http://www.deepchip.com/items/0522-01.html

ESNUG 522 Item 1) ---------- [04/18/13]

From: [Jim Hogan of Vista Ventures LLC]
Subject: Jim Hogan explains the 2013 market drivers for HW emulation growth

Hi, John,

As you already know, for investment purposes I sometimes find it useful to write up my analysis of a specific technology sector to have my thoughts together on it in one place.

Here's my evaluation of the emulation market.

---- ---- ---- ---- ---- ---- ----

EMULATION IS BECOMING UBIQUITIOUS

Emulation emerged in the early 1990s. In its early days, emulation was primarily deployed by a few large corporations to verify large system-level designs -- specifically for microprocessors and graphics processors. The turning point was when audio and video recordings were digitized. Where possible, designs needed to be verified for both functionality and software behavior against voluminous real data before committing to silicon.

By 2013, we are now moving toward emulation becoming ubiquitous:

- Current mobile devices have hundreds of applications. The vast majority have audio and video use as one of their system appliance "must haves".

- Subsystems now are the same size as what entire designs and processors were in the 90s. Teams are now using emulators to verify many of these larger, more complex blocks.

- Because of compute-speed and capacity gains, emulation is becoming an attractive option for more mainstream verification tasks; such as verifying individual IP blocks in the low millions of gates.

- Emulation is even now needed for smaller, under 1 million gate designs -- if there is a lot of control complexity with a large number of cycles -- such as with an H.265 encoder/decoder. In fact, simulating high density videos on an H.265 decoder would be impractical (because it would take weeks to do) if it couldn't be simulated in seconds in an emulator.

It's safe to say that simulation is vital for functional verification. However software simulation (by itself) is way too slow to capture events that occur when an application runs for even a short time on an actual instance of hardware. Emulation is the only practical way to get usefully long SW application runtimes on a hardware instance.

---- ---- ---- ---- ---- ---- ----

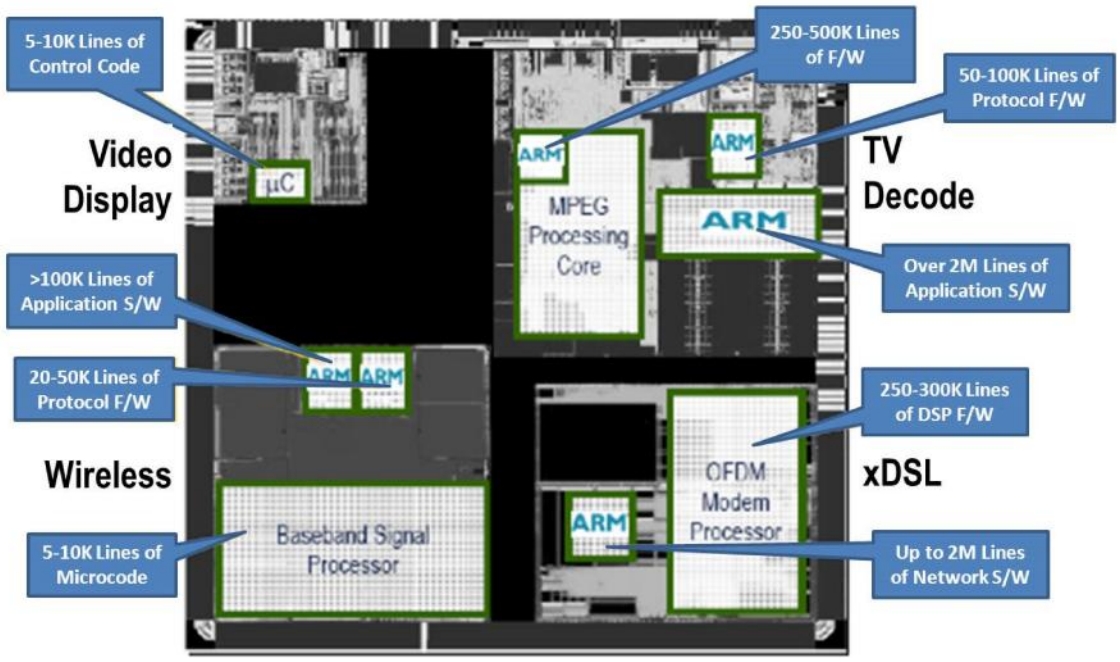
IT'S NOW A SW DOMINATED WORLD:

I see some of the same current top level drivers for emulation as I did for On-Chip Communications Networks: SoC growth is being driven by mobile consumer devices, with the corresponding pressures for:

- Smaller die size and reduced cost
- More functionality
- Increased battery life
- True gigahertz performance

For years, hardware verification was primarily about meeting the hardware design specification. Intel built the processor, Microsoft built the operating system, and Phoenix gave you BIOS. The software conformed to the hardware; meaning that the software developer lived within the constraints of the hardware design.

Fast forward to a 2013 system design world. You'll see it's now software dominated. **Intel**, **Microsoft** and **Phoenix** are still around but struggling; the system design community is now serving consumers by way of **Google Android**, **ARM**, and their SW ecosystems.



Typical SoC - dozens of HW blocks but millions of lines of code

Notice that there's millions of lines of code for roughly a dozen hardware subsystems in a typical SoC. (Source: Texas Instruments)

---- ---- ---- ---- ---- ---- ----

'GOODNESS' IS ABOUT A DESIRED BEHAVIOR

While not explicitly aware of it, consumers' expectations are high with regard to how well the hardware and software work together.

Software is about the experience, not functionality. The software dictates the behavior of an SoC -- in other words, software now forces constraints

on the hardware architecture specification.

Verification encompasses more than "does it do what the specification says?". It now asks "does the specification deliver the end user experience that I'm looking for?". The goal is not just to build the design right, but to build the system that behaves correctly.

How do you ensure your design is up to its intended use? The answer is by using increasingly capable emulation systems.

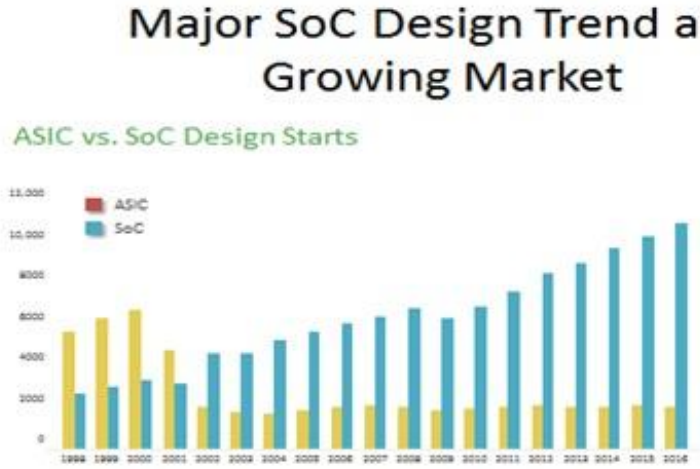
Source: Bluespec, Inc.

You load the operating system and observe the behavior on the hardware; software developers run their application on the virtual machine.

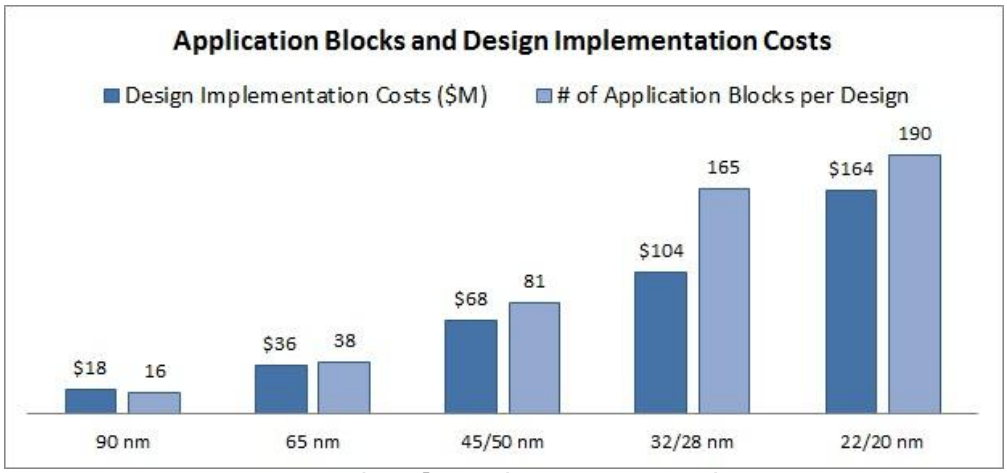
---- ---- ---- ---- ---- ---- ----

IP USE AND REUSE IS EXPLODING

Because today's designs are getting so big and so complex, they're next to impossible to design without using 3rd party IP, IP from fabs, and internal IP reuse.



Semico predicts an average 2013 design will have close to 90 IP cores. This will only grow -- and the more IP that's (re)used, the more each block will demand simulation compute resources -- increasing the need for emulation speed and capacities.



Source: International Business Strategies, 2012

And with each node shrink, 90nm, 65nm, 45nm, 28nm, 20nm, the average number of IP cores used grows -- plus the cost-to-design grows.

---- ---- ---- ---- ---- ---- ----

SW SIMULATION HAS HIT A WALL

SW simulators (like VCS or NC-Sim or Questa) all parallelize across only a small number of x86 processors, so their performance has been scaling only as fast as x86 frequency, which has flattened out. SW simulators have become inadequate for verification within cores and core-to-core.

Emulation is becoming ubiquitous where SW simulation hits a wall. Emulation runs orders of magnitude faster than SW simulation, and its ability to run software in megahertz rather than kilohertz (or even hertz) allows teams to do verification that is otherwise time-prohibitive.

---- ---- ---- ---- ---- ---- ----

EMULATION AND SW SIMULATION COMPLEMENT EACH OTHER

SW simulation is great in the early phases of verification:

- Designers and verification engineers are constantly finding bugs, changing the design, and re-verifying. In this phase, the fast compilation time that SW simulation offers for each new run is critical. And, they're not at the point yet where you need to load the operating system.

- If you have a fixed function or hard-wired integrated circuit, you can just run your Verilog/VHDL to verify that your design does what the specification describes.

What drives emulation use:

- The one downside to emulation is it takes a measurable compile time. But emulation's 1,000X to 1,000,000X faster run time wins out over SW simulation when the number of cycles that must be run offsets the longer compilation time for emulation. As engineers

progress through the design cycle, they must run more and more cycles to identify new bugs.

- The reduced total cost of emulation ownership is another factor helping it grow. Current emulation system cost can go as low as 0.25 to 0.50 cents per gate.

- Even smaller companies can now use it. Emulation used to just be the domain of big companies with huge support teams -- they might spend $2 M on an emulator, and then find they need a service contract for $1 M to make it work.

  Many of today's emulation systems are fairly plug-and-play, with far reduced support requirements. I go into more detail regarding some of these costs of ownership factors in the metrics section of this report.

- Increased static and dynamic probing inside today's much bigger FPGAs has helped emulation grow, too.

---- ---- ---- ---- ---- ---- ----

THE PARTITIONING PROBLEM HAS RECEDED

One significant obstacle to adoption, particularly for FPGA-based emulators, has been the partitioning requirement (also called mapping). A designer would have to partition his register transfer level design into sections which fit inside each FPGA and then connect all the sections via the FPGAs I/Os. It was a tremendous challenge to balance the optimal sizing of the partitioned elements against managing the number of FPGA connections within the limits of the available I/Os.

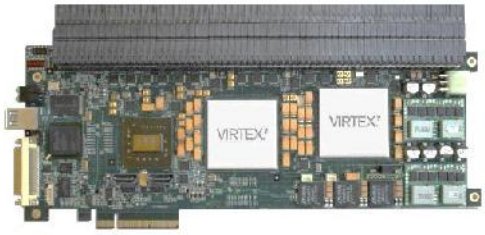
Partitioning software such as what **Auspy Development** offers has helped.

However, it does not entirely eliminate the manual effort. The partitioning process can result in unnatural design hierarchies, and critical timing paths that cross FPGA boundaries can mandate re-partitioning.

One major technology enabler behind the proliferation of emulators relates to Moore's Law: It's now possible to have 30 million gates in just 2 FPGAs.

This higher gate-count-per-FPGA means less design partitioning is necessary. For example, below we see two off-the-shelf FPGA boards.

Source: The Dini Group



Source: Aldec

- The first board has 6 FPGAs; each FPGA is a Virtex-6 LX550T capable of supporting 4 M gates for a total board capacity of 24 M gates.

- The second board has only 2 FPGAs; each FPGA is a Virtex-7 7V2000T capable of supporting 14 M gates for a total board capacity of 28 M gates.

Cutting a design in half is much easier and safer than cutting a design into 6 parts. This lessens a major adoption obstacle for FPGA-based emulators, helping them to close the gap with custom processor based emulators; as well as making emulation more attractive for smaller designs.

Higher capacity FPGAs also push down the cost of the emulation systems; for example, some systems now use off-the-shelf FPGAs at $4 K per FPGA, reducing their core hardware cost. (Further, lower capacity emulators can often add next generation FPGAs shortly after they are released by **Xilinx** and **Altera**.)

---- ---- ---- ---- ---- ---- ----

The rest of my summary covers the terminology, the computer science foundation for different verification approaches, and metrics and rankings for various commercial emulation vendors.

- Jim Hogan
  Vista Ventures, LLC Los Gatos, CA

---- ---- ---- ---- ---- ---- ----

Related Articles

The science of SW simulators, acceleration, protyping, emulation
The 14 metrics - plus their gotchas - used to select an emulator
Hogan compares Palladium, Veloce, EVE ZeBu, Aldec, Bluespec, Dini