MANCHESTER 1824

The University of Manchester

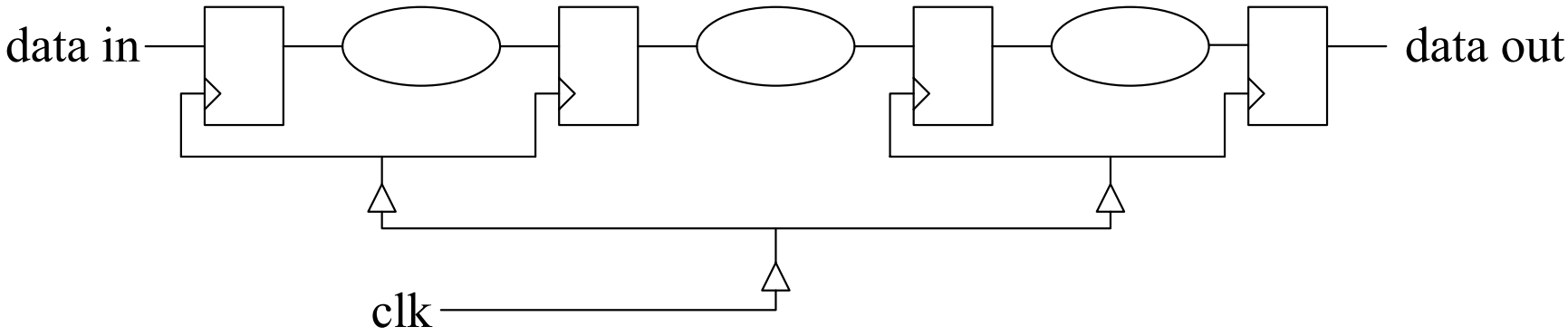
# How to Design Fast Asynchronous Routers for Asynchronous On-chip Networks

Wei Song
Supervisor: Doug Edwards
Advanced Processor Technologies Group

# Index

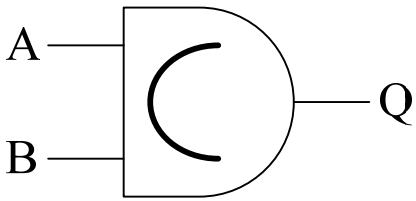
- **What is *asynchronous circuit*?**
- Why to use *on-chip network*?
- Why *asynchronous on-chip network* is slow?
- How can we improve it?
- So, what's next?

# Synchronous Circuit



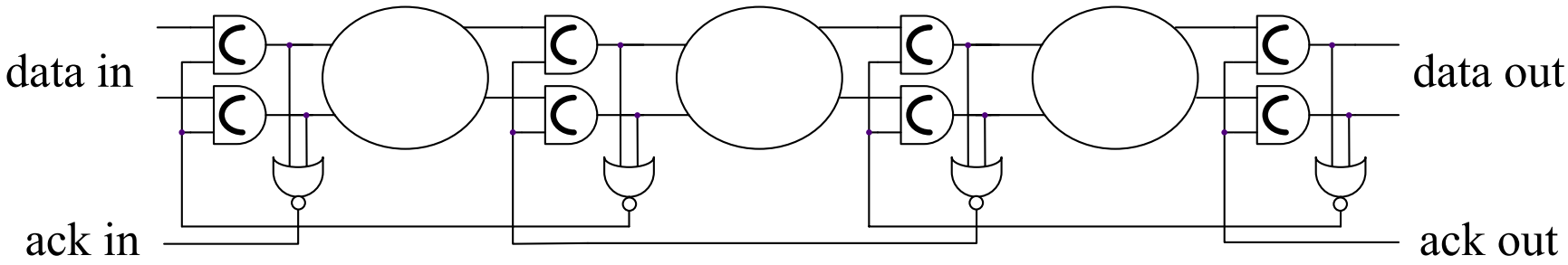

- Pipeline style
- Strict timing assumption
- A global clock driven by a balanced tree

# Asynchronous Circuits – C-element




| A | B | Q’ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | X | Q |
| X | 0 | Q |
| 1 | 1 | 1 |

# Asynchronous Pipeline

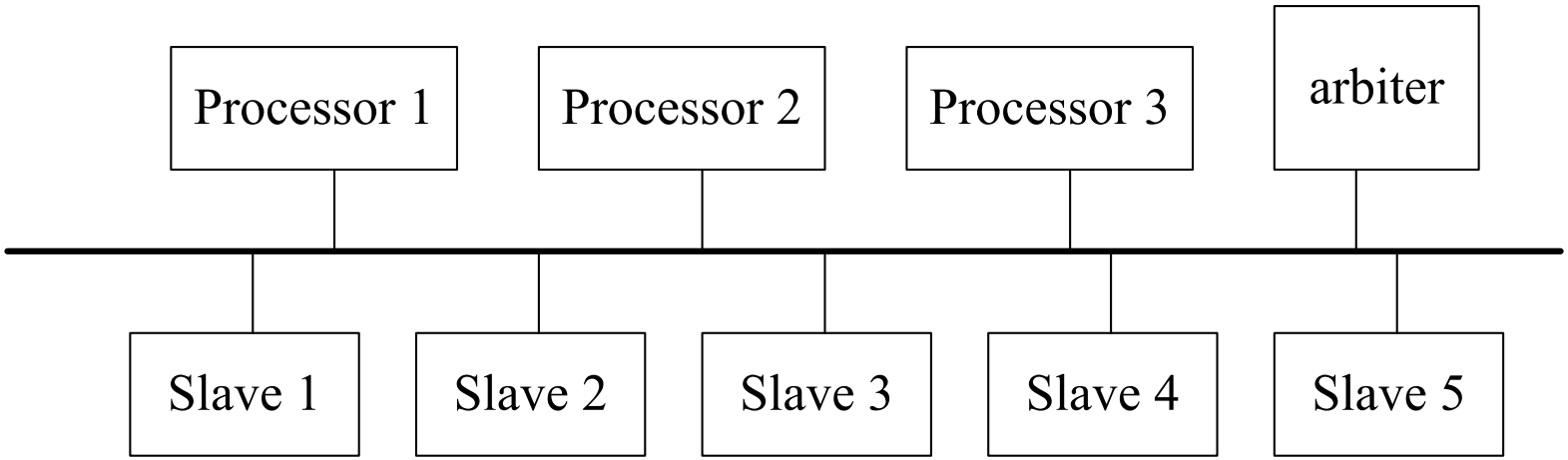



- Handshake
- Nearly delay insensitive (no timing assumptions)
- Power efficient (no global clock)
- Complicated (larger area)

# Index

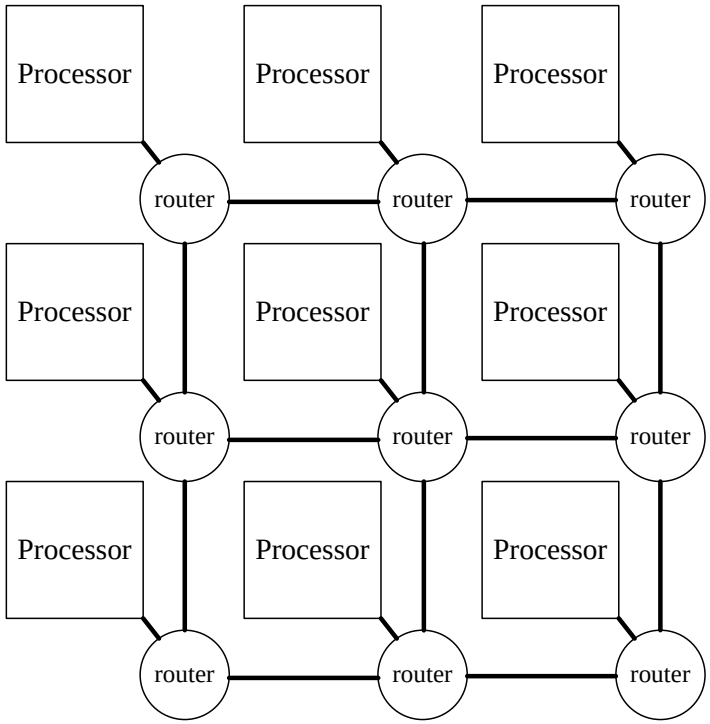
- What is *asynchronous circuit*?
- **Why to use *on-chip network*?**
- Why *asynchronous on-chip network* is slow?
- How can we improve it?
- So, what's next?

# Bus Based Multiprocessor System



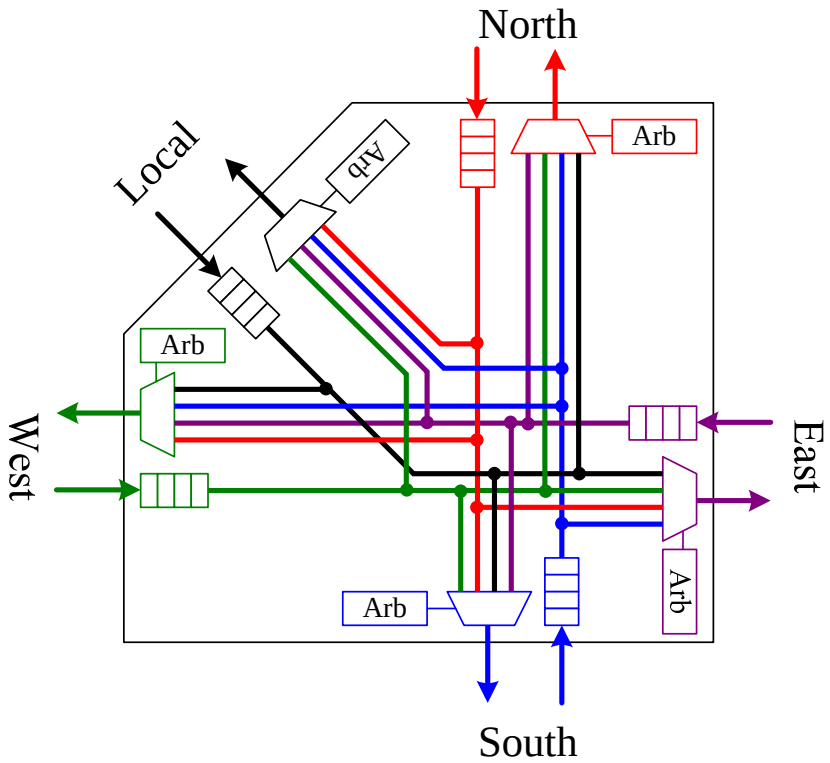

- A shared communication fabric
- One master at one time
- Bandwidth constrained
- Fixed communication latency

# A Mesh Network-on-Chip (NoC)



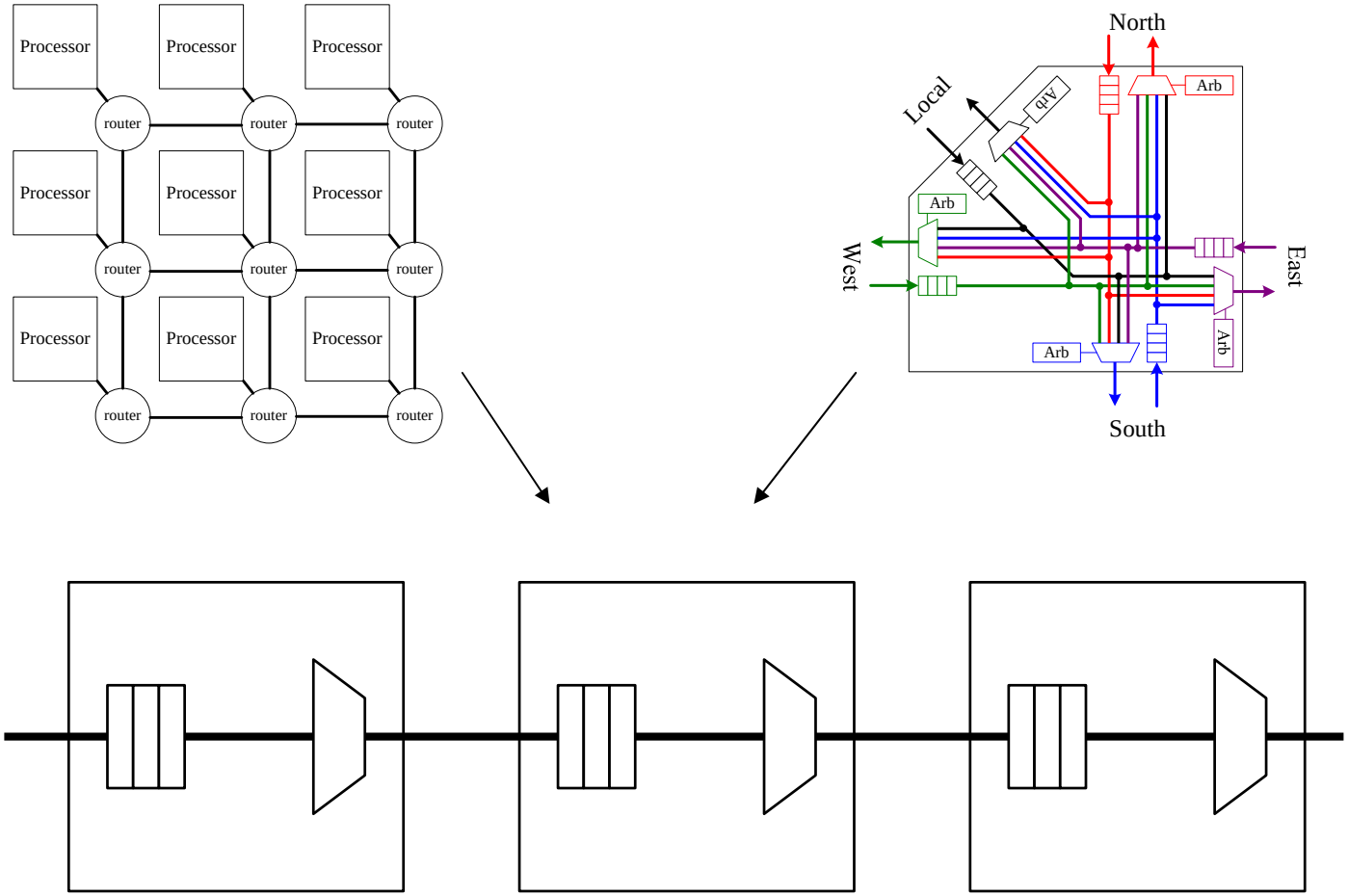

- Distributed communication resource
- Scalable bandwidth
- Multiple master and slave pairs at a time
- Variable communication latency

# The Router for NoC




- 5 ports
- Duplex channels
- Input buffer
- Arbiter
- Crossbar (Muxes)

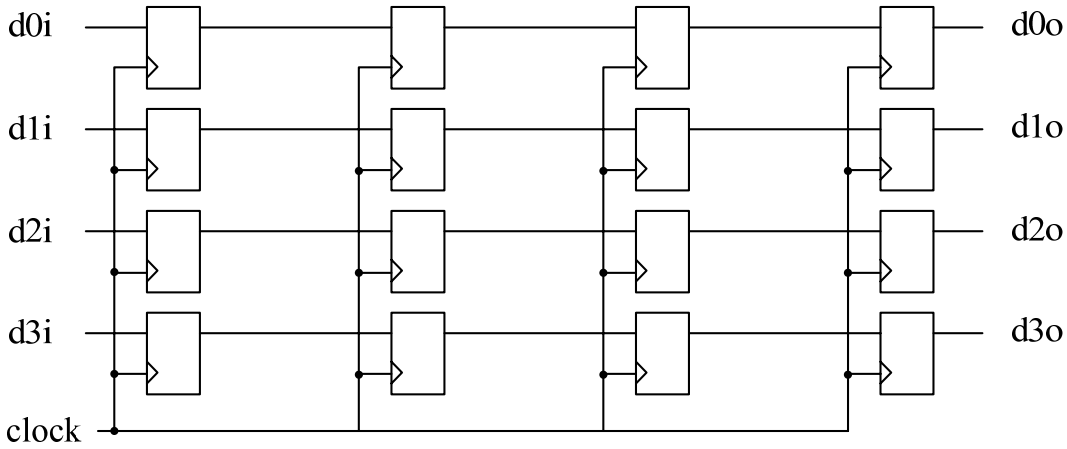
# Data Path of a NoC



# Index

- What is *asynchronous circuit*?
- Why to use *on-chip network*?
- **Why *asynchronous on-chip network* is slow?**
- How can we improve it?
- So, what's next?

# A 4-bit Synchronous Pipeline

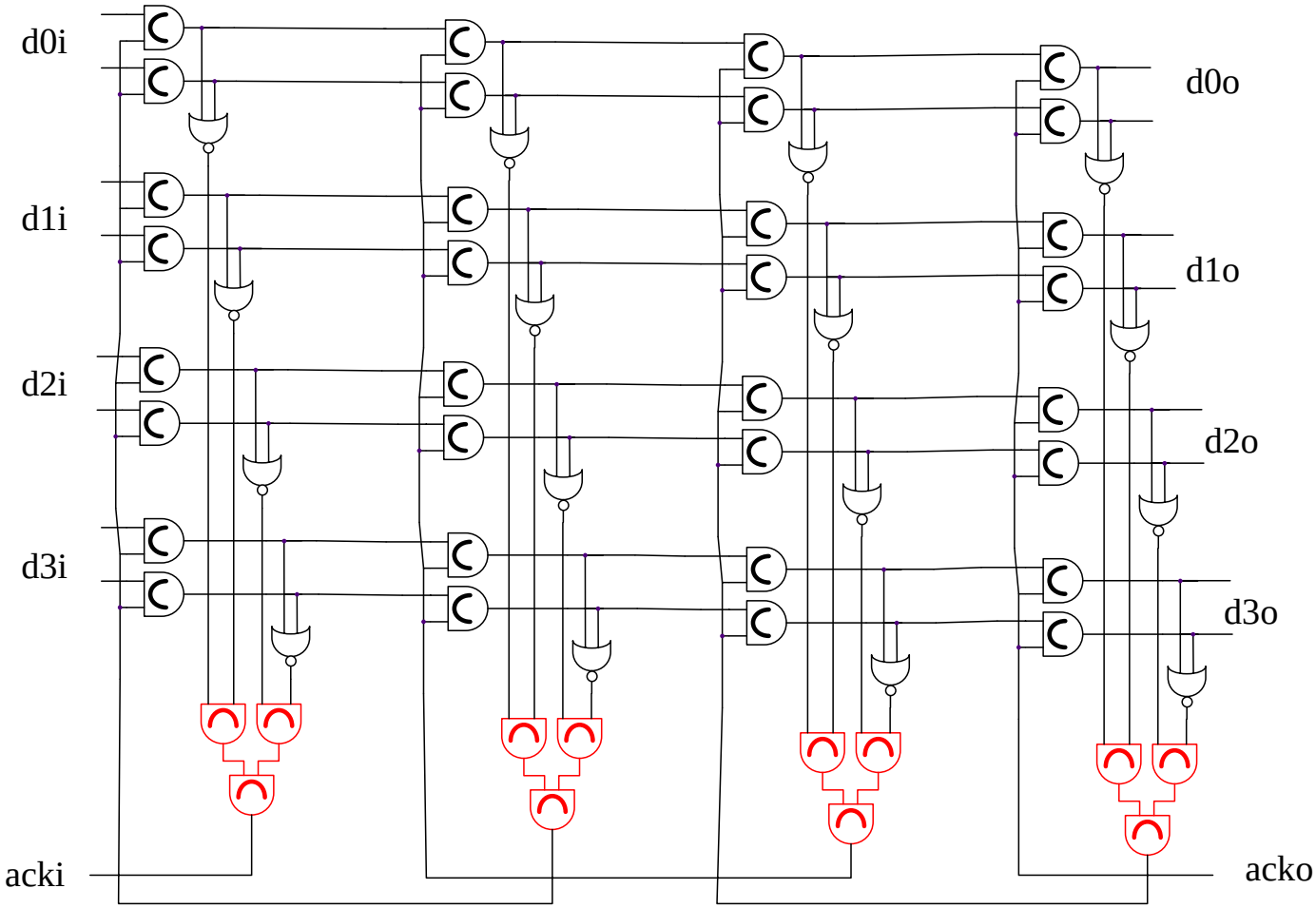



- Data are synchronised by the global clock
- No significant speed difference with the 1-bit pipeline

MANCHESTER 1824

The University of Manchester

# A 4-bit Asynchronous Pipeline



# Reasons of the Low Speed

- Asynchronous pipelines deliberately detect the arrival of data
- A big C-element tree in the loop!

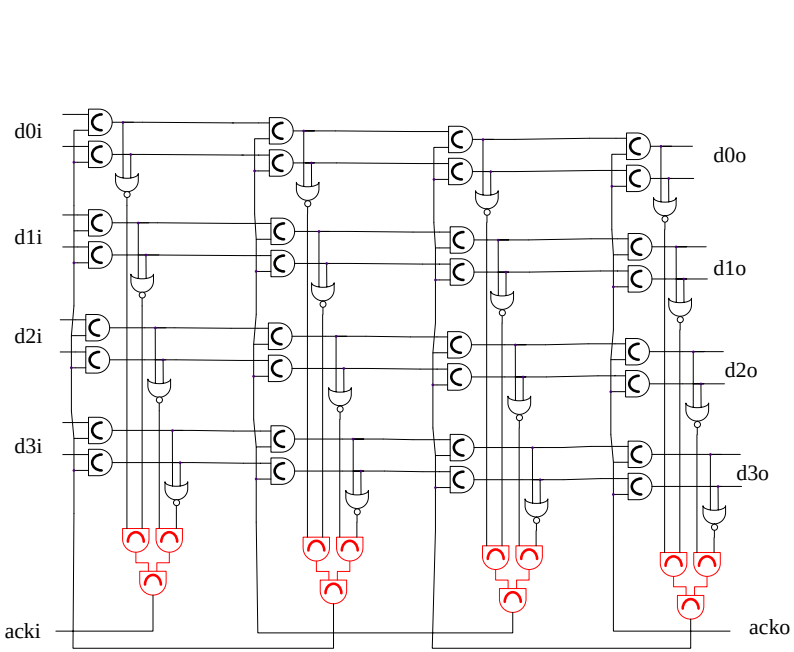
# Index

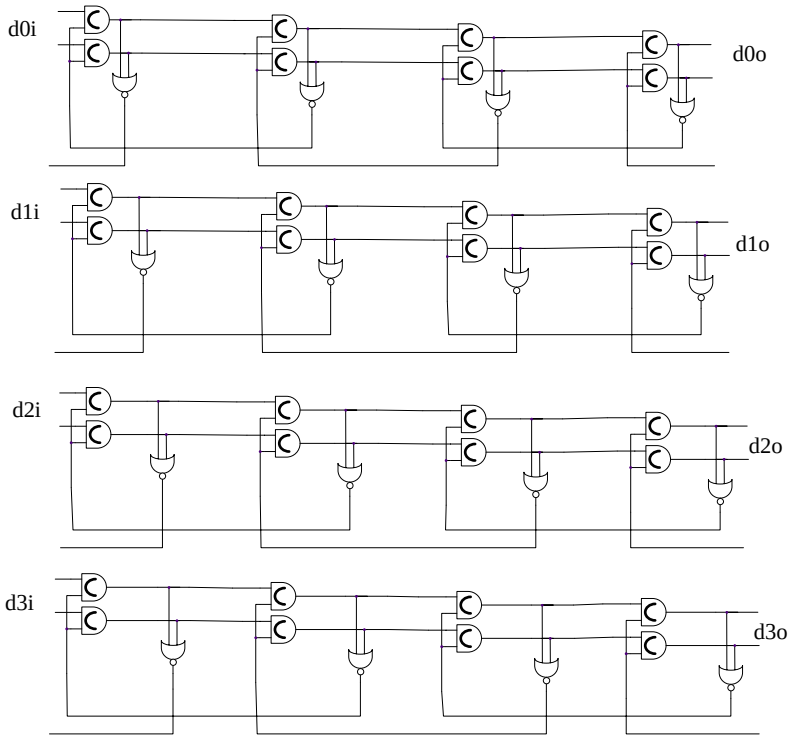
- What is *asynchronous circuit*?
- Why to use *on-chip network*?
- Why *asynchronous on-chip network* is slow?
- **How can we improve it?**
- So, what's next?

MANCHESTER 1824

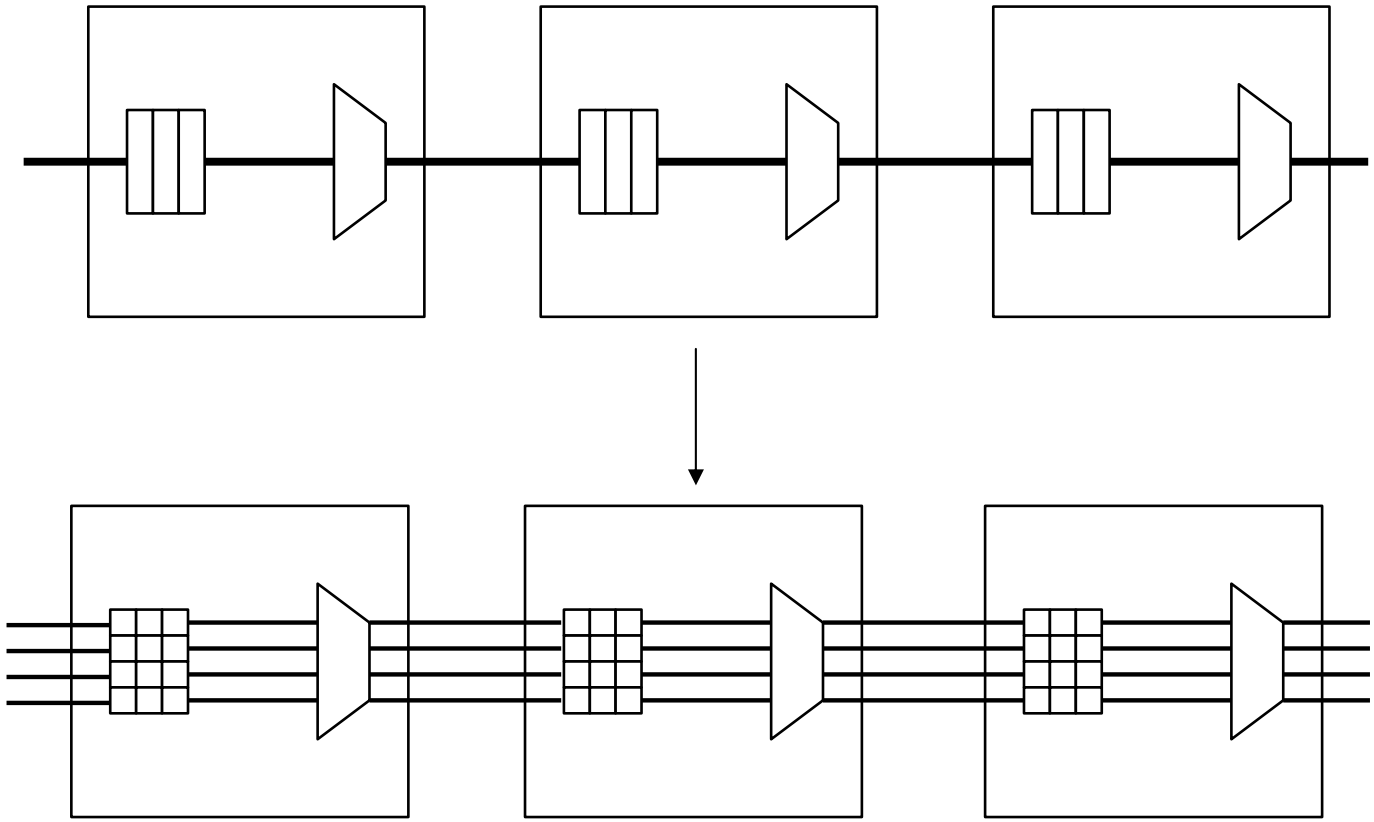
The University of Manchester

# Channel Slicing



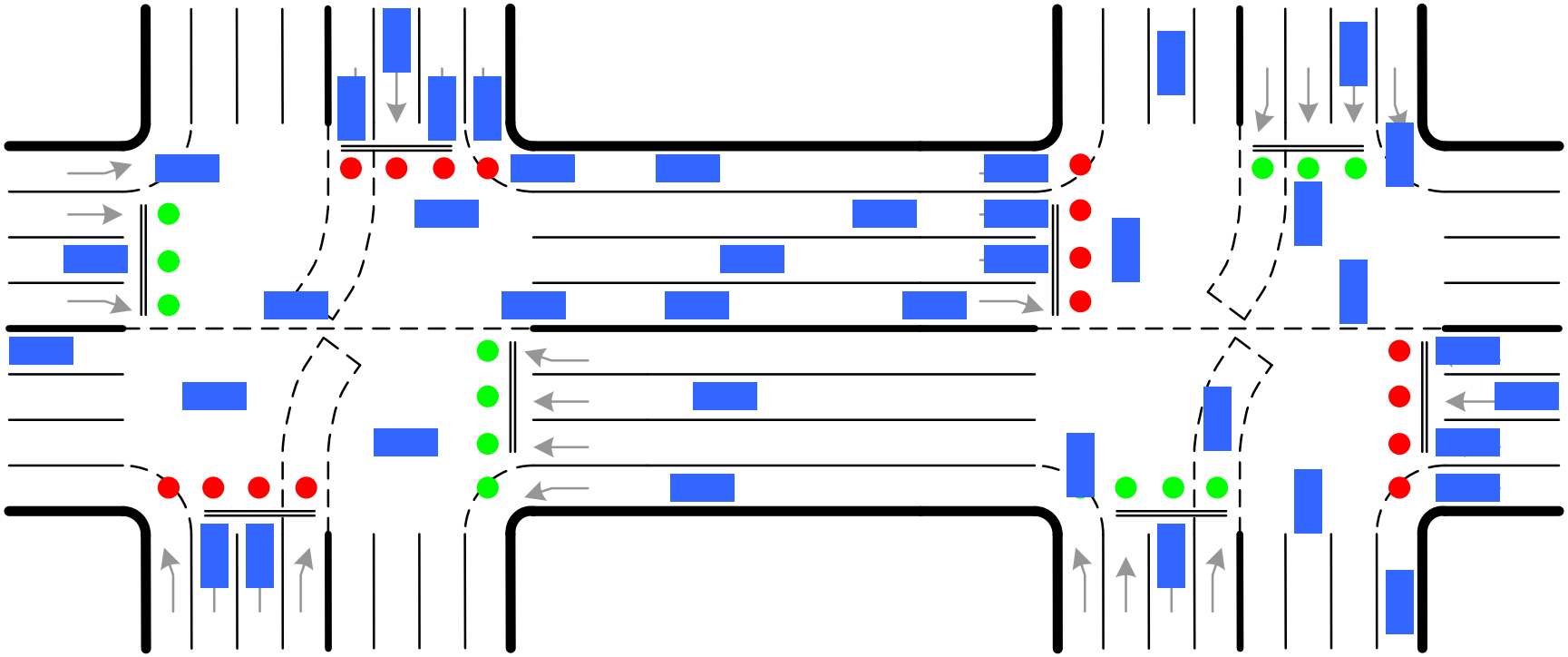
# Re-Synchronisation (1)



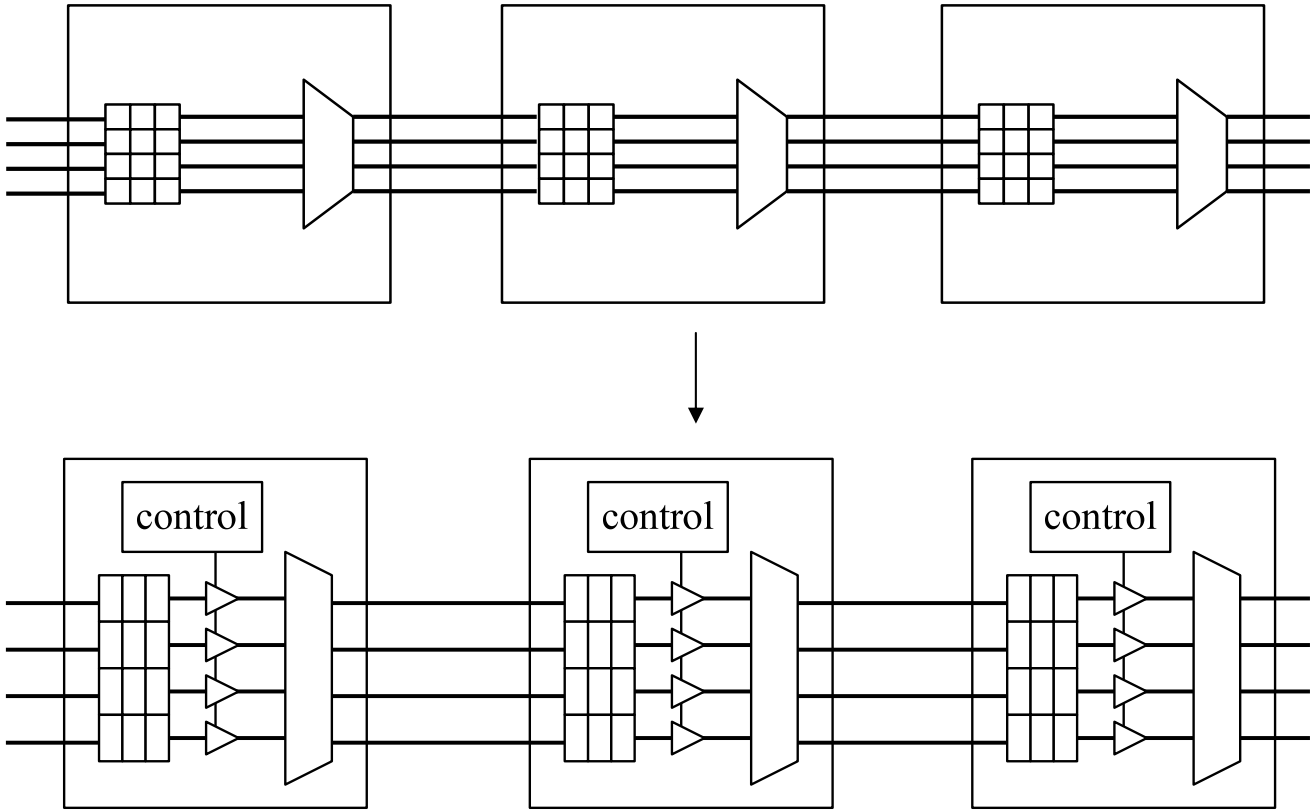
MANCHESTER 1824

The University of Manchester

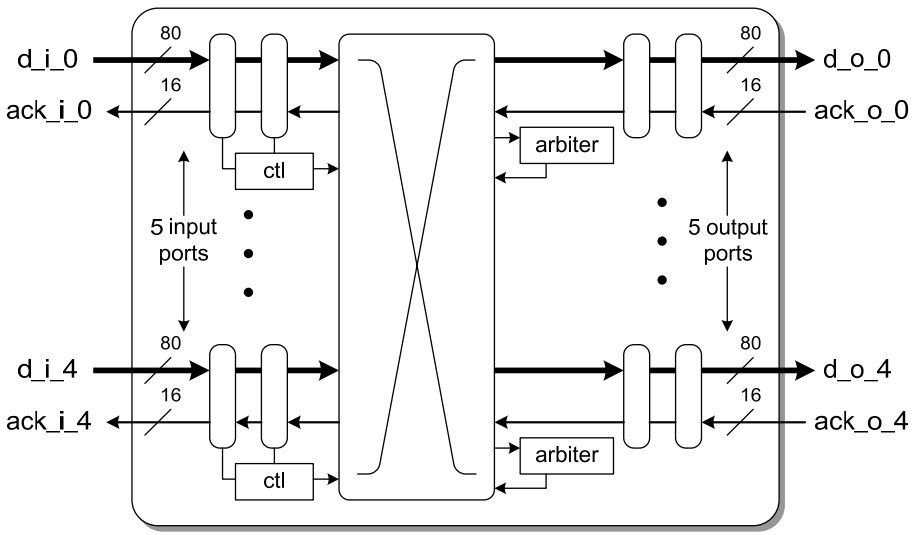
# Re-Synchronisation (2)



# Re-Synchronisation (3)

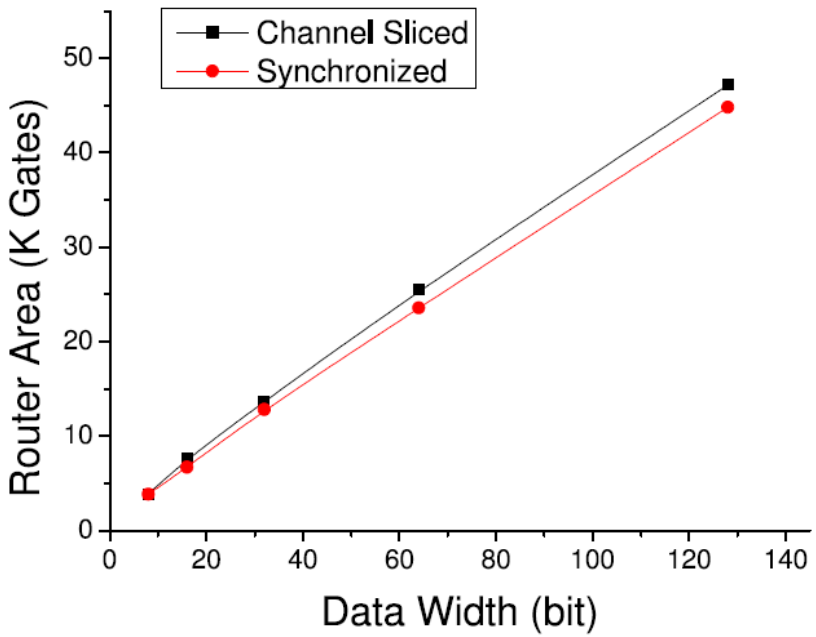


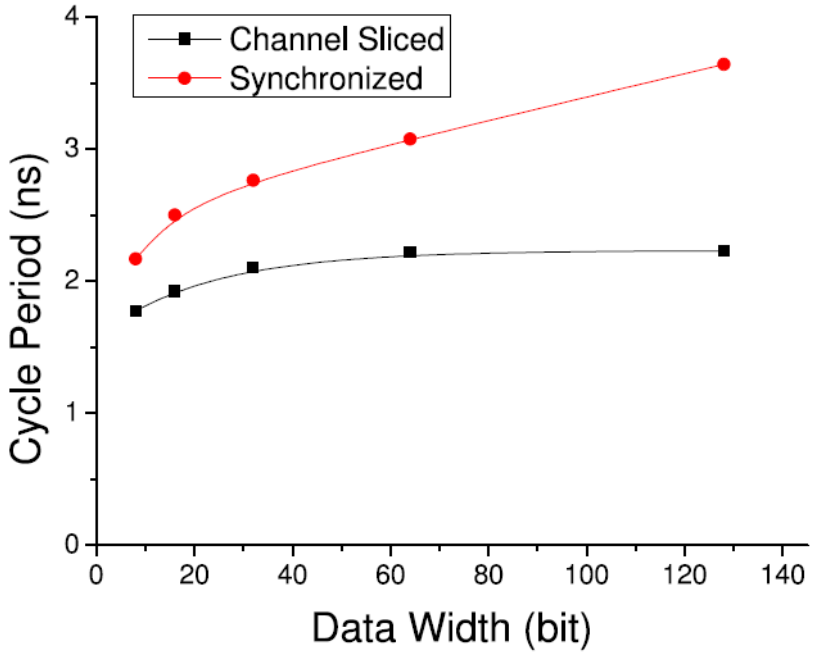
# Hardware Implementation




- Verilog HDL+STG(Petrify)
- Layout Implementation
- Faraday 130 nm Technology
- 12.6K Gates (50,000um$^2$)
- 0.3*0.3mm$^2$
- Channel Sliced 450MHz
- Synchronised 360MHz

# Performance



MANCHESTER 1824

The University of Manchester

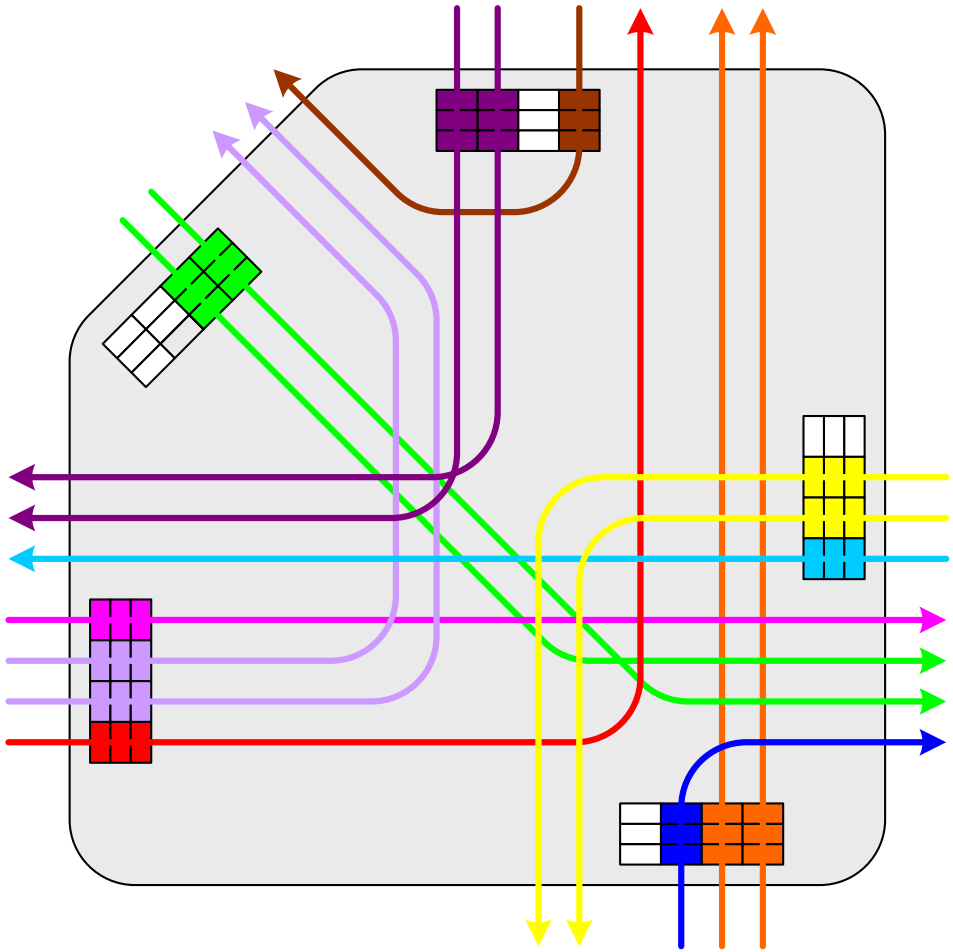
# Index

- What is *asynchronous circuit*?
- Why to use *on-chip network*?
- Why *asynchronous on-chip network* is slow?
- How can we improve it?
- **So, what's next?**

# Spatial Division Multiplex

- Frequently Re-synchronisation will compromise the speed
- Sub-channels should run independently
- Sub-channels could transmit different messages
- Multiple messages could be transmitted by the same channel but on different sub-channels

# Spatial Division Multiplex (con.)



# Conclusion

- Asynchronous Circuits
  - Delay insensitive, low power
- On-chip Network
  - Distributed communication fabric, scalable bandwidth
- Asynchronous On-chip Network
  - The C-element tree in synchronisation compromises speed
- Channel Slicing
  - Let sub-channels run independently, fast
- SDM
  - Let more messages share the fabric simultaneously

MANCHESTER 1824

The University of Manchester

# Thanks!