Máster en Sistemas Electrónicos Avanzados (MSEA)

Co-simulación y verificación funcional con VHDL, C/C++ y Python/m

{main}

Unai Martinez Corral

unai.martinezcorral@ehu.eus umarcor umarcor

Escuela de Ingeniería de Bilbao
Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU)

2021/05

Academic background

- 2013. *Ingeniería Técnica Industrial, Esp. Electrónica Industrial*. Escuela de Ingeniería Técnica Industrial (EUITI) in Bilbao. University of the Basque Country (UPV/EHU).
- 2015. *Máster en Sistemas Electrónicos Avanzados (SIEAV)*. Doctoral School, Faculty of Engineering in Bilbao. University of the Basque Country (UPV/EHU).
- Since 2015. *Doctoral Programme in Engineering Physics*, advised by Prof. Koldo Basterretxea. Digital Electronics Design Group (GDED). University of the Basque Country (UPV/EHU).
- 2018. *Visiting Researcher*, advised by Prof. Mikel Luján. Advanced Processor Technologies Research Group (APT). University of Manchester.

Open Source background

- Organisations in GitHub:

Admin/owner	*Co-maintainer*	*Member*
HDL	GHDL	SymbiFlow
VHDL	VUnit	verilator
DBHI		ITSAS

- *Secretary* of the VHDL Analysis and Standardisation Group (VASG), aka IEEE-P1076.

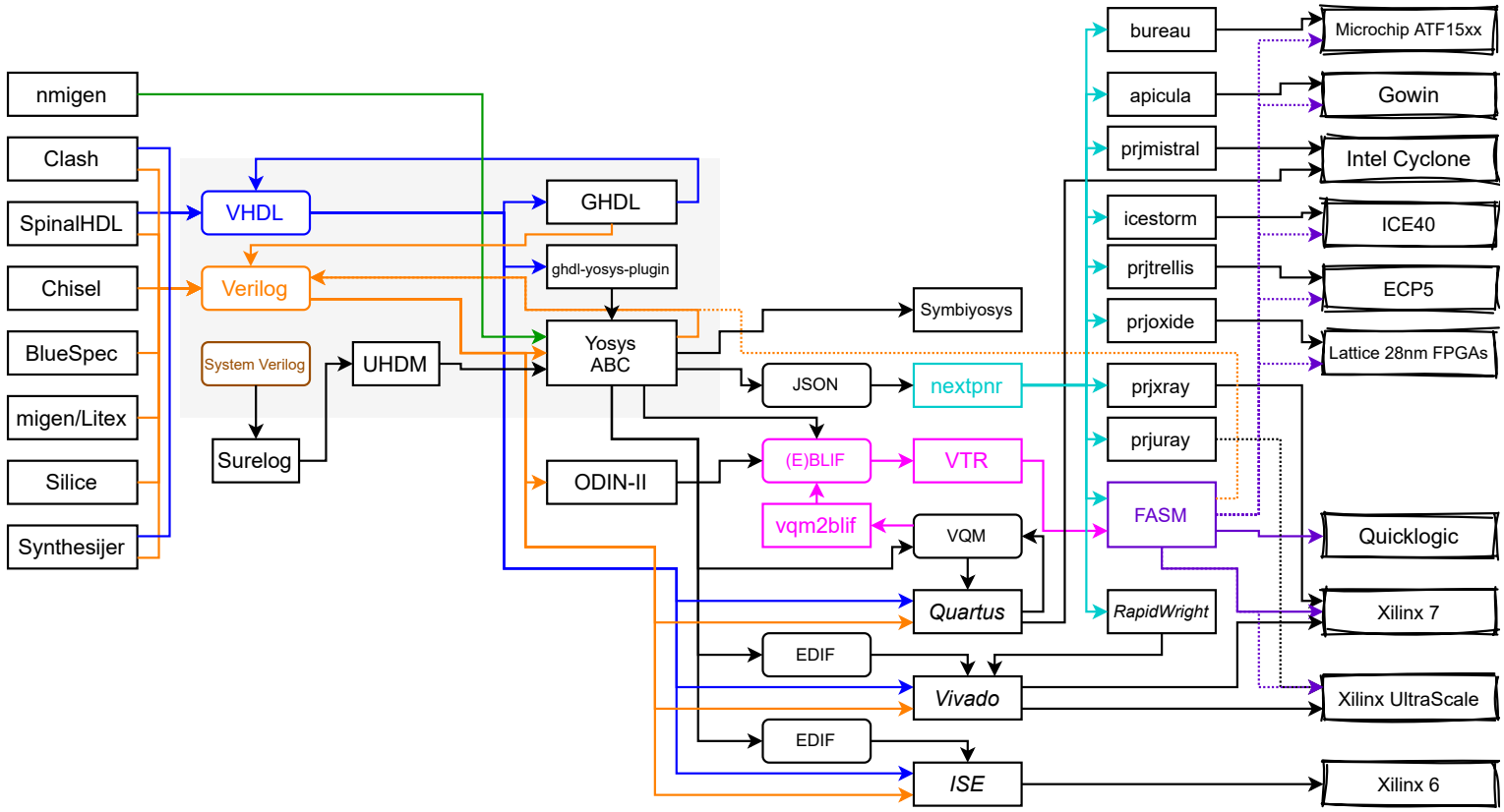
- Contributor to MSYS2, GTKWave, Verilog-to-Routing, Microwatt, I'm TOMU/FOMU, Cobra, etc.

- Awarded the Google Open Source Peer Bonus in Q1 2021.

Software /= Hardware

Software ?= Hardware

EDA tooling ecosystem(s)



hdl/awesome#98

Note: Synthesis flows targeting FPGA bitstreams are shown only. Simulation/verification and ASIC targets are not included in this diagram.

See also: Rodrigo A. Melo. *Free and Open Source Software for FPGA development*. February 2021.

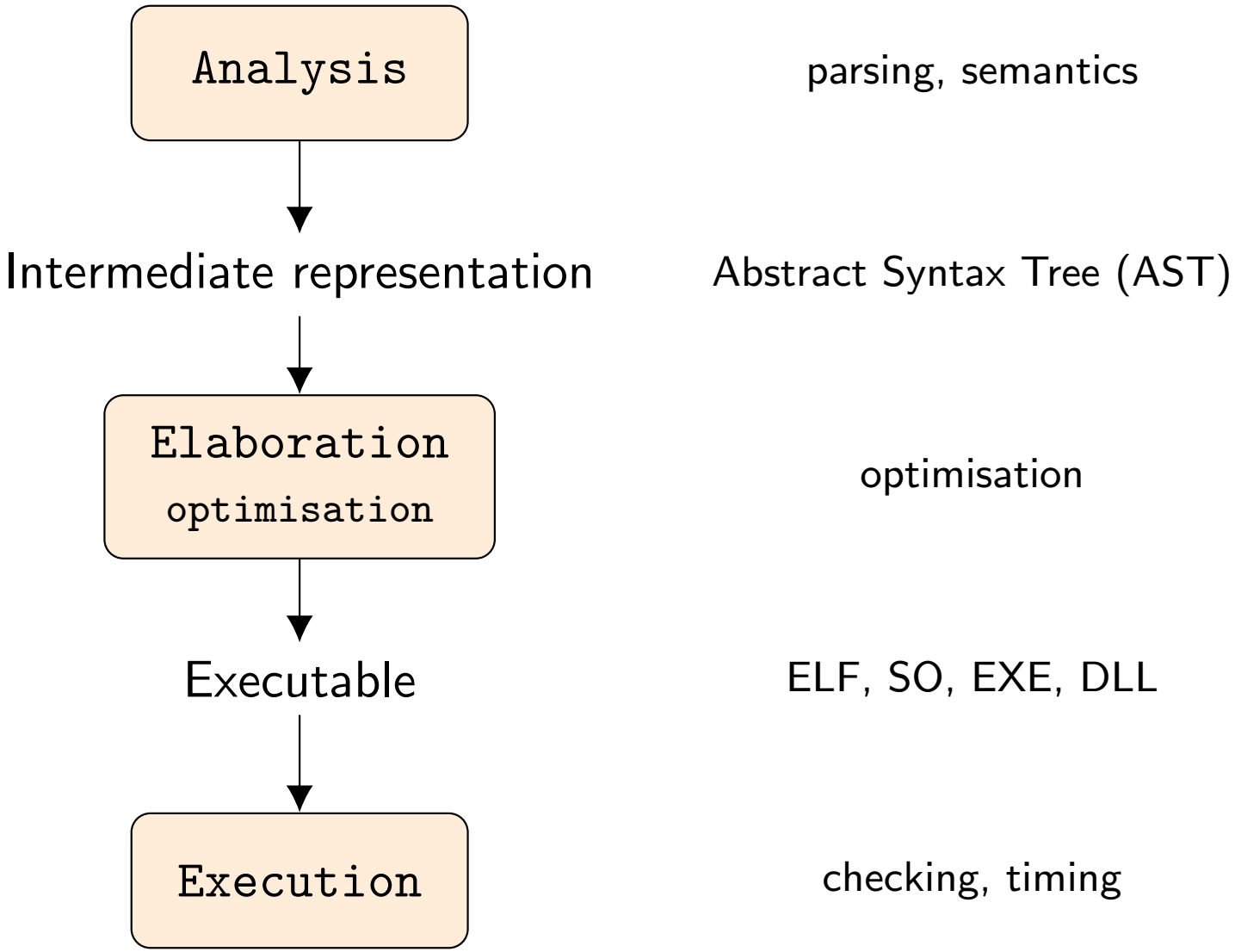
Languages

- Dynamic/static
- Interpreted/compiled
- Strong/loose typing
- Runtime
- Memory management (GC)
- Concurrency/parallelism

- Assembly/ASM
- Ada, C/C++, Java, C#
- go(lang), Rust
- Python, m, Julia, Perl, Ruby
- JavaScript, TypeScript
- shell/bash, powershell, cmd
- LaTeX/Tikz
- MarkDown, reStructuredText
- HTML, CSS/SASS
- ...

W Comparison of programming languages

Languages: compilation and execution



Languages: unified compilation tools

- GNU Compiler Collection (GCC)
- Low-level virtual machine (LLVM); despite its name, LLVM has little to do with traditional virtual machines.

- Java virtual machine (JVM)
- Dynamic Binary Modification/Translation (QEMU)
- WebAssembly

IEEE Hardware Description Languages

- 1076: **VHDL** (1987, 1993, 2000, 2002, 2008, 2019)
- 1076.1: **VHDL-AMS** (1999, 2007, 2017)
- 1364: **Verilog** (1995, 2001 2005)
- 1800: **SystemVerilog** (2005, 2009, 2012, 2017)
- 1850: Property Specification Language (**PSL**) (2005, 2010)
- 1666: **SystemC** (2005, 2011, 2016)

standards.ieee.org

HDL generators

- High-Level Synthesis (HLS) W
- Bluespec SystemVerilog (BSV) git
- Scala: Chisel, SpinalHDL
- Python: MyHDL, Migen, nmigen, Litex
- Haskell: Clash
- Java: Synthesijer
- Silice
- ...

- Flexible Intermediate Representation for RTL (FIRRTL)

HDL simulation

- Mentor Graphics/Siemens, Intel-Altera: ModelSim/QuestaSim
- Aldec: Active-HDL/Riviera-PRO
- Xilinx: ISE ISIM, Vivado XSIM
- Cadence: Incisive, Xcelium
- Synopsys: VCS

W List of HDL simulators

Circuit simulation: open source

HDL simulation:

- GHDL
- nvc
- Verilator
- Icarus Verilog (iverilog)

Analog simulation:

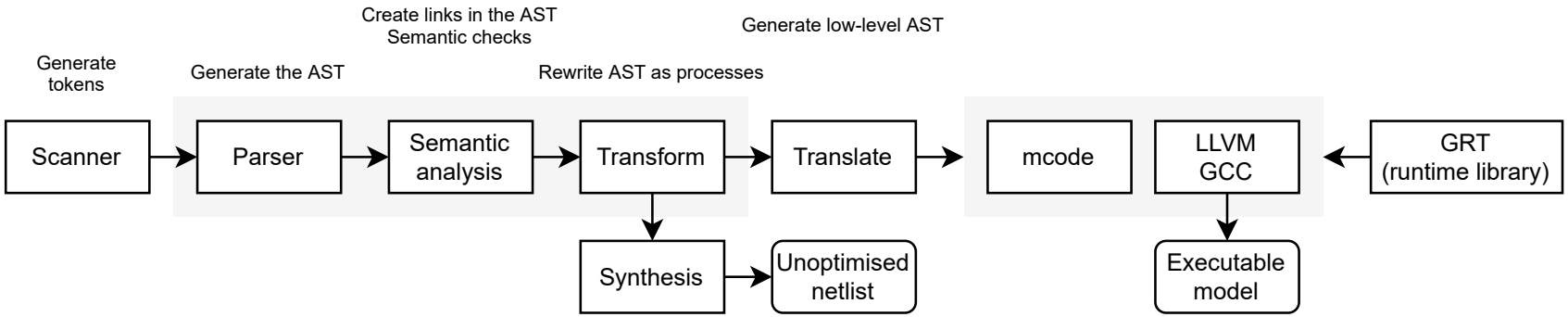
- Xyce
- SPICE
- Ngspice
- Gnucap

Waveform visualisation:

- GtkWave (LXT, LXT2, VZT, FST, GHW, VCD/EVCD)
- dwfv (VCD)

HDL simulation: GHDL

Free and open-source analyser, compiler, simulator and experimental synthesiser for VHDL. GHDL is not an interpreter: it allows to generate machine code from design sources.



- Full support for the 1987, 1993, 2002 versions of VHDL, and partial for 2008. Partial support of PSL.
- Three backends: LLVM, GCC or, x86_64/i386 only, a built-in one (mcode).
- GNU/Linux, Windows and macOS; on x86, x86_64, armv6/armv7/aarch32 and aarch64.
- Can write waveforms (GHW, VCD or FST).
- Synthesis: plain VHDL or yosys (through ghdlsynth).

HDL co-simulation

- Indirect co-simulation:
  - Verilog Procedural Interface (**VPI**), also known as Program Language Interface (**PLI**) 2.0.
  - VHDL Procedural Interface (**VHPI**).
- Direct co-simulation:
  - Specific implementations of (a draft of) VHPIDIRECT, such as Foreign Language Interface (**FLI**) or Xilinx Simulation Interface (**XSI**).
  - Direct Programming Interface (**DPI**).
- Generation of C/C++ models/sources through a transpiler.
  - *Verilated* models.
  - CXXRTL (Yosys).