***Reference number 9***

***High-Level Synthesis for FPGAs: From Prototyping to Deployment***

* The rapid increase of complexity in system-on-a-chip (SoC) design has encouraged the design community to seek design abstractions with better productivity than register transfer level (RTL).
* Nowadays there is a rapidly growing demand for innovative, high-quality HLS solutions for the following reasons.

1. . An automated HLS flow allows designers to specify design functionality in high-level programming languages such as C/C++ for both embedded software and customized hardware logic on the SoC. This way, they can quickly experiment with different hardware/software boundaries
2. Huge Silicon capacity requires a higher level of abstraction
3. Behavioral IP reuse improves design productivity. Behavioral IP can be retargeted to different implementation technologies or system requirements.
4. The wide availability of SystemC functional models directly drives the need for SystemC based HLS solutions, which can automatically generate RTL code through a series of formal constructive transformations.
5. Many SoCs, or even chip multiprocessors move toward inclusion of many accelerators (or algorithmic blocks), which are built with custom architectures, largely to reduce power compared to using multiple programmable processors. These algorithmic blocks are particularly appropriate for HLS.

* Although these reasons for adopting HLS design methodology are common to both ASIC and FPGA designers, we also see additional forces that push the FPGA designers for faster adoption of HLS tools.

1. For FPGA designs, in-system simulation is possible with much wider simulation coverage. Design iterations can be done quickly and inexpensively without huge manufacturing costs.
2. Modern FPGAs embed many predefined/fabricated IP components. These predefined building blocks can be modeled precisely ahead of time for each FPGA platform. It is possible for modern HLS tools to apply a platform-based design methodology and achieve higher quality of results.
3. FPGA platforms are often selected for systems where time-to-market is critical, HLS tools put this tradeoff in the hands of a designer allowing significant reduction in design time.

* A growing number of FPGA designs are produced using HLS tools. Some example application domains include 3 G/4 G wireless systems, aerospace applications, image processing, lithography simulation, and cosmology data analysis.
* HardwareC is one of the earliest C-based hardware synthesis languages for HLS.
* AutoPilot is one of the most recent HLS tools, and is representative of the capabilities of the state-of-art commercial HLS tools available today
* AutoPilot creates synthesis reports that estimate FPGA resource utilization, as well as the timing, latency, and throughput of the synthesized design.