

A Design Style to simplify IP
Integration and Verification

Haresh Amreliya



A Design Style to simplify IP Integration and Verification

The challenge facing system designers today is to use the available silicon to bring new products to market within budget and in a reasonable time. This will become a serious with the trend to incorporate more memory in the future: from 400k to 2M bits. Challenges are also posed due to a tenfold increase in the size of vendor and customer cores from 130k gates to 1.3M gates and in the lines of code per new design. The current islands of IP will be replaced by a sea of IP encompassing most of the chip area, because the increase in random logic will be proportionally much lower.

In order to design larger chips for the future, designers face three major challenges. The first of these challenges is the design productivity gap. To illustrate the design productivity gap, consider that the current design starting at one manufacturer runs about 300,000 gates (1.2 million transistors), while the company can economically manufacture circuits of approximately 5 million gates (20 million transistors.) This divergence is growing every year. As per Moore's Law, this inability to fill the available silicon with meaningful circuitry has an economic consequence due to the disparity between supply and demand.

Using larger circuit building blocks or bringing other technologies such as memory, FPGA, RF or analog can only fill the unused gates on silicon. Hiring larger teams of engineers to work longer hours could result in a large number of gates, but not necessarily within the desired project schedule. A certain structure is necessary in designs of this magnitude. The blocks may come from various sources depending on the particular application- from the semiconductor manufacturer, from the customer or from a third party or IP vendor. A semiconductor supplier offering systems-on-chips will have IP libraries based on one or more micro controllers as well as memory controllers and IP related to the market segments it addresses. The customer may have proprietary IP from projects it has done in the past with the same or other vendors. In the best case this has been proven in applications and will be in silicon or a design language.

The second challenge in design of very large chips is the compound risk dilemma resulting from combining IP multiple blocks. This is based on the probability of each block functioning individually as well as the probability that it can be attached to the bus structure of the chip and will interact successfully with all other IP blocks.

The third obstacle facing designers of very large chips is rigid conformance to ambiguous specifications. This means that a system on a chip must conform to all preceding assumptions and existing applications whether they be good or bad. For example, although a published standard may not define the state of a bit in a particular situation, previous applications may set it and even make further use of this information. It is thus necessary that a new design follow this practice in order to avoid errors. System verification can take so much time as to be practically impossible.

There is a need for a design style that is a methodology of methodologies, to bridge the gap between design and verification on the one hand and manufacturing capabilities on the other. The transistors, which are currently unused, can only be brought to bear with intensive IP utilization. Simply using more IP blocks will fill the space available but in a way that compounds the design risks and makes project planning difficult. The Rapid Silicon Prototyping design style

is a new set of simplifying assumptions to enable the design of very large chips. It addresses the four methodologies of design reuse, chip design, chip debugging and verification, and system level hardware and software co-development. Implementation requires expertise in tools, methodology, and technology. The new design style is based on four basic premises.

The best way to design a new chip is to start with an existing chip:

This means basing new designs on existing prototypes in silicon rather than software descriptions. It is necessary to offer reconfigurable and externally extendable reference designs created specifically to serve as development lab prototypes for custom end products. These designs can be offered as a family such that each end user starts with the IP particularly suited to his requirement.

Build chips from reusable, modular building blocks using standardized interfaces:

The use of modular IP building blocks with standardized interfaces creates a cycle of building block based design procedures, including the ability to amortize IP generation over several generations of end products. Design processes can be simplified until they are much closer to the ideal of plug and play. The problem with pre-designed IP is less in the IP designs themselves than in the need to hand craft interconnections between building blocks. A further complication lies in the lack of standards for block edges and interfaces. Even with the most carefully designed building blocks, without standards the variations in block interfaces guarantee the chip designer will encounter at the very least numerous undocumented features which may act as bugs.

Solving the interconnect problem is at least as important as securing the IP blocks. It requires minimizing the number of non-standard interfaces and using IP blocks, which have already been integrated and tested. This capability is extended to the design software by using standard interface procedures for the IP in order to assist the designer with the trickiest part of chip design and layout.

Every reusable building block on the chip exists as a fully- documented HDL description that can be independently compiled and integrated into an IC. In the other direction, the appearance of the IP blocks on the reference IC tells designers that the building blocks work in real-world silicon and can be tested in full.

Each block has a standardized interface that plugs into an on-chip data bus. The address decoding and arbitration logic is automatically generated, largely eliminating the time and risks involved in designing custom glue logic for each block. Standardized buses provide many benefits for their users, such as reduced risk, improved productivity, and reduced time to market.

The buses themselves have been selected for their robustness and suitability to the wide range of existing IP. The on-chip bus strategy focuses on three open bus architectures. First is the ARM's AMBA System Bus (ASB), suitable for an on-chip processor requiring high-speed random access. Peripherals needing a simple bus with low power and low gate count are served by an extended version of the ARM Peripheral Bus called the VLSI Peripheral Bus (VPB). The third bus is an on-chip version of the industry-standard PCI bus, a structured burst-oriented high-speed bus offering compatibility with existing off-the-shelf devices. For most IP blocks, only one of the buses will be appropriate.

The three buses have been extended off-chip to connect with other peripherals and functional blocks via standardized card slots mounted on a circuit board. To create a production version of

the chip- and board-level prototype, the design team can add IP to the off-chip buses, either in existing chips or in a hardware simulator. The inclusion of Field Programmable Gate Array (FPGA) technology in the prototyping platform architecture adds significant design flexibility, with which engineers can create hardware-based replicas of circuit building blocks that do not yet exist as off-the-shelf silicon. An on-board FPGA can contain either customer or third party IP or be used for additional logic to be developed with an ASIC compatible design flow. A common use is as an interface between external IP and the RSP bus structure.

To compile a manufacturing net list it is necessary to process the hardware descriptions of the prototype chips desired building blocks and on-chip bus structures plus the IP from the plugged-in external chips and FPGA through a standard EDA tool flow.

VLSI has developed the High-level Description Language Integrator (HDLi) to help its customers to bring together the IP on their system-level chips. HDLi is a reusable IP delivery tool that simplifies ASIC design while producing more efficient silicon than is possible with traditional methods. It contains a set of macro compilers, HDL templates and silicon system blocks, which permit the user to specify parameters and compile a function block within seconds.

HDLi delivers HDL models for the IP along with test benches, on-line data sheets, scripts for simulation, synthesis, and test, and firmware, if applicable. Mixing of tools is possible; a demonstration project to prove the methodology used Verilog and VHDL as well as Model Tech for simulation, Synopsys for synthesis and static timing verification, and Compass tools for physical design and verification.

HDLi function compilers are efficient macro compilers that generate both RTL and structural implementations. Examples are multipliers, LFSRs, comparators and register files. HDLi function compilers support true top-down implementation. For example, HDLi views $A*B$ as one function instead of two synthesizable operators. After the user has selected his desired features from the options presented, the HDLi compiler generates the function using the appropriate structures from VLSI's technology offerings. This specialization results in a more highly optimized design using more complex cells. The result of compiling is a structural HDL net list and RTL code. Because HDLi produces library-optimized functions, more of the routing is in lower metal layers, freeing up vertical and horizontal routing resources. HDLi-generated functions typically have one third to one half of the instances and nets of synthesized functions. The result is a denser design and higher operating speed.

Use software-based tools to support the design style, not be the design style.

The use of development tools currently represents a bottleneck in silicon design. Many designs are constrained by the capabilities of the available tools, with verification being one of the most difficult issues. The verification tools remain an important part of the design style, but their role has changed to support prototyping of real-world silicon, not serve as an abstract substitute.

When IP is developed, the hardest task is functional verification, which traditionally involves very large simulations often taking days or even weeks to run. Pre-designed IP enables the IP to be thoroughly validated in hardware using real application code before the customer's target device is designed. Silicon using the manufacturing process of the end product is the best model for that end product. Use of RSP is also the fastest method of verifying the application. It is possible to run much more extensive simulations in the available time than would otherwise be possible. This practice has made possible extensive investigation of the interactions within the system. The task of verifying the implementation of the IP on the target device can be reduced dramatically by using static timing analysis and formal verification rather than extremely

complex system simulations.

After system verification with RSP has been completed, it is easy to use the IP from HDLi. VLSI followed this flow described above when designing the RSP7 chip used in its Rapid Silicon Prototyping system. The designer took compiled IP blocks out of HDLi and instantiated them. HDLi provides the IP in an RTL form, synthesis scripts, simulation scripts, DFT scripts, a test bench, and even the HDL instantiation statement for the designer to copy. Future members of the RSP family will provide the database as a starting point.

At this point scan chains can be inserted and formal verification tools used to check the result. After generation of a testable net list, the designer can use the timing constraints from HDLi to guide timing-driven layout tools. He can use static timing analysis on the layout and iterate until the layout has been optimized. The test pattern generated by the scan insertion tool can be used to create the test pattern for the chip. As a final step these patterns are to be simulated under tester conditions to ensure that they work. This flow maintains functional integrity of the circuit through physical design and test insertion without the need for functional simulation. Test pattern simulation is a much faster way to verify testability because the pattern size is limited and the pattern is only run once. Chip and test pattern are now ready to be handed over from design to manufacturing.

Design application hardware and software simultaneously and interactively. The ability to develop application software simultaneously with the hardware has many beneficial effects. At a minimum, it cuts total product design time by enabling parallel hardware-software development. This parallelism can break down barriers between the departments, leading to more interactive, systems-oriented approaches to end product development where hardware and software engineers work together to optimize the architecture of a design. It is no longer a case of first the hardware, then the software. The platforms create a development lab on a board, giving hardware engineers the opportunity to run a silicon prototype chip through its paces at full speed, writing final specifications for a custom, production version of the silicon prototype IC. The hardware engineering team can run the prototype chip in an application while employing a subset of its functions. The same platform is a target system for the software engineers, who are no longer dependent on the completion of hardware design and a first version of the silicon to start work. This means that they can start work almost immediately. Another benefit is that there is no longer a need for multiple versions of the chip, one to use as a basis for software development and a later one (or more) incorporating the results. It is thus possible to telescope the product development schedule.