Verification tools - A brief idea

Balaji K
Sasken Communication Technologies Ltd.,
Bangalore

1) Simulators:

- Event driven simulators

  Event wheel calculates each event change and updates the database. Good for asynchronous design. Much of the processor cycle is used in updating the event queue and hence the performance in terms of time is less, but in terms of accuracy is high. Example - NCsim, Modelsim, Verilog -XL

- Cycle based simulators

  Event is calculated only on cycle basis. High performance by speed, but less accurate. Good for synchronous design. Could not trap events between cycles. Example - speedsim from Quickturn(Cadence)

- Hybrid simulators

  Mix of both the above. Example - VSS from synopsis

2) Hardware Accelerators:

The design file is accelerated by external hardware (a group of processors) to increase the speed of simulation. Depending up on the complexity, the design can be partitioned and accelerated. We can accelerate our whole design (including the test bench - remember synthesizable test benches). Its normal that designs of considerable complexity in our code are accelerated by a hardware box(where the powerful processors doing acceleration resides) and test bench related codes(sometimes behavioral models of our design) runs on workstation. That is a part of our design for simulation is executed in the workstation and the remaining in the hardware box. The results of these two simulations need to be synchronized. There are several techniques used for this synchronization viz lock step, simulation back plane, ping pong and calaveras. All these techniques are used for synchronizing digital - analog simulators and/or digital - digital simulators. The most used technique for synchronizing a digital-digital simulator is the simulation backplane. Here the simulators communicate with each other at defined time cycles.

- Hardware Event based Accelerator (Ikos NSIM)
- Hardware Cycle based Accelerator (Quickturn CoBALT)

3) Emulators:

Hardware devices are programmed in such a way that it emulates (imitates) the behaviour of the design. FPGAs and CPLs are the widely used programmable hardware devices for emulation. Example - Quickturn ULTRA, Quickturn Mercury. The function of the emulators is to create a hardware model for our design. Since hardware models

(our designs loaded in FPGAs or CPLs) are very fast than our software simulators, much of the time is saved in verifing our design. Consider for an example, in a small design of ours we have 10 process statement (VHDL). As we know these statements are concurrent, our software simulator will track any event change in the sensitivity lists of these processes simultaneously. But in the background these are executed sequentially by the processor (there is a halt and proceed approach followed). That is why in simulating waveforms of 5 ms duration it will take about minutes in simulators while in actual hardware it took the exact 5ms to run the test case.

4) Hardware prototype:

A reconfigurable prototype board is used to emulate the design. It is used for complex SoC designs. It is more powerful and contains low cost hardware. It is a highly ad-hoc technique. Example - Aptix prototype boards

And about CPS !

CPS is a term often used in verification. If one is involved in verification, he needs to bother about the CPS of the tool he is using, because that affects the schedule of his verification very much. As we know much of the time in chip development is spent in verification, its time to bother how fast these verication can be completed for minimizing the time to market. The more and more stuffs added in todays chip needs more and more time to be spent on verification. To fill up this verification gap, many tools emerge today and ultimately its a head ache to choose the best for the design we are involved. The terminology CPS is used to refer the speed of our verification tool. It is the clocks per second or cycles per second. If a tool is claimed that it can handle 100 CPS, means 100 clock cycles of our design is handled in one second. Also 100 clock cycles mentioned dont mean to the clocks we are using in our design, it can refer to the event clock or the resolution clock of the simulator. Greater these CPS, faster our design be simulated. But there is also a cost involved in this speed. Speed of verifying our design increases from Simulator to hardware prototype boards, and so is the cost.

Some more stuffs...

Ikos NSIM is a mixed level accelerator achieving about 2000 CPS (Clocks Per Second). Ikos NSIM contains an EV board where it contains more powerful processors doing (accelerating) the design we have in RTL. Here is the difference between an emulator and an accelerator. Emulator will contain FPGA’s that would contain a prototype model of our design to be implemented in Asic. So where lies our hardware prototyping? Hardware prototyping is used for highly complex design. It also uses FPGAs, but the design is configured in a multiple FPGA fashion and is interconnected as would be in real SoC. This is highly advanced and more powerful and ofcourse is the most costly techniques in verification.

Normally simulators will have simulation CPS from a few tens to a few hundreds where as the accelerators have a CPS upto a few thousands and the emulators have a CPS of a few million CPS and the prototype model can have a CPS of few tens of millions of

CPS, for our design speed of a few hundreds of millions of CPS.

And one more thing to be noted is that there wont be any steep difference between an emulator and a prototype model, if our design is easy to fit into an emulator.

To be in a more easy way let?s go with an example:

Consider we have a design and we expect it to run for 200 Mhz in hardware. That is for a second 200 million cycles of clock has to be processed. Assume there are 50, 000 instructions in our test case and each will take about 4 clock cycles to complete.

Performance Analysis:

Simulator:

Assume a CPS of 40.

$$\text{Time to verify the above design} = \frac{1}{\text{Number of CPS of Verification tool}} * \text{Number of Instructions} * \text{Clocks required per instruction}$$

$$= 1/40 * 50,000 * 4 = 5,000 \text{ seconds}$$

Hardware Accelerator:

Ikos NSIM with a CPS of 2000.

$$\text{Time to verify the above design} = \frac{1}{\text{Number of CPS of Verification tool}} * \text{Number of Instructions} * \text{Clocks required per instruction}$$

$$= 1/2000 * 50,000 * 4 = 100 \text{ seconds}$$

Emulator:

Quickturn’s Cobalt Ultra with a CPS of 6, 00, 000 CPS.

$$\text{Time to verify the above design} = \frac{1}{\text{Number of CPS of Verification tool}} * \text{Number of Instructions} * \text{Clocks required per instruction}$$

$$= 1/6,00,000 * 50,000 * 4 = 0.33 \text{ seconds}$$

Hardware Prototype:

Assume a CPS of 10, 00, 000 CPS

$$\text{Time to verify the above design} = \frac{1}{\text{Number of CPS of Verification tool}} * \text{Number of Instructions} * \text{Clocks required per instruction}$$

$$= 1/10{,}00{,}000 * 50{,}000 * 4 = 0.2 \text{ seconds}$$

Silicon:

$$\text{Time to run the test case} = \frac{1}{\text{Clock frequency}} * \text{Number of Instructions} * \text{Clocks required per instruction}$$

$$= 1/200M * 50{,}000 * 4 = 1 \text{ milli second}$$

Here is the list of tools with their CPS figures.

- Ikos NSIM - 2000 CPS
  - Viking CSM (Simulation Accelerator) uses Modelsim for simulator interface. This is used for VHDL simulations mainly because modelsim is more efficient for VHDL simulations
  - Voyager (VHDL) - Simulator with featured computational power.
- Gemini(Verilog) - Integrates Cadence tools and Synopsys tools
- *Modelsim -300 and above
  *NC - 300 and above
- Quick turn - Cobalt Ultra(Emulator) - 6, 00, 000 CPS

*performance varies with design. These figures are not the actuals. Sometimes these CPS even drop down to 10.

Ikos NSIM is an event based hardware accelerator and Quick turn CoBALT is a cycle based hardware accelerators. If our design contains all synchronous logic then we can go for Quickturn hardware accelerators, and if its more of asynchronous design then its better to go for an event based hardware accelerator. These tools dont guarantee the floorplanning and placement issues, these tools concentrate mainly on the reduced time for simulation(which has much greater impact on time to market). There are various tools the above vendors provide depending upon verification needs which may vary with frequency and gate counts.