The 50 Year History of the Microprocessor as Five Technology Eras

John L. Hennessy, *Stanford University, Stanford, CA, 94305, USA*

The evolution of the microprocessor can be organized into five eras, each distinguished by common trends in the evolution of microprocessors. Most of these eras are around ten years and represent a shift from the previous era. I have had the privilege of being involved in some way for roughly 48 of the 50 years, so this is also a somewhat personal view.

FIRST DECADE 1971–1981

This decade, which began with the birth of the Intel 4004 in 1971, was dominated by three trends:

1. an increase in width as Moore's law-enabled processors to go from 4 to 8 to 16 and eventually 32 bits;
2. a rapid increase in instruction sets, often motivated by assembly language examples and enabled by microcode implementations (see the early Intel and Zilog microprocessors);
3. a rapid increase in clock speeds enabled by faster transistors.

The emergence of the personal computer (Apple II in 1977 and IBM PC in 1979) enabled the shrinkwrap software industry and reinforced the importance of object code compatibility, which the first microprocessors did not exhibit. The Motorola 68000, which appeared in production in the late 1980s, was the first 32-bit microprocessor and offered many of the features associated with minicomputers.

RISC AND PIPELINING YEARS 1981–1995

As Moore's law progressed, it seemed likely that microprocessors would evolve into full blown computers. The accompanying growth in DRAM capacity (from 1 Kib in 1971 to 16 Kib in 1981) reduced the need to hand-optimize code and program in assembly language. The subsequent movement to higher level languages inspired the groups at IBM, Berkeley, and Stanford to explore what became the RISC ideas. In addition to targeting compiler output, rather than handcrafted assembly language, they also emphasized elimination of microcode and compilation to an efficient hardware implementation.

The RISC ideas led to an explosion in the use of pipelining in microprocessors, which generated a rapid increase in clock rate performance. This era was characterized by incredible annual performance growth of approximately 1.5 times, enabled by the inclusion of caches and much faster clocks.

The capstone events in this era were the introduction of 64-bit processors (the R4000, followed by the DEC Alpha, and others) and a growth in pipeline depth from 5 to 7–10 or more stages, which led to clock rates rivaling ECL mainframes.

INTENSIVE ILP YEARS 1995–2005

The third era is characterized by an intensive focus on exploiting instruction-level parallelism and trying to reduce the clock cycles per instruction to less than 1. The ILP-intensive processors fall into two broad categories: superscalar and very long instruction word (VLIW). The superscalar processors used a combination of hardware techniques and software scheduling to issue more than one instruction per clock, while the VLIW approach relied on little hardware support and intensive compiler scheduling to organize independent operations into issue packets. The VLIW approach did not succeed in the end, due to a variety of factors, most importantly the inability to achieve high performance on less structured integer programs.

The initial superscalar processors (such as the MIPS R8000, PowerPC 604, and later Intel Pentium) used static scheduling, meaning that instruction issue was blocked if the next instruction's operands were not yet available. This approach was rapidly followed by a shift to dynamic scheduling (allowing instructions to be executed out of order) and speculation (allowing instructions to be executed before a preceding branch

0272-1732 © 2021 IEEE
Digital Object Identifier 10.1109/MM.2021.3112301
Date of current version 19 November 2021.

was resolved), starting with the AMD K5 and MIPS R10000. For the next eight years, designers would increase the issue rate, add additional execution units, and greatly increase the size of the window that the hardware would examine when looking to execute instructions. For most of this period, the annual performance increase of 1.5 times was delivered, although a slowdown in performance growth appeared in the last few years of this era.

MULTICORE/MULTITHREADING YEARS 2005–2015

By 2005, it was clear that a new direction was needed. The inefficiency encountered when issuing and examining more instructions and the associated power consumption led to the end of the road for the ILP exploitation. Multicore and multithreading approaches shifted the burden for finding parallelism from the hardware to the programmer and programming system. Although some dual-core processors had appeared earlier (e.g., the IBM Power4), by 2005 all new microprocessors were multicore and many incorporated multithreading.

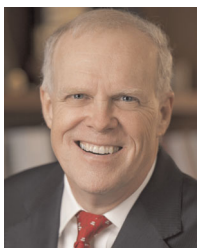
The multicore approach scaled well, at least from a hardware perspective. The challenge was to use the cores efficiently, overcoming the problems presented by Amdahl's law and the overhead of coordinating multiple processes executing on different cores. The breakdown in Dennard Scaling, which began shortly after this era started, increased the power consumption of larger multicores, further eroding the energy efficiency particularly when speed-ups were limited. This led to the era of Dark silicon, when cores would be shut off both to limit power consumption and to avoid overheating. Multicores found many uses when executing independent programs in parallel but achieving consistent and energy-efficient performance growth for integrated applications was harder. The subsequent slowdown in Moore's law, starting near the end of this era, further undermined the strategy of simply scaling up the number of cores

RISE OF HETEROGENEOUS MICROPROCESSORS 2015–20??

The breakthroughs in the application of deep neural networks both for image recognition and complex game playing would require even higher levels of performance both for training and for inference, and a new approach was needed. Initially, that approach used domain-specific architectures (DSAs) as accelerators together with domain-specific languages, which enabled high performance for these architectures. Efficiency was gained by a variety of techniques including reduced precision, SIMD rather than MIMD organizations, and user-controlled memories (versus caches). Initially, these DSAs were implemented as accelerators on adjacent chips (e.g., GPU or TPU), and for training, which is much more computationally demanding, this is still the favored approach.

For inference purposes, which occur at the edge, rather than in a large data center, the DSAs are simpler and are usually integrated with the processor, leading to a heterogeneous design. The Apple M1, recently introduced, is a prime example of this new approach. The M1 includes both high-performance and low-power cores (using an ARM RISC-based instruction set), a graphics accelerator, and AI accelerator for inference. The heterogeneous era has fully arrived.

Achieving performance in the future will require more collaboration between hardware and software architects. New programming models and progress on compilation will be as important as hardware innovations. Fortunately, despite the incredible 50-year run of the microprocessor and its performance growth of close to a million times, researchers and designers seem to have lots of interesting possibilities to continue this incredible story. I look forward to the next 50 years.



JOHN L. HENNESSY is the James F. and Mary Lynn Gibbons Professor of computer science and electrical engineering with the Stanford School of Engineering, Stanford, CA, USA, and the Shriram Family Director of Stanford's Knight-Hennessy Scholars, the largest fully endowed graduate-level scholarship program in the world. He was the recipient of the 2012 Medal of Honor of the Institute of Electrical and Electronics Engineers, the 2017 ACM A.M. Turing Award (jointly with David Patterson), and the 2020 BBVA Foundation Frontiers of Knowledge Award (jointly with David Patterson). Hennessy received a bachelor's degree in electrical engineering from Villanova University, Villanova, PA, USA, in 1973, and master's and doctoral degrees in computer science from the Stony Brook University, Stony Brook, NY, USA in 1975 and 1977, respectively. Contact him at hennessy@stanford.edu.