

# Going up!

Moving up to the next level of abstraction is becoming a compelling trend for chip designers. By **Louise Joselyn**.

Engineers recognised years ago that increasing chip complexity meant they had to move to a higher level of abstraction. The problem has been determining precisely what this 'next level up' is and then making the connection back down to rtl code for physical hardware implementation. A variety of terms has emerged for this process, including electronic system level design, esl, sld, behavioural level design and architectural level design.

So far, esl – as we now call it – has worked well in asic/SoC verification. High level, object oriented programming languages such as C++, SystemC and SystemVerilog are being used with abstract testbenches by a new generation of verification engineers, providing good functional test coverage and overcoming the limitations of simulation.

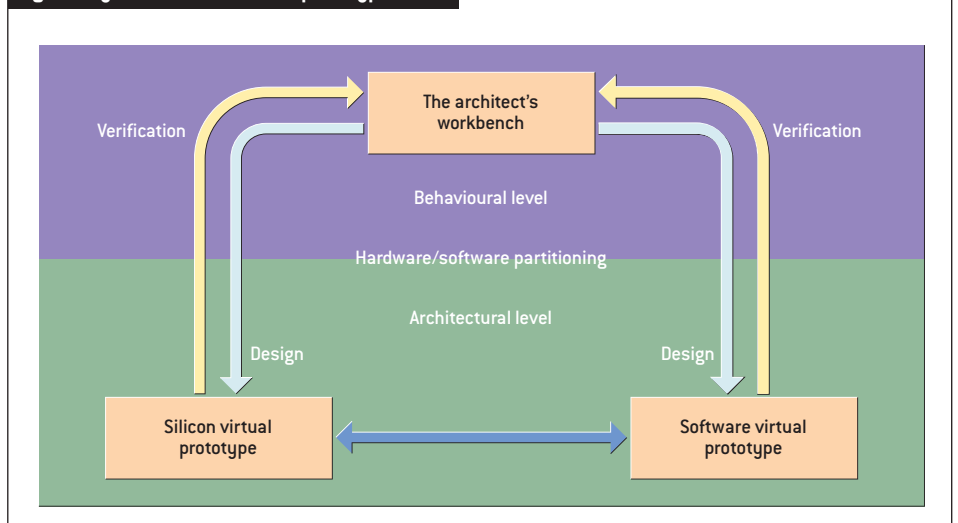
But esl has been slow to take off for the usual reasons – no compelling reasons to change, engineer inertia, lack of an effective tool chain.

Resistance to change has compounded with time to market issues to convince engineers that traditional tools and methodologies can still be used, at least to see the next design through. Many still see esl as adding an unnecessary layer to the design process, without the ability to handle greater complexity. But for leading edge projects, the sheer complexity and an increasing risk of errors which might not get detected until late in the design cycle (or even after production), is forcing change.

Tool chains are beginning to emerge that will enable an automatic, seamless top down design methodology. The previously piecemeal approach, with early point tools such as high level synthesis or C to Vhdl/Verilog compilers, is giving way to more integrated solutions as mainstream vendors recognise the market potential.

Industry watcher Gary Smith said: "We are seeing a mass movement into esl design methodology." He believes systems have become so complex that a 'system architect' can

Fig 1: Gary Smith's three virtual prototype model



no longer, in isolation, predictably define a system. The 'architect' has now become a multidisciplinary team working on system specification and partitioning and determining the impact of trade offs across the various software, firmware and hardware elements. At this behavioural level, system architects model and simulate the system using C, C++ or Mathworks' M language and perform 'what if?' analysis to prove and optimise the system.

After partitioning, software and hardware teams work at the system level to develop silicon and software 'virtual prototypes' to further

optimise and simulate the design and to check it continues to comply with the specification. The use of fpga based virtual platforms and new high level synthesis tools is not only easing the hardware verification overhead, but is also enabling early software development and verification. Eventually, and with much cross communication, the hardware is synthesised and the software compiled.

Speaking at the recent DATE conference in Grenoble, Smith said: "Mentor Graphics saw esl coming a long time before the other vendors." Simon Bloch, general manager of Mentor's design and synthesis division, outlined the key drivers. "ESL design responds to the growing need for software validation, faster verification with fewer bugs and faster time to verified rtl." As a result, he said there has been a dramatic increase in the number of verification engineers. "There are additional benefits, such as the ability to perform architecture trade off analysis, especially for power, and to provide a reference modelling for rtl." Bloch concurs with Smith's vision of



multidisciplinary teams comprising hardware, software and architecture engineers, but all have different needs. "Hardware guys need to develop derivatives fast and simply; they need hardware oriented algorithms and standards based buses." Architects need to optimise for low power, exploit multicore capability and provide scalability, he added. In the software domain, Bloch noted, esl is seen as an aid to meet embedded software needs, reduce lead times and for optimisation.

"Our objective is to create a single transaction level modelling [tlm] platform for all three groups, enabling architectural design, virtual prototyping, system verification and high level synthesis" Bloch said.

Synopsys, meanwhile, jumped into the esl market largely through the acquisition of Synplicity and Synfora, for high level synthesis, and Vast and CoWare, for hardware/software codesign and virtual platforms. Joachim Kunkel, general manager of Synopsys' solutions group, sees increasing use and reuse of IP, especially larger building blocks and memory subsystems. "Today's SoCs are quite different, with a structured design style, reuse of building blocks and standard on and off chip interfaces. This is the key to 'plug and play'," he said.



**Kunkel:**  
**"Virtual prototyping is important to get software development started in parallel with hardware design and to ease the hardware/software integration and validation process."**

But Kunkel gave his own 'reality check'. "We don't yet have a full front to back, fully integrated SoC design flow," he admitted. "To realise this, we need not only a broad base of quality, reusable IP, but also IP reuse and robust eda tools and flows."

Kunkel sees the increased software element, coming with the greater exploitation of multicore and mixed processor designs, creating a growing interest in tools for architectural design and fpga based virtual prototyping. "Virtual prototyping is important to get software development started in parallel with hardware design and to ease the hardware/software integration and validation process," he explained.

No discussion of esl design is complete without reference to Cadence's EDA360 vision; essentially another name for electronic system level design.

John Bruggeman, Cadence's chief marketing officer, participated in the debates at DATE. "What the early proponents of esl missed was that it was disconnected; it has to go right the way down to silicon. ESL was a good idea before its time." He reckons increasing product complexity in the last two years has generated the real need for system level design.

EDA vendors are offering ever more sophisticated esl tools, many based on industry standards. One interesting solution is the Hotspot Paralleliser, developed by Compaan Design, a spin off from Leiden University's Institute of Advanced Computer Science. The tool can be used for exploratory design at a high level, as well as to implement new features quickly into a product development plan. Exploiting parallelism and targeting leading edge hardware, it works in conjunction with third party C to Vhdl tools.

John van Brummen, Compaan's chief commercial officer, added another reason to adopt an esl design methodology. "Moving up to this level of abstraction allows the systems house to be hardware independent. Software and firmware can be developed and tested on reconfigurable fabrics, such as fpgas, and the final hardware solution can be outsourced," he concluded.

