



Switch ON

Ayush Jain (aj2672)

Donovan Chan(dc3095)

Shivam Choudhary(sc3973)

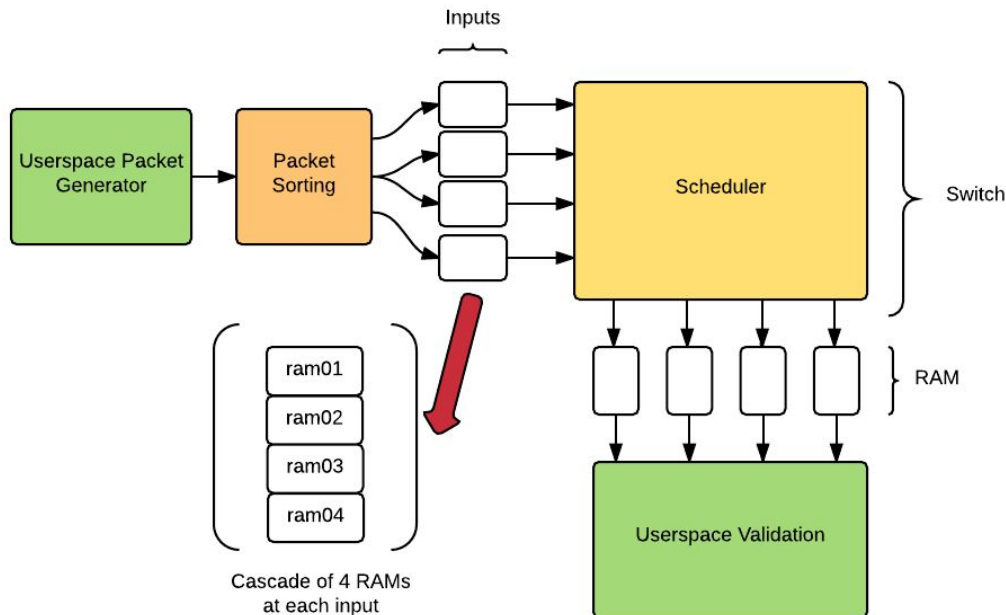


“

An extremely efficient
Hardware Switch!!

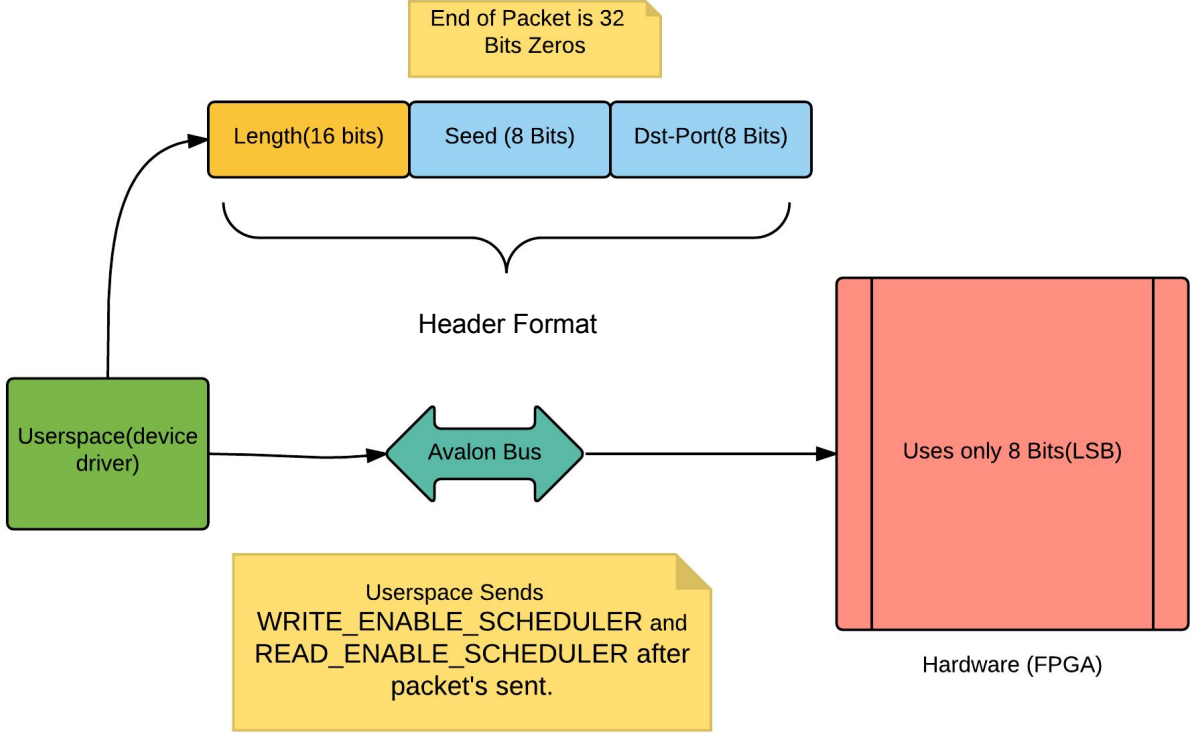
”

Architecture



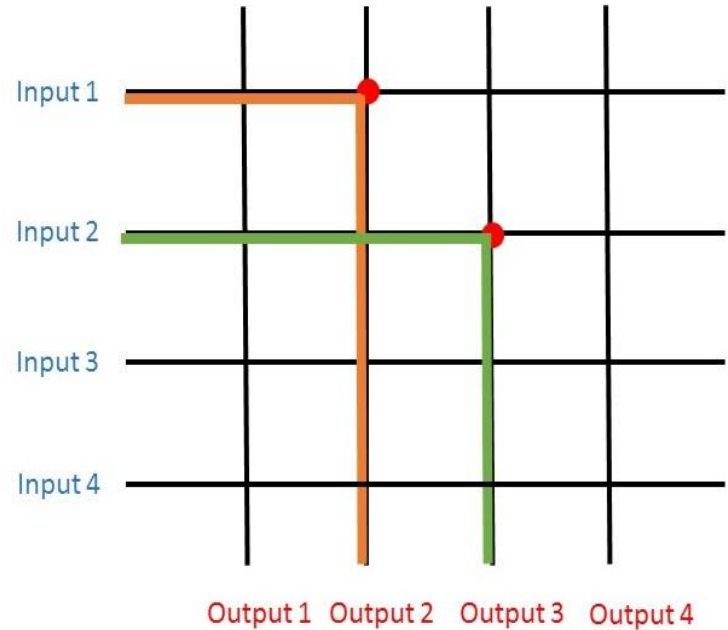
- Userspace generates packets.
- Input module sorts and places in RAM(s).
- Scheduler avoids collisions between packets.
- Buffer stores data in output RAM(s).

Hardware Communication Protocol



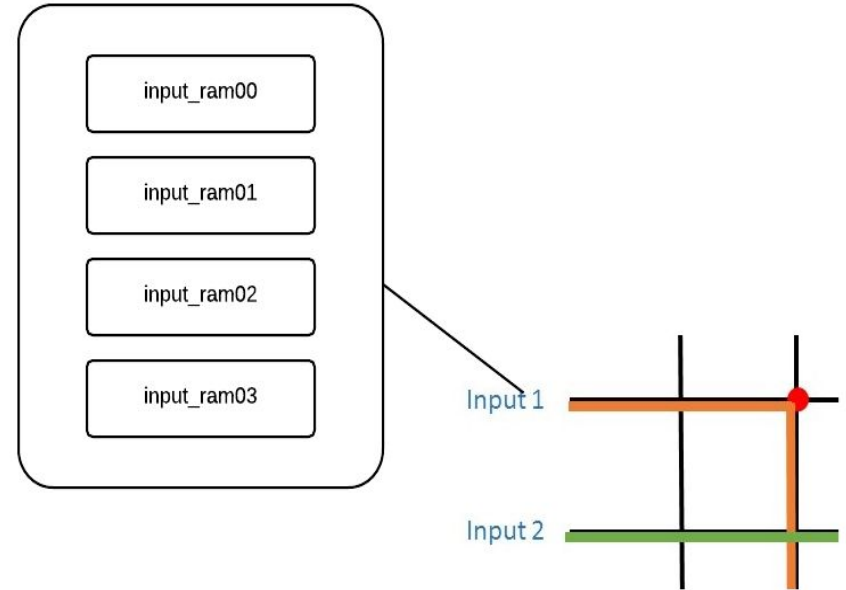
Hardware Scheduler - Single Input Queue

- Source & Destination modelled as 4 RAM(s) each.
- Individual scheduling to prevent collision.
- Greedy, no optimization for head of line blocking.

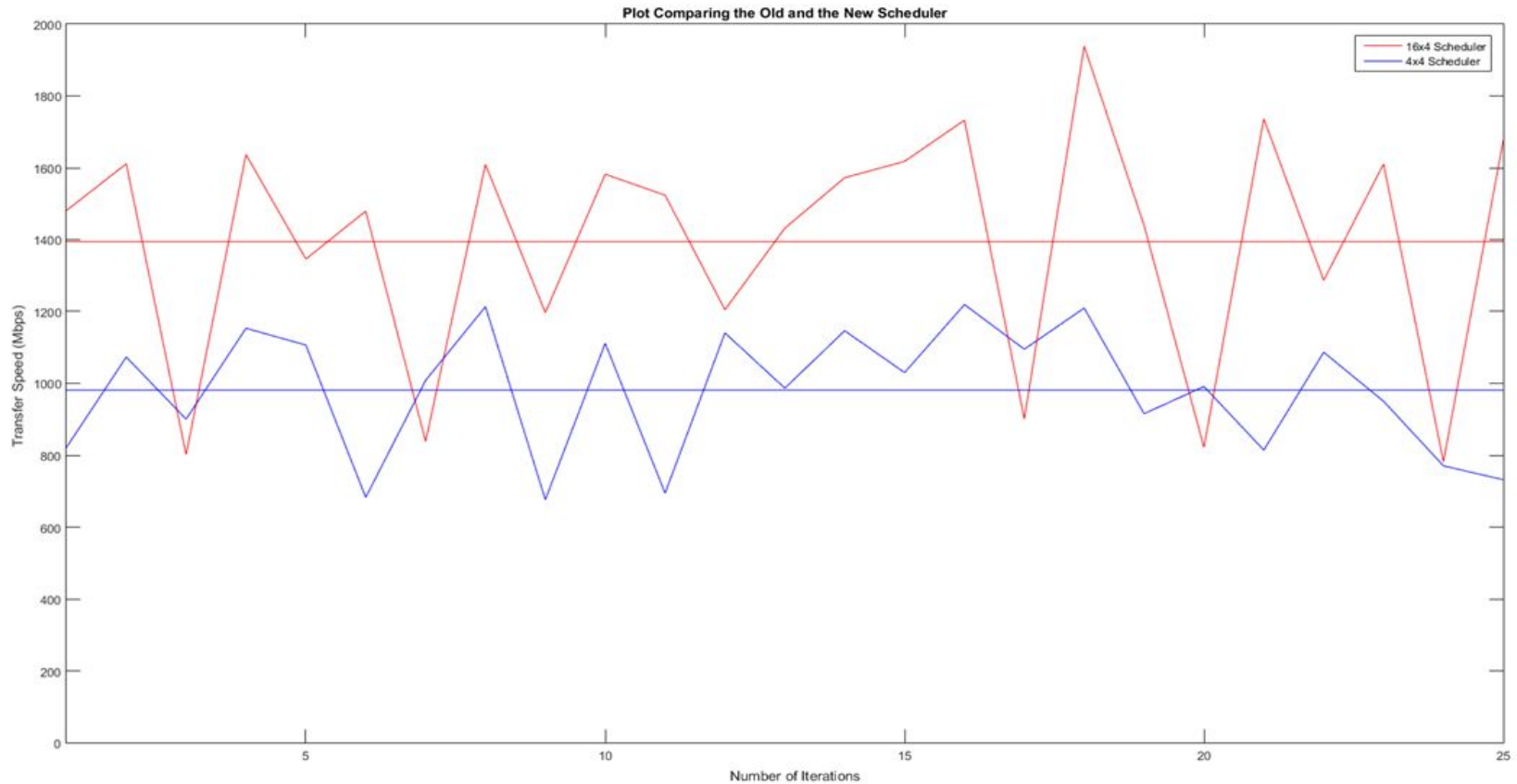


Hardware Scheduler - PPS

- Modeled as 16 RAMs at the input.
(Parallel Packet Switch)
- Destination still modelled as 4 RAMs
- Prevents HOL, hence improves throughput.
- Requires additional hardware complexity and storage.



PPS vs Single Input Queue



PPS vs Single Input Queue (contd.)

Worst-case Performance

$$\text{Speed} = \frac{32}{3} \text{ bits/cycle}$$

$$\text{Speed} = \frac{32}{3 \times 20 \times 10^{-9}}$$

$$\text{Speed} = 0.533 \times 10^9$$

$$\text{Speed} = 508.626 \text{ Mb/s}$$

Average Performance

- Better average case performance for PPS.
- Higher variance.

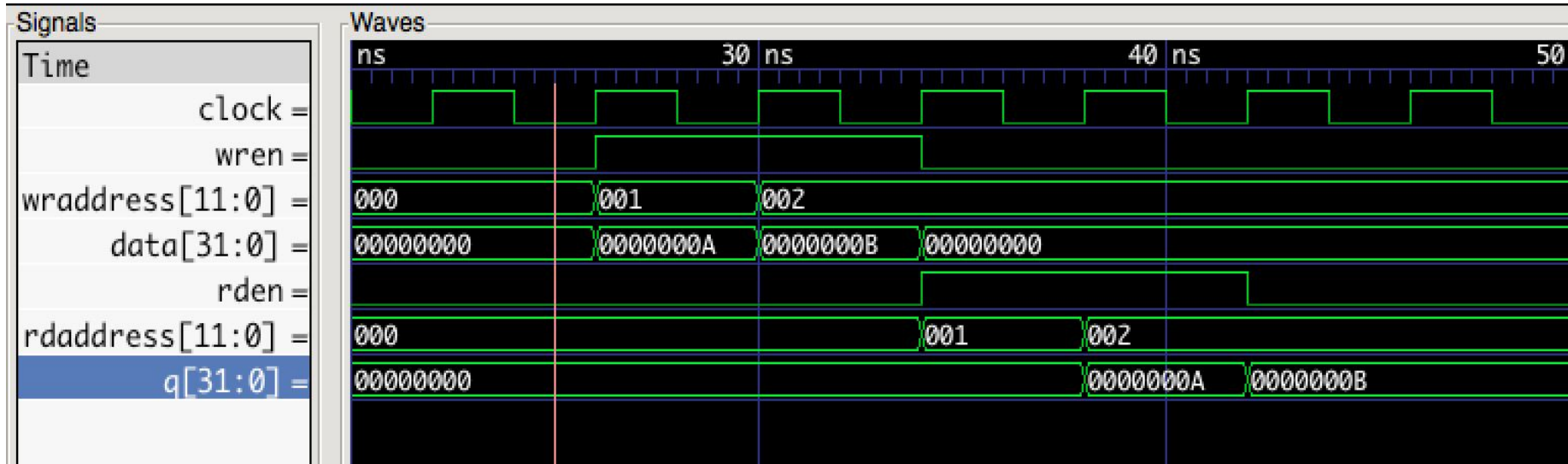
Best-case Performance

- Same for PPS and Single Input Queue.
- Such a case is theoretically less probable to occur.

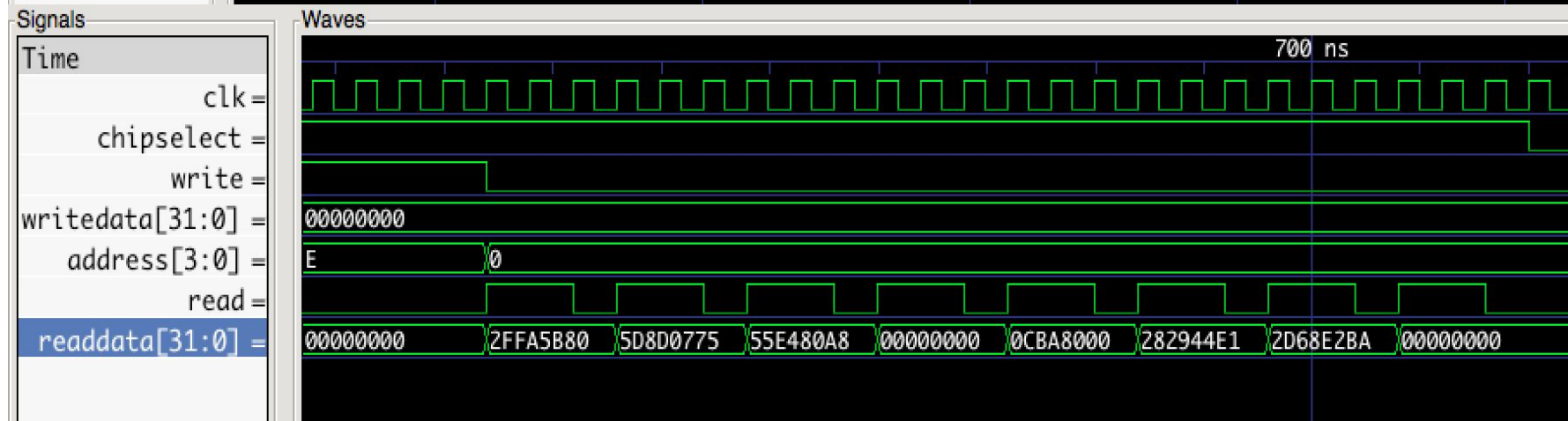
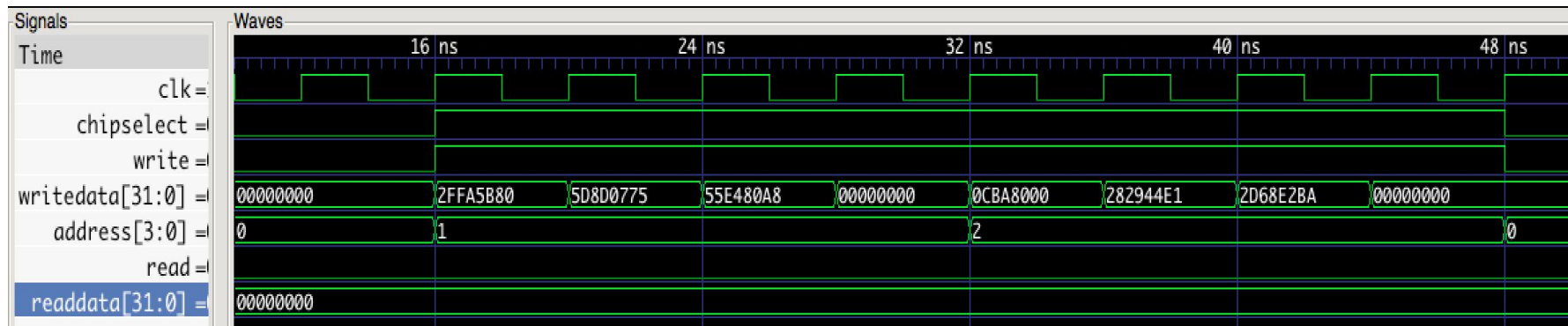
Timing Diagrams - RAM(altsync)

Input Signals - wren, wraddress, data, rden, rdaddress.

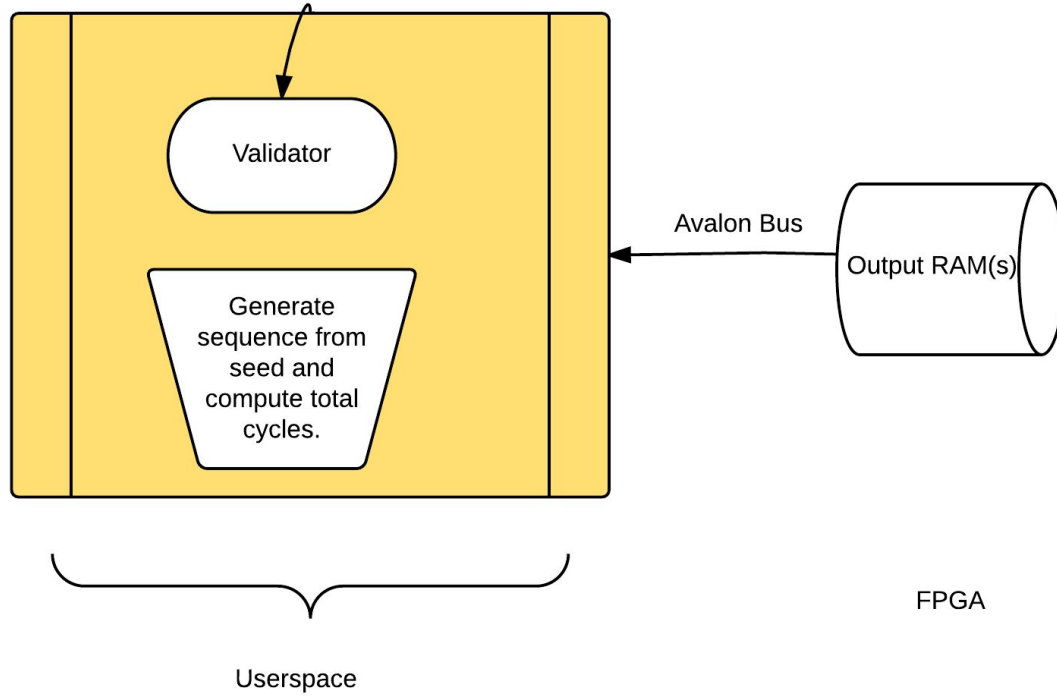
Output Signals - q (Data occurs after one clock cycle)



Timing Diagrams - Full Suite



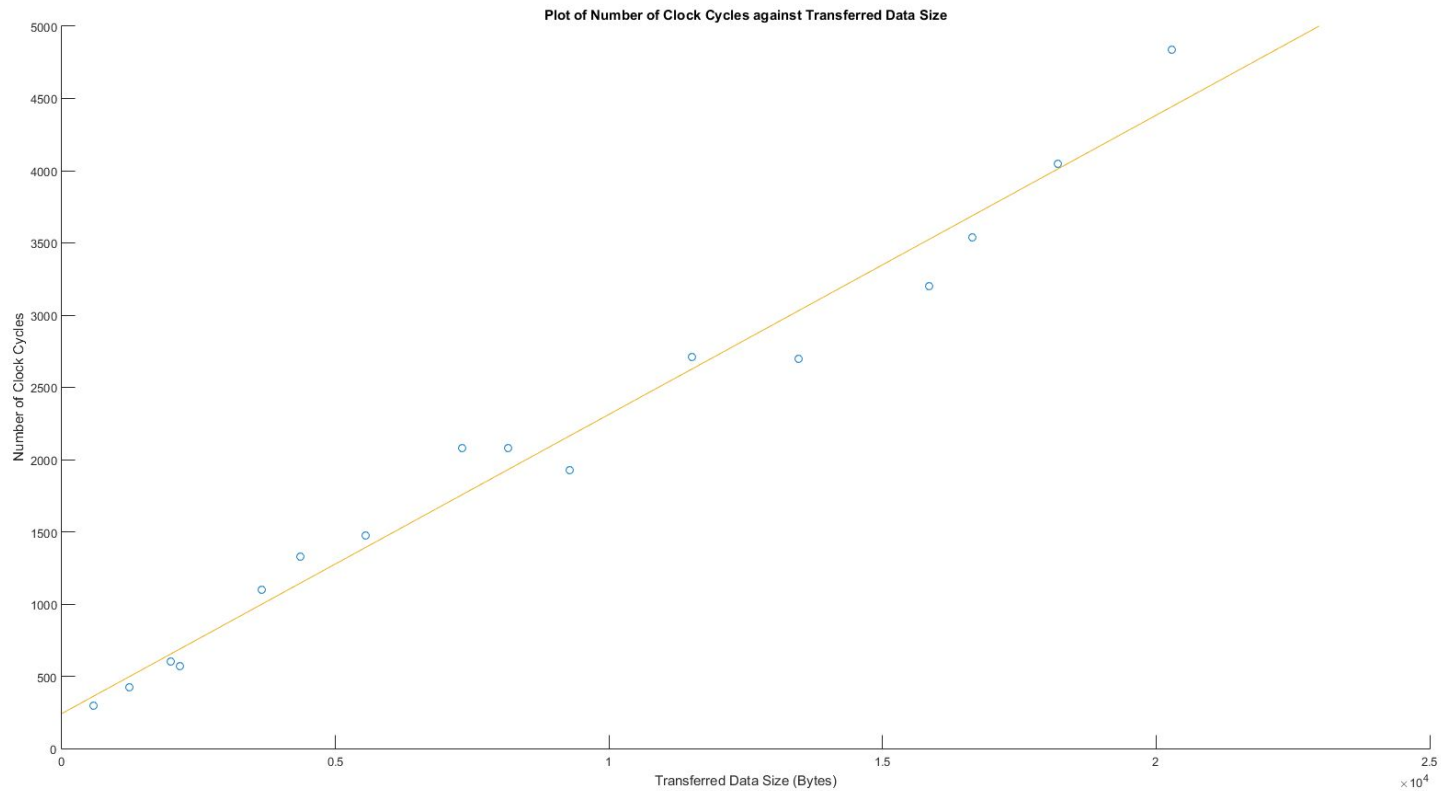
Validator



DEMO



Results



Performance Constraints

- RAMs take up three clock cycles to change from one read location to another.
- Transfers are restricted to 32 bits at a time because of ioctl calls.
- Can also increase performance if we increase the number of parts at the cost of hardware complexity.

Lessons Learned

- It is named hard-ware for a reason.
- Timing diagrams save time.
- Simulations may be far from reality.
- You will often reduce to hard problems in polynomial time.
- Documentations need a lot of work.



Future Work

- Analyze and compare the performance for a maximum bipartite matching solution.
- Implement DMA.
- Produce test results for greater amount of data and different scenarios.
- Interface with Ethernet ports and test with real network.

Thank You!!



Code available on Github: <https://github.com/shivamchoudhary/SwitchON>
<https://github.com/shivamchoudhary/SwitchONHW>