FPGA-Assisted Verification

Team A1: Xiran Wang, Ali Hoffmann, Grace Fieni

Motivation

Hardware design must be thoroughly verified before production

- Traditionally done in simulation, which is **slow**

Our project: run tests through FPGA instead

- Fast hardware clock enables **better performance**
- Tests are run on **actual hardware**

Key requirements

Performance:

- 3x speedup relative to simulation

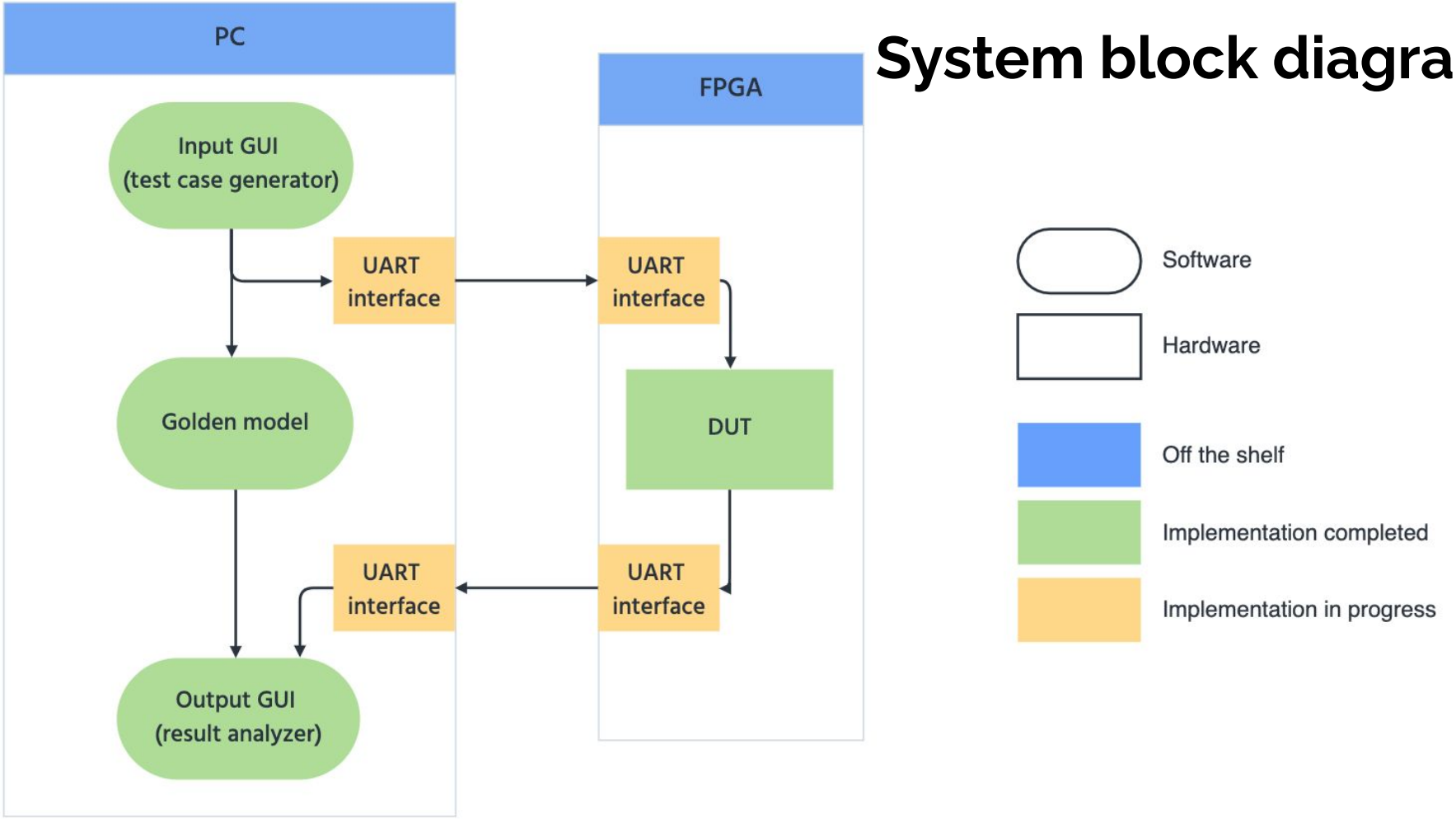
Input end:

- Test cases can have 1 ~ 20,000 instructions
- Test cases can be customized and randomized by user

Output end:

- Provide exactly the instruction that failed

System block diagram



More on the system: GUIs

Web app hosted on anvil.works

Live demo!

More on the system: communication

Protocol: **Universal Asynchronous Receiving-Transmitting** (UART)

- Serially sends a packet from one clock domain (the PC) to another (the FPGA)
- Any number of active high idle bits can be sent in between packets

Our specs

- Can use baud rate of **512,000 bits/second**
- Each packet contains a start bit, 8 data bits, and a stop bit (**10 bits/packet**)

0	1	2	3	4	5	6	7	8	9
START	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	STOP

Testing: performance

For 20,000 instructions,

Simulation: **2 seconds**

Our system: **0.9765 second**

Note: we pipeline sending data to FPGA, processing, and receiving data. Speed is determined by slowest stage (receiving)

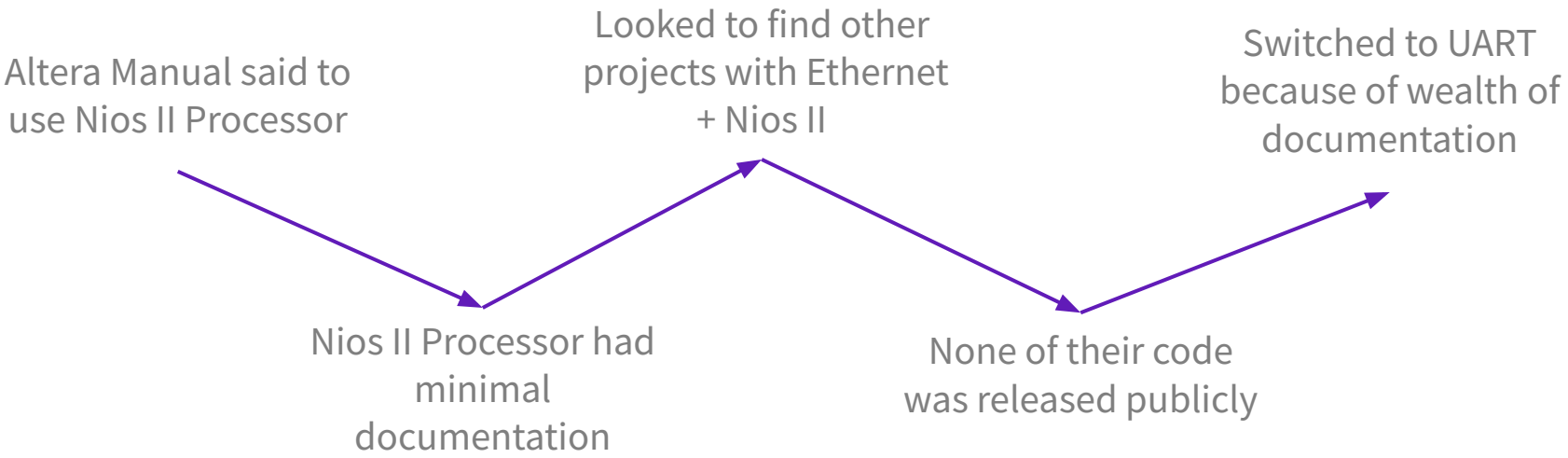
- 20 bits/instr to receive -> 400,000 bits to receive total -> 500,000 bits with start and stop
- 500,000 bits / (512,000 bits/second) = 0.9765 second

=> speedup ≈ 2.04x

An aside on what could have been...

For 20,000 instructions,

Gigabit Ethernet (1000 Mbits/second), could have taken **~0.0004 seconds**



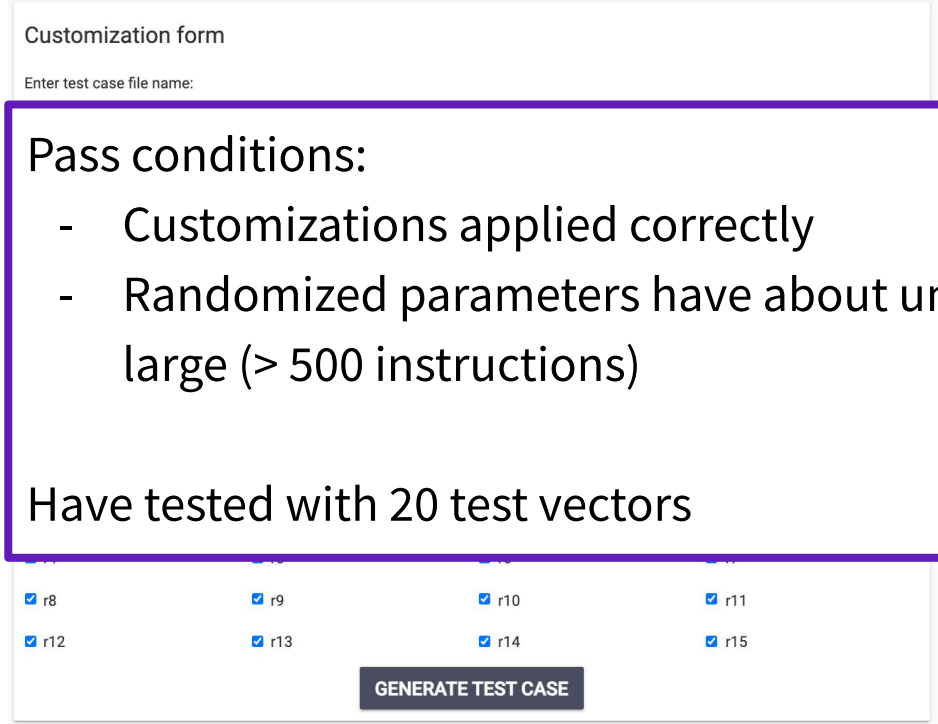
Lessons learned

Don’t be married to one implementation idea!

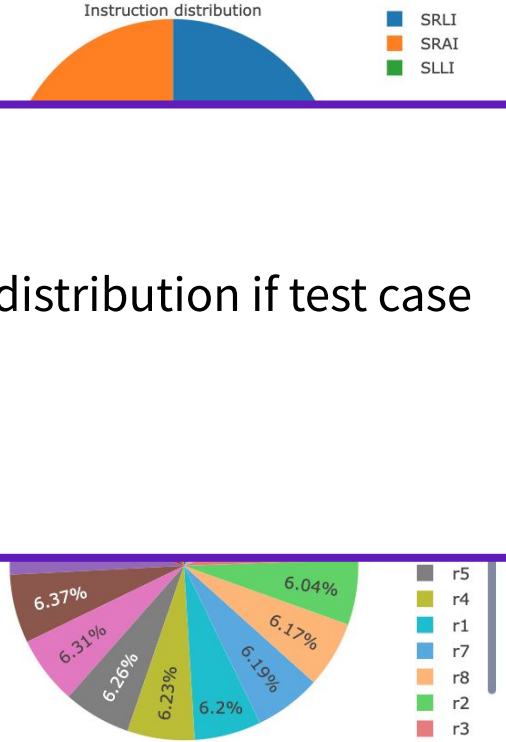
- That idea may take much longer than expected and/or may not work
- Must have backup plans
- Be willing to sacrifice some metrics for other more important ones

Testing: customization/randomization

Example test vector



Example results



Pass conditions:

- Customizations applied correctly
- Randomized parameters have about uniform distribution if test case large (> 500 instructions)

Have tested with 20 test vectors

Testing: failure report

Example test vector

Test case:

```
ADDI r1, r1, 1   # r1 = 1
ADDI r2, r1, 1   # r2 = r1 + 1 = 2
ADDI r0, r1, 1   # no effect (r0 not written to in ISA)
ADDI r3, r1, 1   # r3 = r1 + 1 = 2
```

Bug:

r0 is written to

Example results

Failure at instruction #3

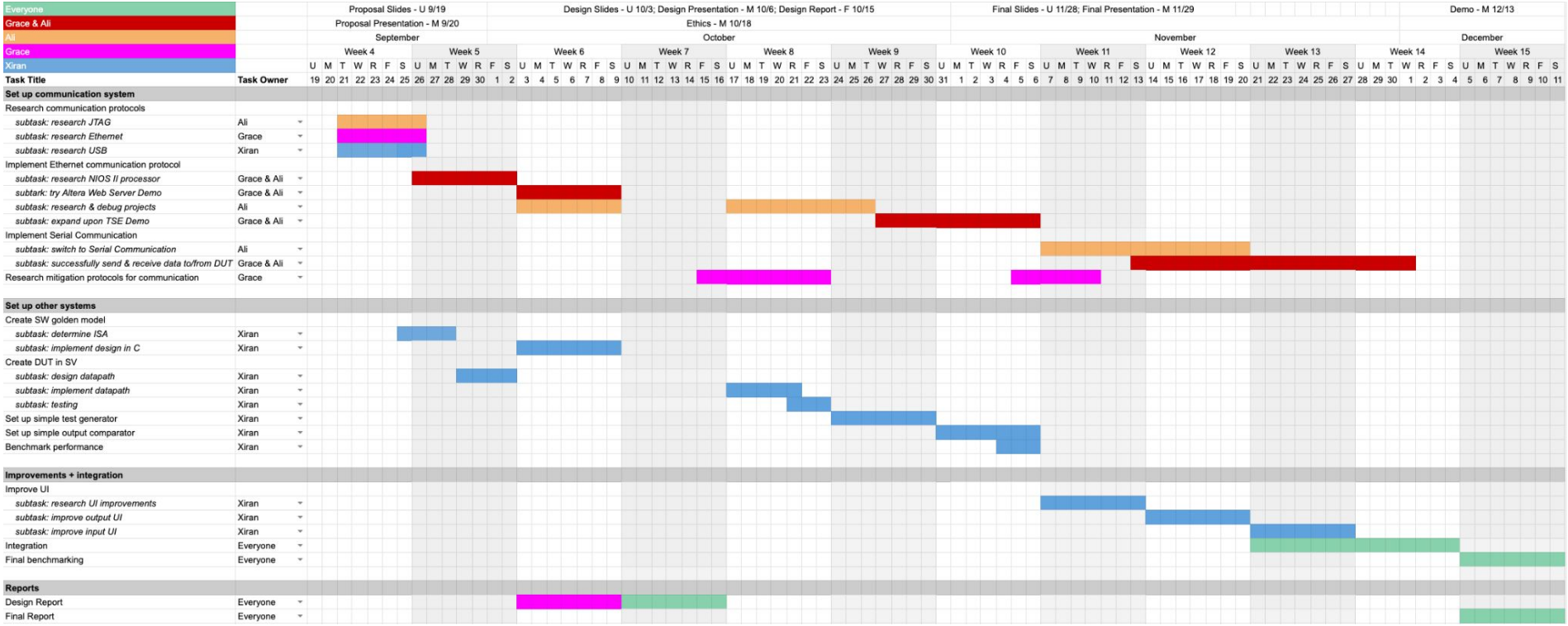
Register comparison table shows r0 is incorrect

Pass condition:

- Failure reported at exact expected location

Have not yet tested due to system being incomplete

Schedule


Everyone
Grace & Ali
Ali
Grace
Xiran
Proposal Slides - U 9/19
Proposal Presentation - M 9/20
Design Slides - U 10/3; Design Presentation - M 10/6; Design Report - F 10/15
Ethics - M 10/18
Final Slides - U 11/28; Final Presentation - M 11/29
Demo - M 12/13
September
October
November
December
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
Task Title
Task Owner
Set up communication system
Research communication protocols
subtask: research JTAG
Ali
subtask: research Ethernet
Grace
subtask: research USB
Xiran
Implement Ethernet communication protocol
subtask: research NIOS II processor
Grace & Ali
subtark: try Altera Web Server Demo
Grace & Ali
subtask: research & debug projects
Ali
subtask: expand upon TSE Demo
Grace & Ali
Implement Serial Communication
subtask: switch to Serial Communication
Ali
subtask: successfully send & receive data to/from DUT
Grace & Ali
Research mitigation protocols for communication
Grace
Set up other systems
Create SW golden model
subtask: determine ISA
Xiran
subtask: implement design in C
Xiran
Create DUT in SV
subtask: design datapath
Xiran
subtask: implement datapath
Xiran
subtask: testing
Xiran
Set up simple test generator
Xiran
Set up simple output comparator
Xiran
Benchmark performance
Xiran
Improvements + integration
Improve UI
subtask: research UI improvements
Xiran
subtask: improve output UI
Xiran
subtask: improve input UI
Xiran
Integration
Everyone
Final benchmarking
Everyone
Reports
Design Report
Everyone
Final Report
Everyone