***Reference number 13***

***Xilinx Vivado High Level Synthesis: Case studies***

* HLS reduces the effort of HDL design capture and debug
* HLS allow flexibility in the final hardware implementation in order to meet design constraints.
* High-level synthesis takes a complete behavioral C++ description of a system along with a series of directives that describe the architectural constraints, and automatically generates a HDL design description.
* RTL code goes lengthy, which is not only costly and time consuming to develop , but also difficult to debug and verify
* HLS is effective in developing and testing early design ideas prior to full HDL generation and RTL simulation.
* A HLS model of a system can be synthesized into many different HDL models, each meeting different timing, clock and area constraints.
* HLS offers opportunities to reduce verification time.
* HLS allows the creation of abstract C++-based functional models, which can be readily tested and verified, prior to generation of HDL design descriptions. This can reduce the testing and debug effort on the resulting HDL models.
* Vivado HLS is the Xilinx HLS engine, accepting descriptions in C, C++ or SystemC and generating VHDL, Verilog and SystemC RTL descriptions.
* Vivado HLS offers a co-simulation feature that can be used to help verify the generated RTL designs.
* HLS offers reduced development time. The generated RTL models are often at least as efficient as those obtained from synthesis of hand-coded RTL.