Chapter 1 - Opening

Digital design evolution and ESL

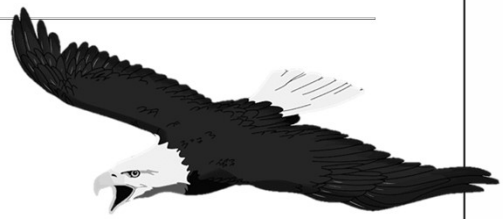
Zainalabedin Navabi

Digital design evolution and ESL

This chapter discusses digital design evolution from transistor-level to system-level. We start with a discussion of each design level and show elements, communications, modeling language, and timing simulation for it. Then, we explain that today's complex hardware structures consist of the integration of many IP cores, hard cores, soft cores, and processing elements require a proper level for describing to hide many unnecessary hardware details. This leads to the discussion of SystemC that properly handles hardware concurrency and timing through a package of C++ classes and its simulation kernel.

Digital design evolution and ESL

+ Transistor Level to ESL
+ System Level Design Concepts
+ Converge in upon SystemC



A BIRD'S-EYE VIEW OF OUTLINE

Digital design evolution and ESL

- Transistor Level to ESL
  - + Transistor Level Design
  - + Gate Level Design
  - + RT Level Design
  - + System Level Design
- System Level Design Concepts
  - Embedded System
  - System-on-Chip (SoC)
  - + Language-based
  - OpenCL
  - HLS
- Converge in upon SystemC
  - What is SystemC
  - Why SystemC

Digital design evolution and ESL

- In this chapter, we will be addressing the following outcomes:
  - Understanding design evolution from transistor-level to ESL
- Relation to the previous chapter:
  - The previous chapter reviewed the prerequisite concepts for this course including digital logic design, VHDL hardware description language, and C++ programing
- Relation to the next chapter:
  - The next chapter will discuss object oriented logic modeling
- Is an assignment due for this chapter? Which one?
  - No