

Máster en Sistemas Electrónicos Avanzados (MSEA)
Co-simulación y verificación funcional con
VHDL, C/C++ y Python/m

Unai Martinez Corral

 unai.martinezcorral@ehu.eus  umarcor

Dpto. Tecnología Electrónica
Grupo de Investigación de Diseño en Electrónica Digital (GDED)
Escuela de Ingeniería de Bilbao
Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU)

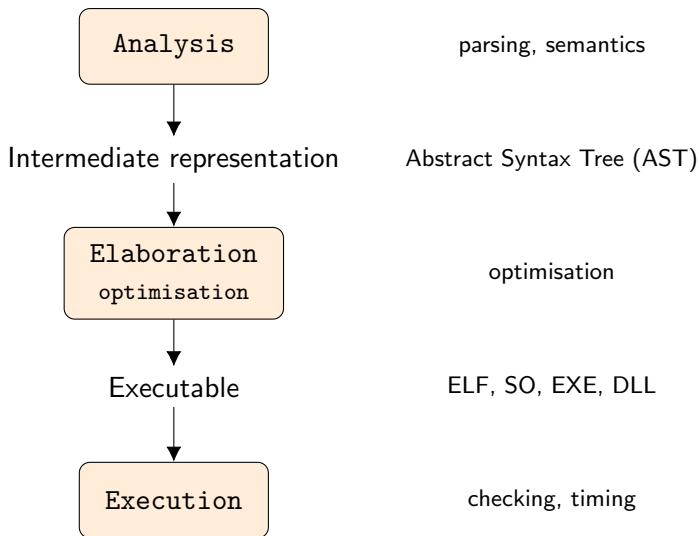
2020/03

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) [W](#)
- ▶ Dynamic Binary Modification/Translation (QEMU) 🌐
- ▶ WebAssembly 🌐 [W](#)

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 [🌐](#) [🐙](#)
 - ▶ Haskell: Clash [🌐](#) [🐙](#)
 - ▶ ...
-
- ▶ 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
- ▶ Synopsys: VCS

W List of HDL simulators

Circuit simulation: open source

HDL simulation:

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

Analog simulation:

- ▶ Xyce  
- ▶ SPICE 
- ▶ Ngspice 
- ▶ Gnuicap 

Waveform visualisation:

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

HDL simulation: GHDL

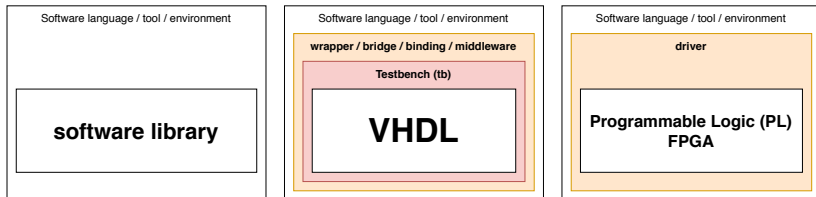
Free and open-source analyser, compiler, simulator and experimental synthesiser for VHDL. GHDL is not an interpreter: it allows to generate machine code from design sources.

- ▶ 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).
- ▶ ghdl-ls implements Language Server Protocol (LSP) in Python.
- ▶ Synthesis: plain VHDL or yosys (through ghdl-synth).

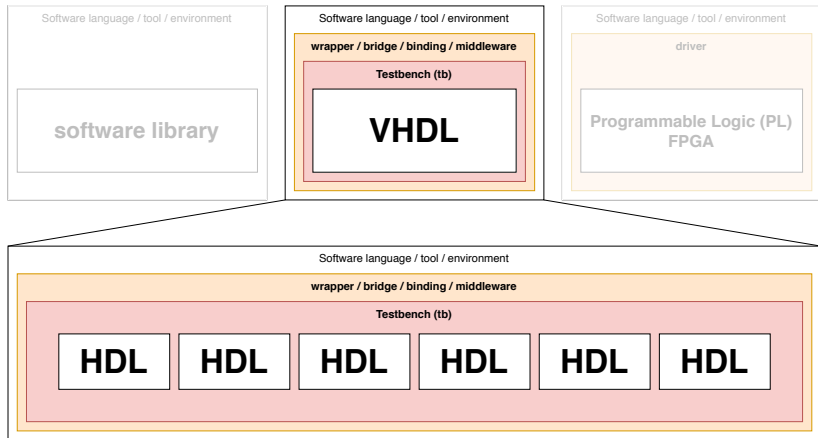
HDL co-simulation

- ▶ Verilog Procedural Interface (**VPI**), also known as Program Language Interface (**PLI**) 2.0.
- ▶ VHDL Procedural Interface (**VHPI**), or specific implementations, such as Foreign Language Interface (**FLI**).
- ▶ Generation of C/C++ models/sources through a transpiler.

Functional verification of an HDL design



Functional verification of a non-trivial HDL design



VHDL simulation: libraries/frameworks for verification

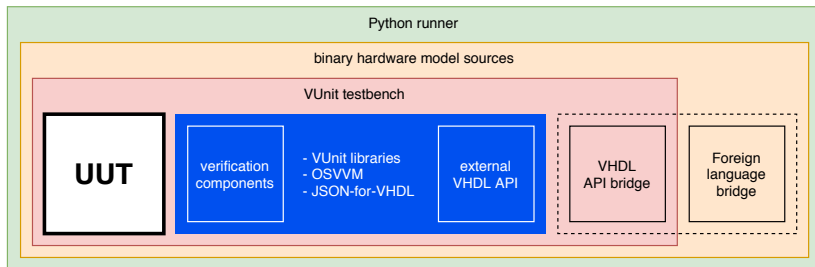
- ▶ UVM: Universal Verification Methodology [W](#)
- ▶ OSVVM: Open Source VHDL Verification Methodology [🌐](#) [🐙](#)
- ▶ UVVM: Universal VHDL Verification Methodology [🌐](#) [🐙](#)
- ▶ cocotb: Coroutine Co-simulation Test Bench [🐙](#) [📖](#) [🌐](#) [</>](#)
- ▶ VUnit: unit testing framework [🐙](#) [📖](#) [</>](#)
 - ▶ VUnit/cosim [🐙](#) [📖](#)

HDL simulation: VUnit

Open source unit testing framework for VHDL/SystemVerilog. It features the functionality needed to realize continuous and automated testing of your HDL code. VUnit complements traditional testing methodologies by supporting a *test early and often* approach through automation.

- ▶ Supported languages: VHDL (93, 2002, 2008, 2019), Verilog, SystemVerilog
- ▶ Supported simulators: GHDL, Aldec Riviera-PRO/Active-HDL, Mentor Graphics ModelSim/Questa and Cadence Incisive (experimental)
- ▶ Requires Python ≥ 3.6 : Python Interface and CLI
- ▶ Supported on Windows, GNU/Linux and macOS.
- ▶ VHDL libraries: OSVVM, JSON-for-VHDL, Run, Check, Logging, Communication, Verification Components, etc.
- ▶ Data Types with an external API for co-simulation

VUnit: overview



Work environment

- ▶ GHDL (LLVM or GCC backends)
- ▶ Python ≥ 3.6
- ▶ Editor: Visual Studio Code (VSC), vim, emacs, Sigasi...
- ▶ GtkWave
- ▶ Language server: ghdl-ls, rust_hdl...

Exercises: introduction

GHDL Quick Start Guide

- ▶ Hello World 

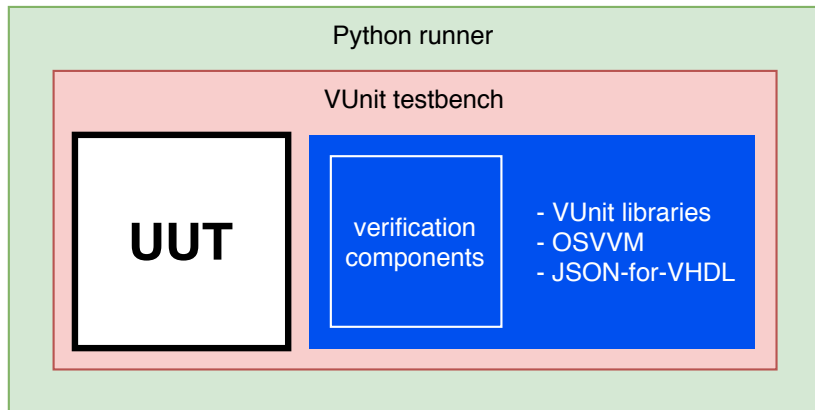
- ▶ Heartbeat 

- ▶ Full-adder 





VUnit User Guide

- ▶ Add run.py to Full-adder

VUnit: tutorial



Exercises: libraries

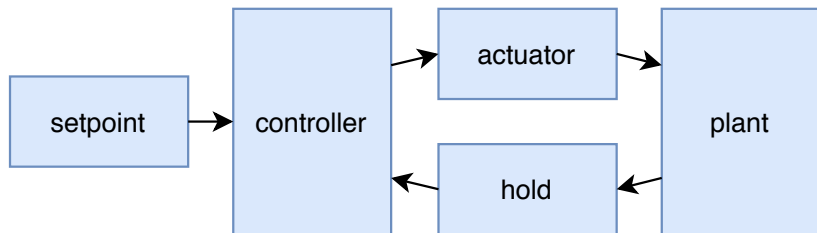
- ▶ Run  `</>`
- ▶ Logging  `</>`
- ▶ Check  `</>`
- ▶ Communication  `</>`
- ▶ OSVVM
 - ▶ array `</>`
- ▶ JSON-for-VHDL
 - ▶ json4vhdl `</>`
 - ▶ composite_generics `</>`

Exercises: verification components

Verification Component Library (VCL)

- ▶ `uart`
- ▶ `array_axis_vcs`
- ▶ `axi_dma`

Exercises: VHPIDIRECT



Exercises: co-simulation

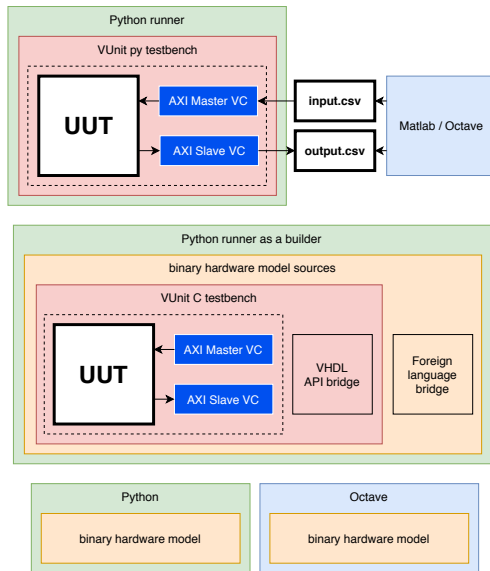
VUnit External VHDL API 

VUnit/cosim 

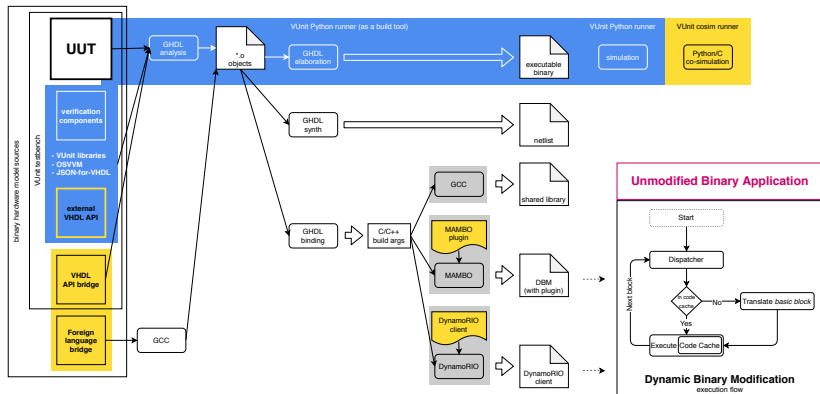
▶ copy  `</>`

▶ buffer  `</>`

Exercises: AXIS co-simulation



Dynamic Binary Hardware Injection (DBHI)



[dbhi.github.io](https://github.com/dbhi)

Other open source projects

Project management:

- ▶ tsfpga   </>
- ▶ fusesoc   </>
- ▶ edalize   </>
- ▶ litex 
- ▶ duh 







Waveform viewer/drawer:

- ▶ wavedrom  

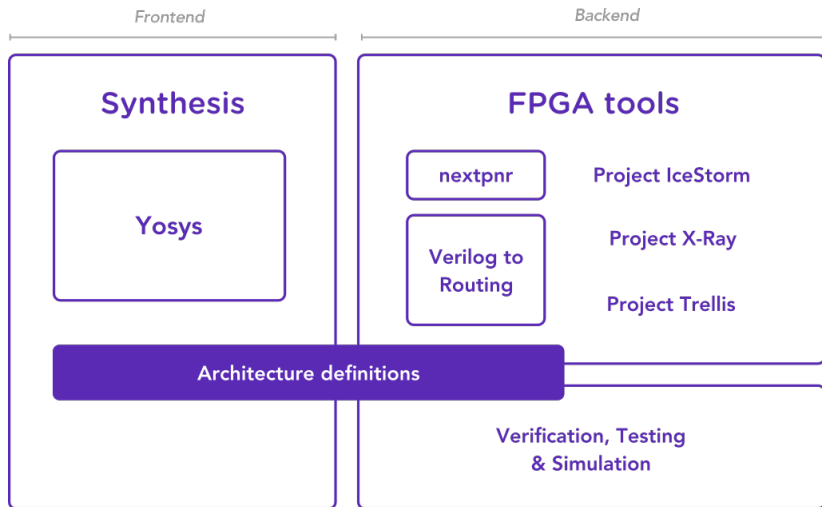
Android:

- ▶ termux  
- ▶ gcc_termux 

Formal verification:

- ▶ ghdl synth-beta 
- ▶ yosys  
- ▶ nextpnr 
- ▶ SymbiYosys  

SymbiFlow



symbiflow.github.io