Modelling Smart FPGA Switches in the Network

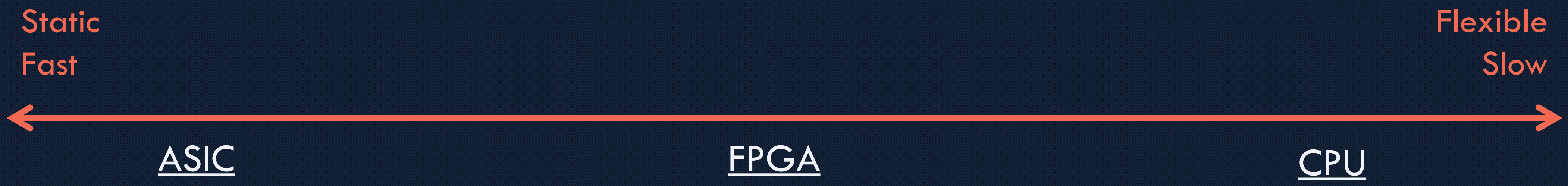
Benji Levine

Supervised by Suhaib Fahmy

What are FPGAs?

Field-Programmable Gate Array

Can be seen as alternative to ASIC	Application-Specific Integrated Circuit
Flexible Logic	LUTs
Flexible routing	Large grid of wires and switch boxes
Flexible IO	Support for 10G Ethernet, SATA, PCIe
Embedded hard modules	Block memory, DSP blocks



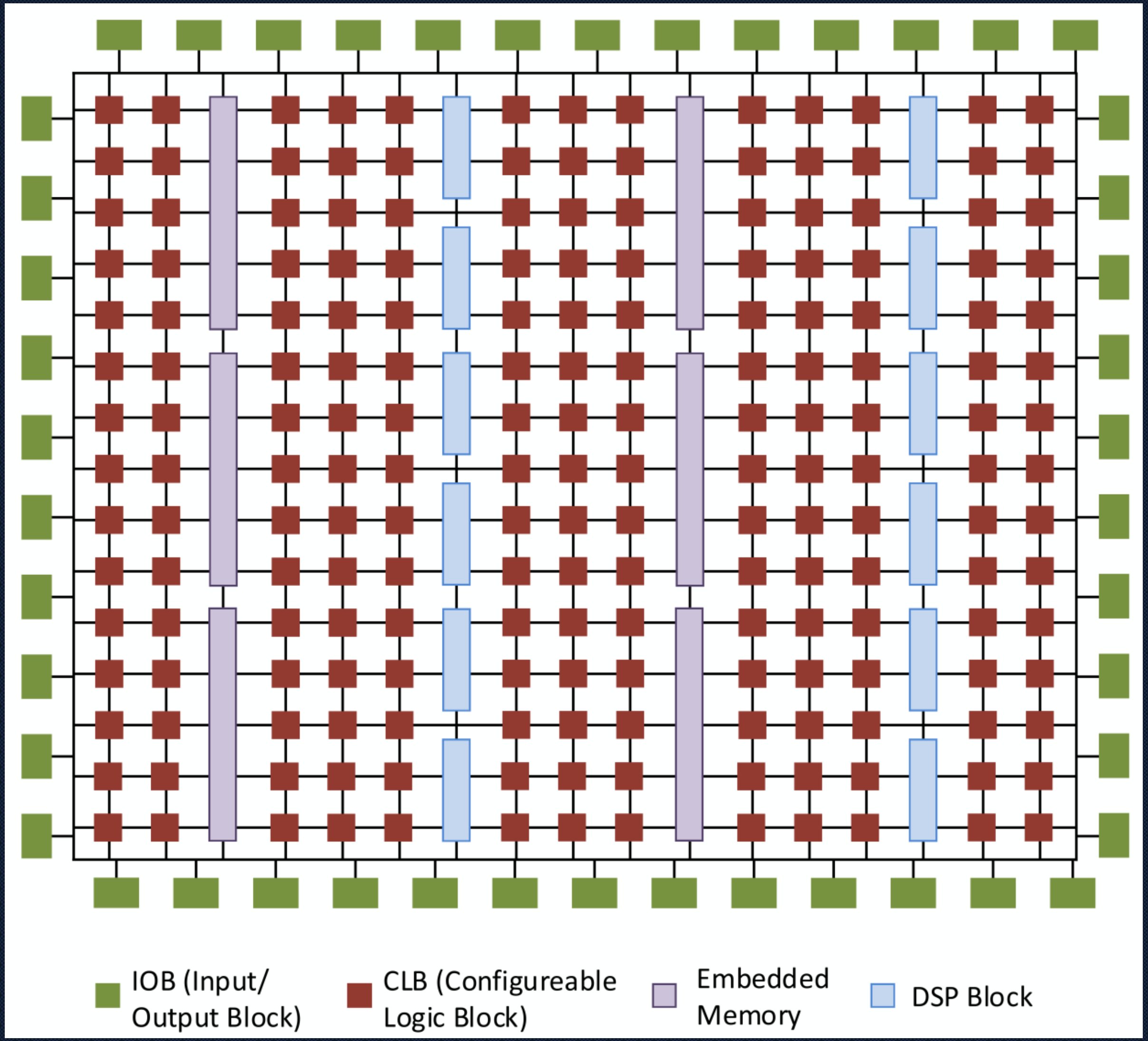
(1)

What are FPGAs?

Field-Programmable Gate Array

Seen as alternative to ASIC

- Flexible Logic
- Flexible routing
- Flexible IO
- Embedded hard modules



(1)

Networking Concepts

- OSI Network Model
- TCP / IP network stack

- Application
- Presentation
- Session
- Transport
- Network Internet
- Data Link Link
- Physical

Networking Concepts

- OSI Network Model
- TCP / IP network stack
- Software Defined Networking

- Application
- Transport
- Internet
- Link

- Control plane
- Data plane
- P4
- OpenFlow (part of Mininet)

Why Smart FPGA Switches?

- Cloud computing

- High latencies
- Large data centres
- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform
- Apache Hadoop

Why Smart FPGA Switches?

- Cloud computing
- Existing solutions

- “Edge” / “gateway” nodes
- Mainframes
- NetFPGA
- Software Defined Networking (SDN)

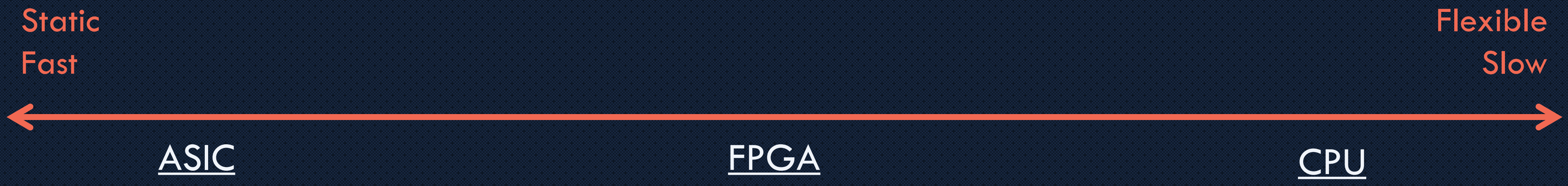
Why Smart FPGA Switches?

- Cloud computing
- Existing solutions
- Drawbacks

- Similar to data centres
- Latency
- Security
- Scalability

Why Smart FPGA Switches?

- Cloud computing
- Existing solutions
- Drawbacks
- Solution
- Do not divert data from existing path
- Appropriate device for computation



Static
Fast

Flexible
Slow

ASIC

FPGA

CPU

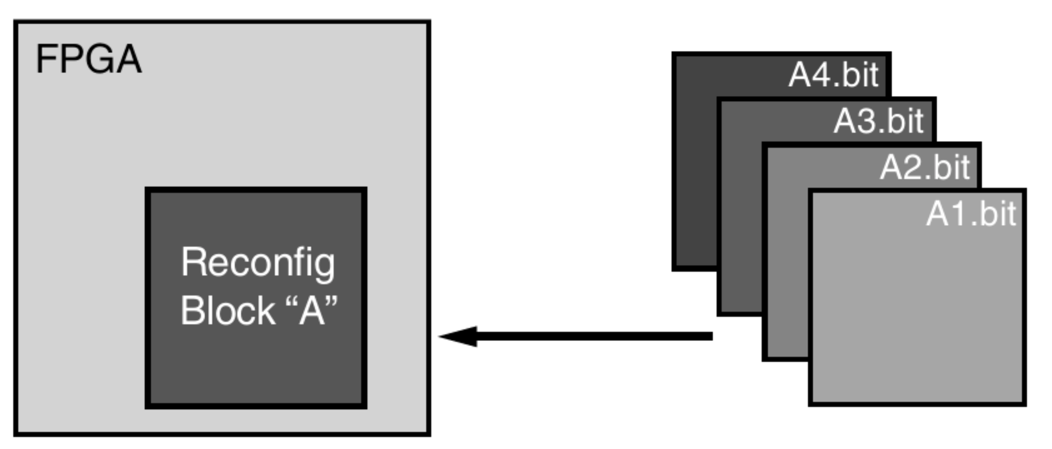
Why Smart FPGA Switches?

- Cloud computing
- Existing solutions
- Drawbacks
- Solution
- FPGAs
- Custom compute architectures – very low latency
- Partial reconfiguration
- Specific hardware for each algorithm

Why Smart FPGA Switches?

- Cloud computing
- Existing solutions
- Drawbacks
- Solution
- FPGAs

- Partial reconfiguration
- Specific hardware for each algorithm



Research and Implementation

- Mininet

- “Emulator for rapid prototyping of Software Defined Networks”
- Open source
- Python API
- Scalable
- Dependency chain

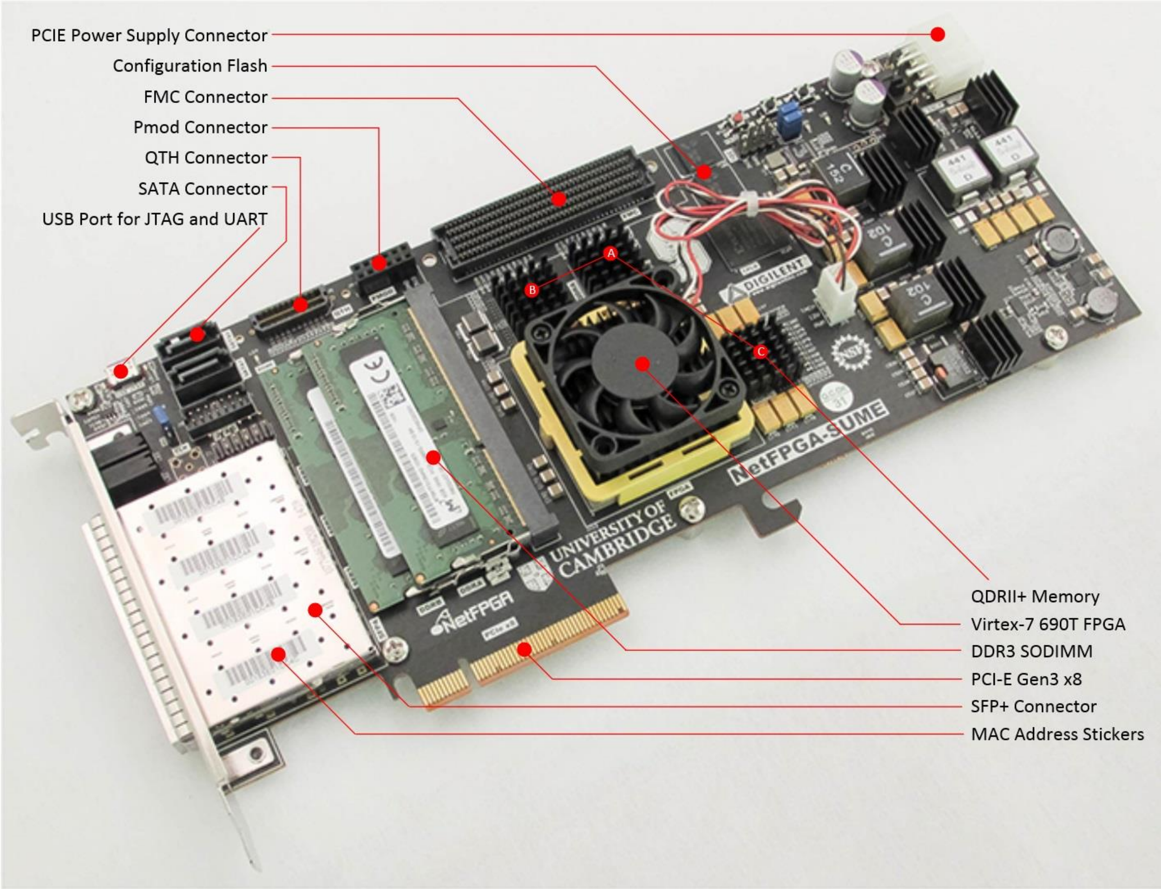
Research and Implementation

- Mininet
- NetFPGA

- PCI Express interface
- Xilinx Virtex / Kintex
- 4 x 1Gbps – 10Gbps Ethernet
- Open source software
- 1G, 10G, SUME, CML
- Standalone or integrated

Research and Implementation

- Mininet
- NetFPGA



Research and Implementation

- Mininet
- NetFPGA
- Software Development

- Python
- Open source (GitHub)
- setuptools
- click
- logging
- git submodules

(6), (7), (8), (9)

Project Management

Agile methodology	Adaptive to change
Kanban Board	Trello
Version control	git
Weekly meetings with supervisor	Comparable to “scrums”
Issues resolved quickly	Access to tools, licensing

(10), (11)

Applications

- Internet of Things (IoT)

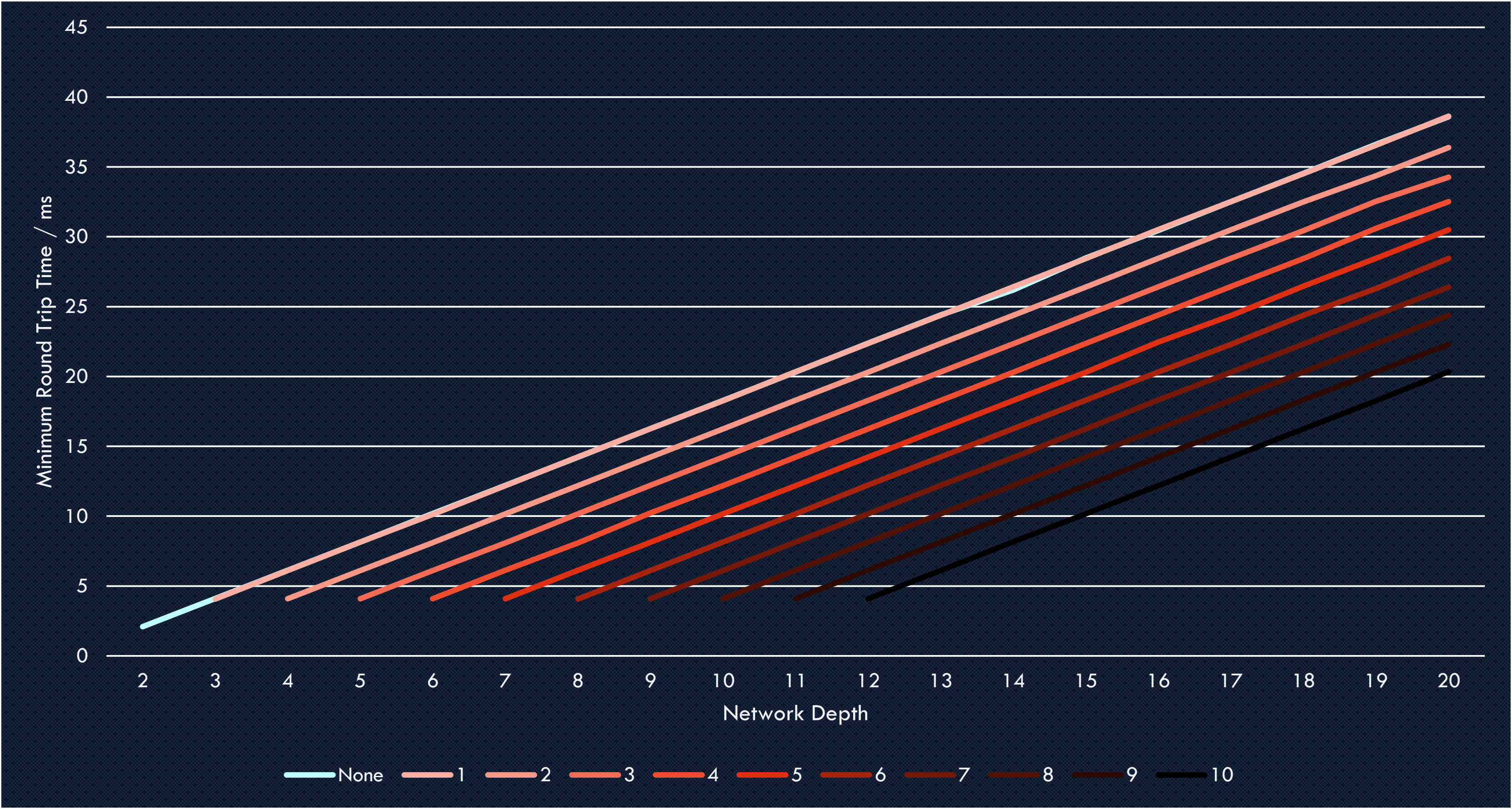
- Education
- Healthcare
- Military
- Finance

Model Results

- Operating Parameters:
- Latency: 1ms
- FPGA latency: 2ms
- Spread: 1 (vertical linear topology)

Model Results

- Operating Parameters:
- Latency: 1ms
- FPGA latency: 2ms
- Spread: 1 (vertical linear topology)



Demo

Questions

References

1. ES3F1 Lectures, Suhaib Fahmy
2. xilinx.com
3. mininet.org
4. netfpga.org
5. reference.digilentinc.com
6. python.org
7. github.com
8. https://github.com/pypa/setuptools
9. https://palletsprojects.com/p/click/p4.org
10. trello.com
11. git-scm.com