Many cores save power

The evolution of the microprocessor; from single cpus to many core devices. By John Goodacre.

Since the days of the mainframe and the desktop pc, processor technology has changed continually to address users’ needs. Now, with mobile computing becoming a dominant market, processor innovation has become a critical driving force for change.

The dominant processor architecture has evolved from being a ‘mains powered performance pusher’ to one that stretches performance within a limited power budget. Processor evolution will rely on energy efficiency as a key component as the industry works to gain ever more performance from a limited power budget.

Once, it was about performance

The first consumer device enabled by the microprocessor was the pc, in which a single threaded cpu ran a single application. Computing devices were stationary and mains powered, which meant mpus could be designed with one thing in mind; performance. And increasing performance was the goal for all developers.

While many early microprocessors were 8bit, the industry had moved on to 16bit and 32bit processing by the mid 1980s, which helped pcs to run multiple applications at once.

Performance increased as the number of transistors on the microprocessor doubled every other year in line with Moore’s Law. Each processor evolution heralded a new computing experience that fuelled consumer and business desires for ever more powerful pcs. It wasn’t until early 2000s that energy efficiency and performance became a focus. Demand for mobile phones was also growing at this time and, as it grew, so too did the need for energy efficient processing.

Although still using single core processing, the industry began to understand the value of energy efficient handheld devices. But, in line with the pc desktop market, users expected certain levels of performance.

ARM was established with the goal of creating an energy efficient processor for use in early handheld computing devices. These products used the reduced instruction set computing (risc) cpu architecture as part of a design strategy. This approach simplified instructions, made execution simpler and reduced the amount of power expended per instruction, making the microprocessor more energy efficient.

The mobile market is not the only area of electronics to benefit from high performance, energy efficient processing. The technology is driving development in many sectors, from smartphones and tablets and digital tvs to printers.

Processors in today’s smart devices must have a number of key features:

- Lowest possible power consumption, to increase battery life and reduce battery weight, while reducing the cost associated with managing high power devices
- Heat dissipation can become an issue and must be addressed for ultra thin, high performance devices

Desktop class user experience and compelling graphics are now essential
The software development process has to ensure the shortest time between processor development and the delivery of product to market.

Multicore reality

Since mid 2000s, it has been accepted that you can’t just build bigger and bigger single cpus for more single thread performance. This is not only hard to do but, more importantly, is not energy efficient. You have to bear in mind that, for every few percent that you increase performance, you incur an even larger energy requirement.

Driven by business and consumer...
In the many core approach, the processing load is shared across ‘many’ smaller processors, rather than multitasking multiple single thread workloads across a single or multicore processor.

Multicore solutions can deliver higher performance at comparable frequencies to single processor designs, while offering significant cost and power savings. It is clear that multicore is now the way to scale beyond a single core and this is especially true as multitasking usage scenarios emerge that allow multiple applications to stack their processing requirements across the cpu.

Multicore solutions can leverage cores with high transistor counts, but optimise the system by not powering them up at the same time; instead, they work on a per specific task basis to save energy. Think of it as intelligent load balancing; not only does it need to look at which processor is best suited to execute a given task, it must also look at the performance requirement of the task and so assign it to the most power efficient processor available.

From an energy efficiency standpoint, load balancing is perhaps the strongest feature of multicore design, enabling specific tasks to be prioritised and specific cores to be used for differentiating tasks. This keeps energy consumption as low as possible without affecting performance, with cores being used or standing idle, depending on the task. When tasks can be balanced across multiple processor cores, each processor may be less than fully used. This allows the voltage and frequency of a multicore processor to be lowered and, as such, to save an exponential amount of power related to the delivered aggregate performance.

For example, today’s smartphones still need to make calls, receive texts and be touchscreen activated. These functions demand small amounts of power compared to video playback, recording or game play, which may also be required in parallel with basic phone and data functionality. It’s an example of why today’s multicore smartphones can deliver increased battery life compared to single core devices executing the same tasks.

This demand for both background and high levels of performance is forcing a significant industry shift; most current smartphones contain multicore cpus and, increasingly, gpus with multiple cores. Higher performance and higher quality graphics are increasingly a ‘must have’.

Many core processing

The many core processing approach requires the processing load to be shared across ‘many’ smaller processors, such as a Cortex-M0, rather than the multitasking of multiple single thread workloads across a single or multicore processor. In either case, we are starting to see clusters of processors working together, sharing data and tasks between caches or groups of the same processor. Many core is becoming interesting.
because smaller processors can deliver a given aggregate performance for much less power than a comparable larger processor multitasking that same workload. As a rule of thumb, based on the delivery of a required single thread performance, there is an exponential cost relative to the increase in performance. However, for multicore processing there is a linear cost associated with that increase in performance. This means that, if you are capable of using ‘many’ cores, the aggregate cost for this performance will be significantly less than either a single core or a ‘few’ multicore processors.

But software is not yet capable of using a many core processor and, as such, devices will need to maintain their ability to execute high performance single tasks.

It is useful to look at current use of cpus and gpus. Many core gpus can deliver significant graphical computation using far less power than multicore cpus. Since the gpu is coherent with the cpu and can look into its caches, external memory bandwidth is reduced, as is cpu load. This move is being recognised with languages such as OpenCL and CUDA.

The future: optimisation
Rather than relying on putting more transistors in processors to get pure performance, the industry now needs to make the most of the limited resources and footprint. It can do this by using both domain specific processors and heterogeneous general purpose computing. It can also achieve this by optimising designs and the design process across multiple core SoCs.

Of course, multicore processing is high on the agenda for many – and rightly so. Optimisation is important; while it makes fewer headlines, it must be considered, especially in smaller footprint applications where there are greater challenges around coherency. Cache coherency is essential in multicore computing applications so the consistency of data stored in local caches of a shared resource can be maintained. Standards and specifications, such as the AMBA 4 bus, must be considered and developed to support system level cache coherency across clusters of multicore processors. This will ensure optimum performance and power efficiency in complex heterogeneous SoCs.

Similarly, the world of debug and trace does not make headlines, but this is another fundamental area from an optimisation point of view. To achieve optimal performance, software developers demand increased visibility within SoC designs. This can be addressed by providing powerful, modular debug and trace infrastructure and tool chain, such as ARM CoreSight SoC. These tools also address the need for greater productivity for multicore SoC designers and reduced time to market.

Conclusion
The future requires more targeted processing, optimisation and differentiation in the design process for systems that contain different types of processor, while supporting the requirements of the software ecosystem. Energy efficiency is the key differentiator in the world of processing and will be the key driver for the future of computing. Anyone who ignores this evolution will simply get left behind.

Author profile:
John Goodacre is director of programme management in ARM’s Processor Division.