



How going back in time can help engineers to debug software.

By **Graham Pitcher**.

“When we talk with engineers, we find that debugging is the largest part of what they do. Almost every time, they say it took a long time because they didn't have the right tools.” So says Christopher Smith, vp marketing for Green Hills Software.

Debugging can, of course, be one of the most demanding tasks for an engineer. The software isn't working as expected because there's a bug. But where is it?

Some years ago, the general way to attack this problem was to use an in-circuit emulator, or ICE, which plugged into where the microprocessor would sit on a board. “It was somewhat intrusive,” Smith noted, “but good enough when processors were slow enough.” ICEs, in general, are yesterday's technology; relegated to the shelf because increased processor speeds meant data couldn't be passed quickly enough over their ribbon cables.

Today, Green Hills is taking a page out of science fiction by developing a Time Machine. This technology allows engineers to step backwards through their programs to find the causes of bugs.

Mike Lindahl is director of engineering for Time Machine with Green Hills. He said: “Time Machine is based on trace data; it gives a history of what a microprocessor is doing and is very much like an audit trail.”

Trace data is the information generated by a microprocessor equipped with a special debug port. ARM was the pioneer of this approach with its embedded trace macrocell. Since then, other manufacturers have equipped their parts with a version of this facility. But if your processor doesn't have trace, there are other ways of accessing this information, albeit intrusive to some degree.

Lindahl gave an example. “If your system is running and suddenly has a memory problem, rather than starting again,



Turning back the clock

you have trace data that's loaded to a file. You can analyse this data to try to find the cause of the problem.”

Time Machine is built on top of this trace data analysis capability. “It allows you to read the data, do analysis and possibly reconstruct what was going on in

the registers or the memory,” Lindahl continued. “At some instruction in history, instead of querying the target, you read from the trace log. You can do single steps forward, going to the next instruction, or set a breakpoint. The tools then go through instruction by

instruction.” Time Machine also lets you replay the trace log so you can step backwards – go to previous, for example.

As well as making it easier to debug software, Time Machine also helps to optimise code. “It's good for performance analysis,” Lindahl claimed. “Path

Analyser gives software engineers an intuitive view of what the system is doing. Often, software engineers will hold a ‘map’ of the system in their heads, but this gives them an insight into system performance,” said Lindahl.

He added that Green Hills has used this

internally and found it of value in making its products run faster. “We've doubled the speed of some systems,” Lindahl noted, “and that's with code we understand and has been there for some time. But it isn't until you get this kind of view that performance problems become obvious.”

He believes the Time Machine approach gives engineers an opportunity to understand software and the relationships between its various components. “You can see what's calling what and how long the processes take,” he noted.

Smith added. “Even if you can get to the code, what a team hasn't been able to do is see how it executes in real time; what its performance really is. Often, by looking at things in this way, you can see the interactions with parts of the system that were unexpected. They can see that code can be written in a different way and speeded.”

Smith said the clever part of Time Machine is the reconstruction and the speed with which you can move through the process. “Before,” he claimed, “it was like looking for a needle in a haystack.”

If you don't have a trace equipped processor, then don't despair. TraceEdge provides ways of collecting Time Machine data. According to Green Hills, this requires as little as one instruction at each instrumentation point in the program.

There are three TraceEdge approaches. TraceEdge-BUS connects a SuperTrace probe directly to the processor's address bus, monitoring a dedicated memory region. Meanwhile, the TraceEdge-PCI card captures PCI and PMC writes, storing in a dedicated TraceEdge memory. This is then uploaded to a host pc for further analysis. Similarly, the TraceEdge-PMC card plugs into a spare PMC site and links to the SuperTrace probe, capturing all writes to the PCI bus. Again, information is uploaded to Time Machine running on a host pc.

The SuperTrace probe can capture up to 1Gbyte of trace data, enabling millions of instructions to be analysed within the Time Machine environment.

“Taking a modest performance hit in order to gain visibility is often something an engineer will do,” Smith concluded. “The alternative today is pretty grim.”