Custom FPGA-Based Emulators Accelerate IC Design Verification

Ray Turner - June 19, 2000

San Jose A new generation of emulators that incorporate custom FPGAs optimized for emulation is beginning to have a dramatic impact on IC design and verification speed.

Custom FPGAs enable new architectures that are highly scalable for increased capacity, incorporate special debugging logic, and provide for higher emulation and compilation speed than off-the-shelf FPGA-based emulators.

The regular, hierarchical structure that can be created in a custom FPGA means you can compile designs on a single workstation five to 20 times faster than traditional emulators. The new emulators also run faster because building special silicon makes it possible to run emulation several times faster with 100 percent internal visibility.

The large amount of time required with conventional emulators built on generic FPGAs to recompile the probes is eliminated by a logic analyzer-switching matrix built into custom FPGAs. These improvements can more than double debugging productivity, making it possible to fix two or three bugs per day compared to one per day today.

The astonishing power of today’s ICs has enabled a tremendous range of products and made those products more powerful, less expensive and easier to use. But the same dramatic increase in power and complexity has also turned the process of designing, verifying, and ultimately, taping-out these chips to silicon into a task of almost unbelievable complexity.

The technology of in-circuit emulation addresses these issues by allowing designers to quickly create a hardware model of a chip design using proprietary emulation software that maps the design onto reprogrammable circuitry. In-circuit emulators simulate the operation of a complex integrated circuit by partitioning the circuit into blocks and compiling the blocks onto hundreds of FPGAs that are linked together to emulate its operation. Up to now, in-circuit emulators have used off-the-shelf commercial FPGAs as building blocks. The obvious reasons are that commercial FPGAs offer a high price-to-performance ratio because their development costs are shared among many other
customers for these devices.

But, in recent years, the commercial FPGA market has sharply diverged from the needs of emulation. The I/O-to-gate ratio of commercial FPGAs is dropping, which is bad for in-circuit emulation because when a large design is mapped into an emulator for verification, the limiting factor on capacity becomes the number of I/O pins to interconnect the devices.

An additional problem is that implementing the debugging capabilities required in emulators into a conventional FPGA requires the construction of scan chains built from flip-flops that clock the data out of the chip into the logic analyzer. This feature consumes quite a bit of logic in a commercial FPGA reducing the capacity of traditional emulators and driving up their cost.

The most advanced emulators built on conventional FPGAs do offer a mode that provides full internal visibility, probing every signal in the circuit, but it consumes a considerable amount of the logic, typically 30 percent. Another problem is that the overhead created by providing 100 percent internal visibility reduces the speed at which signals can be sampled to around 500KHz. Another problem is that the time involved in compiling a design for in-circuit emulators based on commercial FPGAs may be too high to meet fast-track design schedules.

Emulators based on custom FPGA technology can overcome all of these problems. With custom FPGAs, for example, it becomes possible to create a regular, consistent, crossbar-connected architecture. Level-zero blocks are connected with a crossbar matrix to form a level-one block. The structure is replicated throughout the FPGA and level-one blocks are crossbar interconnected to form a uniform structure of level-two blocks. Likewise, the level-two structure is replicated throughout the device and connected with a crossbar matrix. This new architecture means that there is a fixed delay within any logic level, between any blocks at the same level and between blocks of different levels. This makes it easy to compute the delays between any two logic elements in the device. Little computational power needs to be devoted to performance optimization because the equal length of the crossbars means that it makes little difference where blocks are located.

The net result is that compilation time can be reduced by an order of magnitude or more. In the example mentioned above, the time required to compile a 1.1 million gate networking design is reduced from five hours on five workstations to 1.1 hours on three workstations, with the result that a farm of workstations or PCs is no longer required. As a general rule, million-gate designs can be compiled in less than an hour. Engineers compiling large designs on only one or two workstations can usually compile designs from five to 20 times faster than with emulators based on generic FPGAs. In many situations, this makes it possible for the compile to be performed at the exact time when it is needed, rather than having to wait to run it overnight, and even allows multiple compiles to be run in a single day.

Another critical improvement is that custom FPGAs developed for emulation have a very high pin-count relative to the number of gates. The new design uses a hardware-based two-to-one I/O multiplexer ring around a sea of configurable logic blocks that doubles the effective pin-count more than tripling the number of usable gates. These features increase the effective capacity of a system built with custom FPGAs to 20 million ASIC gates.

The use of custom FPGAs also makes it possible to build a hardware-based logic analyzer, a feature that consumes precious resources of off-the-shelf FPGAs. It consists of a matrix of switches that makes it possible to record any signals in the circuit being emulated including I/O. At any point in time, emulators built with custom FPGAs can determine the current state of every signal in the entire design being tested. With the logic analyzer built into the FPGA, designers can move probes instantly, avoiding lengthy delays for compiling probes. The switch matrix can reassign signals to
pins at run time, eliminating what is usually the most time-consuming step of the debugging process.

When you perform emulation, you have to slow down the real world to the speed of the emulator. This is not normally a problem but there is a limit beyond which anomalies begin to occur that make it impossible to accurately emulate a device. The use of custom FPGA's makes it possible to increase the speed of the logic analyzer to 2MHz while still providing 100 percent visibility. This level of speed provides an accurate emulation of virtually any device, eliminating a limitation of generic FPGAs and making it practical to provide 100 percent internal visibility on every emulation. The result is a substantial improvement in debugging productivity.

The use of custom FPGAs also makes it possible to build in the capability to set, force and release any storage element such as a flip-flop or register without compiling. The ability to perform these functions without this delay can significantly increase the speed of the debugging process. During debugging, designers frequently have a hunch of what is causing a problem and need a way to confirm it. The best way is usually to force signal values to change in the same way the proposed fix would. This is much faster than actually changing the design to model a proposed solution to the problem. Instead of adding a gate to turn off a signal at a certain time, you simply run the emulator to that point and force the signal to zero to see whether that really fixes the problem. The fact that custom FPGA-based in-circuit emulators can perform this task in a few minutes rather than a half hour significantly increases the speed of the debugging process.

The inclusion of a complete logic analyzer with instant probe changes, event detectors and pin multiplexing means that designers can move probes instantly, avoiding lengthy delays for recompiling probes. The usual limitations of providing 100 percent internal visibility in terms of sampling frequency, depth and capacity are eliminated. Several other improvements are also simultaneously reaching emulators that are not directly related to the use of custom FPGAs.

Ray Turner is the product line manager for In-Circuit Emulation at Quickturn, a Cadence company.