JUST AS NO ONE would consider building a house without a blueprint and specific details of the floorplan, no chip design team today would begin a new project without first developing a carefully considered verification strategy. And with verification consuming up to 70% of the design cycle it’s easy to understand why implementing an effective strategy has become so crucial.

With all new home projects, the location of the kitchen, number of bathrooms and the square footage of the master suite is likely to change from the original blueprint. The later the changes are made, the greater the cost overrun, fraying everyone’s nerves and wreaking havoc on the homeowner’s budget.

The same is true in chip design, making early and thorough verification essential. If an error is detected late in the design cycle, fixing the problem may require repeating many design and verification steps, slipping schedules and missing valuable market opportunities. If an error is detected after manufacturing, a respin can cost half a million dollars or more. If it is detected in a fielded device, it can cost hundreds of millions of dollars to recall defective devices.

Verification has been evolving over many years. Simulation techniques used since the 1970s are losing their effectiveness since coverage and performance dramatically decrease as designs grow larger. Once a design reaches a quarter of a million gates, gate-level simulation becomes impractical. At the half-million gate level and above, it becomes too risky to verify designs using traditional simulation methods alone. A much needed alternative which meets the challenges of system on chip (SoC) design is offered by formal verification. It’s faster than simulation, requires no vectors and is exhaustive in ensuring more bugs are caught earlier and when they take less time to fix. Equivalence checking, a key element of the methodology, is an automated way of detecting functional inconsistencies, providing a reliable way to ensure that the final design implementation does what the Register Transfer Level (RTL) code specifies. It uses mathematical techniques to determine whether one design representation is functionally equivalent to another. Equivalence checking can functionally verify the implementation of an entire chip design. Now, design teams are taking their verification strategies one step further and modifying them to include assertion-based techniques.

Assertions are checks embedded into the design to verify the designer’s assumptions about how a logic block should operate, both by itself and in concert with surrounding logic blocks. Embedded assertions at the interfaces of intellectual property (IP) blocks preserve the design knowledge needed to verify them as an integrated SoC, essentially making them self-checking. The net result is that assertions help SoC designers find more errors earlier in the design cycle, where they are far easier and less costly to fix. And once assertions are embedded into a design, assumptions the designer made about how the block should operate as part of a system are automatically verified wherever and whenever the block is used in the future.

Specifying assertions can be done through pre-defined assertion monitors from the Open Verification Library (OVL). It consists of an open-source, Verilog Hardware Description Language (HDL)-based library of assertions as a means for capturing design knowledge, which then becomes portable with IP blocks in whatever systems they are integrated.

Formal verification further increases the thoroughness of OVL assertion checks by increasing controllability or exhaustively exploring the state space of the design to determine whether it is possible for the assertions to be triggered under any combination of internal states and external stimuli.

Just as blueprints ensure a home is built to the exact specification of the homeowner, a verification strategy put in place prior to the start of a design project will ensure that a chip works as it is supposed to work.

Kuang Chien Chen is chief technical officer of Verplex Systems Inc.