


FACULTY OF **ENGINEERING**

Building a better VHDL testing environment

Joren Guillaume

FEA
Ghent University

Prelimenary Presentation

Outline

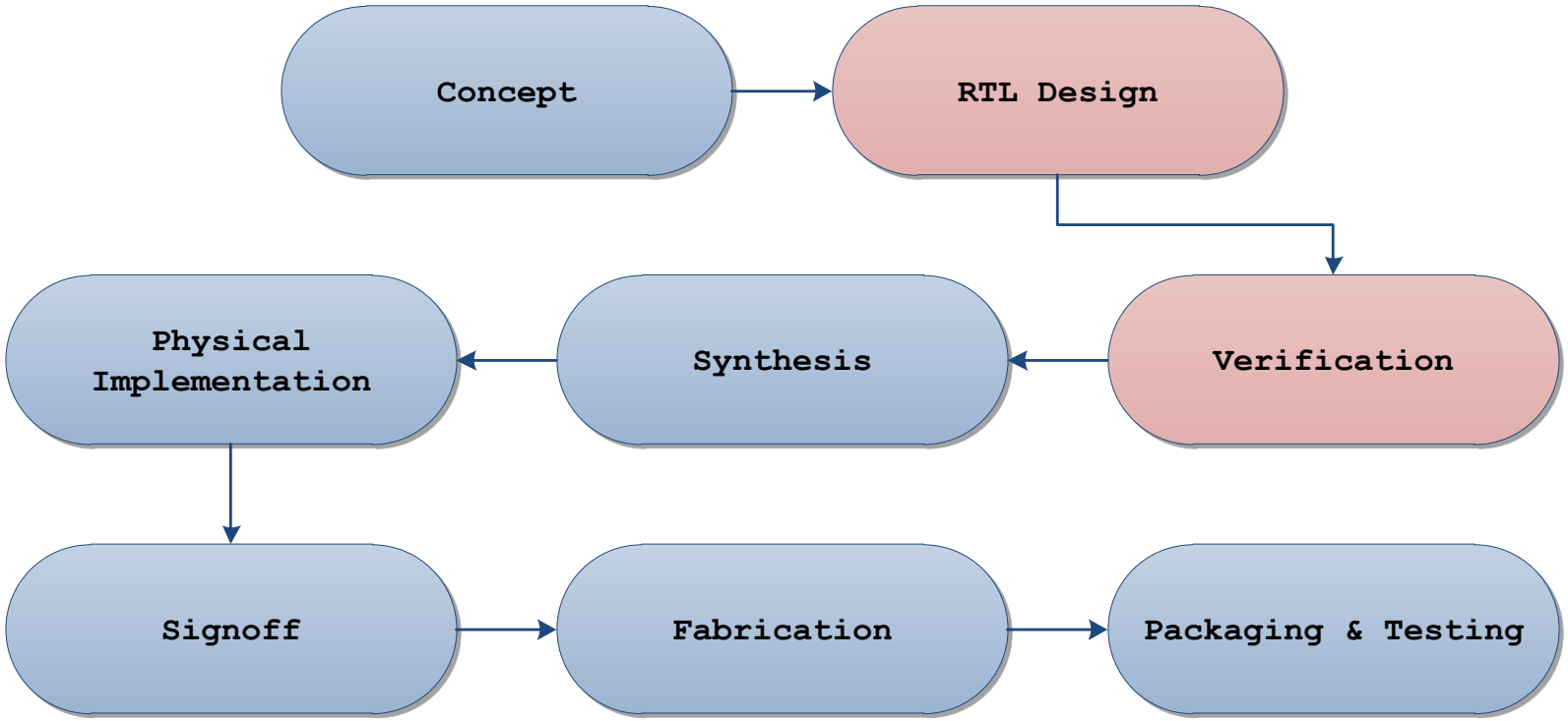
Outline

VHDL

VHDL

- VHSIC Hardware Description Language
- Used for describing digital and mixed-signal systems
- Developed by U.S. Department of Defense
  - Document → Simulate → Synthesize

VHDL - design flow



Outline

Testing VHDL

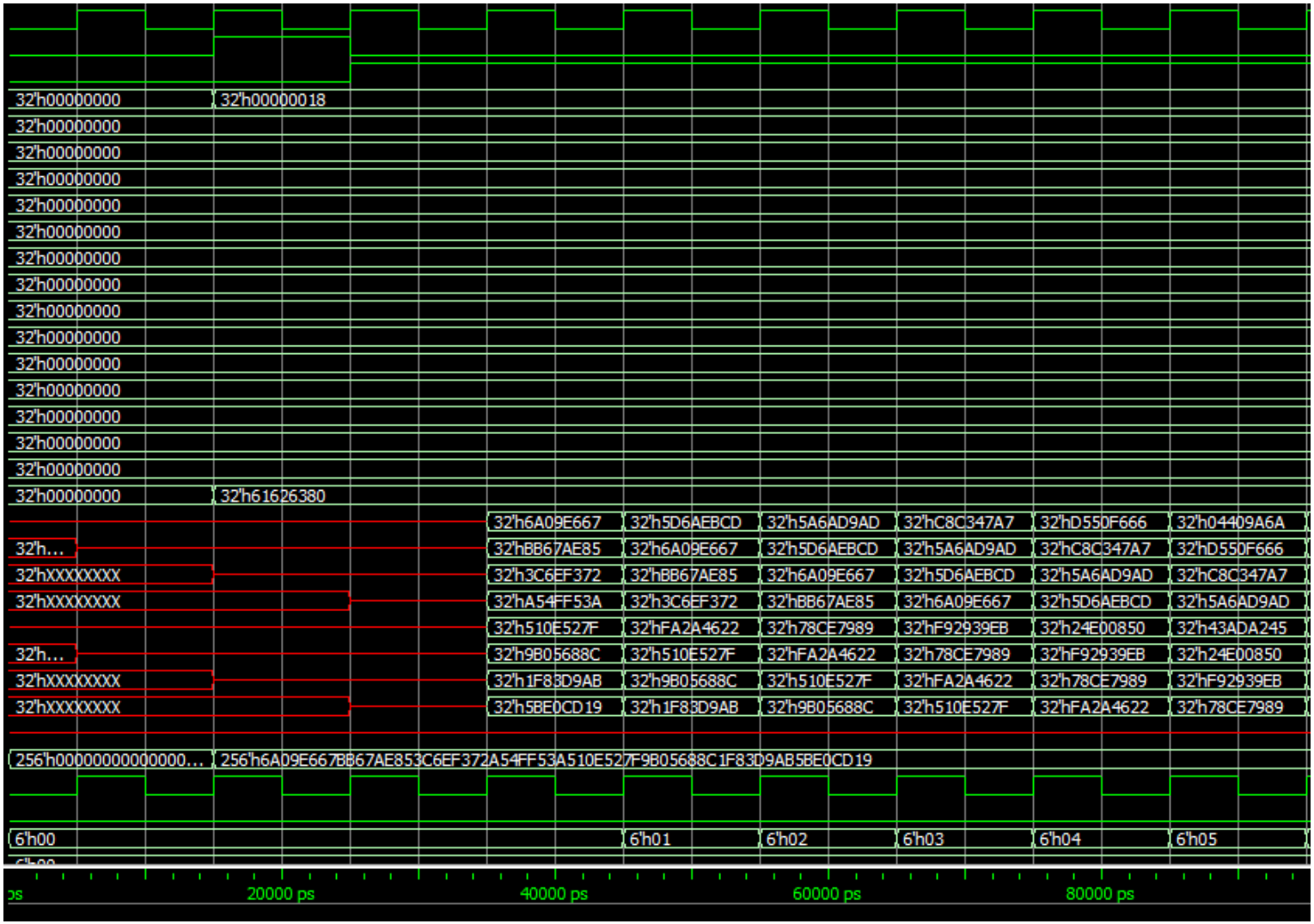
Testbenches

- Unit Under Test (UUT)
- Signal drivers, stimuli & processes
- Assertions and output tracking
  - Comparison to "golden reference"
  - Manual check
  - Wave-check

Problems

- Non-standardized process
- Single point of failure

Modelsim - waves



Outline

VHDL testing framework

Standardized testing framework

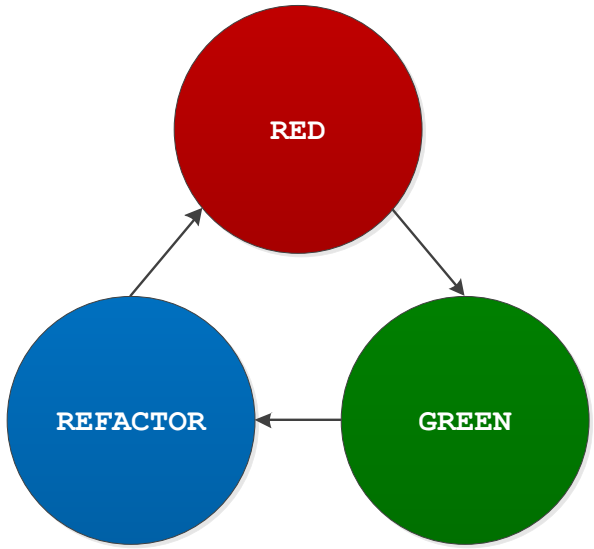
- Based on Test Driven Development (TDD)
- Cross platform
- Utility library
  - Standardized testbenches
  - Swift coding
- Script-based processing → Standardized processing & output
- Continuous Integration (CI) system

Outline

Test Driven Development

Test Driven Development

- Software development technique
- Proven to significantly reduce errors
- All behaviour is tested
- Unit testing & short development cycle
- Red - Green - Refactor



Outline

Specialized library

Bitvis utility library

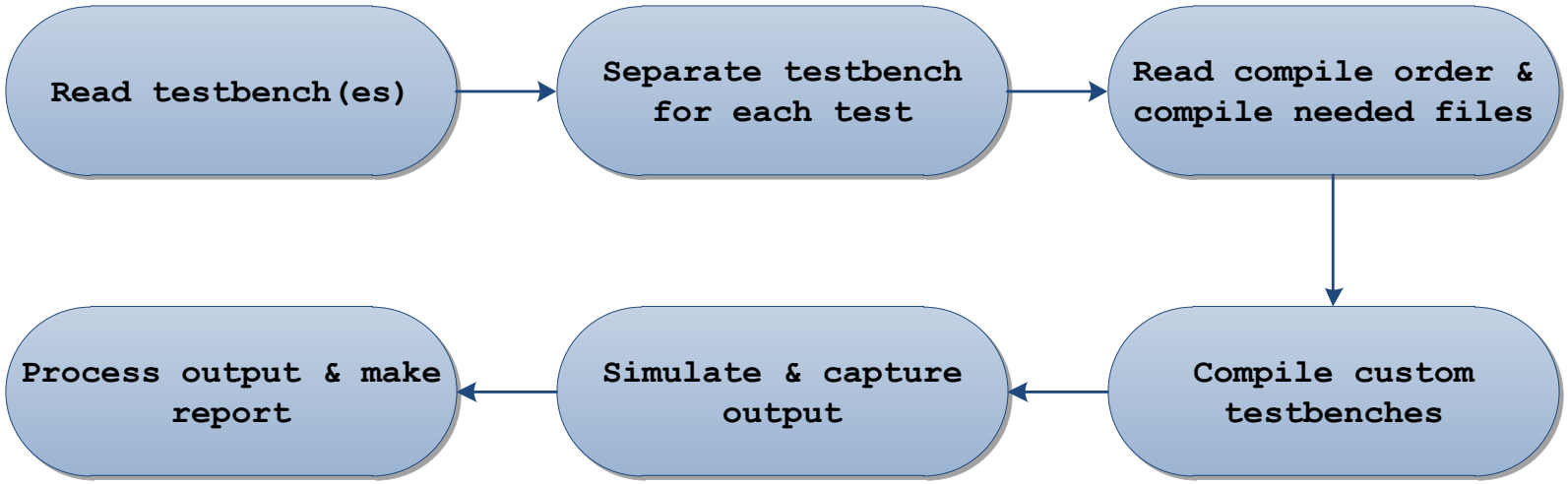
- Expands VHDL functions
  - Easy value checking
  - Clock & pulse generators
  - String handling & random generation
  - Easy output logging
- Quick & uniform coding
- Compatible with all VHDL versions

UNIVERSITEIT GENT

Outline

Script-based processing

Specialized python script



Script-based processing

Features

- Standalone functions
- Customizable process
- Text & JUnit format reports
- Automated file cleanup

Outline

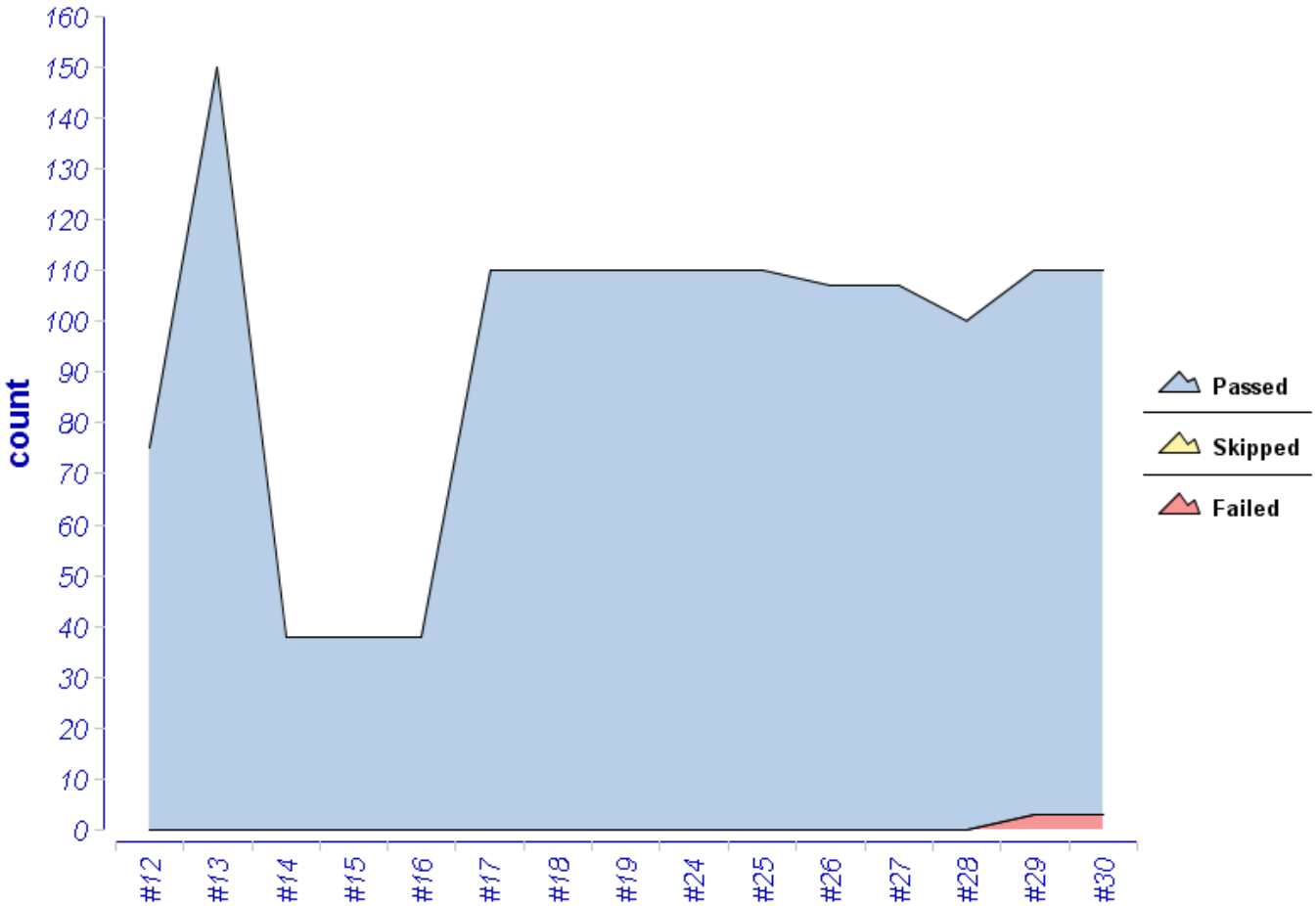
Continuous Integration

Hudson-CI

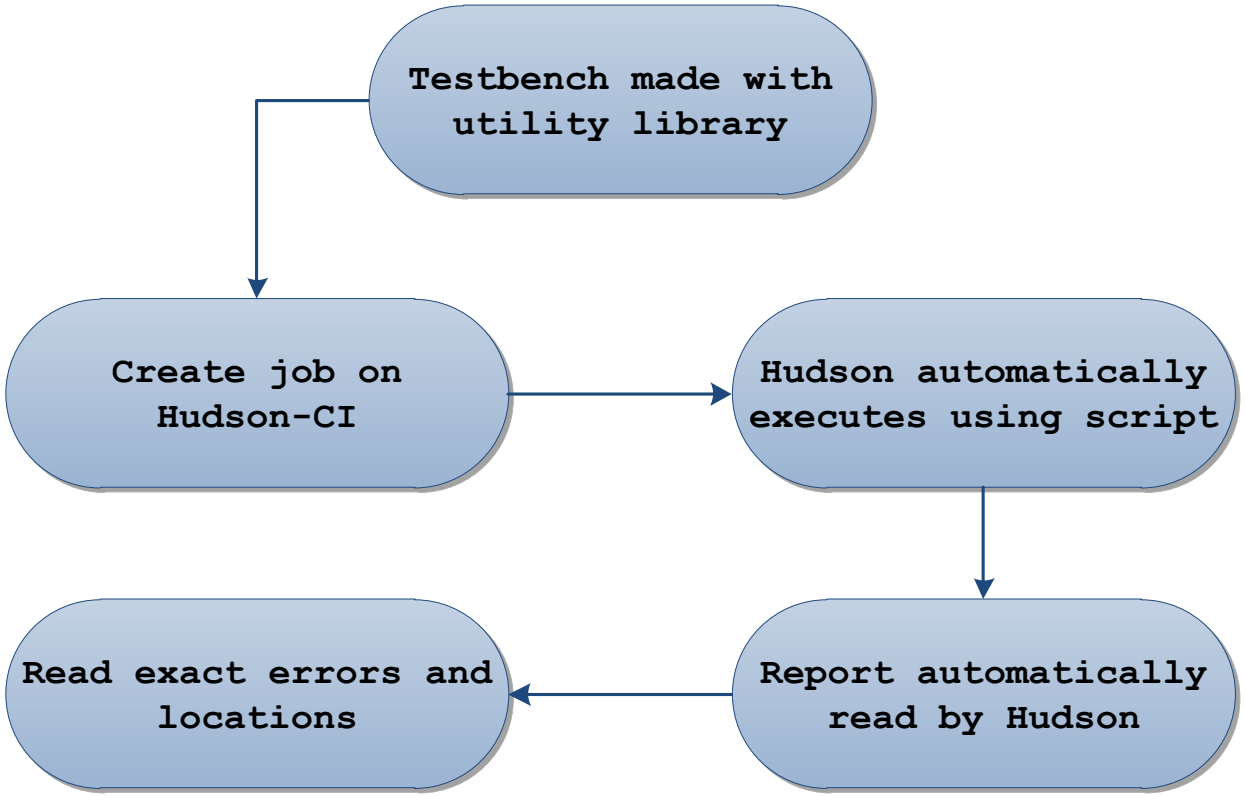
- Centralized, automated testing
- Revision control integration (e.g. Git)
- Statistics on success
- Standardized test reports (JUnit)
- Very customizable

Hudson interface

Hudson statistics



Framework design flow



Outline

Problems

- VHDL has no reflection
  - Circumvent with higher level language
- Code duplication increases compile & simulation time
  - Implement regression testing
  - Smart code filtering
- Simulation is not synthesis
  - Wait statements, wrong sensitivity list ...

Outline

Future work

- Improving base script
  - Better integration utility library
  - More options
- Smart code analysis
- Regression testing
- Proper documentation & examples

Outline

Demo