

XMOS says programmability has no penalty

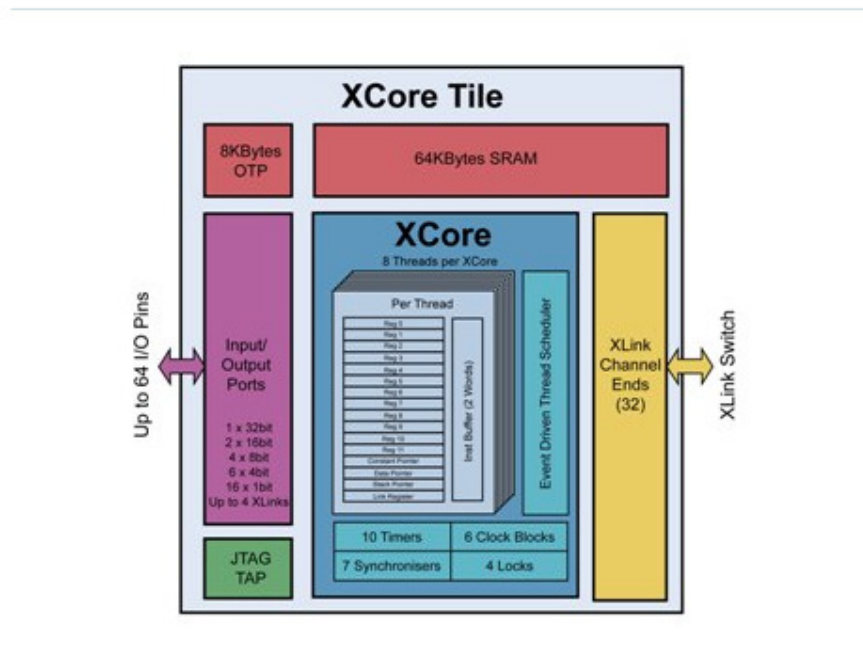
Customising silicon is the proven way to add differentiation to electronic products. When manufacturing volumes are very high, it's easy to see the economic logic of ASICs.

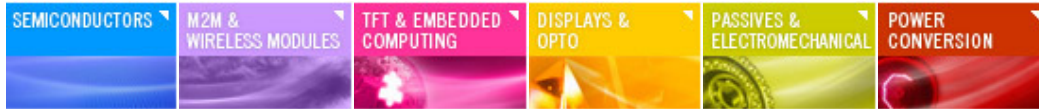
However, for many products demand is either more limited or less predictable, so the ASIC route is fraught with risk and programmable silicon is the only way forward. Programmable hardware comes in the form of programmable logic or microprocessors.

Programmable logic is really flexible but it's expensive because it's built on a gate-based fabric with inefficient memory architectures. However, the FPGA has given a huge boost to ease and speed of product development and opened up new applications to custom silicon.

Microprocessors function in stark contrast to FPGAs and work on the basis of reusing the silicon resources on every clock cycle. If you need all the gates, all the time, the FPGA make sense. However most applications need some of the gates, some of the time, so it's back to the processor.

The limitation of conventional microcontrollers and CPUs is that they either provide a lot of computing power or they're very responsive to outside stimuli. Most applications need both, not one or the other.





An approach that's now bridging the gap, and indeed replacing both FPGAs and traditional microprocessors in many applications, is the multi-tasking event-driven microprocessor. The XCore processor shown in Figure 1 runs up to 8 separate tasks (threads) in round-robin fashion. At a maximum CPU speed of 400MHz these tasks appear virtually concurrent.

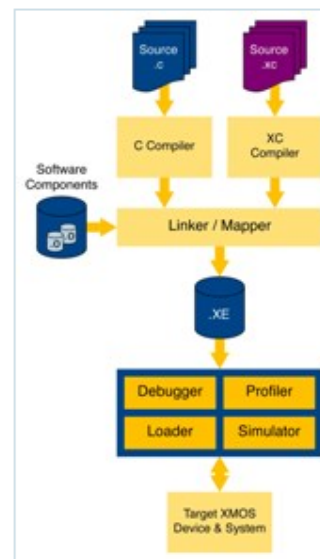
The use of intelligent I/Os means that the processor is not disturbed until all the data that will be demanded by the associated process is collected. The I/Os are timed to within 10ns, and present data to the scheduler in a single clock cycle. This event-driven system eliminates the timing limitations of polling/interrupt systems.

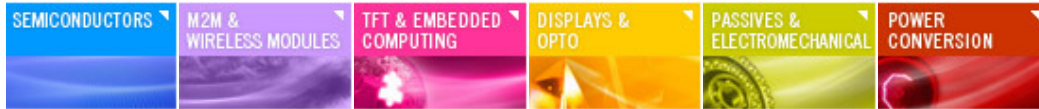
Three important differences characterise this event-driven processor technology. First, by using a high-level software-based design flow shown in Figure 2 and programmability in C, the silicon is accessible and familiar to programmers and design engineers.

Second, the custom electronics use an instruction set based on free reuse of all of the silicon resources, so designers do not encounter the unexpected changes that can occur in a synthesis – fitting - place & route design flow. Third, the hardware kernel provides the services normally associated with an RTOS, but without the memory and complexity penalties. The kernel includes a scheduler, timers, locks and synchronisers to keep everything properly ordered.

As an example, XMOS has implemented an Ethernet AV reference design entirely in the C language. This includes the low-level control of the time-stamp clocks integrated into every I/O port, IEEE-1588v2 time keeping, the MAC and audio packet extraction layers, and DSP functions to add high fidelity features such as filtering.

This takes only two XCore processor tiles with room left for custom features. This same function could be implemented in high-performance processors or high-end FPGAs, but neither would be cost-effective for consumer applications where Ethernet AV will have the highest impact.





Furthermore, the complexity of the design flows would almost certainly require low-level hand crafting to ensure the precision and control required when putting synchronous data over this inherently asynchronous medium.

The same approach will enable new semiconductor businesses to be created quickly and at relatively low cost. Programmers and engineers with a strong product idea and competent programming skills will be able create new devices that are competitive on a global basis.

Most importantly, they will be able to do this on budgets of tens or hundreds of thousands of pounds, rather than the millions or tens of millions with which the industry, and its investors, are familiar.

It has become too slow, expensive and risky to take advantage of the capabilities offered by traditional ASIC and FPGA options. The simpler approach described above leverages the best characteristics of existing programmable technologies through a new architecture and a familiar design flow. It gives designers the freedom to be creative where it makes sense, but use accepted and understood technologies and methods where they are proven best.