The Birth of Arm Multicore Processing

John Goodacre, *The University of Manchester, M13 9PL, Manchester, U.K., and also Research and Innovation Agency, SN2 1FL, Swindon, U.K.*

It was 2002 and my second year at Arm having recently moved back to the United Kingdom from Redmond, WA, USA, having created Microsoft's first real-time data and video collaboration server, which shipped as part of Exchange 2000 Server[1]—guess we're all pleased this early work happened way back then—anyway—having taken the move to cancel the product for which I was the manager, the PrimeXsys platform, I along with five engineers based in the south of France started to look at how Arm might be able to extend beyond their purely mobile processor roadmap, and enable scalable performance along with reduced power consumption. Intel at the time was promising a 10-GHz Pentium 4, and Arm was working on the Arm 1136 and looking toward their first gigahertz-level central processing unit (CPU), so multicore really was not a given, especially for embedded systems in which no operating system (OS)/real time operating system (RTOS) or apps existed for such a device, simply because there had never been an Arm multicore processor before—which needed some effort[2] to make happen for sure!

I knew that we did not want to build the more traditional multicore solution in which each processor snooped the memory bus to keep their caches coherent, so we decided we would build a multicore processor as a single-processor IP core in which we created a new concept called the snoop control unit (SCU) and an integrated interrupt controller, later formalized as the general interrupt controller, which together, would tie each CPU into a cluster; clusters of up to four processors, and yes someone built a three-core multiprocessor—we chose four CPU clusters simply because we wanted to be able to serialize the writes required for coherence in one clock cycle, and for a larger cluster, we would need to add more cycles and end up taking a nasty performance hit on each CPU—but which processor core should we base our multicore investigations was not an easy choice. The ARM926 was very popular at the time, so we pulled that apart and added our multicore magic sauce around it and implemented it in a stack of field-programmable gate array (FPGA) boards—a true 3-D stacked multicore processor, who would have thought—we also modified an early 2.4 Linux kernel to run on it—6 MHz but it was surprisingly spritely, one CPU ended up running X11, another the application, and the others the drivers and background tasks—the concept worked—the architecture did not—we had kept the old cache policies of the 926, and these were causing all sorts of corner case problems as we worked to keep the caches coherent—so back to the drawing board.

By 2003, NEC of Japan became our multicore project's lead partner, as they needed to find a solution to delivery more performance for automotive[3] without being able to increase the megahertz due to fabrication process limitations needed for devices in such environments. NEC, as historians will know, had built multicore chips previously, so together we found a set of memory policies for our SCU and CPU that should work with the Arm architecture. For a short time, we looked at the next maturing CPU from the ARM10 family, but decided we would sync up with the ongoing ARM11 revisions and build the first Arm multicore product around that pipeline. I shared[4] some early thoughts about the potential solution at a conference in 2003, the only conference I attend every year and have done for 20 years, and tested the concept—with unknown to me, the press[5] in attendance too.

The ARM11 MPCore, as opposed to the MPSoC,[6] the name of that conference, was first released in early 2004 as a single IP core, capable of implementing with ease between one and four CPUs (for the price of a single CPU IP), in an SoC that could run independently, without cache coherence, or with some number of CPU as an SMP-coherent multicore. Getting the worlds OS/RTOS to become multicore capable was taking a lot of effort, so much of the initial use was simply as an easy way to build an SoC with multiple processors and it ended up in various devices running imaging pipelines—which as the pixel counts were exploding needed the extra umph and it was easy to split the imaging pipeline over each CPU. It did mean however that these devices during boot could turn into SMP multiprocessors, and this enabled the next revisions of the products to start to share tasks over all the

0272-1732 © 2021 IEEE
Digital Object Identifier 10.1109/MM.2021.3115613
Date of current version 19 November 2021.

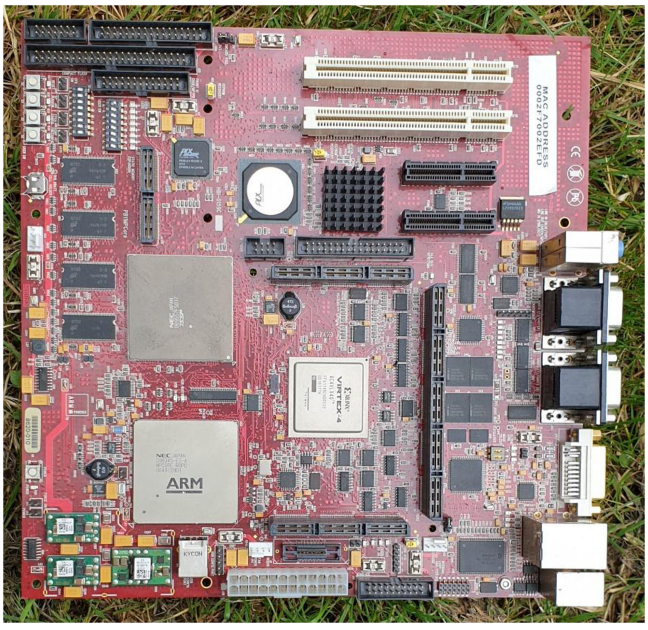


FIGURE 1. ARM11 MPCore first silicon prototype board.

CPU—and with task balancing and dynamic voltage and frequency scaling (DVFS) lowering the power consumption significantly compared to the equivalent number of cycles needed on a single-core processor—made selling the concept pretty easy—I would run a workload, typically playing a video on one CPU, dynamically turn on the other three CPUs and the power consumption would halve while still paying the video at the same quality and speed.

BY 2009, YES YOU GUESSED IT, AT THE ANNUAL MPSoC CONFERENCE, I PUBLICLY SHARED MY GROUP'S WORK FOR TARGETED EXECUTION ON WHAT BECAME KNOWN AS BIGLITTLE MULTIPROCESSING FROM ARM.

Showing[7] that quadcore processor could run the same workload as a single CPU at half the power, which also is able to boost up to four times the peak performance (at yes, four times the power!), was a real hit. It also meant that interrupts could always find some available cycles, something that was about to become important in another market—mobile.

Mobile chips however in the then processes' geometries could not afford that peak power consumption, so we knew dual-core processors would likely be the first devices—but we also knew we could significantly increase the performance and power efficiency of ARM11 MPCore—so started building and marketing the ARM Cortex-A9, released in October 2007; it adopted the other ARM11 technologies of Trustzone and Thumb2—although multicore was still not a concept accepted by all Arm partners, there was especially one very large OEM at the time along with their chip manufacturer that decided to stay with the ARM1176, aimed to boost its megahertz from 400 to 800 MHz, and keep phones running the same Symbian OS—I think history tells us that might not have been the best decision.

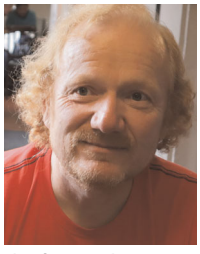
Another lead partner joined the Cortex-A9 project, Nvidia, with the desire to build a mobile chip that became to be known as Tegra. They also had experience of multicore processing and knew that if the power could be kept low for the typical workloads, then the peak "boost" performance significantly improved interrupt responses on which say a side-to-side scroll, or pinch and zoom of the screen occurred, and then, a multicore could be the answer to changing the form factor of what was becoming to be known as the smartphone. By 2008, the prototypes[8] were looking good, and then, the worst news any chip and handset manufacture could get—the OS vendor decided they would no longer be making their OS available, which left another company that controlled their own OS to release this new touch-smartphone concept first—with other vendors waiting on this new thing called Android—Symbian as I mentioned was no longer an option—what a different world it could have been. As Android was based on the Linux kernel, it could also benefit from multicore pretty much from day 1, and it did not take many revisions to start to

stand out—along with that other "leading" company's later move to multicore too—so that the screen could flow without choppiness—while keeping the battery living all day.

The Cortex-A9 opened so many doors to higher performance designs for Arm; there was even a company, Calxeda, that tried to build the first Arm server chip using this core. I however needed to investigate what came next while the billions of multicores filled the world's hunger for technology. In 2004, Kumar *et al.*[9] had published an article "Single-ISA Heterogeneous Multicore Architectures" that caught my attention, and I started to investigate various other approaches that could potentially allow a mix of CPU pipelines run the various tasks of different sizes that were becoming evident in the workloads we were seeing on the A9. By 2009, yes you guessed it, at the annual MPSoC conference, I publicly shared my group's work for Targeted Execution[10] on what became known as bigLITTLE multiprocessing from Arm. I also joined a collaboration, ENCORE, EU FP7-ICT, with the Barcelona Supercomputer Centre around that time to see if the Arm multicore could scale up to HPC—a resounding, err, no, but we did learn it could, and as those that follow the press, Arm multicore is now sitting pretty at the top of the TOP500 of the biggest HPC clusters in the world while my own research agenda[11] at The University of Manchester further investigates beyond multicore on how to take processor and system technology to the next level of performance and power efficiency.

REFERENCES

1. "Microsoft Product Announcement," *Conference Service Enables "Meetings Without Walls,"* 22nd ed. Redmond, WA, USA: New Microsoft Exchange, Feb. 2000. Accessed: Oct. 15, 2021. [Online]. Available: https://news.microsoft.com/2000/02/22/new-microsoft-exchange-2000-conferencing-server-enables-meetings-without-walls/
2. J. Goodacre and A. N. Sloss, "Parallelism and the ARM instruction set architecture," *Computer*, vol. 38, no. 7, pp. 42–50, Jul. 2005, doi: 10.1109/MC.2005.239.
3. M. Yoshida, T. Sugihara, M. Takahashi, Y. Koumoto, and T. Ishihara, ""NaviEngine 1," system LSI for SMP-based car navigation systems," *NEC Tech. J.*, vol. 2, pp. 35–39, 2007. [Online]. Available: https://www.nec.com/en/global/techrep/journal/g07/n04/pdf/070409.pdf
4. *International Seminar on Application-Specific Multi-Processor SoC*, 3rd ed., Hotel Alpina, Chamonix, France, Jul. 7–11, 2003. Accessed: Oct. 15, 2021. [Online]. Available: http://mpsoc-forum.org/archive/2003/lectures.html.goodacre
5. *News article, Semiconductor Business News, Design & Reuse*, Peter Clarke, Jul. 12, 2003. Accessed: Oct. 15, 2021. [Online]. Available: https://www.design-reuse.com/news/5890/arm-four-processor-core.html
6. *International Forum on Multiprocessing System on Chip (MPSoC)*, 2001. Accessed: Oct. 15, 2021. [Online]. Available: http://mpsoc-forum.org/
7. K. Hirata and J. Goodacre, "ARM MPCore; the streamlined and scalable ARM11 processor core," in *Proc. Asia South Pacific Des. Autom. Conf.*, 2007, pp. 747–748, doi: 10.1109/ASPDAC.2007.358077.
8. "Opera Press Release," *NVIDIA and Opera team to accelerate the full Web on mobile devices*, Santa Clara, CA, USA, Sep. 9, 2008. Accessed: Oct. 15, 2021. [Online]. Available: https://press.opera.com/2008/09/09/nvidia-and-opera-team-to-accelerate-the-full-web-on-mobile-devices/
9. R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, "Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction," in *Proc. 36th Int. Symp. Microarchit.*, 2003, pp. 81–92. [Online]. Available: https://www.microarch.org/micro36/html/pdf/kumar-SingleISAHeterogen.pdf
10. "Targeted execution enabling increase power efficiency," in *Proc. 9th Int. Forum Embedded MPSoC Multicore*, J. Goodacre, A Lahiri, A. Shrotriya, and N. Zea, Eds., Savannah, Georgia, USA, Aug. 2–7, 2009. Accessed: Oct. 15, 2021. [Online]. Available: http://mpsoc-forum.org/archive/2009/slides/Goodacre.pdf
11. The University of Manchester Researcher portal. [Online]. Available: https://www.research.manchester.ac.uk/portal/john.goodacre.html



JOHN GOODACRE is a Professor of computer architectures with the Department of Computer Science, The University of Manchester, Manchester, U.K., having previously spent 17 years with Arm Ltd. as the director of technology and systems where he defined and introduced the first multicore processors and other core technologies. He is also appointed by the U.K. government's Research & Innovation agency as the Director of the Digital Security by Design Challenge Fund, a £200 million programme to enable industry and researchers to create a step change in approach to cybersecurity, blocking vulnerabilities by design, and protecting the operation and data by default. His research interests include web-scale servers, exascale efficient systems, and secure and ubiquitous computing. Goodacre received a bachelor degree in computer science from the Univeristy of New York, NY, USA, in 1987. Contact him at john.goodacre@manchester.ac.uk.