The Ansible Machine Prototype Substrate

Andrew Huang

The Problem: Architectural concepts for large computer systems are difficult to validate. Building a dozen variations of a large computer is not an option, but simulations of a large computer system lacks the performance and accuracy necessary to collect a robust set of interesting results. This, combined with the steady performance improvement that desktop processors have experienced for the past decade, has stymied computer architecture research.

Motivation: The Holy Grail of computer simulation platforms would be a system that comes within a factor of 10 to 100 of custom hardware target performance, with perfect scaling of all delays. It would also feature quick and painless reconfigurability between simulations, and good visibility into the design for profiling and debugging.

Previous Work: FPGA-based hardware simulation systems that contain some mix of SSRAM and host connectivity are ubiquitous today. Annapolis Microsystems, Aptix, Virtual Computer Corporation, IKOS, and Microtech, to name a few, all provide viable, high performance simulation solutions. However, none of them provide memory systems with bandwidths approaching those achievable in an ASIC, and thus they remain unsuitable for accurately simulating high performance processor architectures in a manner with sufficient performance to do native code development and performance benchmarking. In addition, almost all of them integrate a relatively low-performance, long-latency network interfaces, such as SCSI or PCI, for host and peer-to-peer interactions, also limiting their usefulness in simulating a large, multi-node system. A concurrent AI Lab project, the Moore Board [1], implements a high-bandwidth to memory simulator with good simulation host connectivity, but it omits adequate peer-to-peer connectivity resources.

Software-only simulations have proven to be quite effective for validating uniprocessor architectural concepts. The abundance of high-performance cross-platform software emulators is a testament to the ability of one microprocessor to emulate another. However, simulations of a multi-processor architecture on a uniprocessor machine should experience a slowdown that is at least proportional to the number of nodes being simulated. This slowdown approaches the point where a software simulation loses its appeal over a hardware-assisted simulation system.

Approach: Advances in FPGA technology have helped close the gap between RTL- and HDL- based simulation and final silicon performance levels. The density of FPGAs has also become sufficient to implement a reasonably-sized microarchitecture on a single device. However, a simulation methodology based entirely on RTL-coded modules often requires too much detail for an architectural simulation. A more top-down approach is often desirable. As a compromise between the high-performance, high engineering effort FPGA-implemented simulations and the lower-performance, quick and dirty software-based simulations, the Ansible machine prototype substrate (figure 1) incorporates both a large FPGA and a tightly coupled embedded strongARM processor subsystem. The FPGA has generous connectivity to other Ansible boards via a bank of twenty-four 16-bit wide links that feature CTT (center-tap-terminated) current-mode signaling and source-synchronous clocks. These links can be utilized to connect to a memory board if less connectivity and more memory is required. The strongARM's memory bus is connected to the FPGA. This connection enables very fast synchronous operation with the FPGA. The strongARM also has low-latency interrupt resources in case asynchronous operation is required. Finally, a smart debug chain interface is provided so that large systems can be tested and debugged through a single common interface.

The Ansible platform is a good research tool because it gives users more flexibility in partitioning their designs between RTL and high-level software simulations. Designs can start life as mostly software simulations and gain performance and cycle-accuracy with time as functions are incrementally implemented in hardware.

Impact: Flexible prototyping substrates such as the Ansible will hopefully be an enabling technology for computer architecture research because of their emphasis on ease of use, debugability, and reconfigurability. They are particularly useful when researching highly parallel architectures, because multi-

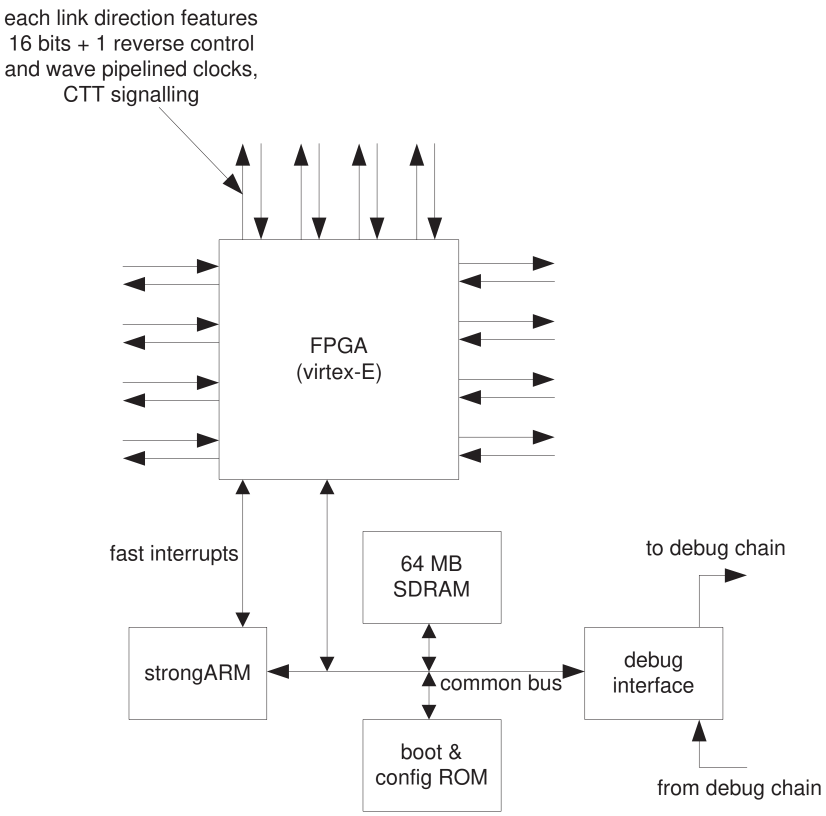

Figure 1: Ansible block diagram.

threaded software simulators running on commercially available off the shelf SMP clusters run out of steam as the simulations start to scale toward interesting (million-node) sized machines. They are also important as a validation tool for the final ASIC production itself, as the cost of mask sets pushes into the million-dollar range.

Future Work: The scope of the Ansible project is not limited to a rapid hardware prototyping substrate. It is part of a broader design methodology embraced by the Aries Decentralized Abstract Machine (ADAM), where an abstraction layer is defined so that both software and hardware development efforts can proceed in parallel. Thus, the Ansible platform is by design capable of implementing behaviorally-equivalent ADAM nodes. If someday an effort is started to re-implement the designs in an ASIC, much of the Ansible RTL can be re-used, and most importantly, the change would be transparent to applications targeting the ADAM platform.

Research Support: Support for this research was provided by the Air Force Research Laboratory, agreement number F30602-98-1-0172, "Active Database Technology".

References:

[1] Andrew Huang. Processor-in-memory systems simulator. Technical report, MIT AI Lab, AI Lab Abstract Book, 2000.