PyMTL: Using Python to Design, Model, Verify, & Tapeout Chips

Multi-Level Hardware Modeling

Applications — Algorithms — Compilers — ISA — Microarch — VLSI — Transistors

Functional-Level Modeling

Cycle-Level Modeling

Reg-Transfer-Level Modeling

Hardware Generation and Simulation Framework

Host Language — DUT → Sim → TB

gen-erate / co-sim

HDL — DUT → Sim

FPGA ASIC

SystemC? BSV? Chisel?
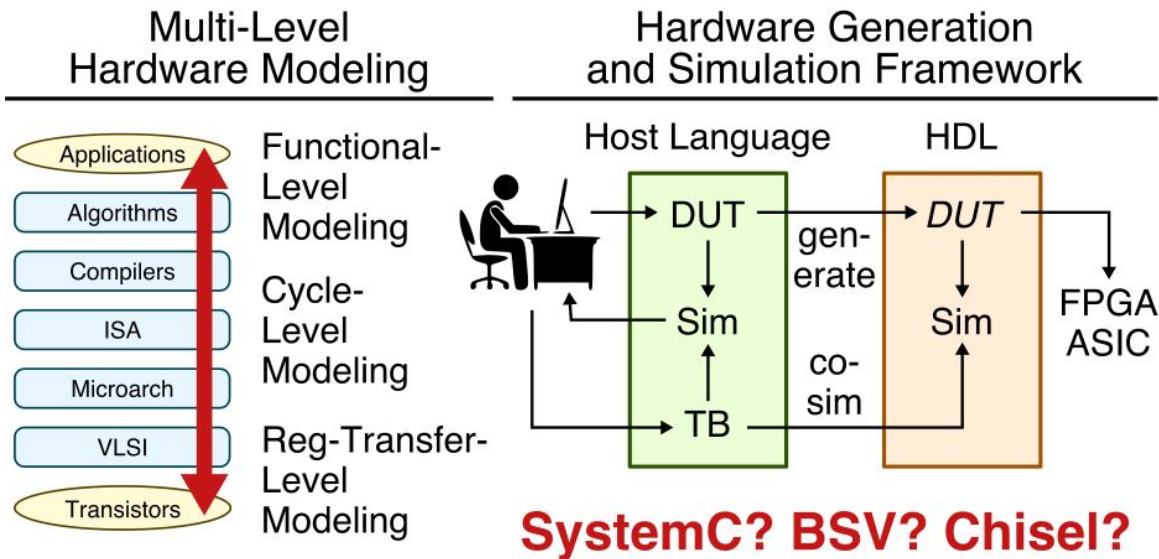
PyMTL: A Python-based hardware generation & simulation framework for building SoCs which enables productive multi-level modeling and VLSI design

- Used by 300+ Cornell students
- Used in several architecture papers from Cornell groups
- Used in three ASIC tapeouts in GF 130nm, TSMC 28nm, TSMC 16nm

LVDS Clock Receiver
RISC Processor
Host Interface
HLS Accelerators
SRAM Subbank (4KB)
SRAM Subbank (4KB)
SRAM Subbank (4KB)
SRAM Subbank (4KB)
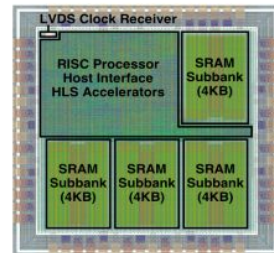
– Derek Lockhart et al., "PyMTL: A Unified Framework for Vertically Integrated Computer Architecture Research," MICRO 2014.
  `https://github.com/cornell-brg/pymtl`
– Shunning Jiang et al., "Mamba: Closing the Performance Gap in Productive Hardware Development Frameworks," DAC 2018.
  `https://github.com/cornell-brg/mamba-dac2018`

Google