An Extensible Framework for Evaluation of Arithmetic Hardware

Zifan Wang
Supervisor: Dr. James Davis

Imperial College London

June 25, 2019

1 Background
Motivation

2 Design & Implementation
System
Hardware
Software

3 Results
Results
Demo

4 Evaluation
Evaluation

5 Backup Slides

Motivation

- Started as a specialised evaluation system for high-radix online arithmetic units
  - At-speed
  - Precision Checking

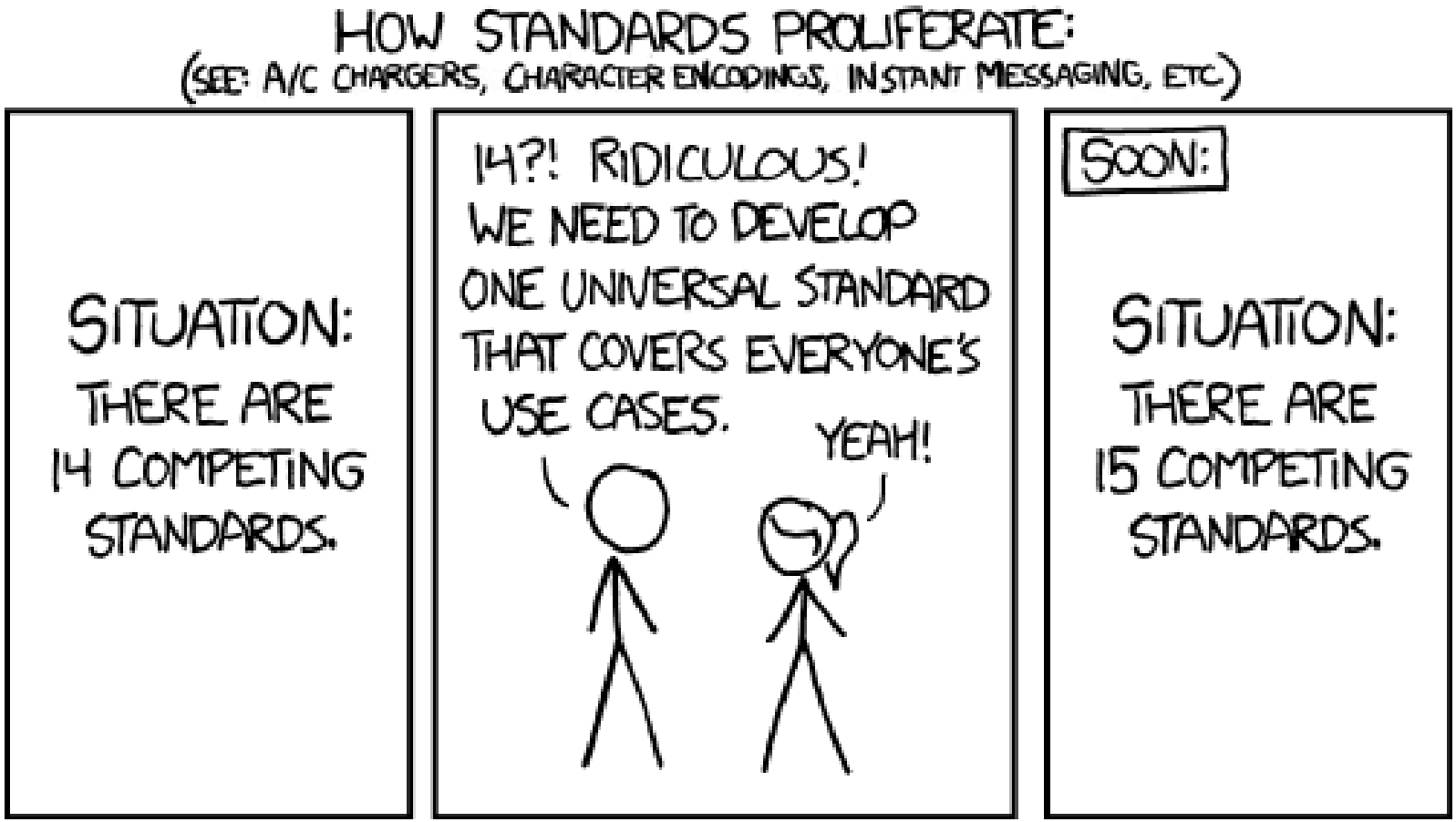
Motivation

- Started as a specialised evaluation system for high-radix online arithmetic units
  - At-speed
  - Precision Checking
- Digital designers all use their own testbenches

Motivation

- Started as a specialised evaluation system for high-radix online arithmetic units
  - At-speed
  - Precision Checking
- Digital designers all use their own testbenches
- Propose an extensible framework

Motivation



xkcd.com/927

Purpose

- Design and Implement an evaluation framework
  - High maximum frequency

Purpose

- Design and Implement an evaluation framework
  - High maximum frequency
  - Controllable frequency

Purpose

- Design and Implement an evaluation framework
  - High maximum frequency
  - Controllable frequency
  - Provide information regarding precision of output

Purpose

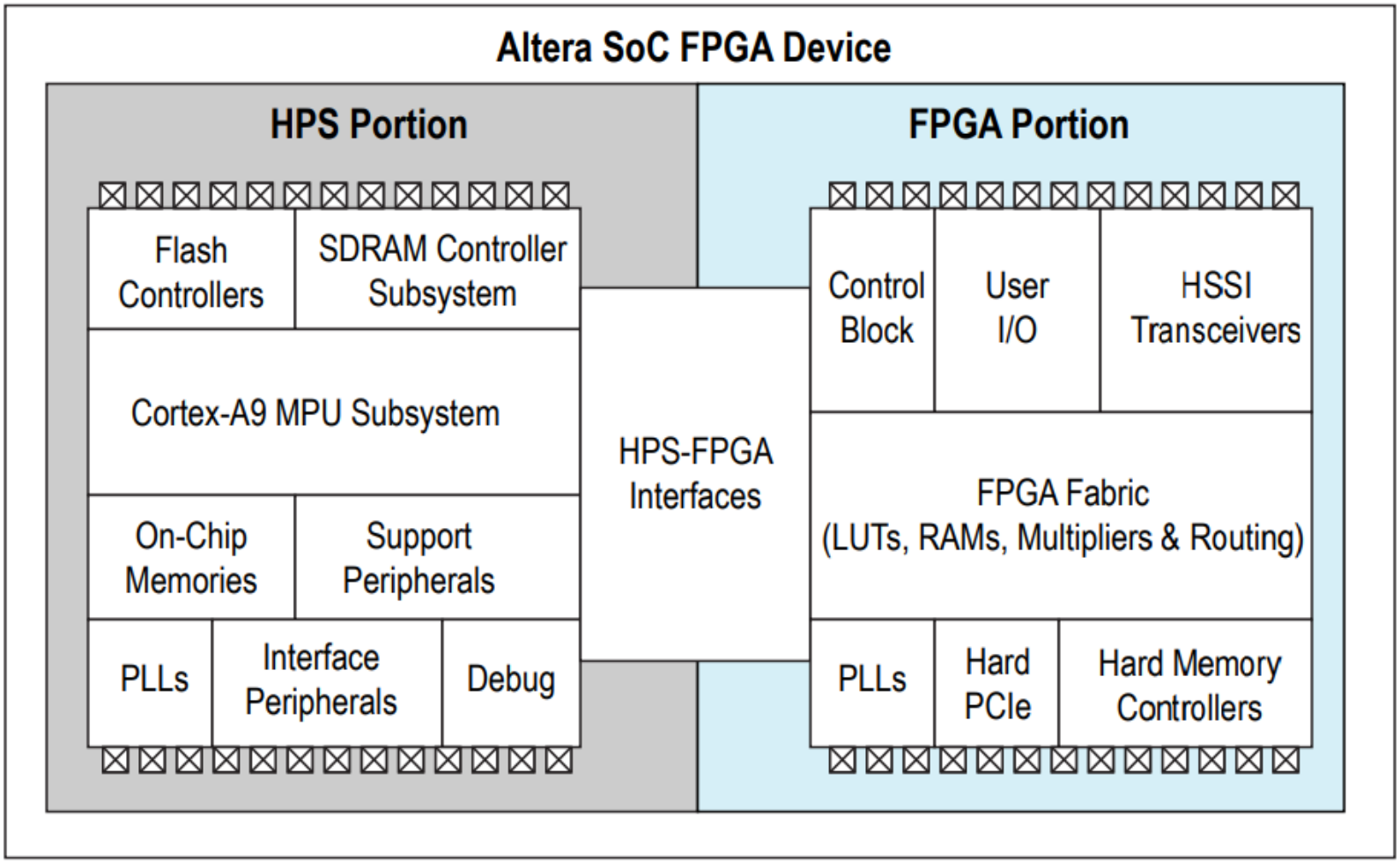
- Design and Implement an evaluation framework
  - High maximum frequency
  - Controllable frequency
  - Provide information regarding precision of output
  - Flexible, customisable

Purpose

- Design and Implement an evaluation framework
  - High maximum frequency
  - Controllable frequency
  - Provide information regarding precision of output
  - Flexible, customisable
  - User-friendly

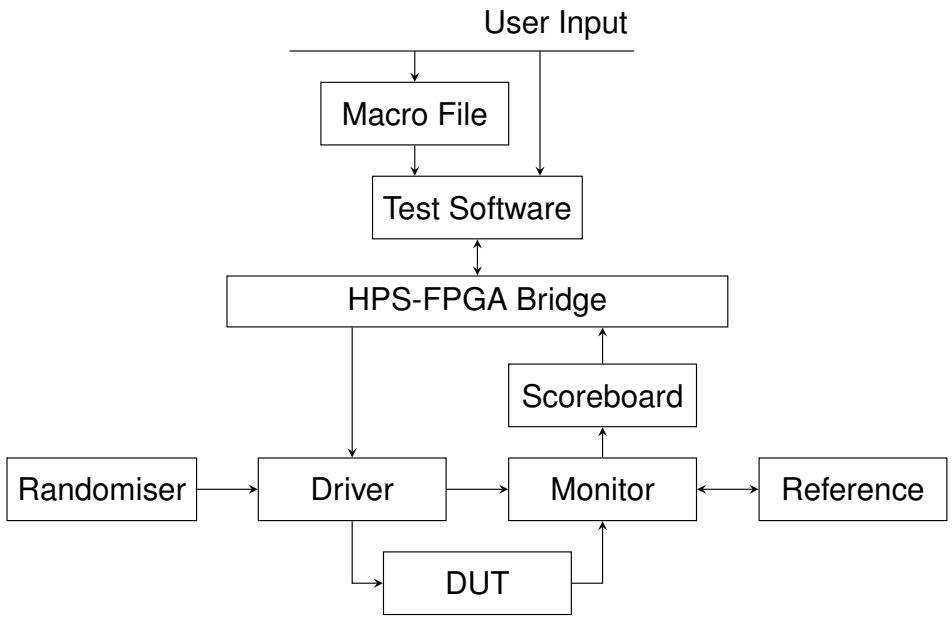
Hardware Choice

Cyclone V SX SoC Development Board

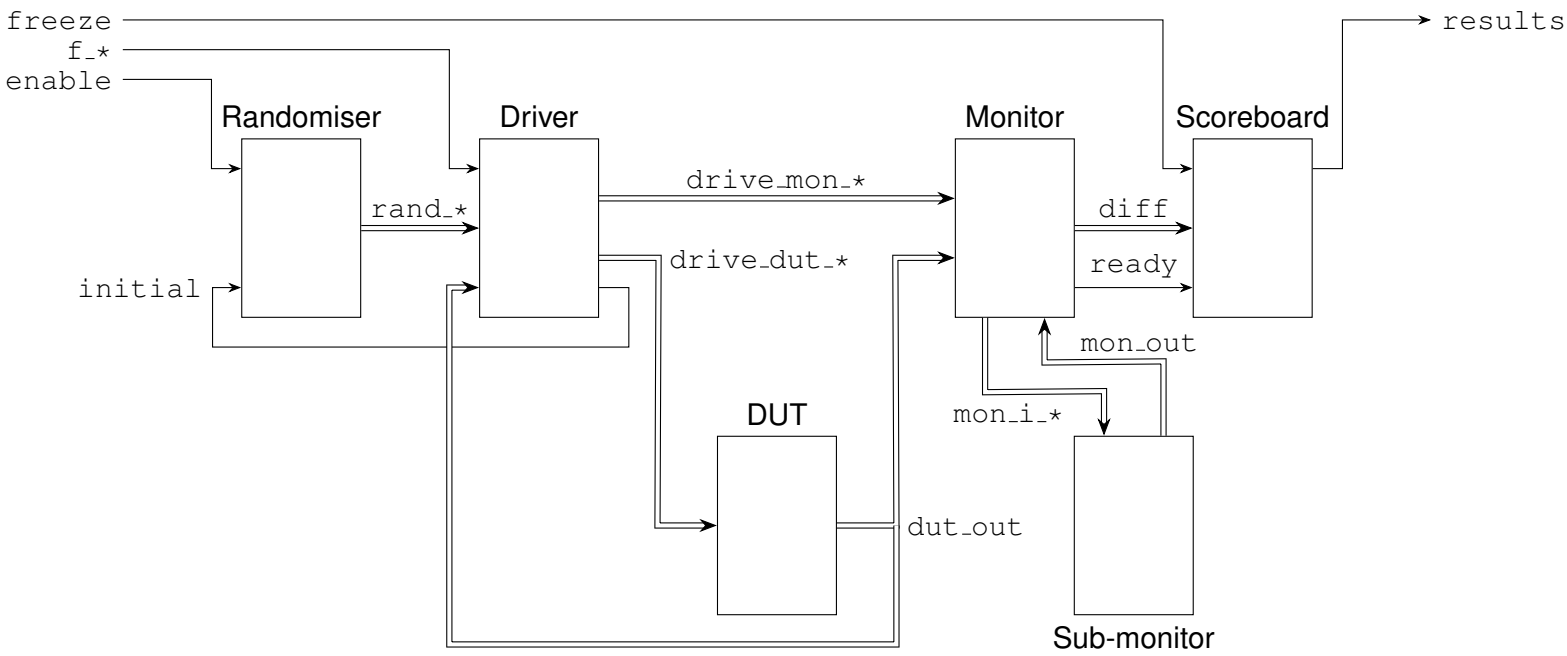


System Architecture

- Inspired by UVM agent
- Modular, thus extensible



Hardware


freeze
f_*
enable
initial
Randomiser
rand_*
Driver
drive_mon_*
drive_dut_*
Monitor
diff
ready
Scoreboard
results
mon_out
mon_i_*
DUT
dut_out
Sub-monitor

Software

Results

- Flexible
- Robust
- User-friendly

Demo

- Shows the configuration process
- Shows software interaction

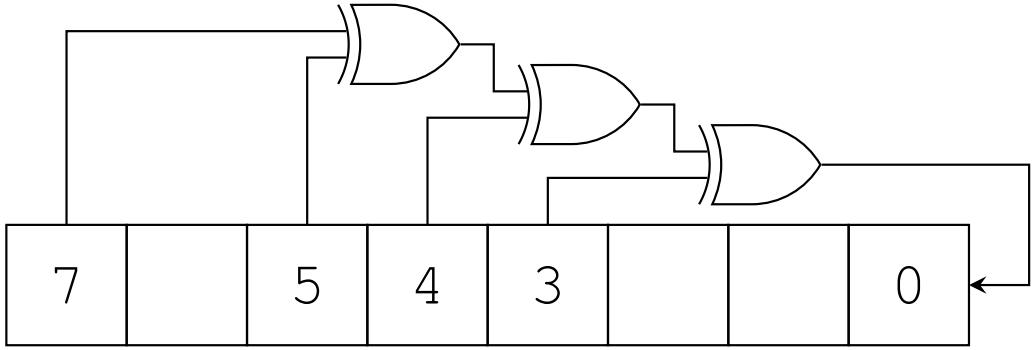
Limitations

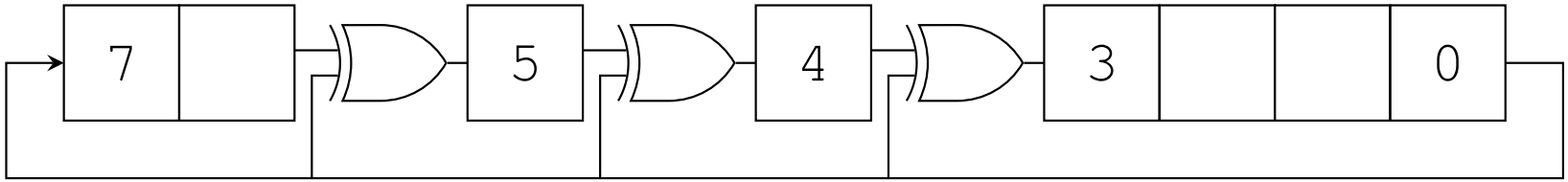
- Limited customisability of current implementation
- Tedious and error-prone
- Could be overcame with 2 improvements
- Unified software system + Verilog preprocessor
- Set up a more powerful HPS-FPGA communication system
- Not limits to the extensibility of the framework

Thank you

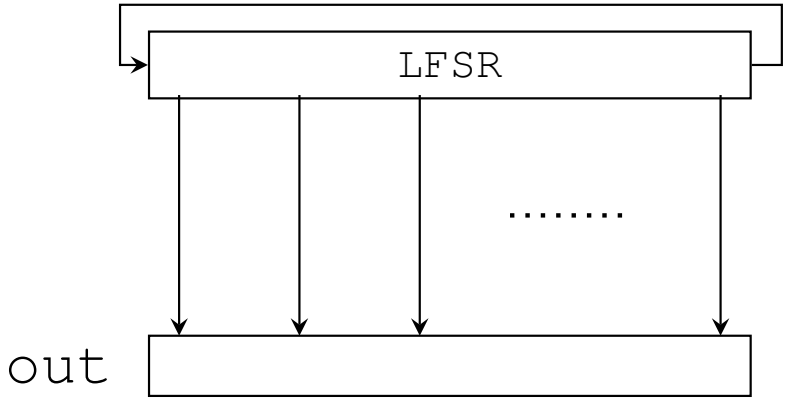
Questions?

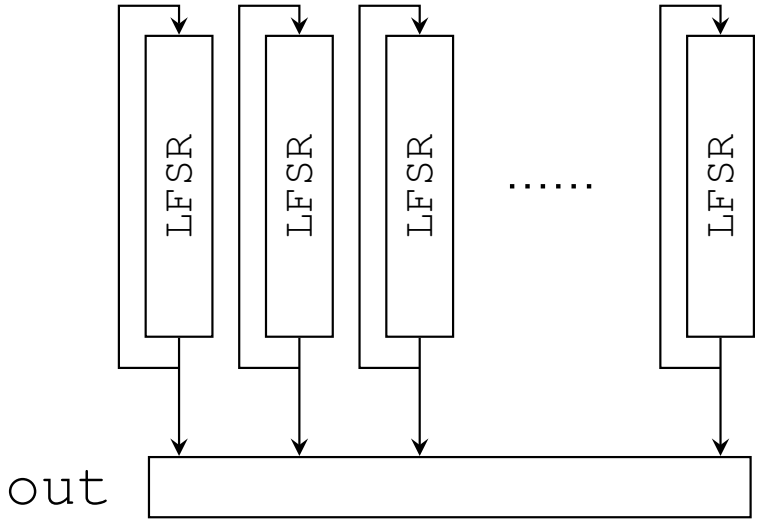
LFSRs





Randomiser Structure





Monitor Structure