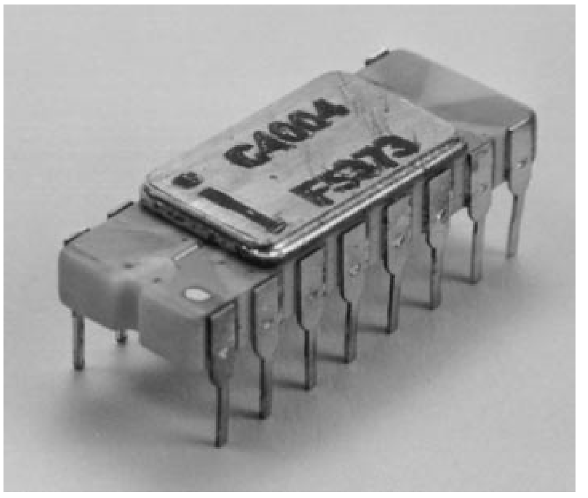

Figure 1. The world's first microprocessor, the Intel 4004, ca. 1971. Originally designed to be a less expensive way to implement the digital logic of a calculator, the chip instead spawned a computing revolution that still shows no signs of abating.

From the 4004's humble beginning, the microprocessor has assumed an importance in the world's economy similar to that of the electric motor or the internal combustion engine. Microprocessors now supply more than 90% of the world's computing needs, from small portable and personal desktop computers to large-scale supercomputers such as Intel's Teraflop machine, which contains over 9000 microprocessors. A variant of the microprocessor, the microcontroller, has become the universal controller in machines from automobile engines to audio systems to wristwatches.

MICROPROCESSORS AND COMPUTERS

Microprocessors are the processing units or the "brains" of the computer system. Every action that microprocessors perform is specified by a computer program that has been encoded into "object code" by a software program known as a compiler. Directed by another software program known as the operating system (e.g., Microsoft's Windows 95), the microprocessor locates the desired application code on the hard drive or compact disk and orders the drive to begin transferring the program to the memory subsystem so that the program can be run.

Digital electronic computers have at least three major subsystems:

- A memory to hold the programs and data structures
- An input/output (I/O) subsystem
- A central processor (CPU)

A microprocessor is the central processor subsystem, implemented on a single chip of silicon.

In microprocessor-based computer systems, the I/O subsystem moves information into and out of the computer system. I/O subsystems usually include some form of nonvolatile storage, which is a means of remembering data and programs even when electrical power is not present. Disk drives, floppy drives, and certain types of memory chips fulfill this requirement in microprocessor-based systems. Keyboards, trackballs, and mice are common input devices. Networks, modems, and compact discs are also examples of I/O devices. The memory subsystem, a place to keep and quickly access programs or data, is usually random-access memory (RAM) chips.

Microprocessors and microcontrollers are closely related devices. The differences are in how they are used. Essentially, microcontrollers are microprocessors for embedded control applications. They run programs that are permanently encoded into read-only memories and optimized for low cost so that they can be used in inexpensive appliances (printers, televisions, power tools, and so on). The versatility of a microcontroller is responsible for user-programmable VCRs and microwave ovens, the fuel-savings of an efficiently managed automobile engine, and the convenience of sequenced traffic lights on a highway and of automated bank teller machines.

Microprocessor software is typically created by humans who write their codes in a high-level language such as C or Fortran. A compiler converts that source code into a machine language that is unique to each particular family of microprocessors. For instance, if the program needs to write a character to the screen, it will include an instruction to the microprocessor that specifies the character, when to write it, and where to put it. Exactly how these instructions are encoded into the 1s and 0s (bits) that a computer system can use determines which computers will be able to run the program successfully. In effect, there is a contract between the design of a microprocessor and the compiler that is generating object code for it. The compiler and microprocessor must agree on what every computer instruction does, under all circumstances of execution, if a program is to perform its intended function. This contract is known as the computer's instruction set. The instruction set plus some additional details of implementation such as the number of registers (fast temporary storage) are known as the computer's instruction set architecture (ISA). Programs written or compiled to one ISA will not run on a different ISA. During the 1960s and 1970's, IBM's System/360 and System/370 were the most important ISAs. With the ascendancy of the microprocessor, Intel's x86 ISA vied with Motorola's MC68000 for control of the personal computer market. By 1997, the Intel architecture was found in approximately 85% of all computer systems sold.

Early microprocessor instruction set architectures, such as the 4004, were designed to operate on 8-bit data values (operands). Later microprocessors migrated to 16-bit operands, including the microprocessor in the original IBM PC (the Intel 8088). Microprocessors settled on 32-bit operands in the 1980s, with the Motorola 68000 family and Intel's 80386. In the late 1980s, the microprocessors being used in the fastest servers and high-end workstations began to run into the intrinsic addressability limit of 4GB (four gigabytes, or four billion bytes, which is 2 raised to the power 32). These microprocessors introduced 64-bit addressing and data widths. It is likely that 64-bit computing will eventually supplant 32-bit microprocessors. It also seems likely that this will be the last increase in addressability that the computing industry will ever need because 2 raised to the power 64 is an enormous number of addresses.

Prior to the availability of microprocessors, computer systems were implemented in discrete logic, which required the assembly of large numbers of fairly simple digital electronic integrated circuits to realize the basic functions of the I/O, memory, and central processor subsystems. Because many (typically thousands) of such circuits were needed, the resulting systems were large, power-hungry, and costly. Manufacturing such systems was also expensive, requiring unique tooling, hand assembly, and a large amount of human debug effort to repair the inevitable flaws that accumulate during the construction of such complex machinery. In contrast, the fabrication process that underlies the microprocessor is much more economical. As with any silicon-integrated circuit, microprocessor fabrication is mainly a series of chemical processes performed by robots. So the risk of introducing human errors that would later require human debugging is eliminated. The overall process can produce many more microprocessors than discrete methods could.

Figure 2. The 1998 successors to the line of microprocessors started by the 4004, Intel's Pentium II processor, mounted within its Single-Edge Cartridge Connector (SECC). This picture shows the cartridge with its black case removed. On the substrate within the cartridge, the large octagonal package in the center is the Pentium II CPU itself. The rectangular packages to the right and left of the CPU are the cache chips. The small components mounted on the substrate are resistors and capacitors needed for power filtering and bus termination.

MOORE'S LAW

In 1964, Gordon Moore made an important observation regarding the rate of improvement of the silicon-integrated circuit industry. He noted that the chip fabrication process permitted the number of transistors on a chip to double every 18 months. This resulted from the constantly improving silicon process that determines the sizes of the transistors and wiring on the integrated circuits. Although he made the initial observation on the basis of experience with memory chips, it has turned out to be remarkably accurate for microprocessors as well. Moore's Law has held for well over 30 years. Figure 3 plots the number of transistors on each Intel microprocessor since the 4004.

These improvements of the underlying process technology have fueled the personal computer industry in many different ways. Each new process generation makes the transistors smaller. Smaller transistors are electrically much faster, allowing higher clock rates. Smaller wires represent less electrical capacitance, which also increases overall clock rates and reduces power dissipation. The combination of both permits far more active circuitry to be included in new design. Constant learning in the silicon fabrication plants have also helped drive up the production efficiency, or yield, of each new process to be higher than its predecessor, which also helps support larger die sizes per silicon chip.

The impact of this progression has been profound for the entire industry. The primary benefit of a new microprocessor is its additional speed over its predecessors, at ever better price points. The effect of Moore's Law has been for each new microprocessor to become obsolete within only a few years after its introduction. The software industry that supplies the applications to run on these new microprocessors expects this performance improvement. The industry tries to design so that its new products will run acceptably on the bulk of the installed base but can also take advantage of the new performance for the initially small number of platforms that have the new processor. The new processor's advantages in price/performance will cause it to begin to supplant the previous generation's volume champion. The fabrication experience gained on the new product allows its price to be driven ever downward until the new design completely takes over. Then an even more advanced processor on an even better process technology is released, and the hardware/software spiral continues.

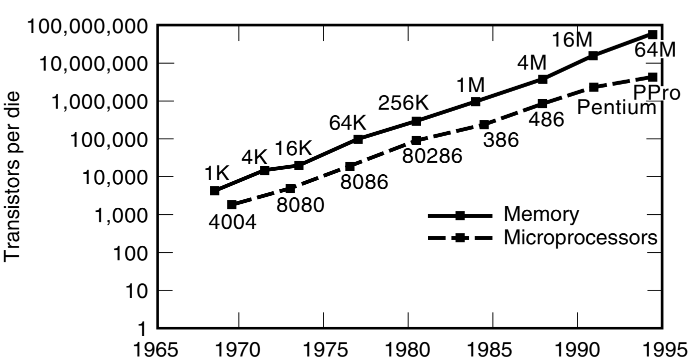

Figure 3. Moore's Law has accurately predicted the number of transistors that can be incorporated in microprocessors for over 25 years. Since this transistor count strongly influences system performance, this remarkable "law" has become one of the central tenets in the field of computers and integrated electronics. It guides the design of software, hardware, manufacturing production capacity, communications, and corporate planning in nearly every major area.

MICROPROCESSOR ARCHITECTURES

Another factor in the performance improvement of microprocessors is its microarchitecture. Microarchitecture refers to *how* a microprocessor's internal systems are organized. The microarchitecture is not to be confused with its instruction set architecture. The ISA determines what kind of software a given chip can execute. The earliest microprocessors (e.g., Intel 4004, 4040, 8008, 8080, 8086) were simple, direct implementations of the desired ISA. But as the process improvements implied by Moore's Law unfolded, microprocessor designers were able to borrow many microarchitectural techniques from the mainframes that preceded them, such as caching (Intel's 486, MC68010), pipelining (i486 and all subsequent chips), parallel superscalar execution (Pentium processor), superpipelining (Pentium Pro processor), and out-of-order and speculative execution (Pentium Pro processor, MIPS R10000, DEC Alpha 21264).

Microprocessor designers choose their basic microarchitectures very carefully because a chip's microarchitecture has a profound effect on virtually every other aspect of the design. If a microarchitecture is too complicated to fit a certain process technology (e.g., requires many more transistors than the process can economically provide), then the chip designers may encounter irreconcilable problems during the chip's development. The chip development may need to wait for the next process technology to become available. Conversely, if a microarchitecture is not aggressive enough, then it could be very difficult for the final design to have a high enough performance to be competitive.

Microarchitectures are chosen and developed to balance efficiency and clock rate. All popular microprocessors use a synchronous design style in which the microarchitecture's functions are subdivided in a manner similar to the way a factory production line is subdivided into discrete tasks. And like the production line, the functions comprising a microprocessor's microarchitecture are pipelined, such that one function's output becomes the input to the next. The rate at which the functions comprising this pipeline can complete their work is known as the pipeline's clock rate. If the functions do not all take the same amount of time to execute, then the overall clock rate is determined by the slowest function in the pipeline.

One measure of efficiency for a microarchitecture is the average number of clock cycles required per instruction executed (CPI). For a given clock rate, fewer clocks per instruction implies a faster computer. The more efficient a microarchitecture is, the fewer the number of clock cycles it will need to execute the average instruction. Therefore, it will need fewer clock cycles to run an entire program. However, the desire for high microarchitectural efficiency is often in direct conflict with designing for highest clock rate. Generally, the clock rate is determined by the time it takes a signal to traverse the slowest path in the chip, and adding transistors to a microarchitecture to boost its efficiency usually makes those paths slower.

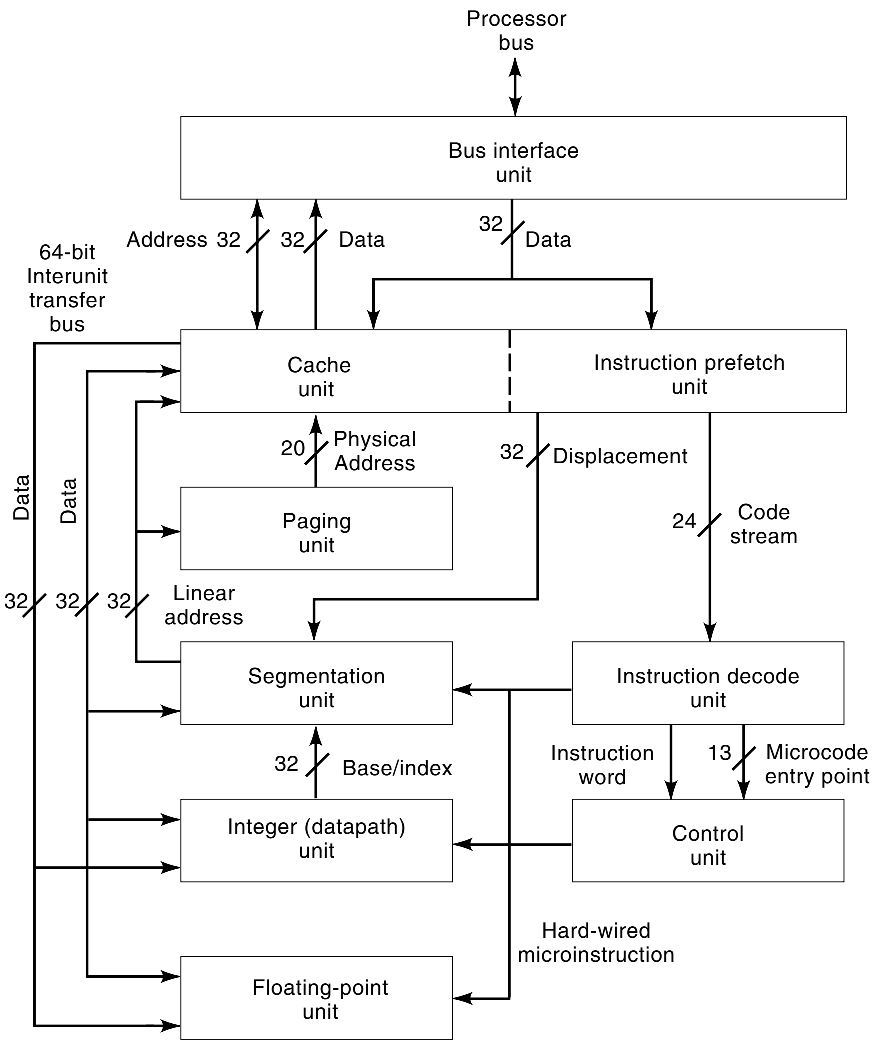

Figure 4. Block diagram of the most popular microprocessor of the early 1990s, the Intel i486. The various blocks shown work together to execute the Intel Architecture instruction set with approximately 1.1M transistors. Newer designs, such as the Pentium processor, or the P6 microarchitecture at the core of the latest Pentium II processor, are much more complicated.

Figure 4 illustrates the functional block diagram of the Intel 486, a very popular microprocessor of the early 1990s. (The microarchitectures of microprocessors that followed the 486, such as the Pentium processor or the Pentium Pro processor, are too complex to be described here.) The prefetch unit of the 486 fetches the next instruction from the instruction cache at a location that is either the next instruction after the last instruction executed or some new fetch address that was calculated by a previous branch instruction. If the instruction requested is not present in the cache, then the bus interface unit generates an access to main memory across the processor bus, and the memory sends the missing instruction back to the cache. The requested instruction is sent to the instruction decode unit, which extracts the various fields of the instruction, such as the opcode (the operation to be performed), the register or registers to be used in the instruction, and any memory addresses needed by the operation. The control unit forwards the various pieces of the instruction to the places in the microarchitecture that need them (register designators to the register file, memory addresses to the memory interface unit, opcode to the appropriate execution unit).

Certain very complex instructions are implemented in an on-chip read-only memory called the microcode. When the instruction decoder encounters one of these, it signals a microcode entry point for the microcode unit to use in supplying the sequence of machine operations that correspond to that complex macroinstruction.

Although it is not obvious from the block diagram, the Intel 486 microarchitecture is pipelined, which allows the machine to work on multiple instructions at any given instant. While one instruction is being decoded, another instruction is accessing its registers, a third can be executing, and a fourth is writing the results of an earlier execution to the memory subsystem.

See References 1–4 for sources of more details on designing microarchitectures.

THE EVOLUTION OF ISAS

Although microprocessor ISAs are crucial in determining which software will run on a given computer system, they are not static and unchangeable. There is a constant urge to develop the ISA further, adding new instructions to the instruction set or (much more rarely) removing old obsolete ones. Almost all old ISAs have many instructions, typically hundreds, some of which are quite complicated and difficult for compilers to use. Such architectures are known as Complex

Instruction Set Computers (CISC). In the early 1980s, substantial academic research was aimed at simplifying ISAs [Reduced Instruction Set Computers (RISC)], and designing them with the compiler in mind, in the hopes of yielding much higher system performance. Some important differences remain, such as the number of registers, but with time the differences in implementations between these two design philosophies have diminished. RISC ISAs have adopted some of the complexity of the CISC ISAs, and the CISC designers borrowed liberally from the RISC research. Examples of the CISC design style are the Intel x86, the Motorola MC68000, the IBM System/360 and /370, and the DEC VAX. RISC ISAs include MIPS, PowerPC, Sun's Sparc, Digital Equipment Corp. Alpha, and Hewlett-Packard PA-RISC.

COPROCESSORS AND MULTIPLE PROCESSORS

Some microprocessor systems have included a separate chip known as a coprocessor. This coprocessor was intended to improve the system's performance at some particular task that the main microprocessor was unsuited for. For example, in the Intel 386 systems, the microprocessor did not implement the floating-point instruction set; that was relegated to a separate numerics coprocessor. (In systems that lacked the coprocessor, the microprocessor would emulate the floating-point functions, albeit slowly, in software.) This saved die size and power on the microprocessor in those systems that did not need high floating-point performance, yet it made the high performance available in systems that did need it, via the coprocessor. However, in the next processor generation, the Intel 486, enough transistors were available on the microprocessor, and the perceived need for floating-point performance was large enough, that the floating-point functions were directly implemented on the microprocessor.

Floating-point coprocessors have not reappeared, but less-integrated hardware for providing audio (sound generation cards) and fast graphics are quite common in personal computers of the 1990s, which are similar to the coprocessors of the past. As the CPUs get faster, they can begin to implement some of this functionality in their software, thus potentially saving the cost of the previous hardware. But the audio and graphics hardware also improves, offering substantially faster functionality in these areas, so that buyers are tempted to pay a small amount extra for a new system.

HIGH-END MICROPROCESSOR SYSTEMS

Enough on-chip cache memory and external bus bandwidth is now available that having multiple microprocessors in a single system has become a viable proposition. These microprocessors share a common platform, memory, and I/O subsystem. The operating system attempts to balance the overall computing workload equitably among them. Dedicated circuits on the microprocessor's internal caches monitor the traffic on the system buses, in a procedure known as "snooping" the bus, to keep each microprocessor's internal cache consistent with every other microprocessor's cache. The system buses are designed with enough additional performance so that the extra microprocessors are not starved.

In the late 1990s, systems of 1, 2, and 4 microprocessors became more common. Future high-end systems will probably continue that trend, introducing 8, 16, 32, or more microprocessors organized into clusters. As of the mid 1990s, the fastest computers in the world no longer relied on exotic specialized logic circuits but were composed of thousands of standard microprocessors.

FUTURE PROSPECTS FOR MICROPROCESSORS

From their inception in 1971, microprocessors have been riding an exponential growth curve in the number of transistors per chip, delivered performance, and growth in the installed base. But no physical process can continue exponential growth forever. It is of far more than academic interest to determine when microprocessor development will begin to slow and what form such a slowdown will take.

For example, it is reasonable to surmise that the process technology will eventually hit fundamental limitations in the physics of silicon electronic devices. The insulators most commonly used in an integrated circuit are layers of oxide, and these layers are only a few atoms thick. To keep these insulators from breaking down in the presence of the electric fields on an integrated circuit, designers try to lower the voltage of the chip's power supply. At some point, the voltage may get so low that the transistors no longer work.

Power dissipation is becoming an increasingly important problem. The heat produced by fast microprocessors must be removed so that the silicon continues to work properly. As the devices get faster, they also generate more heat. Providing the well-regulated electrical current for the power supply, and then removing the heat, means higher expense in the system. With the 486 generation, aluminum blocks with large machined surface areas, known as heat sinks, became commonplace. These heat sinks help transfer the heat from the microprocessor to the ambient air inside the computer; a fan mounted on the chassis transfers this ambient air outside the chassis. With the Pentium processor generation, a passive aluminum block was no longer efficient enough, and a fan was mounted directly on the heat sink itself. Future microprocessors must find ways to use less power, transfer the heat more efficiently and inexpensively to the outside, and modulate their operations to their circumstances more adroitly. This may involve slowing down when high performance is temporarily unnecessary, changing their power supply voltages in real time, and managing the program workload based on each program's thermal characteristics.

Microprocessor manufacturers face another serious challenge: complexity, combined with the larger and less technically sophisticated user base. Microprocessors are extremely complicated, and this complexity will continue to rise commensurate with, among other things,

- Higher performance
- Higher transistor counts
- The increasing size of the installed base (which makes achieving compatibility harder)
- New features to handle new workloads
- Larger design teams
- More difficult manufacturing processes

This product complexity also implies a higher risk that intrinsic design or manufacturing flaws may reach the end user undetected. In 1994, such a flaw was found in Intel's Pentium processor, causing some floating-point divides to return slightly wrong answers. A public relations debacle ensued, and Intel took a $475 million charge against earnings, to cover the cost of replacing approximately 5 million microprocessors. In the future, if existing trends continue, microprocessor manufacturers may have tens or even hundreds of millions of units in the field. The cost of replacing that silicon would be prohibitive. Design teams are combating this problem in a number of ways, most notably by employing validation techniques such as random instruction testing, directed tests, protocol checkers, and formal verification.

What really sets microprocessors apart from the other tools that humankind has invented is the chameleonlike ability of a computer to change its behavior completely under the control of software. A computer can be a flight simulator, a business tool for calculating spreadsheets, an Internet connection engine, a household tool to balance the checkbook, and a mechanic to diagnose problems in the car. The faster the microprocessor and its supporting chips within the computer, the wider the range of applicability across the problems and opportunities that people face. As microprocessors continue to improve in performance, there is ample reason to believe that the computing workloads of the future will evolve to take advantage of the new features and higher performance, and applications that are inconceivable today will become commonplace.

Conversely, one challenge to the industry could arise from a saturated market that either no longer needs faster computers or can no longer afford to buy them. Or perhaps the ability of new software to take advantage of newer, faster machines will cease to keep pace with the development of the hardware itself. Either of these prospects could conceivably slow the demand for new computer products enough to threaten the hardware/software spiral. Then the vast amounts of money needed to fund new chip developments and chip manufacturing plants would be unavailable.

However, negative prognostications about computers or microprocessors have been notoriously wrong in the past. Predictions such as "I think there is a world market for maybe five computers" (Thomas Watson, chairman of IBM, 1943) or "photolithography is no longer useful beyond one micron line widths" have become legendary for their wrongheadedness. It is usually far easier to see impending problems than to conceive ways of dealing with them, but computer history is replete with examples of supposedly immovable walls that turned out to be tractable.

In its short life, the microprocessor has already proven itself to be a potent agent of change. It seems a safe bet that the world will continue to demand faster computers and that this incentive will provide the motivation for new generations of designers to continue driving the capabilities and applications of microprocessors into areas as yet unimagined.

ACKNOWLEDGMENTS

Various trademarks are the property of their respective owners.

BIBLIOGRAPHY

1. D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach, 2nd edition,* San Francisco: Morgan Kaufmann, 1996.
2. G. A. Blaauw and F. P. Brooks, Jr., *Computer Architecture Concepts and Evolution,* Reading, MA: Addison-Wesley, 1997.
3. D. P. Siewiorek, C. G. Bell, and A. Newell, *Computer Structures: Principles and Examples,* New York: McGraw-Hill, 1981.
4. M. S. Malone, *The Microprocessor: A Biography,* Santa Clara, CA: Springer-Verlag, 1995.

ROBERT P. COLWELL
Intel Corporation