Máster en Sistemas Electrónicos Avanzados (MSEA)

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

{cosim}

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

VHDL co-simulation with GHDL

ghdl.github.io/ghdl-cosim

- Indirect co-simulation:
  - Partial Verilog Procedural Interface (**VPI**), also known as Program Language Interface (**PLI**) 2.0.
  - ~~VHDL Procedural Interface (**VHPI**)~~.
    There is AVHPI, and work in progress by Marlon James for adding VHPI support.
- 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**)~~.
    The VASG expects to standardize a direct interface (VHDPI or VHFFI) in the next revision of the standard (202X).

Direct VHDL co-simulation with GHDL

- Type declarations
- Linking object files
- Notebook: Common mistakes

Exercises:

- 'rand' from stdlib </>
- 'sin' from libmath </>
- custom C </>

Direct co-simulation with GHDL: wrapping

- Wrapping a simulation
- How to use GHDL from an external C program?

Exercises:

- basic </>
- time </>
- exitcb </>
- Command-Line Arguments </>
- Setting parameters in C through VHDL generics </>

Direct co-simulation with GHDL: arrays and matrices

- Constrained/bounded integer arrays
- Constrained multidimensional arrays of doubles/reals

Exercises:

- Vector of std_logic </>
- Array and AXI4 Stream Verification Components </>

Direct co-simulation with GHDL: shared/dynamic loading

- Dynamic loading

Exercises:

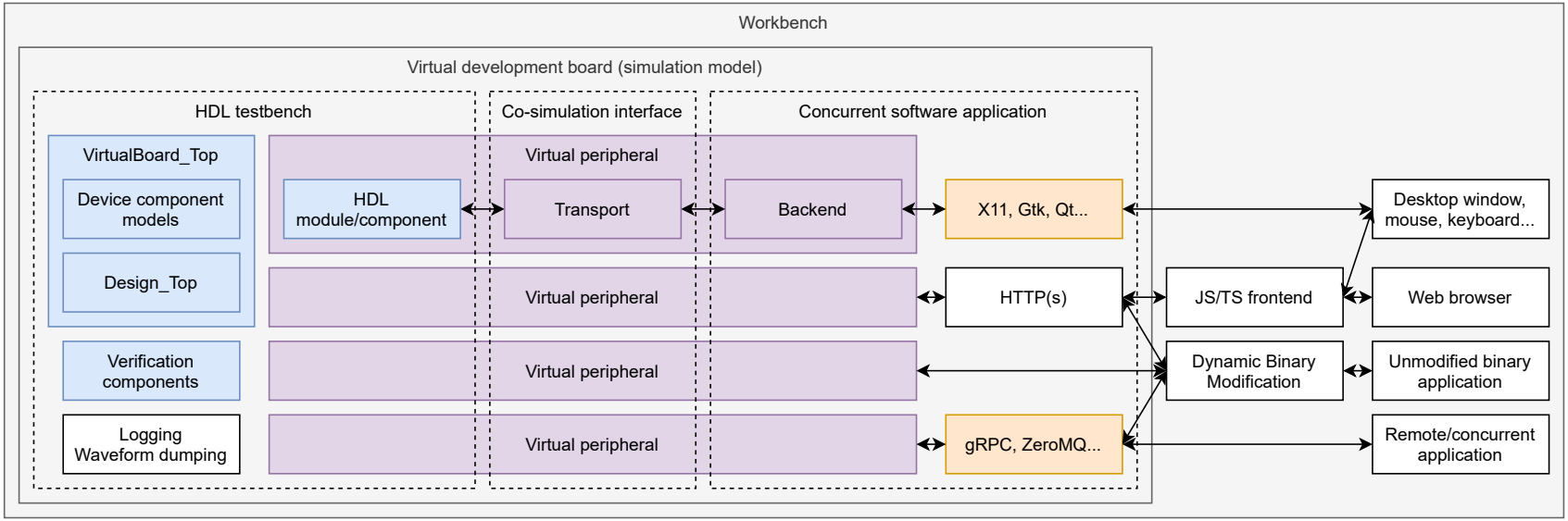
- shlib </>
- dlopen </>
- shghdl </>
- py </>
- py/vunit </>
- pycb </>

Standard Direct VHDL interface proposal (VHDPI/VHFFI)

The VHDL Analysis and Standardization Group (VASG) is working towards defining a direct co-simulation interface in the next revision of the standard. The interface is inspired on GHDL's direct interface, FLI, XSI and System Verilog's DPI. In fact, one of the original motivations is allowing standardized VHDL and System Verilog co-simulation.

- IEEE-P1076/VHDL-Issues#10
- P1076/DpiProposal
- umarcor.github.io/ghdl-cosim/VHDL202x
- VHDL/Compliance-Tests: cosim

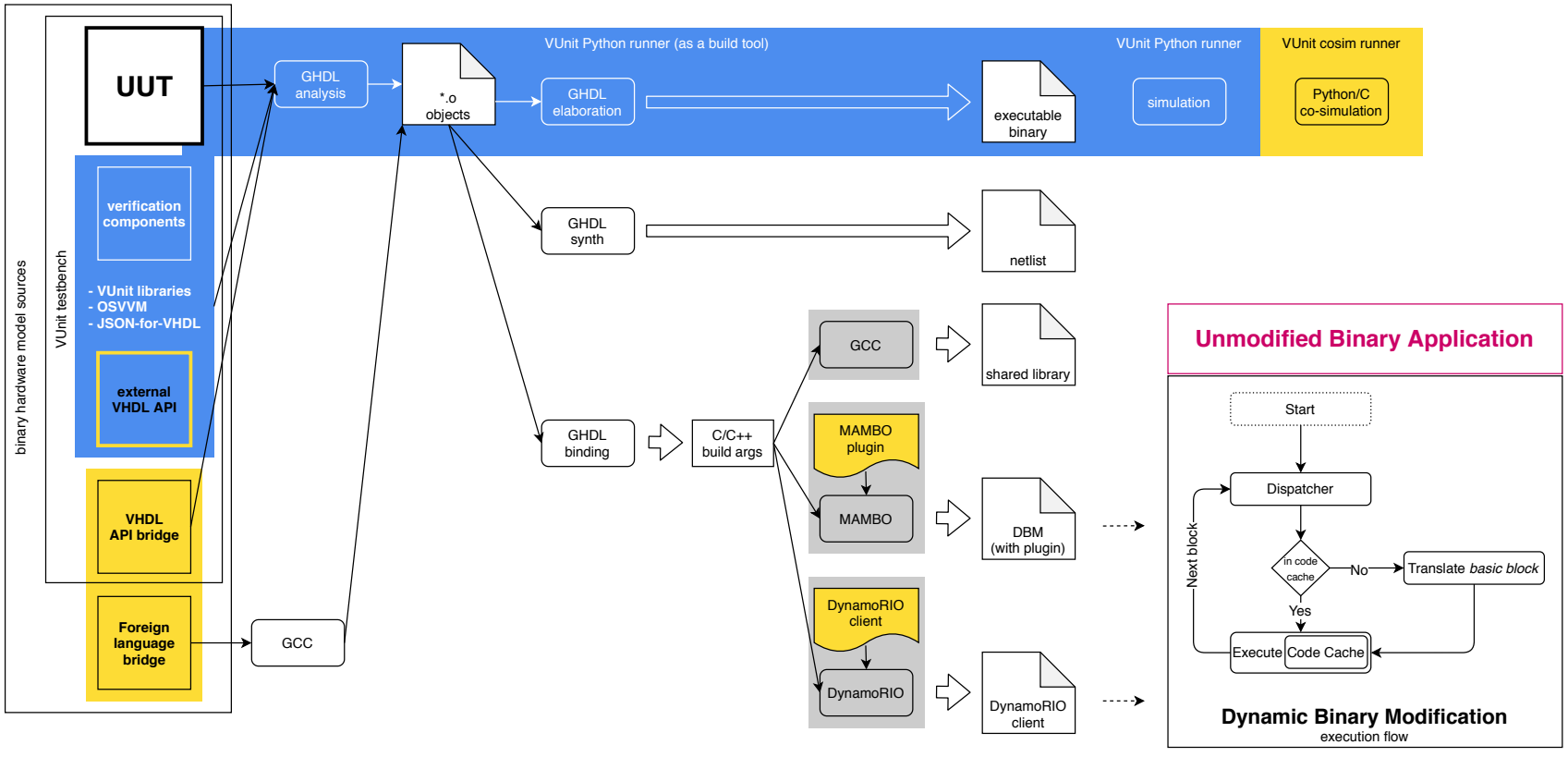
Virtual development board for HDL design



dbhi/vboard

</> dbhi/vboard: vga

Dynamic Binary Hardware Injection (DBHI)



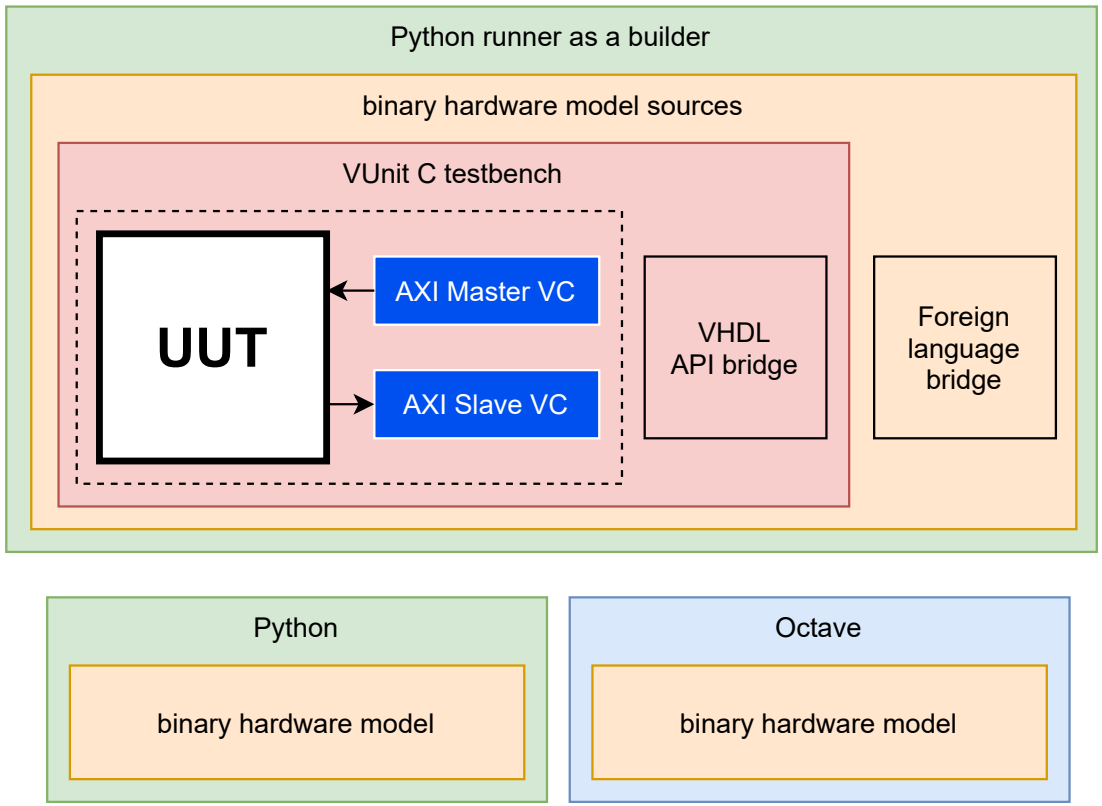
dbhi.github.io

VUnit's external API

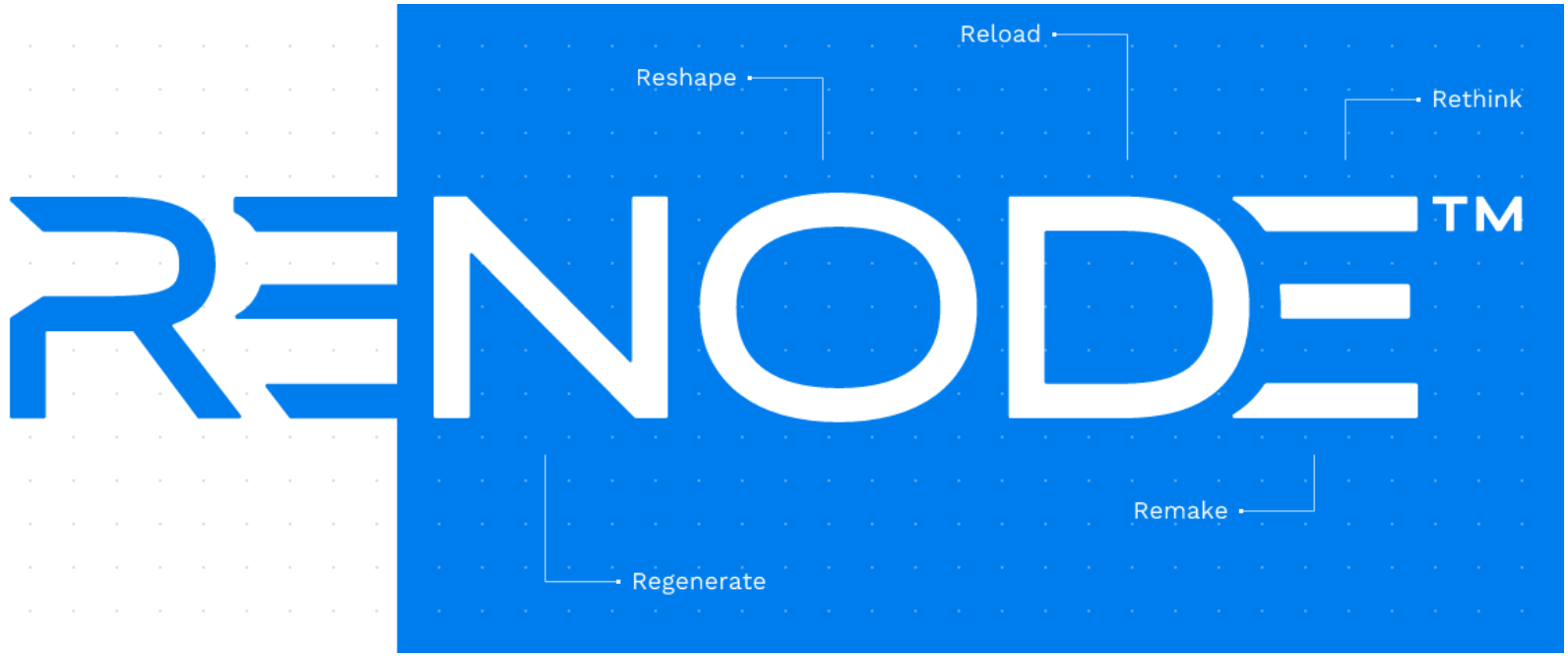
- VUnit External VHDL API
- VUnit/cosim

Exercises:

- copy </>
- buffer </>



Renode by Antmicro



renode.io

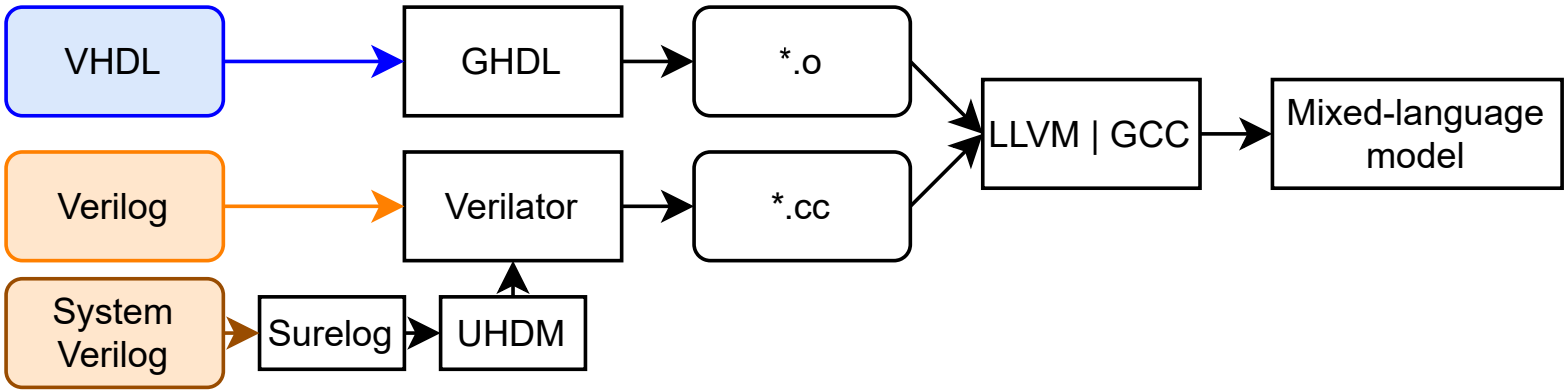
Indirect co-simulation with GHDL: VPI

</> umarcor/osvb: fpconv

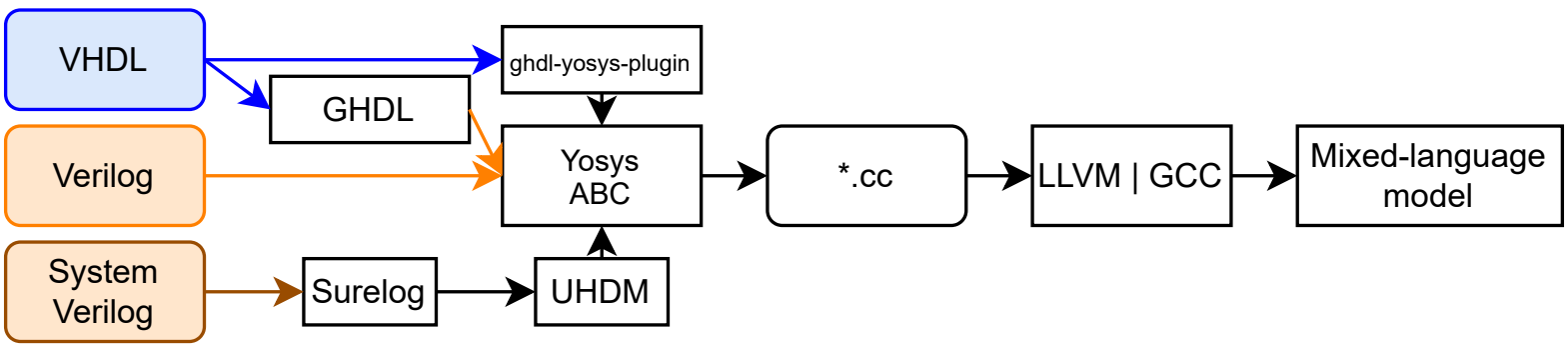
 umarcor.github.io/osvb/notebook/fpconv

Mixed-language co-simulation

- *Verilated* models



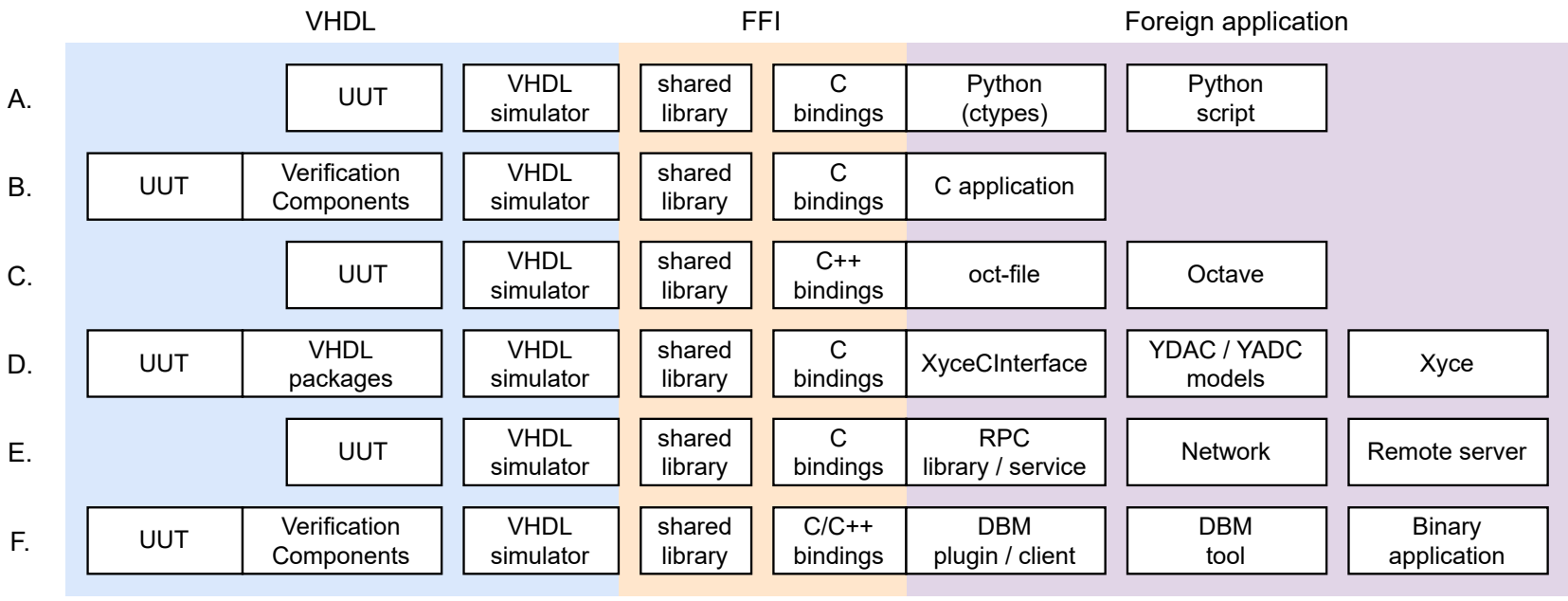
- CXXRTL (Yosys)



Mixed-signal co-simulation

- ghdl-cosim: vhpidirect/examples/vffi_user/Xyce

Use cases


VHDL
FFI
Foreign application
A.
UUT
VHDL simulator
shared library
C bindings
Python (ctypes)
Python script
B.
UUT
Verification Components
VHDL simulator
shared library
C bindings
C application
C.
UUT
VHDL simulator
shared library
C++ bindings
oct-file
Octave
D.
UUT
VHDL packages
VHDL simulator
shared library
C bindings
XyceCInterface
YDAC / YADC models
Xyce
E.
UUT
VHDL simulator
shared library
C bindings
RPC library / service
Network
Remote server
F.
UUT
Verification Components
VHDL simulator
shared library
C/C++ bindings
DBM plugin / client
DBM tool
Binary application