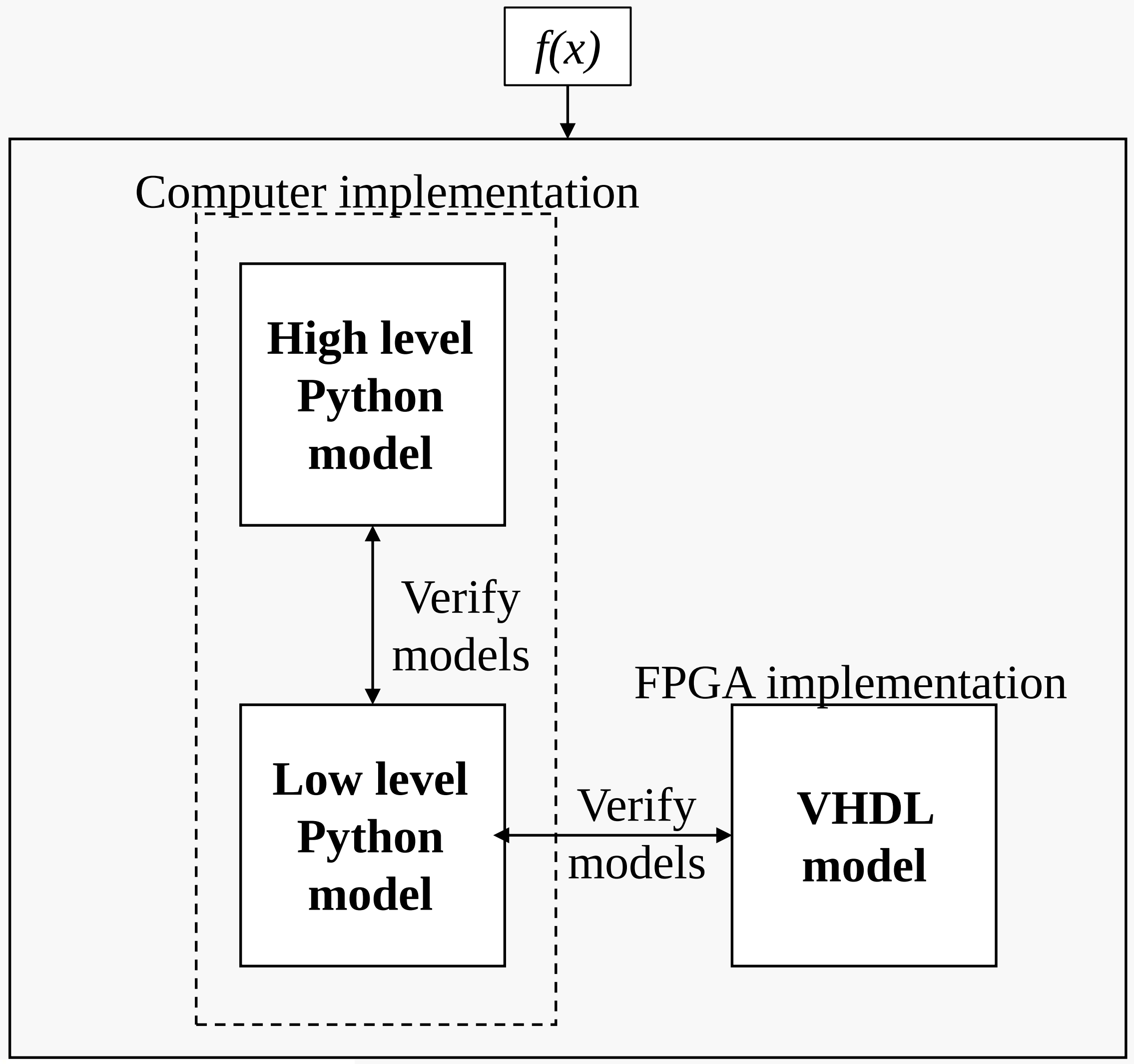
EES 424 Practical 1

FPGA communication system

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Research project model implementation



High level Python model

Implemented using Python functions working with methods far removed from bits, nibbles and bytes. This will typically use float data types and several pre-defined functions.

Low level Python model

The same functionality as implemented in the high level model, using low level arithmetic functions implemented on nibble and byte level operation. Logic features such as AND, OR, NOR, XOR, and bit shifting are expected

VHDL model

The functionality seen in the low level Python model, written in VHDL and implemented on an FPGA.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Objectives for Practical 1

- Implement a basic mathematical function or process with at least one input on FPGA
- Design and implement a communication system for data communication between FPGA and PC
- Design and implement a verification strategy in Python to test whether data transmitted from FPGA to PC is error free

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Minimum requirements (50-60%)

1.1. Function model with either one or two inputs

a) Demonstrate a standard arithmetic function* supported in VHDL (addition, multiplication, etc)

b) Code must be functional

c) Code must be synthesizable (implemented in hardware on a FPGA)

d) Evidence in the design approach that the function can be replaced by the models implemented by group members

* May implement non-standard VHDL arithmetic or increase complexity of implemented function for more than minimum requirements

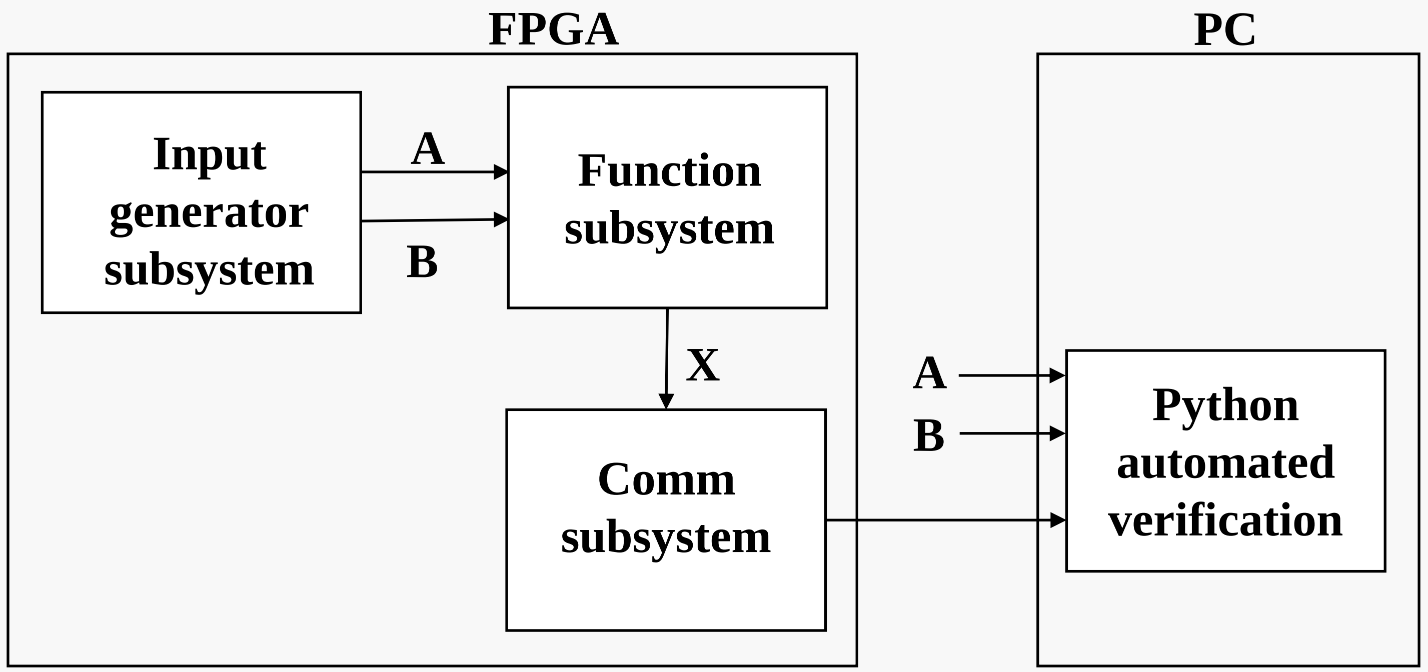
Minimum requirements (50-60%)

1.2 Communication model/subsystem for data communication

a) Communication between FPGA and PC*
b) Code must be functional
c) Code must be synthesizable (implemented in hardware on a FPGA)

* Scope for improvement includes bidirectional communication to meet more than minimum requirements

Communication system



Input generator subsystem
Subsystem that generates inputs that can be replicated in Python for testing if error free data transmission achieved

Function subsystem
Takes inputs and generates outputs in accordance with the implemented basic function (see 1.1)

Communication subsystem
Takes functional subsystem outputs and prepares them for transmission to PC

Python automated verification
Python script that uses the same inputs as the functional subsystem to compare thousands of outputs and verify error free communication

Minimum requirements (50-60%)

2.1. System integration

a) Communication between FPGA and Python must be demonstrated. Data dump in secondary files are considered bare minimum*

3.1. Verification

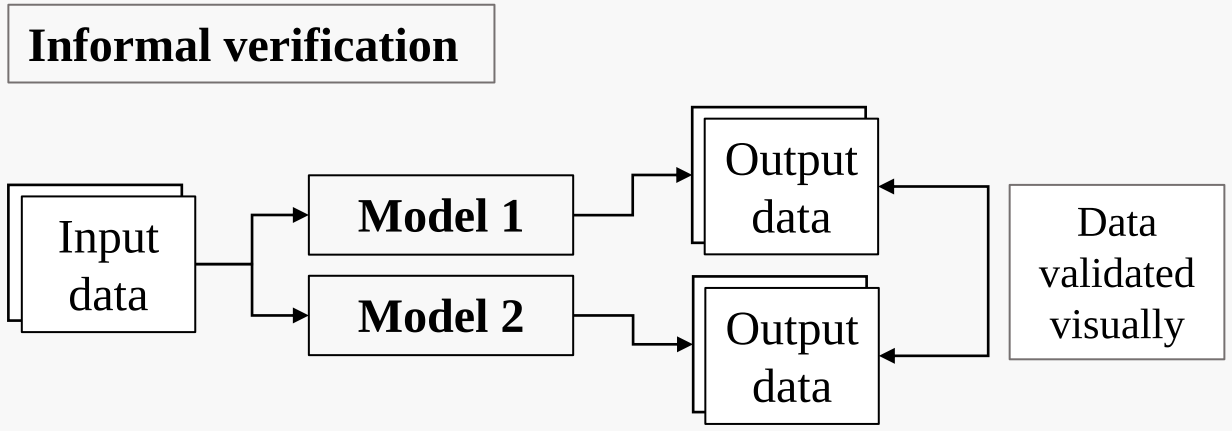
a) Results for visual inspection has to be demonstrated (ModelSim or similar)

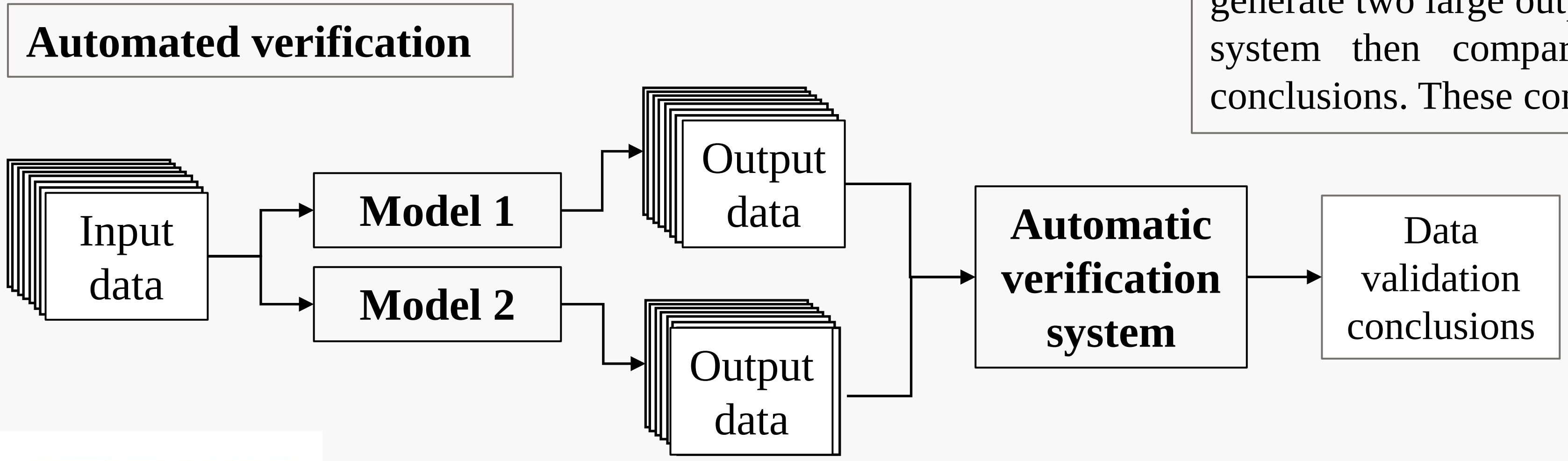
b) Implemented verification strategy for automated testing

* Scope for improvement includes seamless integration between Python and FPGA, with no extra steps required

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Model verification





Informal verification
The two models of interest take a small dataset (typically 10s of samples – numbers, bits, bytes, etc) and the results are compared visually as waveforms, numbers, bytes, etc. This is to confirm that there are no glaringly obvious mistakes in the implementation

Formal verification
The two models of interest take a much larger dataset (hundreds of thousands, or millions of samples) and generate two large output datasets. An automated validation system then compares the two models and delivers conclusions. These conclusions may be statistical.

Admin

- Demonstrations
  - Likely video-based

- Lab book
  - Design methodology used for design and implementation of FPGA system: software flow diagrams, calculations, pseudocode, etc
  - Design methodology and considerations used for Python verification system
  - Results

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi