Concept System Level Design Languages

Norbert Tremurici, Christian Töpfl

What are System Level Design Languages in general

- Difference to "normal" programming languages (software, RTL code)
  - what do "normal" programming languages try to achieve?
  - what can be expressed in "normal" programming languages?
  - what system level design/high level synthesis languages and tools have been created and which are in use? many apparently: https://ieeexplore.ieee.org/document/7368920 (A Survey and Evaluation of FPGA High-Level Synthesis Tools)
- Benefits? e.g. Design of hardware only with C knowledge
  - benefit: design stage can be much quicker with good tool usage and tedious, error prone tasks avoided
  - caveat: using system-level design languages sometimes requires intimate knowledge of both the software and hardware domain
- Goal: Automate split in hardware and software & disadvantages because of that
  - caveat: how to constrain design to hit specific targets such as performance, resource and real-time requirements...
  - can time to market be improved by trying a hw/sw co-design methodology using system level design? apparently yes: https://ieeexplore.ieee.org/document/8356004 (Are We There Yet? A Study on the State of High-Level Synthesis)
  - can tools automate the decision on what to accelerate using hardware? somewhat, but difficult to automate: https://ieeexplore.ieee.org/document/8847448 (High-Level Synthesis Design Space Exploration: Past, Present, and Future)

Details of SystemC, Intel High Level Synthesis Compiler, Xilinx (Vitis) High-Level Synthesis and possible others

- Basic Methodology, Syntax, Base Language...
- Usage in the industry, academia, ...
- How does Intel approach high level synthesis when building their tools? https://youtube.com/watch?v=nYbw9k7KNJ4&list=PL0pU5hg9yniYUd4ez4Rypi7QHWVUcpaxY&index (Introduction to High-Level Synthesis (Part 1 of 7))

Code Examples (~5)

- C++ vs SystemC vs VHDL
- Spade and others(?)
- C vs {Intel HLS, Xilinx HLS} vs VHDL