



Right here, right now

The growing demand for complex architectures is making it increasingly more difficult for hardware and software developers to keep up with new chip innovations. These designs are driving the requirement for a more creative debugging and analysis philosophy to accelerate the hardware development process and to offer a more collaborative synergy between hardware and software developers.

Not only are devices becoming more complex, but the product time to market is becoming shorter too. System on chip (SoC) designs appeared on the market out of the blue and immediately created an issue for debug specialists. Now the latest innovation in the shape of multi-core has arrived and the industry is asking itself how can it debug such a system? The combination of these two issues is beginning to create an expectation in the industry as companies strive to solve these problems: namely, how do you debug such complex systems and still get the product to market in a reasonable time?

Esterel Technologies' technical director, Central Europe, Jakob Gärtner confirms that today's developers do indeed face the challenges of short development cycles, increasing hardware and software complexity and bad software quality – or 'let the tester fix all problems'. "We ensure clear separation of tasks and a high degree of automation in order to reduce the amount of necessary testing through a 'correct by construc-

tion' approach."

Today's customers want it all – right now. But increased innovation escalates the complexity of the development process.

By **Mike Richardson**.

tion' approach."

Esterel takes a 'model based' approach to design and verification so debugging is clearly separated into two distinct domains: functional, which ensures the functionality of the system and software are correct with respect to the requirements – a step ideally done at the model level; and debugging in the system integration phase, which ensures the functionally correct parts of the system are fitting together on the embedded system – a step ideally done with a tool suite that integrates traditional source code level debugging and model level debugging.

The time to market issue is probably most relevant to the faster paced consumer electronics industry and there are many multicore projects currently being developed specifically for this industry where the timescales are very tight. This also includes the telecoms sector, where multiple devices with dozens of cores on a single device are a regular occurrence. How do you debug them and still get the product to market in a reasonable time?

"When you develop multicore and multi-processor systems, there are a number of factors that you need to consider,"

Wind River's system architect, Paul Tingey explained. "Firstly, how do you communicate between your development environment and to all these different devices? When you debug systems of this kind, you need tools that can debug multiple cores. Wind





"Wind River's Workbench solution has features built into the product with the capability to debug 'multiple anything!'"

Paul Tingey, **Wind River**

River offers its Workbench solution and one of the features built into this product is the capability to debug 'multiple anything!' You can connect multiple cores, multiple processors and multiple operating systems (OS) transparently and debug these complex systems in a much simpler way than previously."

Tingey's view is the ability to connect to multiple devices and debug them at the same time really is key to developing large, complex systems. Wind River's ICE performs run control across multiple cores to effectively synchronise the debugging of multiple cores. Imagine a system where two cores are communicating with each other and are time dependent. If you debug one of these cores, you stop half of the system, while the other half still runs – not an ideal scenario.

However, with advanced run control, you can synchronise all these events and debug the entire system in a synchronous way.

"There are many systems that our customers are developing – not just multiple cores, but running them asynchronously and with different OSs in different cores," Tingey continued. "This has been a problem in the past because you have two different OSs with two different tool suites. But with Workbench, you can use a single integrated development environment (IDE) to develop on either Linux based target or VxWorks based target, or both.

"Another benefit is that, by using a common IDE, two different sets of developers with a distinct knowledge of different targets can transfer and share this information easily between each other."

Gärtner identifies a trend towards model based design and verification approaches and, more specifically, to formal methods. "There is a trend to replace test based debugging with formal verification technology such as Esterel's Design Verifier. This offers mathematical proof of absence of bugs on functional level. Further developments will allow customers to formally analyse system models for component failure effects, list all critical components (single points of system failure) and derive fault trees automatically."

Tingey sees the problems associated with multicore as the most important trend driving debugging technology. "We also have customers who are interested in profiling and analysing systems at the earliest possible stage. Traditionally, there was a quality assurance stage that came some time after the development stage, but now companies have

become more astute, put the two together and as the software is developed, it is also validated at the same time. This requires some highly specialised tools."

His view is that the market is becoming more aware of multicore and its associated problems. Consequently, development tools will evolve with the emphasis on multicore being the norm.

"Virtualisation is another technology to consider," he added. "Customers are starting to look at the ways of creating a virtualisation layer on top of an OS, so that you can 'virtualise' an environment for another OS. This is going to become increasingly popular and could pose questions as far as debugging is concerned.

"Virtualisation technology creates an environment where the OS believes it is running on a piece of hardware, but it is actually running on a 'virtualised' or simulated piece of hardware. There are many ways of using this technology and what we may see in the future are systems incorporating both multicore and virtualisation at the same time."

Meanwhile Esterel's Gärtner envisages that model based development, automation and certification needs will forever change the life of the debugging specialist.

"Full integration, with state of the art software IDEs and debugging environments, will ensure rapid turnaround times and further reduce debugging workload.

In addition, integration with worst case execution time analysis tools, will help to avoid runtime issues long before the user goes embedded.

"Standardised platforms and compositional methodologies will focus their work on functional model level debugging on one hand and embedded integration testing on the other.

But the general idea is to avoid bugs before they arrive on the target," Gärtner concluded. ☺

