Microprocessors Follow Mainframe Path

Many Popular Techniques Developed First by Mainframe Designers

by Bernard L. Peuto, Consultant

In the first part of this article (see MPR 3/31/97, p. 19), I compared the development of microprocessors and mainframes. In this second part, I further illustrate my belief that, having focused on personal data processing, microprocessors copied 50 years of mainframe development in roughly half that time. Will the new focus on virtual reality and information sharing take development down a different road?

Available Knowledge in 1971

Early microprocessors were very primitive and had badly designed architectures, which seems somewhat surprising, since the computer-science education of the day was excellent. Most major universities had started computer-science programs, funded by large federal-government research grants, in the late 1950s and early 1960s. After 10 years, they were well established.

Students there studied the major computers of the time: the IBM 360 and 370, the PDP-8, and the PDP-11 (introduced in 1970), as well as the dominant operating systems: OS/360, Multics, and Unix (after 1970). But with few exceptions, the knowledge gained was not used in microprocessor design until 1976, when the architects of what would become the Z8000 and 68000 were hired.

Among the best explanations for this state of affairs is the background of the initial designers, most of whom specialized in IC technology development and had to learn the requirements for a microprocessor from their customers. To their credit, they did learn, but it took time. The scarcity of transistors available for design has also been given as a reason, although it is more a pretext to justify the architectural weaknesses of early microprocessors. The limitations of available packaging may have played a role, too.

More interesting was the debate between those who felt that microprocessors would be a programmable replacement for logic and those who felt that microprocessors would be a data-processing engine. By 1985, the mainframe data-processing viewpoint had won, and fully trained computer scientists had key design positions.

Early Challenges: 1971–1989

Beyond the architectural issues, early challenges for mainframes had to do chiefly with implementation technologies. For logic, relays were used first, then tubes, then transistors, and finally ICs. For memories, delay lines were used, then Williams tubes, core memories, and transistor memories. A whole family of electromechanical peripherals was perfected: tape drives, card readers, drums, and disks. All these accumulated experiences were well understood in 1971, benefiting microprocessor-based systems.

Thus, the principal challenges for early microprocessors were architectural. The designers of the Intel 4004 have said that their first challenge was how to adapt the structure of the PDP-8 to the new silicon environment. The PDP-8, of course, was itself a fairly crude machine.

This copying of minicomputer-like structures took place between 1971 and 1974 and spanned the reigns of the 4004 and the 8080, including other microprocessors such as the Motorola 6800. (Here, as in the rest of this short paper, we can only give examples and not be exhaustive.) By the time the 8080 shipped in 1974, the focus had shifted to minimizing glue logic and providing a family of peripherals with each CPU, getting closer to a one-board computer.

The 16-bit generation of microprocessor-based systems became competitive with minicomputers, and the new challenges dealt as much with getting design wins as with architectural issues. One of the best-known symbols of this era is Operation Crush, the campaign to gain market share for Intel's 8086 over competing products from Motorola and Zilog, which ultimately helped Intel win the IBM PC design with the 8088.

The next architectural phase was the embodiment in microprocessors of the lessons learned from PCs that were on the path to becoming personal data processors. From 1978 to 1985, this phase started with the 8086 and ended with the 80386, which finally implemented virtual memory—an advance that had been implemented by other companies before Intel, including Zilog and National.

The last architectural phase of the early challenges closed with the introduction of on-board caches in the 80486 in 1989. Intel, again, was not the first company to introduce this feature.

Microprocessors Import Mainframe Concepts

Some concepts that had been established for mainframes were adopted without many questions by microprocessor designers. Most prominent was the so-called von Neumann architecture, named for John von Neumann's 1945 report embodying the lessons learned from ENIAC and introducing the EDVAC architecture. It established the importance of binary coding, sequential execution of instructions, and stored program memory, and it identified the differences between architecture (he called it logical design) and implementation.

The second readily adopted concept dealt with customer focus. Mainframes were first designed as numerical table calculators (for example, military ballistics tables) and quickly evolved into scientific and commercial products.

Real-time applications rapidly became an important product focus, too. As computers were sold in greater numbers, providing a complete system solution to all customer needs became a necessity, as did the requirement, after the IBM 360, for binary software compatibility.

Microprocessors started out as the core of a desktop calculator. They maintained the distinction between scientific and commercial applications and had a very significant focus on real-time applications as well. Providing a complete family of peripherals and development tools became a necessity as early as the 8080. Binary software compatibility with the first 16-bit microprocessors was a requirement for all subsequent products.

The most intriguing case of the unquestioned adoption of mainframe concepts deals with the software development cycle. Mainframe vendors at first considered software a minor component of their product, which they provided free to their customers. It did not take them long to realize the true cost of software development. IBM was the first to unbundle software in 1968. The massive delays and difficulties behind the development of OS/360 from 1964 to 1967 led also to the realization that software projects were different. Out of that experience, Frederick P. Brooks, Jr., the program manager for the 360, wrote the famous book, *The Mythical Man-Month*. By the 1970s, structured programming was the rage.

The simplest explanation for this unquestioned adoption of software concepts is that, by 1976, most microprocessor vendors were hiring software designers with a background in computer science. These employees naturally used the knowledge they had acquired with existing computers. The path was set for an accelerated appropriation of mainframe concepts, simply because there did not seem to be any justification for doing it differently.

Major Hardware Imports

Like mainframes, microprocessors started with an ALU- and instruction-centric view of computer architecture, so the first focus of their designers was on instruction formats and addressing modes. By 1979, some of the most complex formats had been adopted by the Z8000 and 68000. In 1982, when memory protection and capabilities were hot subjects, Intel's 80286 went overboard and adopted descriptor formats very similar to those of Multics and the Honeywell 6000 line. We still suffer today from this unfortunate decision.

As early as 1954 (in the IBM 701), floating-point was one of the data types supported in hardware for scientific computers. Primitive at first, it took its current look in 1961 with STRETCH, a very important IBM computer that implemented many advances later used in the 360.

Having learned from the problems of similar but not identical floating-point standards, microprocessor designers, with the collaboration of academics such as Professor Kahan at UC Berkeley, created the IEEE floating-point standard. It was first used by Intel in 1980 for the optional 8087 coprocessor. As with mainframes, optional floating-point units eventually gave way to internal floating-point units, as in the 80486 in 1989.

As memory speed reached 2 µs in 1961, STRETCH was the first to implement a simple form of instruction prefetch. Years later, it appeared in microprocessors: the Intel 8086 in 1978 and the Motorola 68000 in 1979.

Microcode was first suggested by Maurice Wilkes but became very prominent when the IBM 360 line used it to implement one architecture at multiple price/performance points and to emulate previous computers (the 1401 and 7094) to achieve software portability. The integrated version of microcode was first used by the 8086 and 68000 16-bit microprocessors.

Wilkes in 1965 suggested the concept of "slave memory," which we now call cache memory. IBM made it a well-known feature of the high-end 360 Model 85 in 1969. The first microprocessors with internal caches were the Motorola 68020, shipped in 1984, and the Zilog Z80000, shipped in 1986. The MIPS R2000 in 1986 directly controlled an external cache, and the Intel 80486 integrated its cache in 1989. Now, most high-speed microprocessors integrate an L1 cache and use an external L2 cache.

Sophisticated Pipelines

Another very important product from IBM was the 360/91, shipped in 1967 but originally introduced in 1964. It was IBM's fastest computer, in competition with the CDC 6600, the reigning supercomputer of the day. Most of the advanced pipelining concepts of today were invented for the 360/91.

The MIPS R2000 in 1986 was the first microprocessor to implement a simple pipeline with branch prediction favoring the branch-taken path. This pipeline was adopted for the 80486 in 1989. Register renaming and instruction scheduling, both concepts from the 360/91, were used by Intel for Pentium Pro in 1995, after several RISC microprocessors had already adopted them.

Virtual Memory

In the 1960s, virtual memory, protection, and the generalization from descriptors to capabilities dominated the mainframe's architectural innovations.

Paged virtual memory had been invented in 1962 as the one-level store for the English computer Atlas. Segmentation had been introduced in 1960 for the Burroughs B5000. The combination of segmented and paged virtual memory, as well as improved memory protection, was at the root of the Multics operating system. This system was based on a modified General Electric computer, the 645, later implemented as the Honeywell 600 and 6000 lines. Multics dominated computer-science education in the 1960s and early 1970s.

Rings of protection and the descriptors used to implement them gave way to fine-grained protection of objects, generally referred to as capabilities. They were first implemented in software in the CAL time-sharing-system in 1973.

Capabilities encountered severe implementation overhead problems and have been revisited only recently, after a long period of disfavor.

Confronted with the need to extend the 16-bit address spaces of early 8-bit microprocessors, both the 8086 in 1978 and the Z8000 in 1979 used segmentation. The Z8000 MMU also supported memory protection in 1979. The Intel 80286 implemented hardware protection rings *à la* Multics in 1982. A paged-virtual-memory version of the Z8000 MMU was also introduced. The Z80000 implemented a segmented and paged virtual memory in 1983. A similar system was introduced by Intel for the 80386 in 1985.

Changing Application Focus

At first, it seemed that microprocessors were being used mainly in calculators and similar logic-replacement applications. But the pressure to imitate existing computers was too strong, and so the first period, from 1971 to 1978, should be labeled the phase of poor computer implementations. As each subsequent generation was introduced, architectural weaknesses and mistakes were eliminated. By the time the 8086 rolled out in 1978, a microprocessor with a full complement of peripheral chips represented a credible minicomputer implementation.

With the introduction of the IBM PC in 1981, the application focus shifted to making better PCs, which is to say, better personal data processors. This period lasted from 1981 to 1989, when the 80486 shipped—the first time that a high-volume microprocessor could be considered a full mainframe CPU on a chip. During that period, IBM was a major investor in Intel, and its impact on the 386 and 486 architecture was probably quite significant.

The PC market quickly supported a very large software industry that produced killer applications like word processing, spreadsheets, and later desktop publishing, graphics, and telecommunications. These applications did not differ enough from mainframe applications to justify a different architecture. This similarity reinforced the value of copying the proven architectural concepts of mainframes.

The first difference in application focus between mainframes and microprocessors, the graphical user interface (GUI), appeared in the next period, from 1983 to 1990. For the first time, the computational focus of a microprocessor-equipped PC shifted away from being a CPU-centric data-processing application toward spending a significant fraction of cycles on the graphical manipulation underlying the implementation of a graphically oriented human interface.

Graphical User Interface

The search for a better interface between computers and users began in 1960 with an important pioneer, J.C.R. Licklider, who wrote a very influential paper, "Man-Computer Symbiosis." Licklider exercised an enormous influence on the development of computers through his control of the ARPA government funding that dominated all computer-science research at major universities from 1958 to 1975. Time-sharing systems (like Multics at MIT), AI (at Stanford and MIT), the ILLIAC IV, and the ARPAnet were some of the beneficiaries of ARPA's largesse.

Among the laboratories supported was SRI, where in 1968 Doug Englebart presented the electronic office demo: the first use of a GUI and a mouse to interface with a computer. This work, and the work done by Alan Kay at the University of Utah on the Dynabook in 1969, greatly influenced the direction of the user-computer relationship. The Dynabook was probably the first description of a portable computer. The University of Utah, one of the ARPA-funded labs specializing in graphics, pioneered many of the concepts used today in computer graphics.

In 1975, the Xerox Palo Alto Research Center (PARC) introduced the first personal computer with a GUI, the Alto. The workstation business felt its impact in the early 1980s. But the Apple Lisa in 1983 and the Macintosh in 1984 became the first examples of high-volume personal computers with a full GUI implementing the desktop metaphor.

In the DOS world, several products competed as GUI-enhanced operating systems. GEM from Digital Research in 1984 was one of them. But none survived Microsoft's lock on the DOS world. Work on Windows 1.0 started in late 1981; the system did not ship until 1985. Hampered by underpowered PCs and the lack of standard graphics support, the IBM PC-compatible world had to wait until Windows 3.0 in 1990 for the GUI environment to explode.

The differences between a GUI-enhanced PC and early mainframes are significant. But the architectural impact is reduced to the need to support a bitmap memory and a simplified multiprocessing architecture encompassing the graphics processor. Mainframes would not be used today without a network of workstations, PCs, or terminals implementing a graphically oriented interface. So even there, the gap between the two environments has been reduced.

New Application Focus

The current evolution of microprocessors points to emerging application foci different from the existing personal-data-processing focus. Will this result in new architectural concepts different from those of mainframes?

The number of logic transistors available for high-volume devices is now around 5 million and rapidly heading toward 10 million. With so much logic available for one product, perhaps we should abandon compatibility and design the architecture differently. This argument has been used to justify the absence of x86 compatibility in PDAs. But no significant performance gain will come from doing so, and given the significant penalty for developing new tools, I do not see any reason for significant differentiation.

Perhaps emerging new applications such as computing appliances (PDA-like devices, WebTV controllers, TV set-top boxes) could result in a different architectural optimization. The impact of consumer electronics on software could be significant, too. But again, I do not see any reason for significant differentiation at the architectural level.

Virtual reality clearly requires many more floating-point cycles than current architectures provide. But there too, the architectural impact seems more likely to be restricted to the design of specialized coprocessors.

Multimedia has introduced, for the first time in many years of computer development, the need to process a new data type. In response, new instructions have been introduced for the x86 architecture (MMX) as well as for others, such as the PA-7100LC. DSP capabilities are becoming a stronger element of PC products. But here again, the architectural concepts are not really new, and the additions, although significant, do not change the overall architecture.

I believe that the greatest change looming on the horizon comes from the current revision of the computing utility, i.e., the Web. In his famous paper, Licklider described a computing utility as ubiquitous as electrical utilities, only a wall plug away and available for all users. In fact, his computing utility would have provided two forms of sharing: processor sharing, which leads to time sharing, and data sharing, which is being reinvented with the World Wide Web.

Computational utilities, first implemented as time-sharing utilities, did not succeed in the long term and were replaced by the equivalent solution of dedicated personal computers. Information sharing, long a minor component of the computing industry (with dedicated services like Lexis/Nexis), is making a grand comeback with the Web, the ultimate information-sharing utility. One can only express admiration for Licklider, who predicted it in 1960! Unfortunately, we are too early in the development cycle of the Web to forecast any significant architectural differentiation for microprocessors that may result.

Recommended Readings

I have not yet found a good book on the history of microprocessors. The Campbell-Kelly and Aspray volume is an excellent history of computers, with good chapters on microprocessors. The importance of IBM makes the three excellent books written or co-written by Pugh fascinating "must reads." Goldstine and Stern have written excellent accounts of the early days of computers.

Finally, everybody interested in computer architecture should own copies of the Bell and Newell books, which reprint articles on some of the most important computers.

Bashe, Charles J.; Johnson, Lyle R.; Palmer, John H.; and Pugh, Emerson W. *IBM's Early Computers*. MIT Press, 1986.

Bell, Gordon C., and Newell, Allen. *Computer Structures: Reading and Examples.* McGraw-Hill, 1971.

Campbell-Kelly, Martin, and Aspray, William. *Computer: A History of the Information Machine*. Basic Books, 1996.

Goldstine, Herman H. *The Computer: From Pascal to von Neumann*. Princeton University Press, 1972.

Pugh, Emerson W.; Johnson, Lyle R.; and Palmer, John H. *IBM's 360 and Early 370 Systems*. MIT Press, 1991.

Pugh, Emerson W. *Building IBM: Shaping an Industry and Its Technology*. MIT Press, 1995.

Siewiorek, Daniel P.; Bell, Gordon C.; and Newell, Allen *Computer Structures: Principles and Examples*. McGraw-Hill, 1982.

Stern, Nancy. *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*. Digital Press, 1981.

Missing From Microprocessors

A few architectural innovations available on mainframes have not been copied in microprocessors. Fairly early in their development, mainframes provided support for multiple independent processors. The Honeywell 6000 line in the 1970s is an example. We have yet to see multiple processors on a chip, due in part to the lack of support for this feature in Windows until now. I expect this situation to change.

A few more esoteric pipelining structures have been implemented in mainframes, like multiple branch-path execution. These structures, too, will be copied.

The greatest differences between mainframes and microprocessors are in input/output. Mainframes, as complete data-processing systems, provided an elaborate set of channels and I/O interface structures. One of the great advances of the 360 in 1965 was its standardization of these interfaces. IBM has since considerably increased the sophistication and speed of its I/O subsystems, including complete local and remote storage hierarchies based on disk arrays.

Because microprocessors are only the central processing unit, personal-computer systems have not yet focused on these issues. Much of the current I/O support can be traced to the primitive support of the 8086. As integration increases, this support will change. Already, USB and other "channels," like IEEE 1394, are likely candidates for integration.

Conclusions

With a past focus on personal data processing, microprocessors have rapidly copied 50 years of mainframe development, although the new focus on virtual reality and information sharing could take development down a different road. But there are no new programmable computing structures ready to emerge today that could replace the microprocessor or separate its architectural evolution from that of mainframes. M

I would like to acknowledge the help provided by Len Shustek and by the editorial board of Microprocessor Report *in the preparation of this article.*

Dr. Bernard L. Peuto was the architect for the Z8000, the Z8, and several other Zilog products. Now president of Concord Consulting, he develops business strategies for microprocessor-based products. He can be reached at 415.851.0194 or by e-mail at blpeuto@peuto.com.