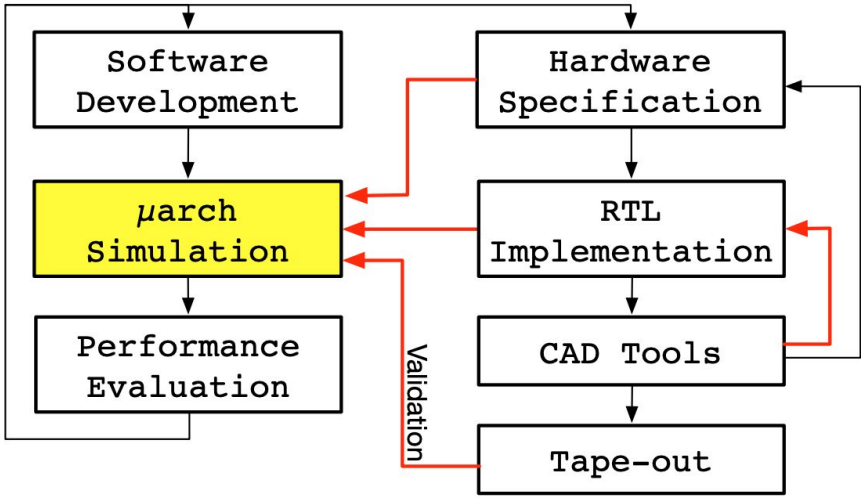
FPGA-Assisted Verification

Team A1: Xiran Wang, Ali Hoffmann, Grace Fieni

Verification: bottleneck of chip-design process

Hardware design must be thoroughly verified before production

- Traditionally done using software **simulation** tools, which are **slow**
- This phase of development process can take months

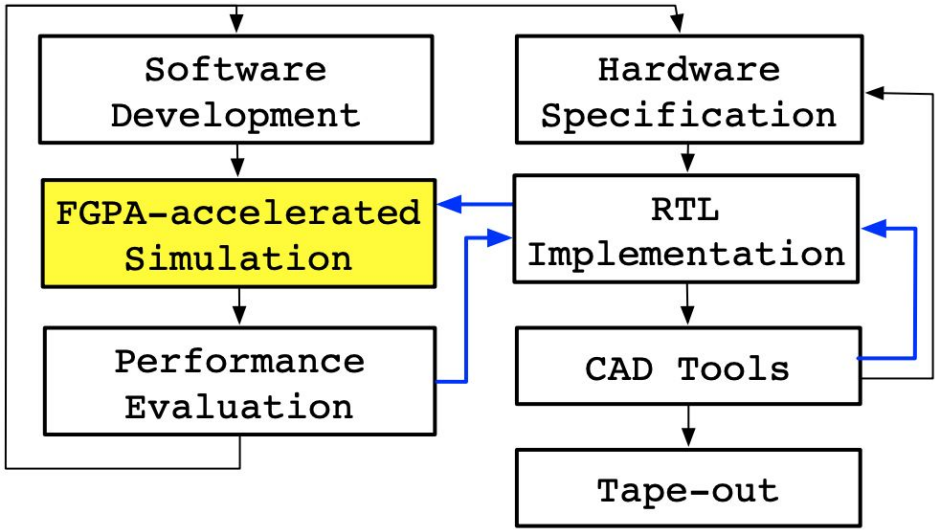


Our project: run tests through FPGA instead

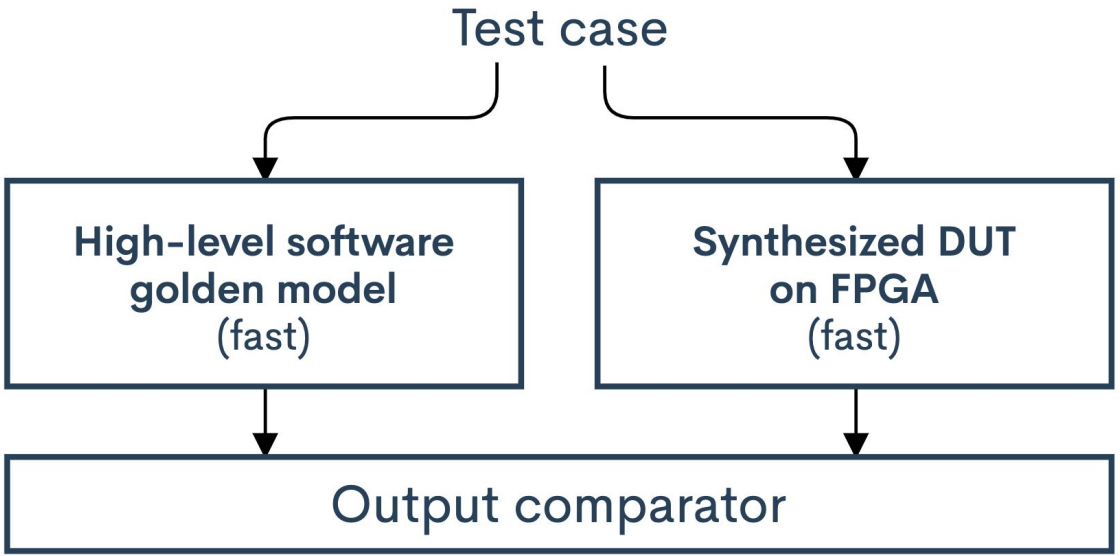
Fast hardware clock enables **better performance** than software simulators

Why FPGA?

- Low cost
- Low power
- Reconfigurable



More concretely...



The DUT will be a single-cycle RISC processor with a limited instruction set.

Requirements

- Execute arbitrary test cases comprised of valid instructions from ISA
- Randomized test cases
- Runtime speedup of 3x relative to simulation
- Cycle-by-cycle correctness checks

Why these requirements?

- Randomization
  - Finds unanticipated bugs
  - Required for framework to be useful
- 3x speedup
  - e.g. 24 hrs -> 8 hrs (1 day -> overnight)
- Cycle-by-cycle checks
  - Indicating origin of failure eases debugging

Challenges

Need to send test cases to and obtain results from the FPGA

FPGAs have **limited memory** and **I/O bandwidth**

→ Balance what is stored on FPGA vs. send across communication channel

→ Carefully choose communication protocol

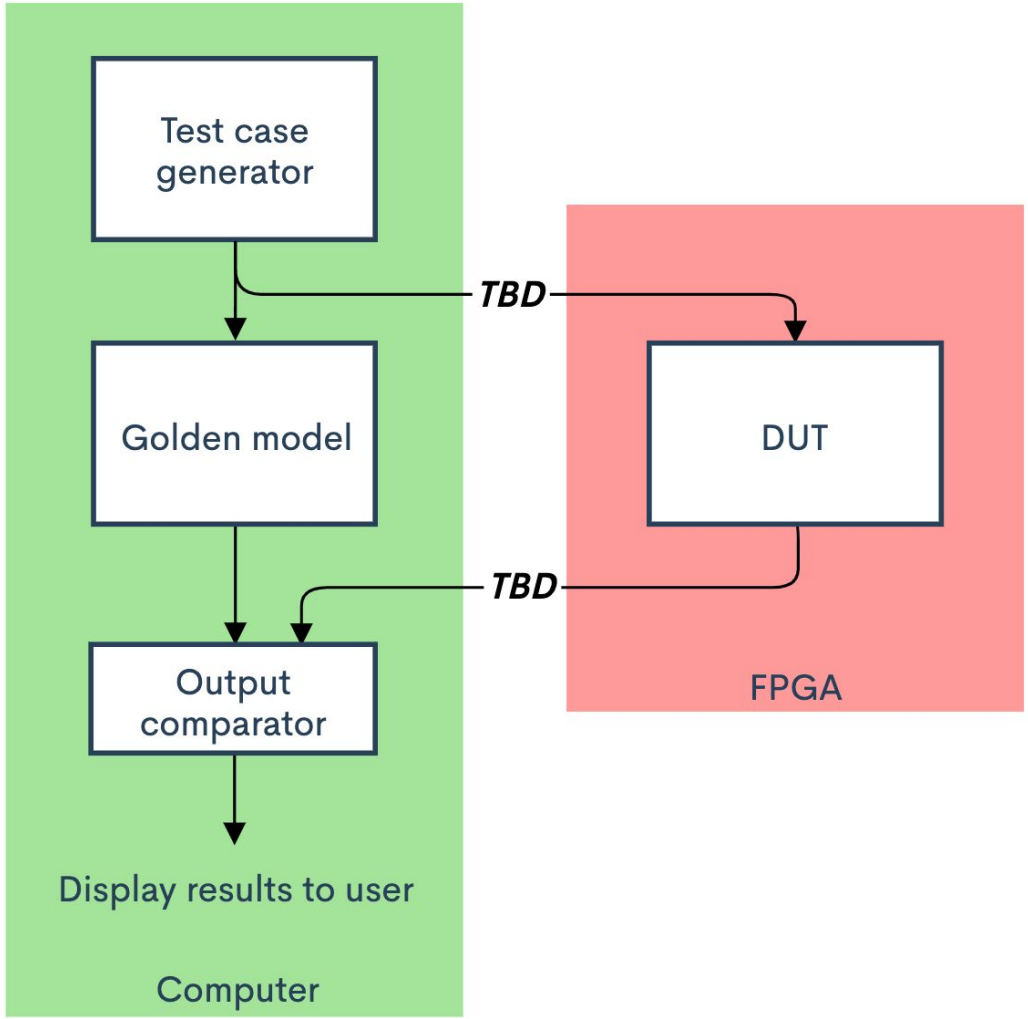
Challenge mitigation

Find a communication protocol that meets the speedup requirement through benchmarking and research

Protocols under consideration:

- JTAG
- USB
- Ethernet

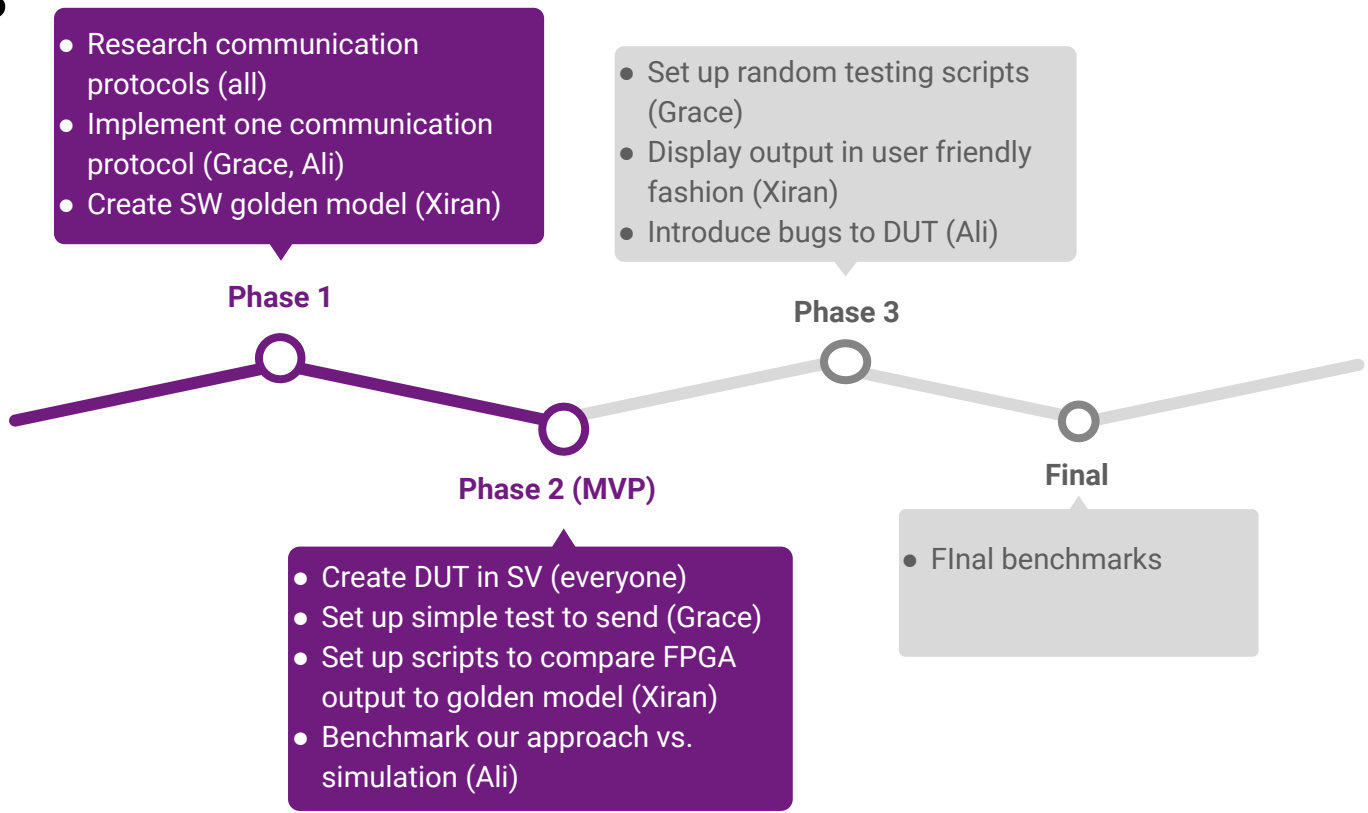
Solution


Test case generator
TBD
Golden model
DUT
TBD
Output comparator
FPGA
Display results to user
Computer

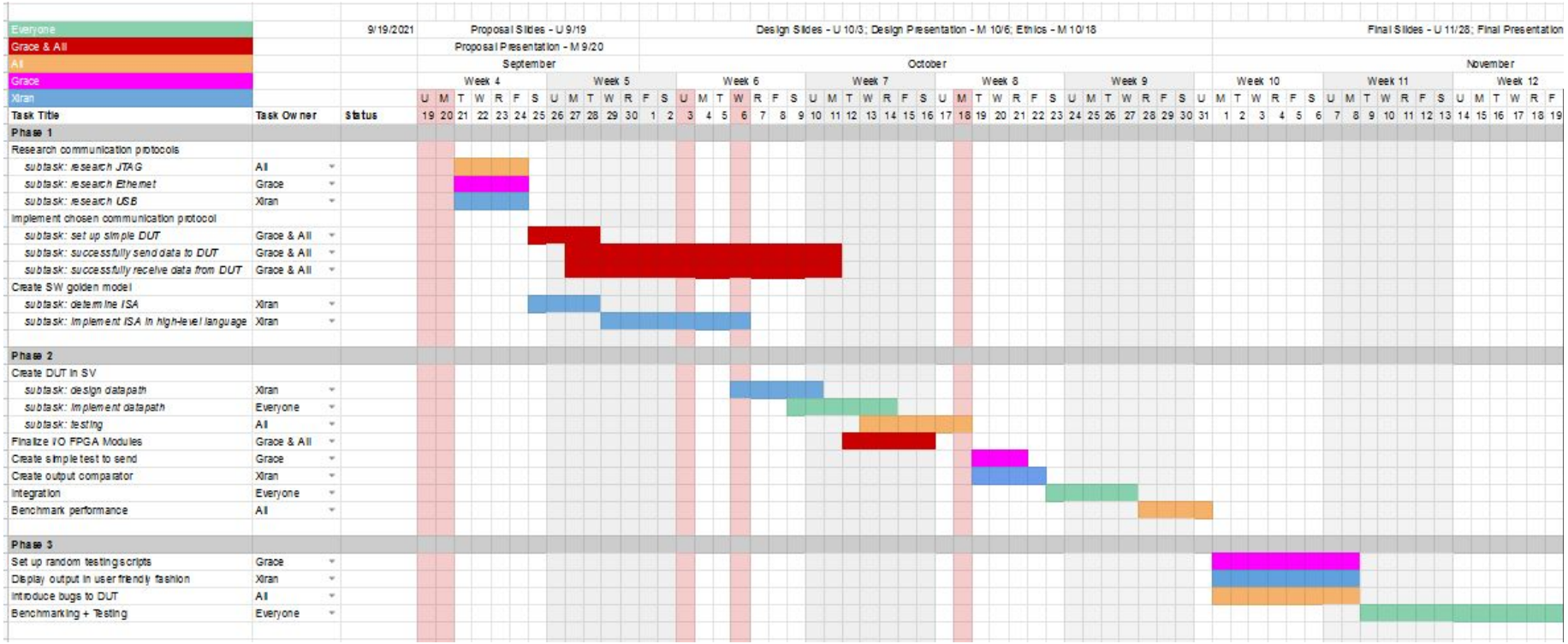
Testing + Benchmarking

- Randomization
  - look at test cases to see if different values are generated
- 3x speedup
  - time simulation and FPGA approach
- Cycle-by-cycle checks
  - insert bugs into DUT and check failure at correct clock cycle

Tasks


Research communication protocols (all)
Implement one communication protocol (Grace, Ali)
Create SW golden model (Xiran)
Phase 1
Set up random testing scripts (Grace)
Display output in user friendly fashion (Xiran)
Introduce bugs to DUT (Ali)
Phase 3
Final
FInal benchmarks
Phase 2 (MVP)
Create DUT in SV (everyone)
Set up simple test to send (Grace)
Set up scripts to compare FPGA output to golden model (Xiran)
Benchmark our approach vs. simulation (Ali)

Schedule


Everyone
Grace & All
All
Grace
Xiran
9/19/2021
Proposal Slides - U 9/19
Proposal Presentation - M 9/20
Design Slides - U 10/3; Design Presentation - M 10/6; Ethics - M 10/18
Final Slides - U 11/28; Final Presentation
September
October
November
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Task Title
Task Owner
Status
Phase 1
Research communication protocols
subtask: research JTAG
All
subtask: research Ethernet
Grace
subtask: research USB
Xiran
Implement chosen communication protocol
subtask: set up simple DUT
Grace & All
subtask: successfully send data to DUT
Grace & All
subtask: successfully receive data from DUT
Grace & All
Create SW golden model
subtask: determine ISA
Xiran
subtask: implement ISA in high-level language
Xiran
Phase 2
Create DUT in SV
subtask: design datapath
Xiran
subtask: implement datapath
Everyone
subtask: testing
All
Finalize I/O FPGA Modules
Grace & All
Create simple test to send
Grace
Create output comparator
Xiran
Integration
Everyone
Benchmark performance
All
Phase 3
Set up random testing scripts
Grace
Display output in user friendly fashion
Xiran
Introduce bugs to DUT
All
Benchmarking + Testing
Everyone