

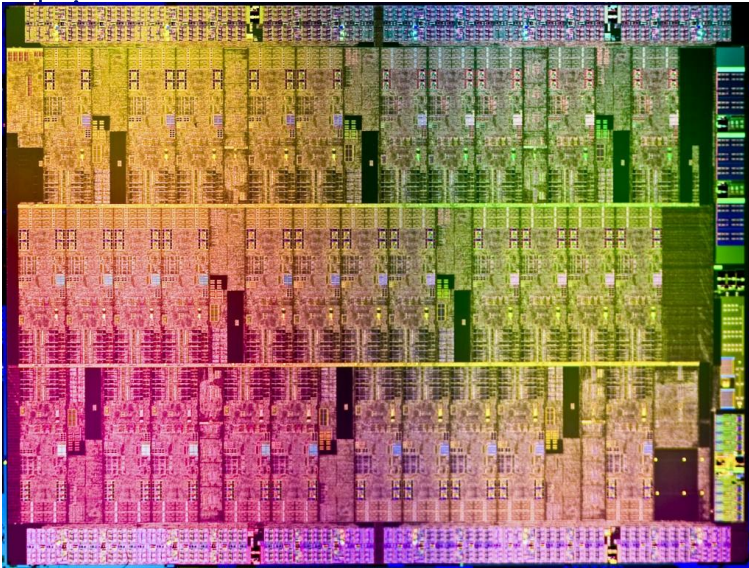
# From software to hardware

Martijn Bastiaan <[martijn@qbaylogic.com](mailto:martijn@qbaylogic.com)>



# What's "hardware"?

This:



Not

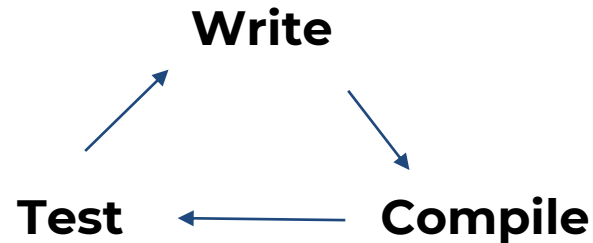


# Why talk hardware in a software cluster?

Hardware and software development have a lot in common:

- FPGAs are reconfigurable
- Virtually all design time spent on “plain text”
  - e.g. VHDL, Verilog

- Typical work day:



# Why talk hardware in a software cluster?

FPGAs have a lot to offer

- Exact control over timing
  - Latencies measured in nanoseconds
- Enormous performance boosts for specific applications



# Hardware architectures

- Fundamentally:
  - Logic elements (AND, XOR, OR, ..)
  - Memory elements (flip-flop)
  - Connected to each other by wires
- Clever combinations do actual work!
- Conventional languages: Verilog and VHDL



# Better abstractions

Software:

- **Assembly** to **C** to **Java, Python, C++, Scala**

Embedded:

- **C** to **MicroPython, Rust**

Hardware:

- **VHDL, Verilog** to **????**

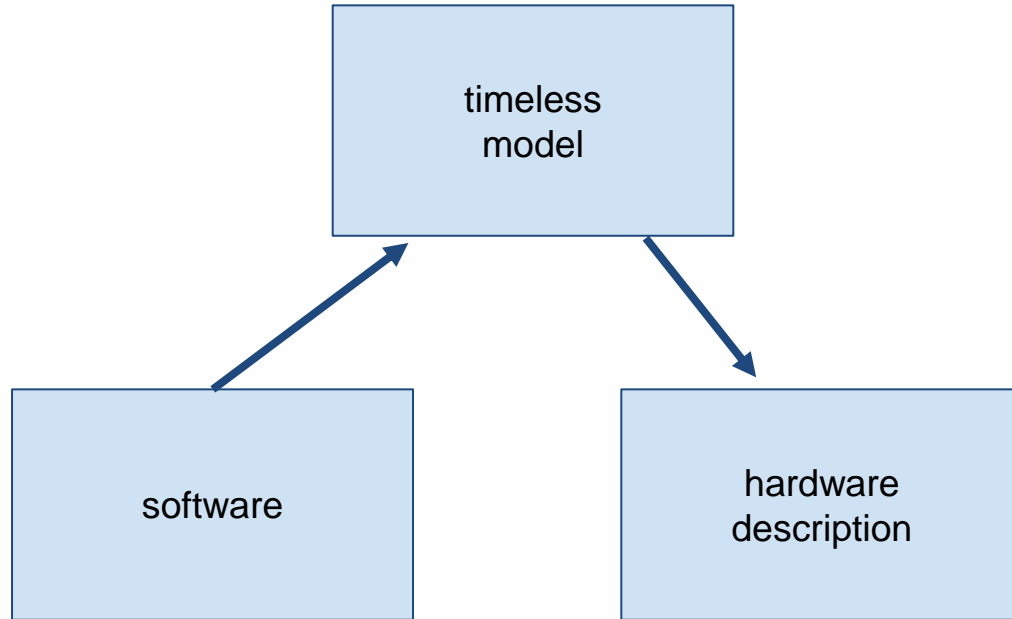


# Software languages

- E.g., C, Python
- Fundamentally:
  - Fetch instruction
  - Execute
  - Store result
- In other words:
  - Sequential operation
  - Critically depends on (large) *Random Memory Access*

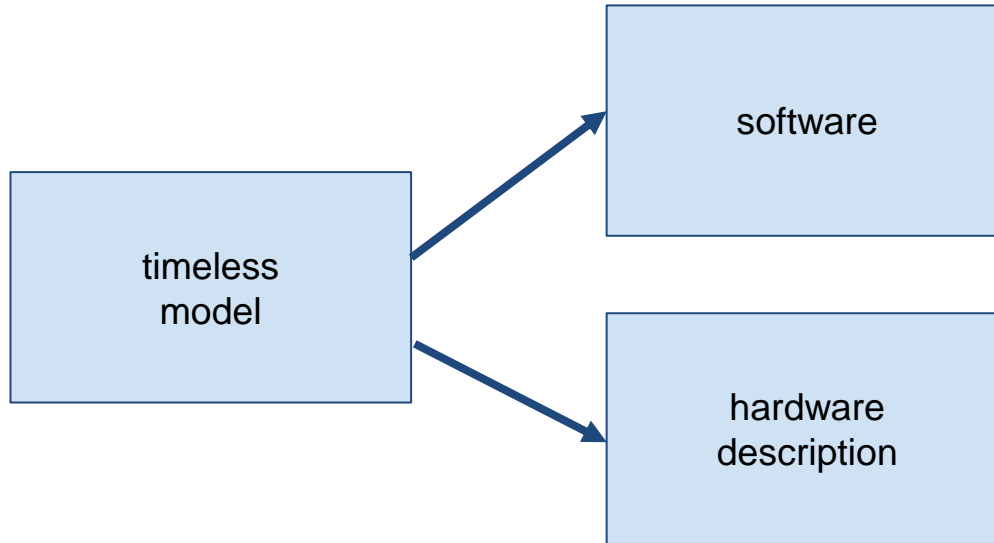


# Software to hardware





# What if..



# Functional prog.: Haskell

- No deals with the devil:
  - A function's output only depends on its input
  - No explicit memory model / pointer logic
  - Effectively parallel
- Decades of engineering:
  - Software performance on par with Go/C#
  - Strong mathematical foundations
- Very expressive type system



# Clash: Haskell => hardware

- The missing piece:
  - Translates Haskell to hardware descriptions
  - Reuses large parts of Haskell's optimizing compiler: GHC
  - Full control over hardware, like VHDL and Verilog
- Virtually all of Haskell's features translatable to hardware
- Free, OSS, BSD-2 licensed
- Developed by community and QBayLogic
- <https://clash-lang.org/>

# Clash in the real world

- Myrtle.ai
  - FPGA based neural network accelerator
- Demcon-Focal
  - Satellite - ground laser communication
- “FAANG” Research
  - Cycle predictable, self-synchronizing data centre applications
  - <https://github.com/bittide/bittide-hardware>

# Conclusion

- Functional programming ideal start for architecture specifications
- Haskell is a high performance, general purpose programming language
- Clash adds hardware description backend to Haskell
- **Software engineers should not fear hardware designs!**



# Questions

- QBayLogic, FPGA-ASIC design house
- Present today:
  - Maarten Kuper <maarten@qbaylogic.com>
  - Martijn Bastiaan <martijn@qbaylogic.com>

