C++HDL

Hardware Description Language for rapid digital design

Mike Reznikov, 2025

Chipmakers boom

- Thousands companies are doing their own chips in 2025
- Most of them are doing AI applications
- Market already contains hundreds of AI accelerators of different size and speed
- GPU and CPU vendors are still fundamentally on AI market keeping their part
- Many countries try to develop their own chip and AI technologies to be independent and to satisfy growing demand
- Particular bottlenecks of the world chip making process appeared

Bottlenecks of chip making

- Production capacities and resources – few countries have technologies for semiconductor ICs production and big part of capacities and resources is already in game
- Human resources – large part of semiconductor professionals are already taken by large companies and having many zeroes ending salaries, making starting up extremely expensive and almost impossible

- **Time** – nobody still has 10 or 5 years to develop their devices. The most relaxed estimation is 2 years of Time-To-Market with 2 iterations or you lose

Time consumptive processes

- Hiring, setup & management takes months, especially if you're looking something below market prices
- Software development – usually time consumptive but can be compressed if you have large and professional team
- **Hardware development** – eats up to 90% of time of hardware vendors. **Design. Verification. Physical design. Floorplanning. Tapeout**. Each of these stages can take years for large and complex designs.

New vendors have two options: find their way to develop hardware faster or buy tools and components at million dollar prices.

Digital vs Analog

- Your chip design will contain both Digital and Analog part
- Analog part development is much slower and more complex than digital, but there are almost all components developed and achievable in libraries – this is your entrance price
- Complex digital filling of your chip, which usually takes 90% of crystal, is much harder part in comparison to small pieces of analog interfaces
- Digital algorithms operate up to 512, 1024 and larger bitwidths on frequencies up to 2GHz implementing such multistage blocks as RAM, CACHE, ALU, CPU, VECTOR UNIT, GEARBOX, multiplexers etc
- **Digital part will eat 90% of your budget and years of development and simulation time**

C++HDL and simulation

- Simulation eats hundreds and thousands years of CPU time for large RTL designs during development and testing (daily/weekly) stages
- RTL simulation tools are usually slow and extremely expensive (there are open source replacements like Verilator which require special skills like C++ programming)
- When average size company runs maximum 100-1000 RTL tests nightly, it needs to run 10k tests each night usually
- If test iteration after RTL changes takes 5-10 minutes of time, company already cant afford it (but probably is still not aware of this)

C++HDL runs tests and simulation 100x faster

C++HDL approach

- C++HDL approach is a C++ replacement for Verilog/SystemVerilog RTL
- C++HDL defines cycle accurate register-to-register aligned RTL, same way as SystemVerilog
- C++HDL allows building of a large C++ digital processing ecosystem (classes, types, templates, methods, inheritance, OOP state/behavior)
- C++HDL provides extremely fast simulation of blocking Verilog-style expressions during development and during testing
- C++HDL generates SystemVerilog output then which is 100% register to register reflection and can be converted backwards

C++HDL source code

- C++HDL author is Mike Reznikov (https://www.linkedin.com/in/mike-reznikov)
- I’m using C++HDL in digital designs I’m developing for AI acceleration and hardware calculations in different projects
- I’m partially open sourcing C++HDL and some examples with MIT license https://github.com/mirekez/cpphdl
- If you need rapid digital development consider using C++ and try C++HDL to be able to easily convert the results to SystemVerilog