PROCEEDINGS A

rspa.royalsocietypublishing.org

Perspective

Cite this article: Furber S. 2017 Microprocessors: the engines of the digital age. *Proc. R. Soc. A* **473**: 20160893. http://dx.doi.org/10.1098/rspa.2016.0893

Received: 7 December 2016
Accepted: 8 February 2017

Subject Areas:
computer-aided design

Keywords:
microprocessor, Moore's Law, Intel, ARM, SpiNNaker

Author for correspondence:
Steve Furber
e-mail: steve.furber@manchester.ac.uk

An invited Perspective to mark the award of the Royal Society's Mullard Award to the author in 2016.

THE ROYAL SOCIETY
PUBLISHING

Microprocessors: the engines of the digital age

Steve Furber

School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK

SF, 0000-0002-6524-3367

The microprocessor—a computer central processing unit integrated onto a single microchip—has come to dominate computing across all of its scales from the tiniest consumer appliance to the largest supercomputer. This dominance has taken decades to achieve, but an irresistible logic made the ultimate outcome inevitable. The objectives of this Perspective paper are to offer a brief history of the development of the microprocessor and to answer questions such as: where did the microprocessor come from, where is it now, and where might it go in the future?

1. Introduction

A computer requires memory to hold programs and data, a processor to execute those programs using the data, and I/O (input/output) capabilities to interface to the outside world. The intense action takes place within the processor, and the microprocessor achieves integration of all of the processing functions on a single microchip. The introduction of the microprocessor represented a breakthrough in terms of the size and cost of a computer system, and was one of the advances that made the personal computer (PC) revolution, and later the mobile revolution, come about. The next revolution in computing in which the microprocessor will play a central role is IoT—the Internet of Things.

Today, thanks to the exponential progress in the number of transistors that can be fabricated on a single chip (Moore's Law [1]), the term 'microprocessor' has become less clear in its precise meaning. The processor chip in a typical PC is a formidable beast with several processor 'cores', complex cache memory hierarchies (though the main memory is still off chip) and very high-performance I/O interfaces (though most of the I/O components are still off chip). The nearest analogue to the original microprocessor is the individual processor core, and this is the interpretation that will be used in this paper.

The key benefit of the microprocessor results from integrating all of the components of a computer that are involved in executing instructions together on the same microchip. Instructions are fetched from external memory (though often today this is cache memory on the same chip) and data are loaded and stored from external memory (again, often using on-chip caches), but the instruction decode and execute logic is all collocated, resulting in significant performance and energy benefits compared with splitting the processing functions across two or more chips, as was done prior to the arrival of the microprocessor. These benefits accrue because on-chip connections incur much lower parasitic capacitance than do off-chip connections, and most of the delays and energy consumed by a processor result from driving capacitive loads up and down during execution.

In this Perspectives paper, I will offer a personal view of the key developments in the history of the microprocessor, which can be divided quite cleanly into decade-by-decade progress. This is not an exhaustive history, but an attempt to highlight the key issues as they emerged, and it starts in the 1970s.

2. The 1970s: emergence

Back in 1969, Nippon Calculating Machine Corporation approached Intel with a proposal for Intel to build 12 custom chips for its new Busicom 141-PF* range of calculators. Intel came back with a counter-proposal to develop just four chips, one of which could be programmed to meet the needs of the range. That programmable chip was the Intel 4004. Intel bought the rights to those chips back from the customer and launched the Intel 4004 and its accompanying chipset with an advertisement in the 15 November 1971, issue of Electronics News: 'Announcing a New Era in Integrated Electronics'. The microprocessor was born.

The Intel 4004 [2] used 2300 transistors on a 10 µm pMOS process. It could be clocked at frequencies up to 740 kHz and would execute up to 92 600 instructions per second. It was a 4-bit device (defined by the 4-bit width of the data bus), with 8-bit instructions and a 12-bit address bus, all integrated into a 16-pin dual-in-line package. From this modest start, a new era did, indeed emerge!

Through the 1970s, a diverse range of microprocessors were developed, the great majority of which were 8-bit devices with 16-bit address buses packaged in 40-pin dual-in-line packages. These included direct descendants of the 4004 such as the Intel 8008 and the 8080, the Signetics 2650 (my first microprocessor, now largely forgotten!), the Motorola 6800, the National Semiconductor SC/MP (from which Acorn was bootstrapped), the MOS Technology 6502 and the Zilog Z80. The 6502 drove down the price to new levels of affordability, and together with the Z80 was largely responsible for the emergence of the computer hobbyist movement which in turn led to the home computer revolution of the 1980s.

Thanks to the 8-bit microprocessor, the computer was now out of the hands of the white-coated computer operator employed by the large corporation, and into the hands of the young enthusiast—students and entrepreneurs. When those young enthusiasts included the likes of Steve Jobs and Steve Wozniak creating the Apple 1, the seeds of change were well and truly sown.

3. The 1980s: RISC versus complex instruction set

By the beginning of the 1980s, the PC market was established, and it was beginning to break out from its hobbyist origins into the wider home market, with basic computer familiarity and gaming being the primary uses in the home. These machines used 8-bit microprocessors, but there was a clear roadmap up to 16-bit microprocessors.

In 1981, IBM introduced the IBM PC, also powered by an 8-bit microprocessor—the Intel 8088—clearly targeting desk-tops in business. It was not an especially ambitious machine, but the IBM name carried a lot of weight, and the IBM PC came to set the standard for most of the PC market up to this day (though IBM itself no longer makes PCs). Only Apple offered some credible degree of competition.

Thus, the PC was established, and the scene was set for the microprocessor manufacturers to move their customers up to 16-bit machines, as more performance would clearly sell more machines. But how should a 16-bit machine be architected? The established microprocessor manufacturers were all large semiconductor companies who knew a lot about making chips but far less about computer architecture. 8-bit microprocessors were relatively simple to design, but 16-bit architectures were a completely different kettle-of-fish.

There was a readily available source of architectural insight into how to configure a 16-bit machine as the minicomputer business had been there before. The leading minicomputer in the 1970s was the (32-bit) DEC VAX 11/780. It did not use a microprocessor, but it showed how to architect such a machine using a multi-chip processor, and why should not a microprocessor do something similar? The 11/780 architecture was very complex, and reflected the desire at the time to ‘close the semantic gap’ between the high-level language and the machine instruction set. 16-Bit microprocessors should follow this trend, within the constraints of the limited transistor resource on a single chip.

But some folk had other ideas! In 1980, David A. Patterson and David R. Ditzel published their seminal paper ‘The case for the reduced instruction set computer’ [3]. This paper made the very strong case that optimizing an architecture for the limited resource on a single chip was quite different from optimizing it for a multi-chip processor such as that on the VAX 11/780. Their arguments were strong, and were backed up by a real chip design—the Berkeley RISC I—that was being designed by a postgrad class in one session. There were other designs around at the time that reinforced this message, most notably the IBM 801 [4] and soon thereafter the Stanford MIPS (Microprocessor without Interlocking Pipeline Stages) [5]. The fundamental case was that an architecture based on a 1970s minicomputer would have a very complex instruction set (CISC!), which incurred a lot of chip area for the microcode ROM to map all of those instructions into basic instruction elements. With RISC that complexity was reversed; by keeping the instruction set as simple and regular as possible, no microcode ROM would be required, so there were more transistors available for architecture features that gave more benefit, such as a full 32-bit instruction set and pipelined execution (which was also facilitated by the regular instruction set).

The mainstream microprocessor manufacturers were unconvinced by all of this academic argument, and indeed spent most of the 1980s expressing their firm opposition to the concept (though by the end of the 1980s most had succumbed and had some sort of in-house RISC project underway). However, away from the mainstream, smaller companies considering designing their own processors lapped this all up. One such company was Acorn Computers in Cambridge, UK, who were responsible for the design of the very successful BBC Microcomputer, and were struggling to see how they should move up to 16-bit processing. At its peak, Acorn employed around 400 staff and some in-house chip design expertise, and they had worked closely with VLSI Technology, Inc., of San Jose, California, so they were developing some experience in designing chips from scratch.

Acorn started by looking at the 16-bit microprocessors emerging from mainstream industry, but found that they had two principal drawbacks:

- Real-time performance: the BBC Micro made extensive use of the 6502’s good real-time response to handle a lot of complex I/O in software. The emerging 16-bit microprocessors had significantly inferior real-time capabilities.
- Memory bandwidth utilization: Acorn’s engineers had formed the view that the primary determinant of performance was the processor’s ability to access memory at high bandwidth. The most expensive component in a PC was the memory, but the 16-bit processors of the day could not make full use of the bandwidth offered by those memories—surely a mistake?

So the Acorn team had started to think about designing their own microprocessor to overcome these perceived deficiencies. The RISC philosophy from Berkeley found an eager audience, and the Acorn RISC Machine (later simply ARM) project was born. The ARM was designed from the outset as a 32-bit machine, so Acorn largely skipped the 16-bit generation. Why not? 32 bits should give you twice the bandwidth of 16 bits, and hence twice the performance!

Acorn was, of course, not alone in pursuing the RISC idea in the commercial domain. Stanford University spun out a company to commercialize derivatives of their RISC work, and again in the UK, Inmos Ltd pursued a somewhat parallel path (though not influenced by RISC) to develop the transputer—a single-chip machine with processor, memory and communications on the same chip, in a form that enabled easy scaling up to large-scale parallel machines. And there were others [6].

4. The 1990s: clock wars and Systems-on-Chip

Through the 1990s, Moore's Law delivered ever-increasing transistor resource that enabled full 32-bit RISC and CISC microprocessors to be delivered with ever-more complex microarchitectures, including cache memories, translation look-aside buffers, etc. —the full gamut of tricks and features that had been developed in a different age in mainframe and minicomputers, and then some.

The characteristic that drove microprocessor design more than anything else was the desire for faster clock rates. This became the major marketing differentiator between high-end processors, leading to the 'clock wars', where selling microprocessors to a broad and largely non-technical market depended increasingly on claims for extreme clock rates that often had little or no direct relationship to any performance benefit realizable by a typical user.

At the same time, in a very different market, mobile systems were developing in a very different way. The 1990s saw the introduction of the mobile System-on-Chip (SoC) where, instead of using the growing transistor resource to improve the performance of the microprocessor, it was used to bring more and more system functions onto the same microchip as the microprocessor. For this purpose, a small and simple microprocessor left the maximum space for other functions, and the newly formed ARM Ltd had the smallest and simplest microprocessor offering in this newly emerging market [7]. Nowhere was this more vital than in the digital mobile phone handset market, where Europe had stolen a lead over the rest of the world, and when the leading handset manufacturer of the time—Nokia—adopted the ARM for its products, the consequences were highly disruptive.

The Nokia deal was not a foregone conclusion. ARM's RISC heritage conveyed many advantages in terms of performance and simplicity, but it had one significant drawback: the fixed 32-bit instruction set architecture resulted in worse code density than was achieved by the variable-length instructions used by the CISC competition, and code density matters here. Poor code density leads to larger code memories and more power dissipated in code fetching from those memories.

ARM Ltd addressed the code density issue with an imaginative leap. They introduced the Thumb 16-bit instruction set [8], where each Thumb instruction is a compressed form of a 32-bit ARM instruction. Thumb instructions are decompressed using simple combinatorial logic at the front of the ARM execution pipeline, which is otherwise pretty much unchanged. A mode change switched the processor between 32-bit ARM and 16-bit Thumb execution, and the code density problem went away. Nokia was convinced, and ARM's route to domination of the mobile phone handset business was established.

All this SoC development attracted little attention from the high-end microprocessor companies. ARM was not even pretending to play the clock-wars game; it was in a different business at a very different performance level.

5. The 2000s: many cores make efficient work

Shortly after the turn of the Millennium, the world of the high-end microprocessor changed beyond all recognition. The clock wars of the 1990s had run into a wall—power. Chips were simply getting too hot, and the problems of cooling them were becoming insurmountable.

High-profile development programmes for very high clock-rate products were scrapped, presumably at considerable expense, and another way forward had to be found. Fortunately, there was a simple solution. Instead of using the (still) ever-increasing transistor resource to make a single microprocessor go faster, use that resource to put two (or more) identical microprocessors onto the same microchip. Owing to the fundamental physics that defines the power consumption of a CMOS circuit, two half-speed processors use about half the power of a single double-speed processor. The only down-side is that the two processors are much harder to program than is the single one, but that is a problem for the user, not the manufacturer; in any case, there is no longer any choice in this matter. Go parallel, or accept performance stagnation!

Meanwhile, in the mobile phone backwater, things were also changing. Simple mobile telephone handsets were becoming smarter, with pretensions to functionality that had previously been the domain of the computer. Text messaging had always been a mobile phone function, but now users wanted to read their emails on their phones, and maybe even browse the web. Mobile phones needed more compute power, but still on a very tight power budget (a mobile phone handset can accommodate around 3 W of processor dissipation before it becomes uncomfortable for the user), so multicore solutions also emerged here.

The real turning point came with the introduction of the first Apple iPhone in 1997. Apple reinvented the smartphone concept in the way that only Apple can, and introduced a user interface that gave the smartphone user access to most of the functionality of a desktop machine, albeit somewhat restricted by the size of screen that could be fitted onto a product designed to fit into a pocket, in addition to phone capability and additional functionality resulting from a range of sensors built into the machine.

All of this required the humble mobile processor to step up to a much higher performance mark. A smartphone incorporates many processors, not just the frontline application processors, but those frontline processors were now being asked to deliver the performance of a 1980s supercomputer just to keep the user interface smooth and responsive. In effect, smartphone processors were being asked to deliver performance approaching that of a desktop processor but on a smartphone power budget.

6. The 2010s: mobility, performance and machine learning

The present decade has seen dramatic growth in mobile technology, with smartphones dominating the mobile phone handset market, driving up the demand for mobile data bandwidth. Alongside the smartphone, its bigger brother, the tablet has gained market acceptance, again driven initially by Apple's innovative product developments and marketing. The tablet's successful format has emerged as a large-screen variant of the smartphone rather than as a mobile version of a PC, previous attempts to introduce which largely floundered.

With the growth of the market for smartphones and tablets, screen resolution has improved and processor performance has been pushed hard within the very restrictive power and battery-life limits. Who would have imagined that a humble mobile phone would ever require more processing power than could be delivered by a state-of-the-art 32-bit processor? Yet 64-bit machines now dominate high-end smartphones and tablets. The overwhelming majority of these processors are based on designs from ARM Ltd, who by the beginning of 2016 had seen 85 billion ARM-powered chips shipped by their global partnership, giving them absolute numerical domination of the world microprocessor market.

This decade has also seen dramatic growth in 'cloud' computing—vast warehouses full of servers and storage systems [9], predominantly powered by Intel's high-end microprocessors. The scale of these largely invisible systems beggars belief, with global corporations such as Amazon and Google having multiple datacentre installations around the world, collectively using millions of high-end Intel microprocessors to provide Internet services such as search and video streaming, with individual power budgets in the region of 100 MW—a significant proportion of the electrical power output of a small power station.

Some of the key user functions available on smartphones—such as speech recognition—are, in fact, currently delivered through cloud services. This synergy of low-power user-friendly mobile platforms, high-speed digital wireless communication and huge cloud compute resources underpins today's consumer smartphone experience.

Several of the cloud applications employ 'deep learning' [10] techniques—machine learning using large-scale multi-layered neural networks. There is a lot of interest around accelerators for the neural networks as employed by deep learning systems, but the jury is still out as to the best approach to take here. Graphics processors (GPUs) have become increasingly usable for highly parallel numerical computation, and this is currently the leading approach to accelerate deep learning. An innovative approach is demonstrated by Google's Tensor Processing Unit [11], a custom chip designed to accelerate machine learning by exploiting the reduced computational precision typically required for such algorithms. Brain-inspired 'neuromorphic' technologies [12] offer greater energy-efficiency, but are relatively unproven. All manner of intermediate solutions are emerging, and this is an area that is ripe for start-up companies and entrepreneurial innovation.

7. Technology trends

Underpinning the progress in microprocessors is the progress in semiconductor technology over the last half century. This progress has been exponential in almost all respects, including:

- the growth in the number of transistors that can be integrated on a single microchip, which has followed Moore's Law [1] from a few thousand in the early 1970s to a few billion today;
- the matching progress in memory technology in terms of the number of bits that can be stored on a chip, from a few thousand in the early 1970s to billions today;
- the shrinkage of the transistor feature size which has made this growth possible, from 10 µm in the early 1970s to around 10 nanometres today;
- the increase in clock speed from just under 1 MHz in the 1970s to around 3–4 GHz today, a figure that is limited by power rather by the technology itself; and
- the cost of the design of, and the manufacturing facility for, a state-of-the-art microprocessor.

For much of the history of the microprocessor, there has been a win–win scenario whereby smaller transistors have delivered faster, more efficient and cheaper functionality, and the rate of progress has been governed principally by the time required to recoup enough revenue from one generation to cover the costs of developing the next generation. This virtuous circle has now come to an end—the cost per function is now increasing as transistors shrink beyond about 30 nm, and the economics of design are under significant strain.

If Moore's Law is no longer delivering progress, what other options are there? Microchips are still two-dimensional, with all of the action close to the surface of a thin sliver of silicon. There is growing interest in three-dimensional packaging [13]—stacking microchips on top of each other, thereby reducing the distance that signals have to travel from one chip to another, for example, from the microprocessor to its memory. To a first approximation, one can consider the energy required to perform a given computation to be determined by the number of bits of data that are moved in the course of that computation multiplied by the distance that they move. Three-dimensional packaging therefore offers the prospect of improving energy-efficiency, but it brings with it quite severe thermal considerations: memory chips, in particular, cannot be allowed to get too hot if they are to operate reliably, so stacking sensitive memories on top of power-hungry microprocessors is not a recipe for reliable operation.

8. The future: Internet of Things, heterogeneity and dark silicon

Although there is always the risk of technology disruptions rendering any attempt to predict the future futile, the next decade or two of developments of mainstream microprocessor technology seem to be fairly settled. It seems unlikely that radical new technologies (such as the much vaunted quantum computing) will have a significant impact on mainstream computing within the next 20 years, so the world of the microprocessor will likely be dominated by trends that are already visible today.

Among these visible trends, IoT (the Internet of Things) [14] will affect the design of the largest volume of microprocessors, and is largely unaffected by the end of Moore's Law. IoT envisages a world where local connected intelligence is inbuilt in everything, epitomized by the 'Internet light bulb' and the connected home. Sensors are everywhere, relaying their sensed information back into cloud services. The world will have a very fine-grained nervous system. The microprocessors that power the periphery of this network will need to be small and efficient, and operate with no maintenance, often harvesting energy from their environment (since replacing batteries will be out of the question). They will number in the hundreds of billions. In many ways they may be similar to today's microcontrollers, but security will be a huge issue. There have already been issues with computer systems in cars being hacked; when the whole planet is available for hacking, security will be paramount and innovation will be required to achieve this within the very modest scale and resources of an IoT device.

Higher up the microprocessor family tree, energy-efficiency will dominate in a way that has a quite different manifestation. Efficiency considerations, together with the end of Moore's Law, will drive microprocessors towards heterogeneous solutions, with tuned accelerators available to reduce the energy demands of particular classes of application. GPUs were an early example of an application-specific accelerator, though they were motivated primarily by performance requirements rather than by energy-efficiency. The current trend in seeking efficient accelerators for deep learning, probably alongside newly emerging approaches to artificial intelligence, will continue and will be joined by other application-specific accelerators.

The overall picture that emerges, then, is that future microprocessors will have a few complex 'fat' cores to maximize the performance of code that will not parallelize, together with many simpler 'thin' cores to run parallelizable code more efficiently, and a range of special-purpose accelerators to support important computational kernels even more efficiently than can the thin cores. Heterogeneity will become a common feature of high-end many-core microprocessors.

With all of this compute resource available, the run-time system will have to manage the power and performance levels of each computational unit to keep power under control, and at any time a significant proportion of the computational resource will have to be powered down, since running it all at the same time will result in excessive power consumption. These powered down units will form the 'dark silicon' [15].

9. Progress over one career

I have offered a personal view of the key stages in the progress of the microprocessor over the half century of its existence, and will close by offering two specific data points that represent the start and towards the end of my own career in terms of microprocessors and microprocessor-based systems into which I have had significant input:

— The first ARM processor (figure 1), retrospectively called ARM1 since there have been many successors, where the design started late in 1983 and the first silicon chip was operational on 25 April 1985.

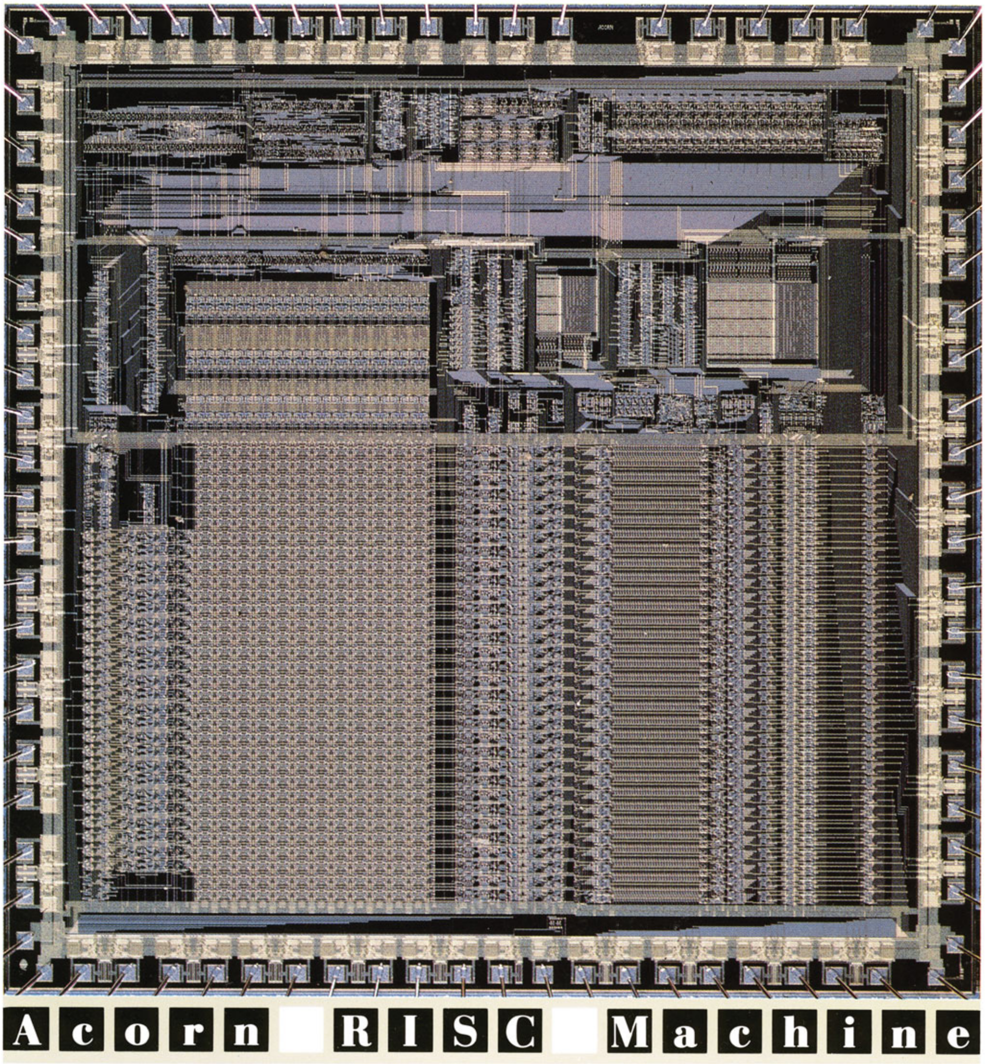

Figure 1. The first ARM chip, delivered in 1985. The regularity of the layout—particularly of the 32-bit datapath that occupies over half of the chip area at the bottom of the figure—betrays the manual process used for the design. The chip incorporates 25 000 transistors on a 7 × 7 mm area. Connections to external circuitry are made through the outer ring of input/output circuitry. (Online version in colour.)

— SpiNNaker (figure 2), where the design started in 2006 and the first silicon chip was operational in 2011. The large-scale machine (figure 3) was turned on under the auspices of the European Union Human Brain Project in April 2016.

The development of the ARM was described earlier in the historical account. The first ARM microprocessor was designed on a 3 µm CMOS process, using around 25 000 transistors on a 7 × 7 mm silicon chip. It would execute at up to 6 MHz, processing six million instructions per second while using about 0.1 W of electrical power.

A quarter of a century later, the SpiNNaker processor was designed on a 130 nanometer CMOS process (an old, but economic, technology by 2006 standards), using around 100 million transistors on a 1 × 1 cm silicon chip [16]. The SpiNNaker processor incorporates 18 ARM processor cores, each with a local memory system (which account for the great majority of the

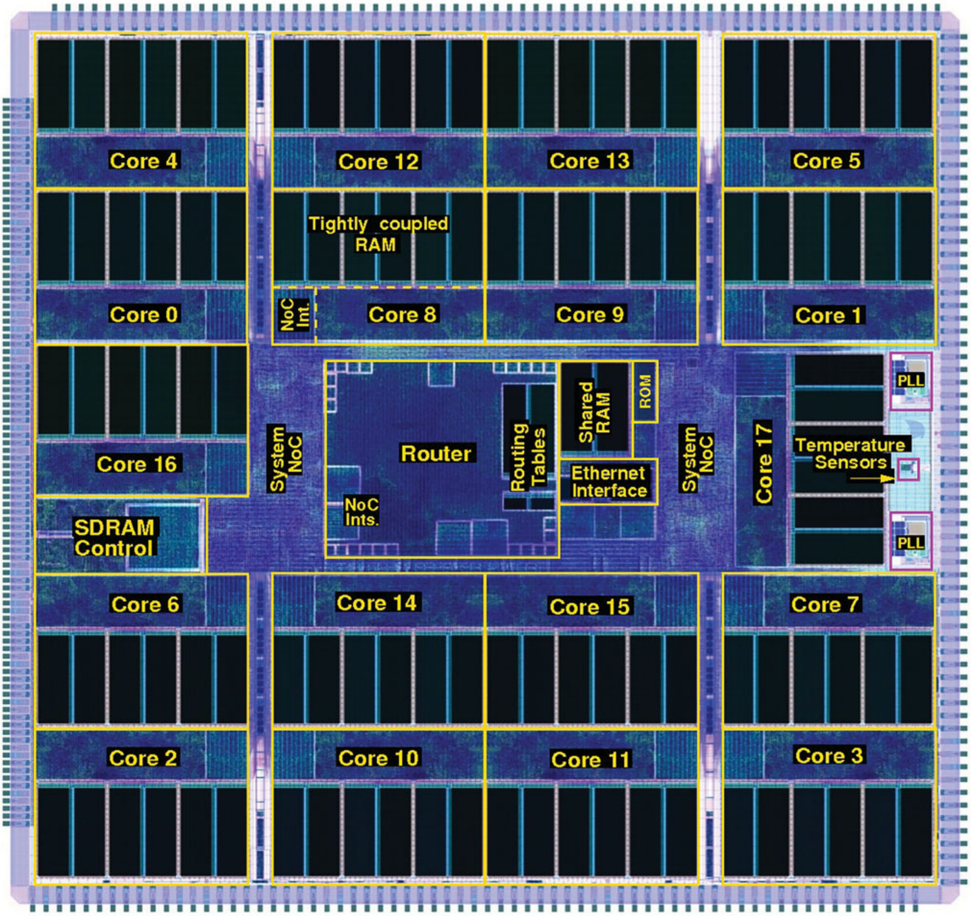

Figure 2. The SpiNNaker chip, delivered in 2011. Each of the 18 cores includes an ARM processor and sundry additional circuitry within the lighter rectangular area. The darker area within each core is memory (RAM). The chip incorporates 100 000 000 transistors (the great majority of which are in the memories) on a 1 × 1 cm area. Again, connections to external circuitry are made through the outer ring of input/output circuitry. (Online version in colour.)

transistors), and a wide range of system support components. The ARM cores operate at up to 200 MHz, giving the chip a total throughput of up to 3.6 billion instructions per second, with a total electrical power consumption (including the 128 Mbyte memory incorporated into the same package) of 1 W.

SpiNNaker was designed to support large-scale brain and brain-like computational models in biological real time, and as such the processor is a node in a large machine [17]. The Human Brain Project SpiNNaker platform uses 28 800 SpiNNaker processors to yield a machine with half a million ARM processor cores occupying a total active silicon area approaching 5 m^2.

SpiNNaker exemplifies some of the principles outlined earlier for future trends in computer design:

- massive parallelism using many ‘thin’ cores (without the fat cores, as the SpiNNaker target application area is ‘embarrassingly’ parallel;
- three-dimensional packaging (which is fairly basic in SpiNNaker’s case) to minimize the distance over which data moves;
- prioritizing energy-efficiency over performance.

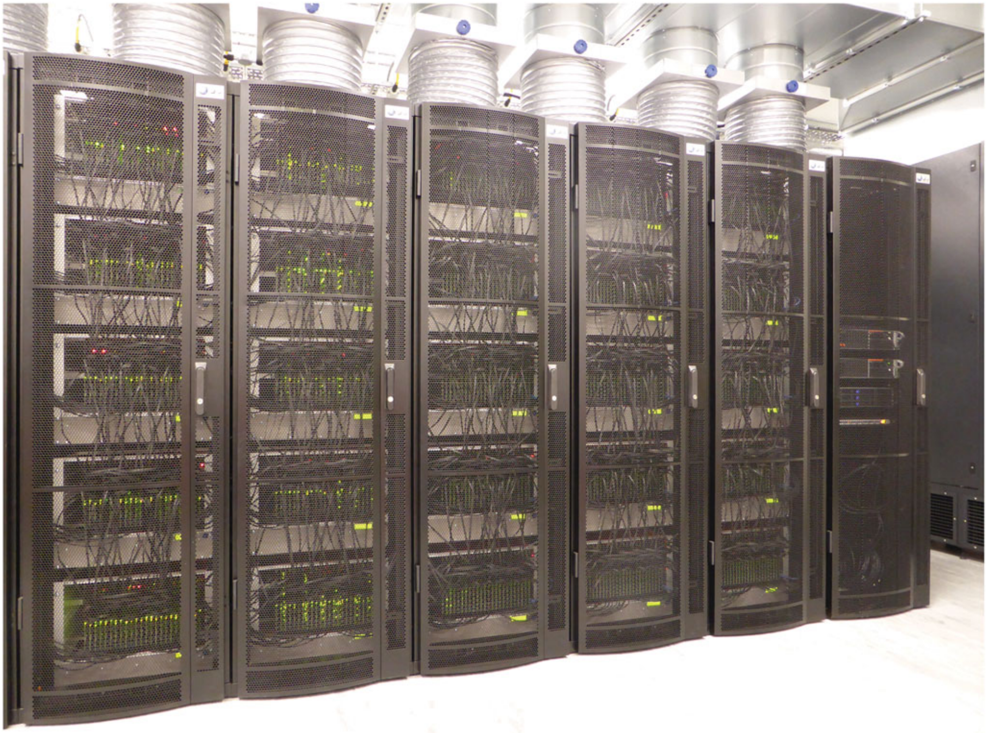

Figure 3. The SpiNNaker machine, delivered in 2016. The machine incorporates 28 800 SpiNNaker chips—half a million ARM processor cores in total—connected into a two-dimensional toroidal surface, contained in five 19″ rack cabinets. The sixth cabinet contains the associated server systems. The total active silicon area in the machine amounts to 5 m^2. (Online version in colour.)

The instruction set architecture supported by the ARM processors on SpiNNaker has seen many extensions and enhancements since that used on the first ARM processor, but it is still highly recognizable as a descendant of that original instruction set. In contrast, the silicon technology and the resulting performance, functional density and energy-efficiency have changed by many orders of magnitude over that quarter of a century.

10. Conclusion

The microprocessor has seen formidable growth in its influence on humanity since its humble origins in 1969 as a proposal to optimize the cost of building a range of desktop calculators. The inescapable cost, performance and efficiency benefits of integrating all of the central functions of a computer onto a single microchip have driven the entire computer business into the microprocessor's clutches, and in the process the market for products with embedded computational intelligence has diversified in many directions to the point where now there are already more than ten computers for every human on the planet—a number which will increase by another order of magnitude over the coming decade as IoT becomes a pervasive reality.

Since the spread of the PC through the 1980s, followed by the introduction of the worldwide web in the 1990s (making the Internet, which had already been around for a couple of decades, usable by the wider population) and concurrent developments in mobile (wireless) communications, computer and communications technology has become part of the infrastructure of human society everywhere. We have truly entered into a digital age, and the microprocessor is the engine of that digital age.

Competing interests. The author is a founder, director and shareholder of Cogniscience Ltd, a University of Manchester spin-out company that owns a SpiNNaker patent and related IP.

Funding. The design and construction of the SpiNNaker machine was supported by EPSRC (the UK Engineering and Physical Sciences Research Council) under grant nos. EP/D07908X/1 and EP/G015740/1, in collaboration with the universities of Southampton, Cambridge and Sheffield and with industry partners ARM Ltd, Silistix Ltd and Thales. Ongoing development of the software is supported by the EU ICT Flagship Human Brain Project (FP7-604102), in collaboration with many university and industry partners across the EU and beyond and exploration of the capabilities of the machine is supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement 320689.

Acknowledgements. The instruction set architecture of the original ARM microprocessor was developed principally by my colleague at that time, Sophie Wilson, with whom I share the Royal Society 2016 Mullard Award. The Acorn team included several folk who wrote software test programs, and a VLSI design team who implemented the silicon chip and last, but not least, support and encouragement from Acorn's senior management at that time, most notably Hermann Hauser. SpiNNaker has been 15 years in conception and 10 years in construction, and many folk in Manchester and in various collaborating groups around the world have contributed to get the project to its current state. I gratefully acknowledge all of these contributions.

Author profile

Steve Furber CBE FRS FREng is ICL Professor of Computer Engineering in the School of Computer Science at the University of Manchester, UK. After completing a BA in mathematics and a PhD in aerodynamics at the University of Cambridge, UK, he spent the 1980s at Acorn Computers, where he was a principal designer of the BBC Microcomputer and the ARM 32-bit RISC microprocessor. Over 85 billion variants of the ARM processor have since been manufactured, powering much of the world's mobile and embedded computing. He moved to the ICL Chair at Manchester in 1990 where he leads research into asynchronous and low-power systems and, more recently, neural systems engineering, where the SpiNNaker project is delivering a computer incorporating a million ARM processors optimized for brain modelling applications.

References

1. Moore GE. 1965 *Cramming more components onto integrated circuits*. In Electronics, 19 April. pp. 114–117.
2. Intel's museum archive, 'i4004 datasheet', http://www.intel.com/Assets/PDF/DataSheet/4004_datasheet.pdf
3. Patterson DA, Ditzel DR. 1980 The case for the reduced instruction set computer. *ACM SIGARCH Comput. Archit. News* **8**, 25–33. (doi:10.1145/641914.641917)
4. Radin G. 1982 *The 801 minicomputer*. In *Proc. 1st Int. Symp. on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS-I*), *Palo Alto, CA, 1–3 March*, pp. 39–47. New York, NY: ACM.
5. Hennessy JL, Jouppi N, Przybylski S, Rowen C, Gross T, Baskett F, Gill J. 1982 *MIPS: A Microprocessor Architecture*. In *MICRO-15: Proc. 15th Ann. Symp. on Microprogramming, Palo Alto, CA, 5–7 October*, pp. 17–22. Washington, DC: IEEE Computer Society Press.
6. Furber SB. 1989 *VLSI RISC architecture and organization*. New York, NY: Marcel Dekker.
7. Furber S. 2000 *ARM system-on-chip architecture*. Reading, MA: Addison Wesley.
8. Jagger D (ed). 1997 *Advanced RISC machines architecture reference manual, 1st edition*. Englewood Cliffs, NJ: Prentice Hall.
9. Barroso LA, Clidaras J, Hölzle U. 2013 The datacenter as a computer: an introductions to the design of warehouse-scale machines (2nd edition). *Syn. Lect. Comp. Archit.* **8**, 1–154. (doi:10.2200/S00516ED2V01Y201306CAC024)

10. Hinton GE, Osindero S, Teh Y-W. 2006 A fast learning algorithm for deep belief nets. *Neural Comput.* **18**, 1527–1554. (doi:10.1162/neco.2006.18.7.1527)
11. Jouppi N. 2016 Google supercharges machine learning tasks with TPU custom chip. See https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html
12. Furber S. 2016 Large-scale neuromorphic computing systems. *J. Neural Eng.* **13**, 1–14. (doi:10.1088/1741-2560/13/5/051001)
13. Pavlidis VF, Friedman EG. 2009 *Three-dimensional integrated circuit design*. San Francisco, CA: Morgan Kaufmann.
14. Atzori L, Iera A, Morabito G. 2010 The internet of things: a survey. *Comput. Networks* **54**, 2787–2805. (doi:10.1016/j.comnet.2010.05.010)
15. Esmaeilzadeh H, Blem E, St. Amant R, Sankaralingam K, Burger D. 2011 Dark Silicon and the End of Multicore Scaling. In *Proc. 38th Int. Symp. on Computer Architecture* (*ISCA'11*), *San Jose, CA, 4–8 June*, pp. 365–376.
16. Painkras E, Plana LA, Garside JD, Temple S, Galluppi F, Patterson C, Lester DR, Brown AD, Furber S. 2013 SpiNNaker: A 1 W 18-core System-on-Chip for Massively-Parallel Neural Network Simulation. In *IEEE J. of Solid-State Circuits*, pp. 1943–1953.
17. Furber SB, Galluppi F, Temple S, Plana LA. 2014 The SpiNNaker project. *Proc. IEEE* **102**, 652–665. (doi:10.1109/JPROC.2014.2304638)