The Hardware Design Toolchain

Approaches and State of the Art

Fredo Erxleben

September 17, 2014

We will hate the tools (FCCM 1996 prediction for 2001)

We will still hate the tools (FCCM 1998 prediction for 2003)

We will merely dislike the tools (FCCM 2000 prediction for 2005)

We [will] hate the tools more (FCCM 2007 prediction for 2012)

Outline

Why bother?

- Complexity of . . .
  - . . . designs
  - . . . the design process
- Tools are . . .
  - . . . rarely known to developers
  - . . . of unknown usefulness
  - . . . (not?) adequate?

HDLs: General

Everything below RTL is often specific to the HW-vendor
$\rightarrow$ HDLs also serve as interfaces

- VHDL
- Verilog
- SystemC, ABEL, JHDL...

HDLs: VHDL and Verilog

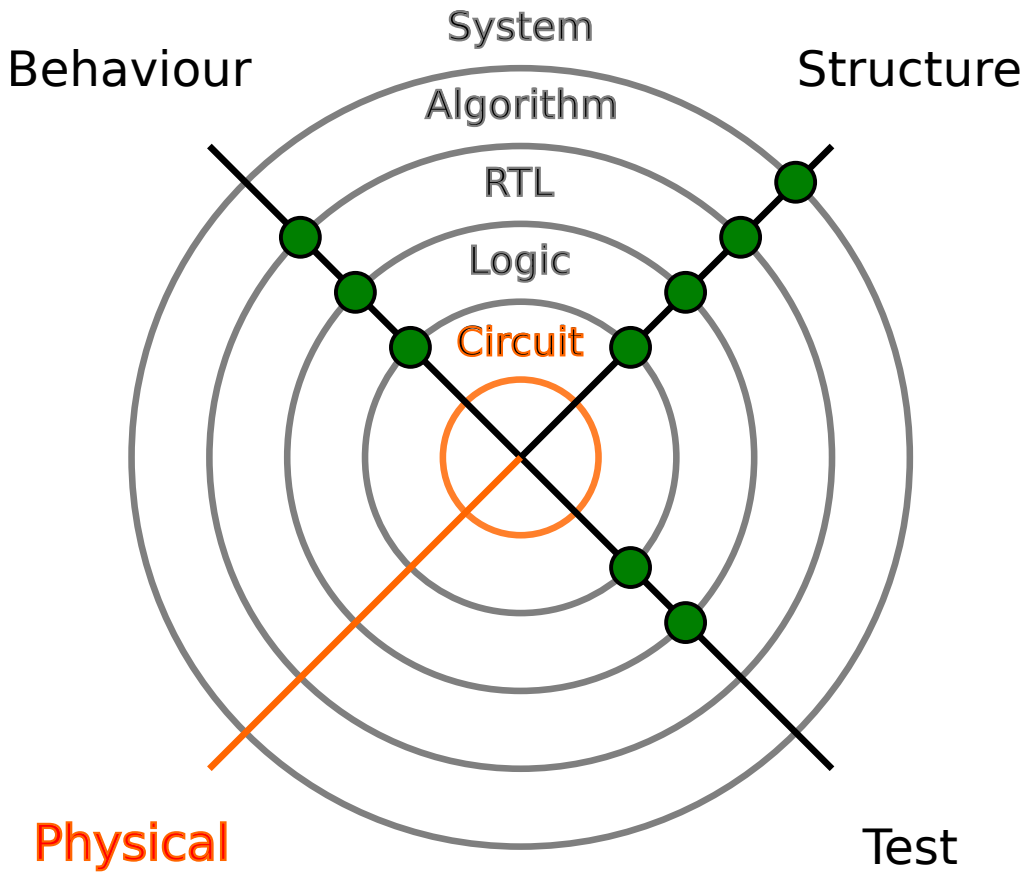
Most commonly used HDLs.
VHDL...

- ...originated from *Ada*
- ...can cover structure, behavior and some testing
- ...is consequently very complex

Verilog...

- ...is extended to *SystemVerilog*
- ...resembles *C*

The extended Gajski-Kuhn Chart



Criteria for Tool Evaluation

- The field of application [5].
- The design of human interaction
- Availability and openness
- Tool-chain integration capabilities

Specification: Why dedicated tools?

Specification is tightly linked to verification.
Verification should be done by automated tools.
→ Specification should be formalized and machine-readable.

Specification: Where are we now?

- State of the art
  - Text, tables, graphics
  - not formalized at all
- Approaches
  - try UML [1]
  - embed spec. in *VHDL* [11]
- . . . purely academic

HLS from High-level Languages

Idea

Write in an HLL, like you do with software.
Let the tools do the rest...

- *C/C++* [13]
- *Haskell* [3][12]
- *Java* $\rightarrow$ JHDL
- *Matlab* [2]

Approach: Evolutionary Algorithms

Idea

Only describe the environment conditions.
Let the algorithm figure out the best solution.

- Might work for optimization
- Not useful for new designs
- Purely academic for now. [8]

HLS by Dedicated Tools

Idea

Have a tool for each specific part of the design.
Control it by providing some parameters.

FloPoCo[4]

. . . is a generator for arithmetic cores for FPGAs

HLS from Visual Representations

Idea

Design by placing and connecting components using a GUI.
Translate the resulting layout into an HDL.

Show-off

Qucs

Editing Tool Requirements

Editing tools should help to tame the complexity of HDLs.

- Different views on design
- *Code* navigation
- *Code* completions
- Shortcuts for repeated tasks
- *Code* refactoring
- Enforcement of conventions

Show-off

SigasiPro

Also: documentation

High design complexity
→ extensive user and developer documentation required
Documentation quality impacts productivity!

Available tools
VHDLDoc, *doxygen-VHDL*, *VHDocl*

Considerations on T+V

- No tools focused on HW-design T+V
- Tool-assisted verification requires formalized spec
- Testing very much limited to
  - runtime assertions
  - waveform inspection
  - (JTAG)

State of the Art

Testing

Whatever your HDL offers you. . .

Verification

SPIN [10], HDL features

Chaining: Why bother?

One tool can not cover the whole design process.
But. . .

Behold!

One-vendor-to-rule-them-all-policies

Figure: Source: wikipedia

Difficulties when creating Toolchains

- Communication between tools in the chain → exchange formats
- Specialized tools rarely support many formats
- Chain needs to be set up manually

The true chore

Which tool is the right one?
Does my tool fit in the chain?

Exchange formats

- EDIF
  - Attempt to create industry standard (1985)
  - State: abandoned
- BLIF
  - Attempt to create academic standard (1992)
  - State: rarely used at best
- some HDL
  - limited to capabilities of the HDL

Open Source: Why bother?

- Tools can get widespread
- Closer interaction Developer $\leftrightarrow$ User
- Less effort for customization
- High number of potential developers

Disclaimer

Complex tools need high initial effort

What is out there?

Design	Qucs, fritzing
HLS	JHDL + JHDL-CAD [7]
Simulation	GHDL, FreeHDL
Below RTL	ABC [6], open HW-Platforms [14]

Suggested Improvements (by others)

- Formulation (Specification)
- Resource and Performance prediction
- Modeling techniques
- Bridge between formulation and design phase
- Reduced translation and routing times
- also see [9]

Suggested Improvements (by me)

- user interaction, usability
- content awareness
- chaining capabilities
- openness...

Some Papers I


Terry Bahill and Jesse Daniels.
Using objected-oriented and uml tools for hardware design: A case study.
Systems Engineering, 6(1):28–48, 2003.


P. Banerjee, D. Bagchi, M. Haldar, A Nayak, V. Kim, and R. Uribe.
Automatic conversion of floating point matlab programs into fixed point fpga based hardware design.
In *Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on*, pages 263–264, April 2003.


Per Bjesse, Koen Claessen, Mary Sheeran, and Satnam Singh.
Lava: Hardware design in haskell.
SIGPLAN Not., 34(1):174–184, September 1998.


Florent de Dinechin and Bogdan Pasca.
Designing custom arithmetic data paths with flopoco.
IEEE Design & Test of Computers, 28(4):18–27, 2011.


Douglas Densmore, Roberto Passerone, and Alberto Sangiovanni-Vincentelli.
A platform-based taxonomy for esl design.
IEEE Design and Test of Computers, 23(5):359–374, 2006.


Berkeley Logic Synthesis Verification Group.
ABC: A System for Sequential Synthesis.

Some Papers II

B. Hutchings, P. Bellows, J. Hawkins, S. Hemmert, B. Nelson, and M. Rytting.
A CAD suite for high-performance FPGA design.
In *Field-Programmable Custom Computing Machines, 1999. FCCM '99. Proceedings. Seventh Annual IEEE Symposium on*, pages 12–24. IEEE, 1999.

Paul Layzell.
A new research tool for intrinsic hardware evolution.
In Moshe Sipper, Daniel Mange, and Andrés Pérez-Uribe, editors, *Evolvable Systems: From Biology to Hardware*, volume 1478 of *Lecture Notes in Computer Science*, pages 47–56. Springer Berlin Heidelberg, 1998.

S.G. Merchant, B.M. Holland, C. Reardon, A.D. George, H. Lam, G. Stitt, M.C. Smith, N. Alam, I. Gonzalez, E. El-Araby, P. Saha, T. El-Ghazawi, and H. Simmler.
Strategic Challenges for Application Development Productivity in Reconfigurable Computing.
In *Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National*, pages 209–218. IEEE, 2008.

Budi Rahardjo.
Spin as a hardware design tool.
In *Proc. First SPIN Workshop*. INRS Quebec, Canada, 1995.

R. Reetz, K. Schneider, and T. Kropf.
Formal specification in VHDL for hardware verification.
In *Design, Automation and Test in Europe, 1998., Proceedings*, pages 257–263. IEEE, 1998.

Some Papers III

Mary Sheeran.

Hardware design and functional programming: a perfect match.
Journal of Universal Computer Science, 11(7):1135–1158, jul 2005.
|http://www.jucs.org/jucs_11_7/hardware_design_and_functional—.

S. Vernalde, P. Schaumont, and I Bolsens.

An object oriented programming approach for hardware design.
In *VLSI '99. Proceedings. IEEE Computer Society Workshop On*, pages 68–73, 1999.

Aaron Weiss.

Open source hardware: Freedom you can hold?
netWorker, 12(3):26–33, September 2008.