THE UNIVERSITY OF UTAH

Design of High Speed Synchronous Elastic Flow Network-On-Chip Router

Why Network on Chip

- Busses have limitations
  - Architecture can be slow depending on Intellectual Property (IP) core count
    - IP range from processors, memory, or specific processors such as Digital-Signal-Processors (DSPs)
  - Prevents true concurrency
    - If System-on-Chip (SoC) has more than one processor core only one core can communicate on the bus at a given time
  - Long wire delays
  - Scalability[1]
    - Multi layer bus
      - Timing closer challenging
      - Hard to predict wire delays during design phase until physical layout

---

1. Sharma, Swati, Chandra Mukherjee, and Ashish Gambhir. "A Comparison of Network-on-chip and Buses.“

Advancement In Technology

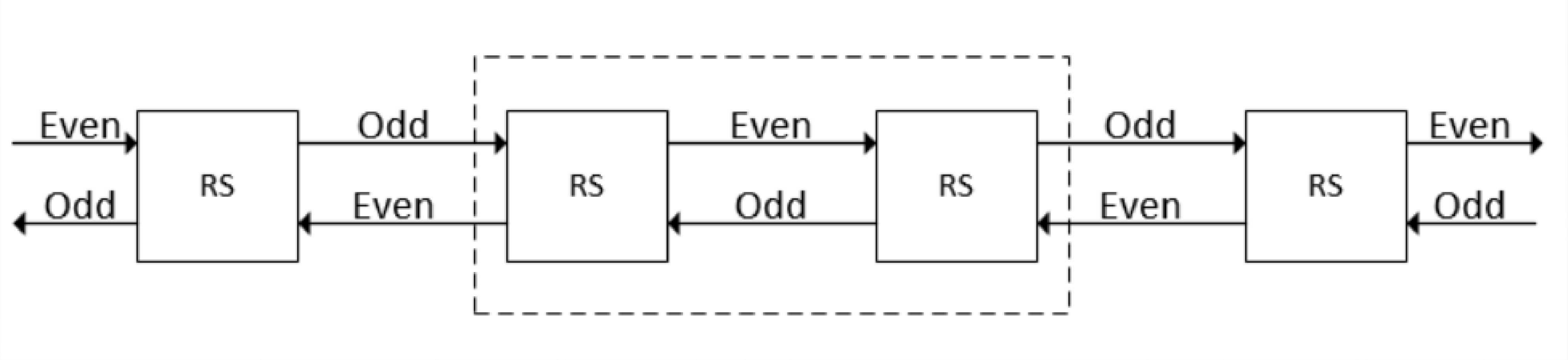
- New process nodes allow for increase in chip density
  - More logic
  - Integrate more IP into SoC
- Wire delay is not decreasing in respect to gate delay
  - Each new generation gate delay scales (decreases) linearly
  - Wire delay increases each new generation compared to gate delay

Research

- Previous work
  - Deign and Integration of High Speed Relative Timed Network-on-Chip Routers[1]
    - Asynchronous RT NoC
- New Work
  - Design of a high speed synchronous Network-on-Chip (NoC) that is similar to asynchronous NoC
    - Implemented asynchronous behavior
      - Latch vs DFF
        - Decrease area, power, delay
      - Control Handshaking

1. D. S. Takur, Design and integration of high speed relative timed network-on-chip routers, Master's thesis, The University of Utah, 2016.

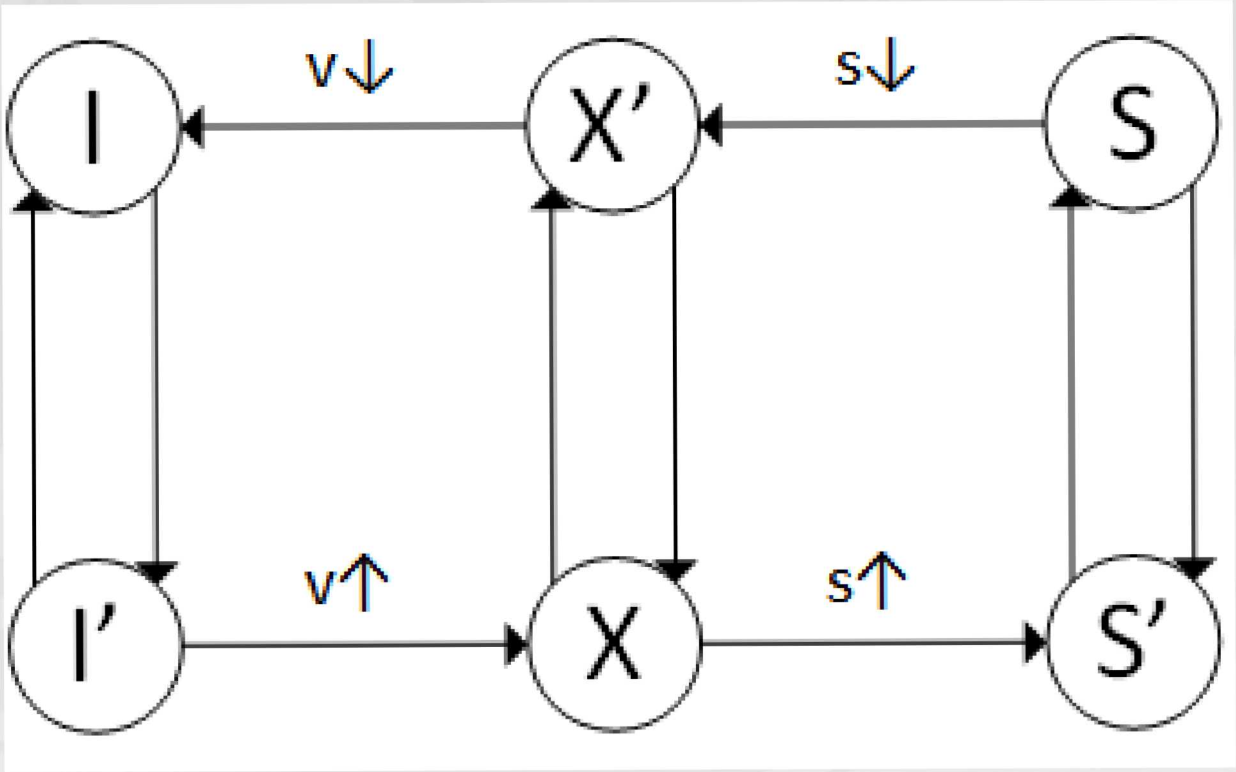
Synchronous Elastic Flow (SELF) Protocol



- Interleaved SELF Channel
  - Inputs on one clock edge, outputs on other, which interleaves data flow
  - Left and right channels are active on alternate edges
  - Controllers are implemented on opposite clock polarity

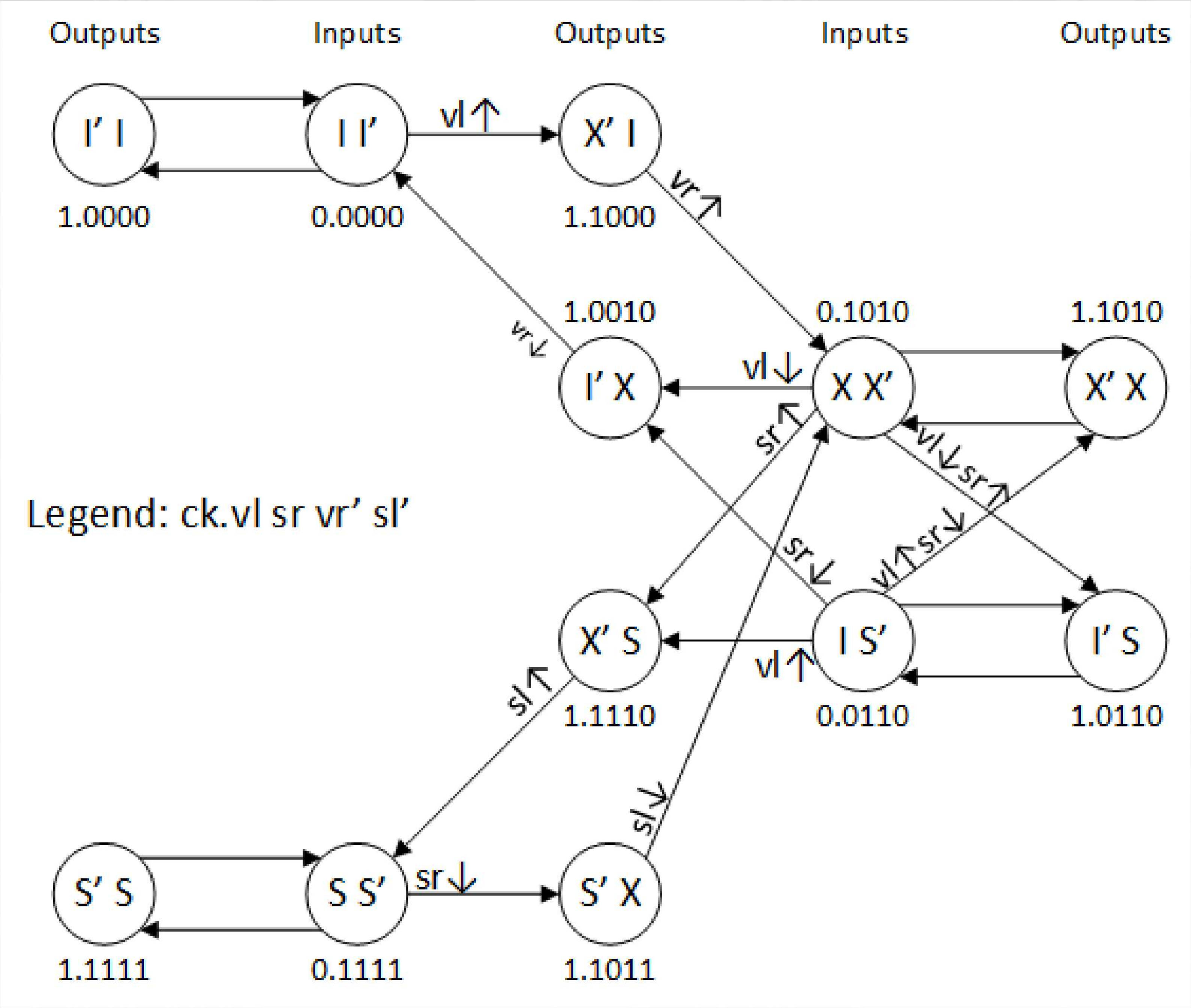
SELF Continued

- Interleaved Channel Protocol
  - Channel signals:
    - v: valid
    - s: stall
  - Three states:
    - I: Idle
    - X: Xfer
    - S: Stall



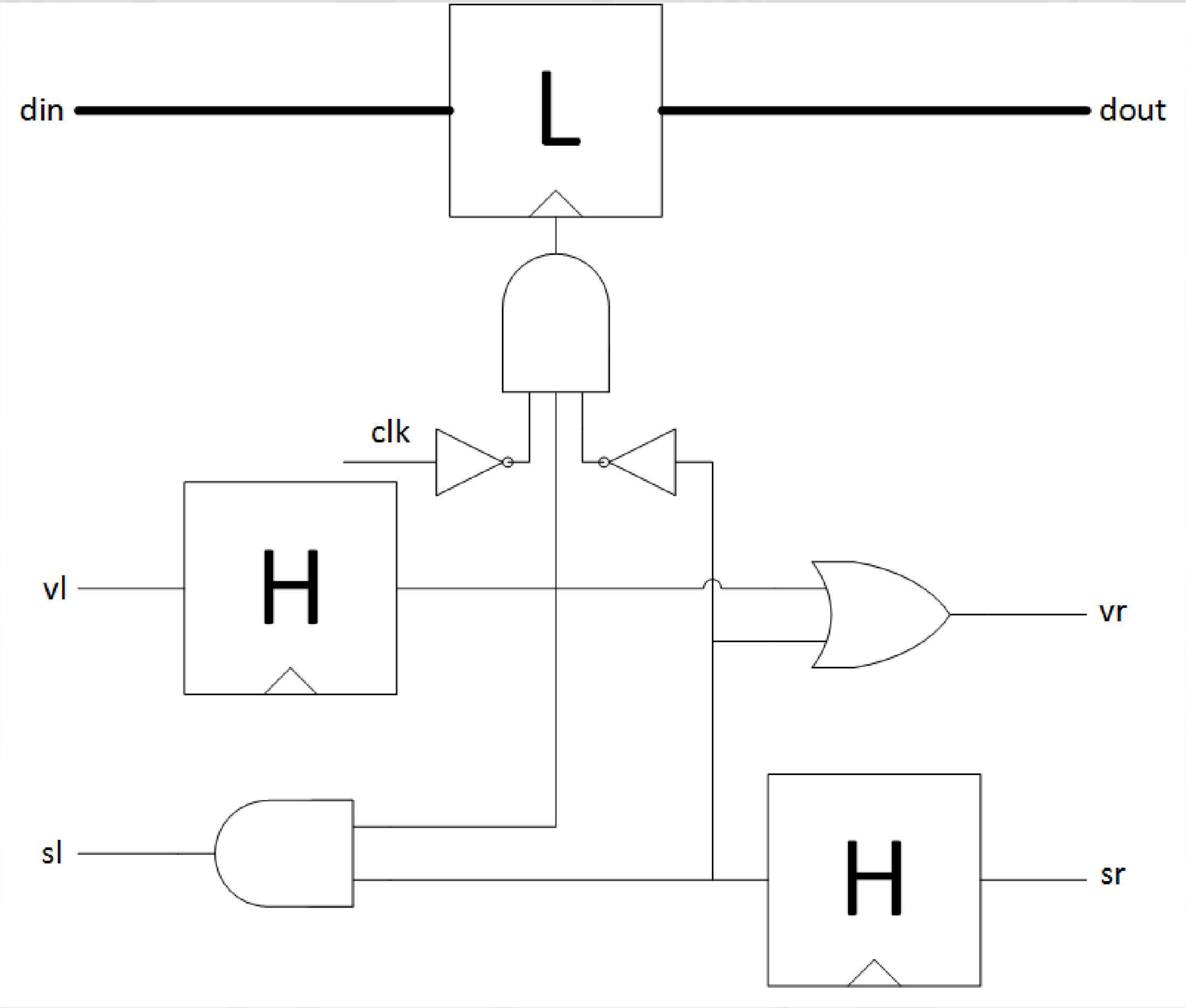
SELF Continued

- State graph composed of left and right channels

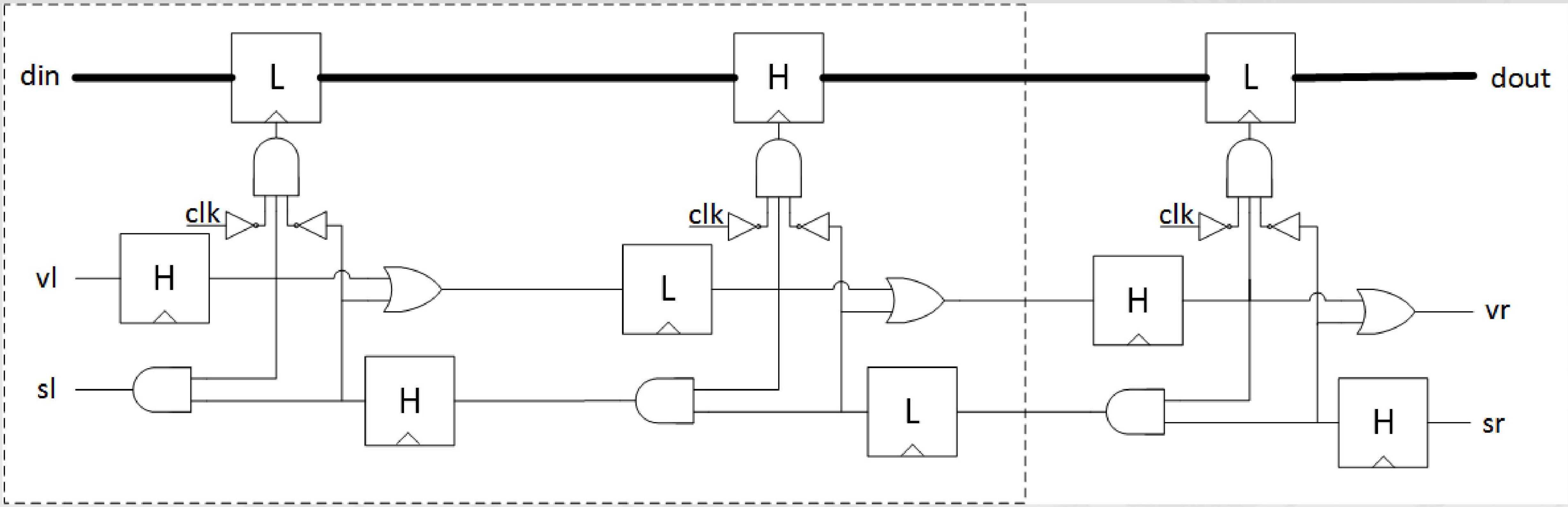


SELF Protocol Implementation

- Latency-insensitive (LI) systems[1]
  - Separates communication from computation
- Synchronous interlocked pipelines
  - Asynchronous like protocol
    - request/acknowledge handshaking
    - Valid left (vl), and stall right (sr) are analogous to req/ack signals

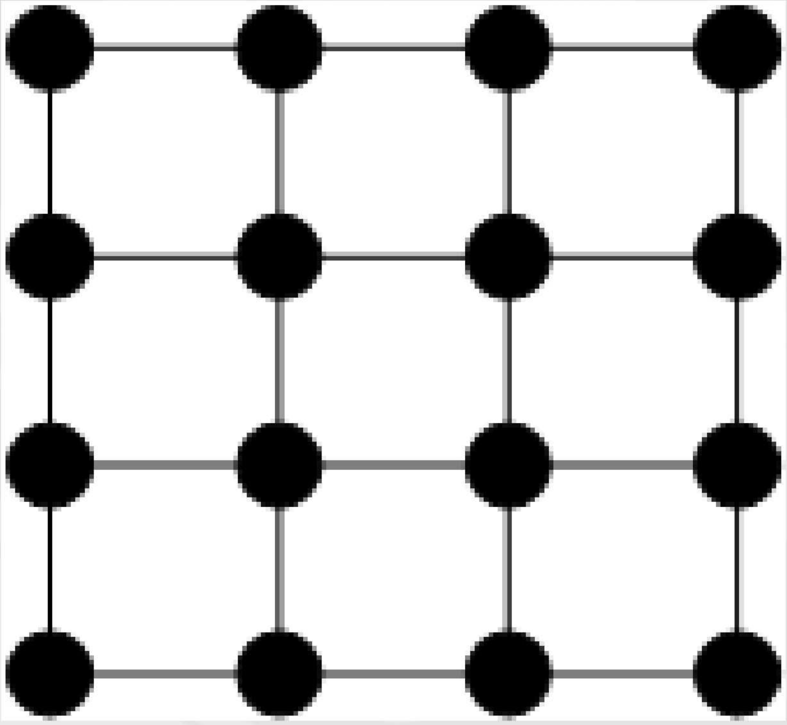


SELF Protocol Implementation Continued



- Shows implementation of interleaved full buffer partition
- Two half-buffer combined into a full “flop-level” buffer

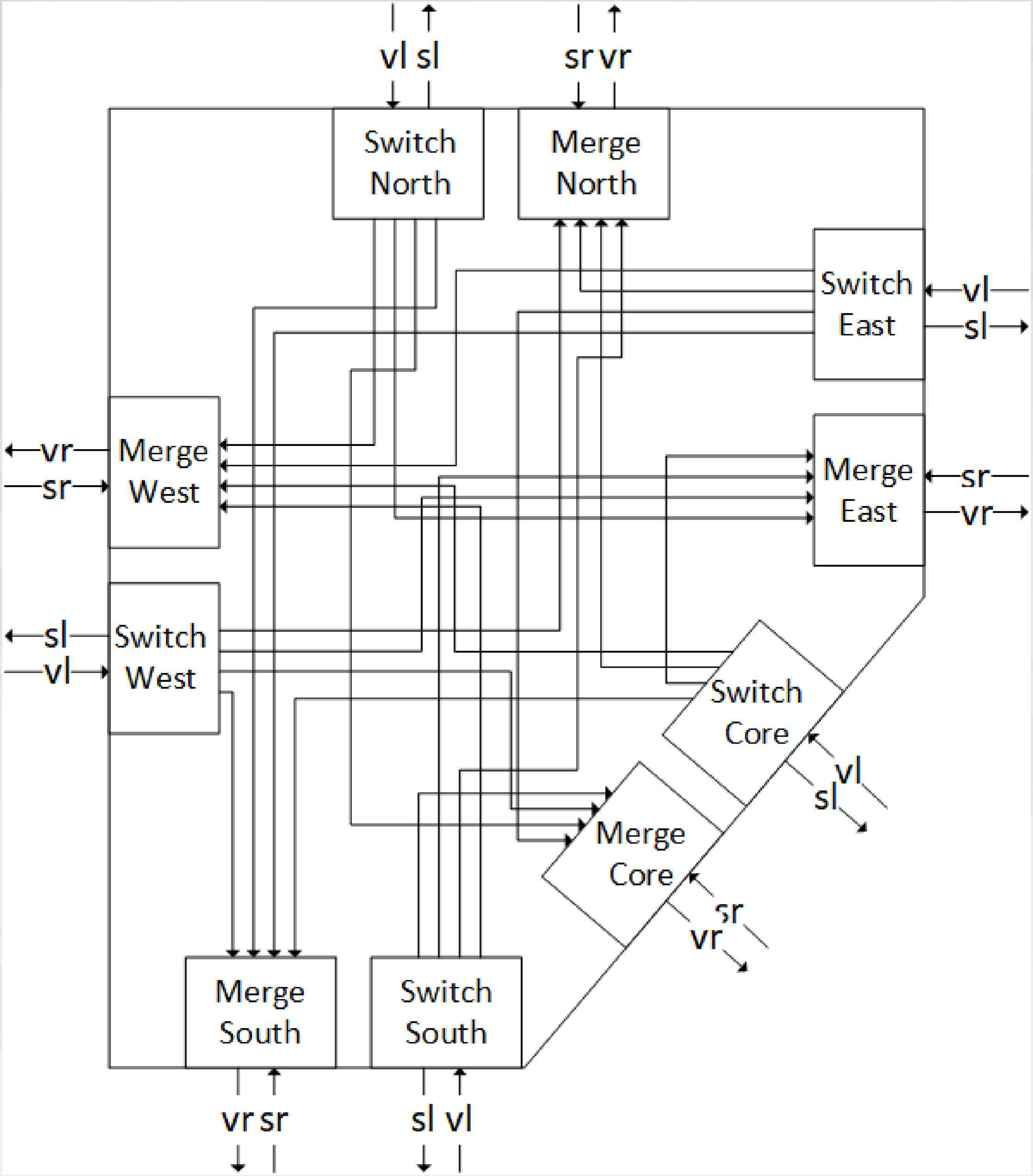
Network Overview



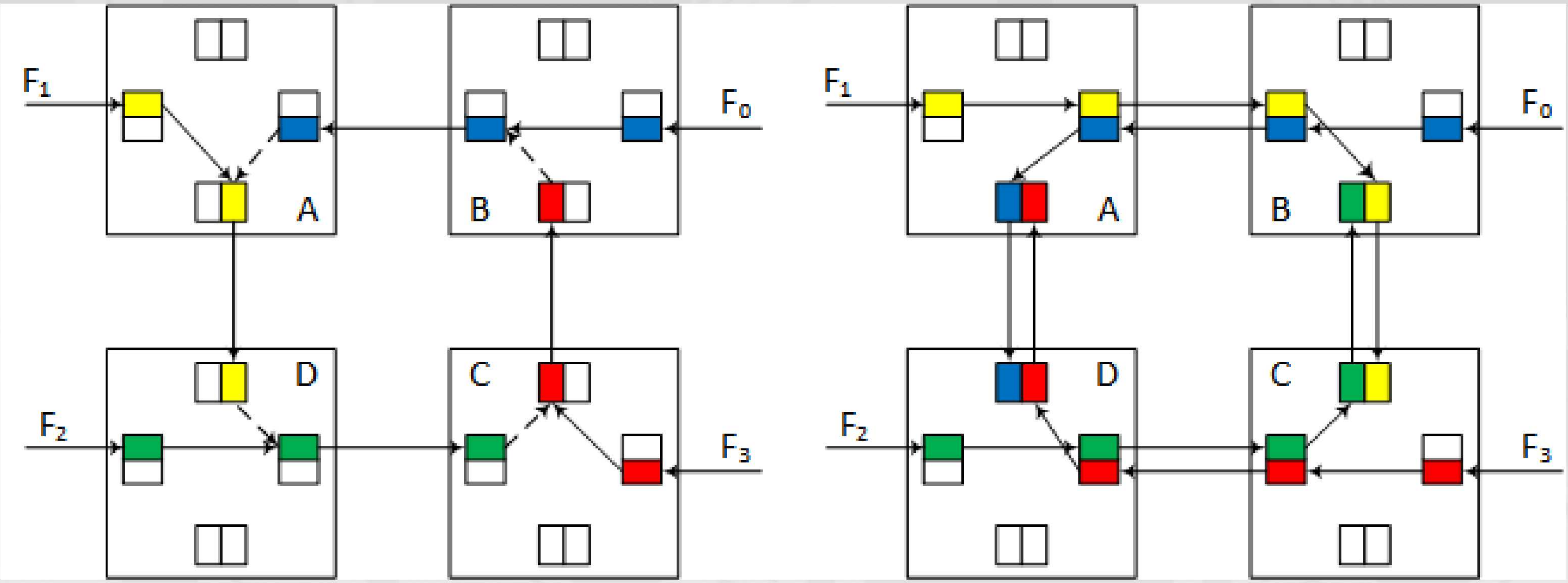
- Distributed Memory Multiprocessors
- Connected by mesh based array of NoC
- XY Plane routing
  - Traverses X, then Y

SELF Router

- 5 port router
- Modules
  - Switch – forks incoming data to the merges internal to the router
  - Merge – merges internal data from each fork

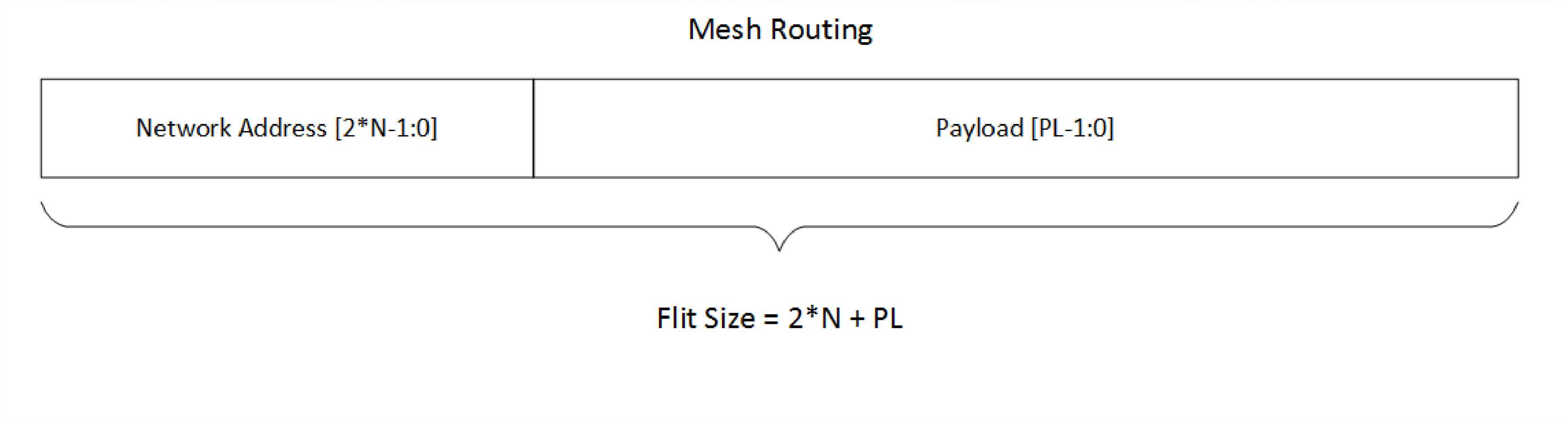


XY Routing Prevents Deadlock



- XY routing algorithm is deadlock free
  - Implemented as lookup table
  - Route predefined before being inserted into network
  - Traverse X then Y

Address Routing



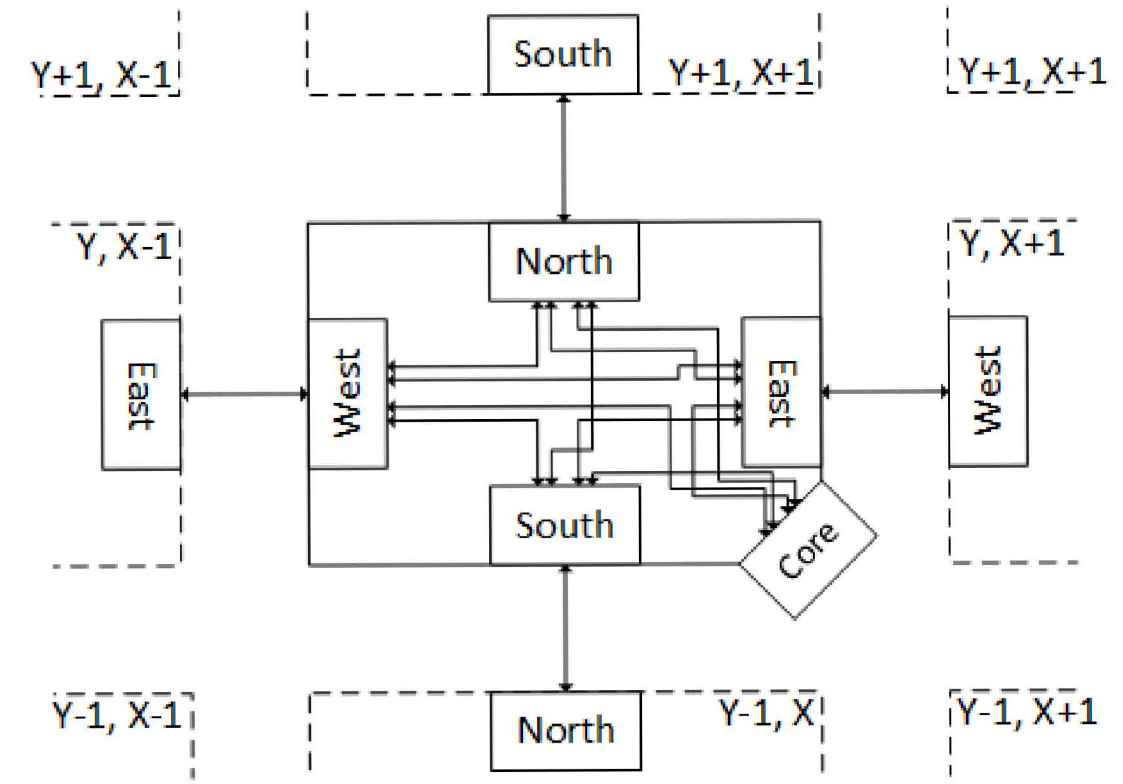
Mesh Routing

Flit Size = 2*N + PL

- For 5 Port Router, Top two bits in Network Address range are current port address
- N = worst case hop count to traverse network

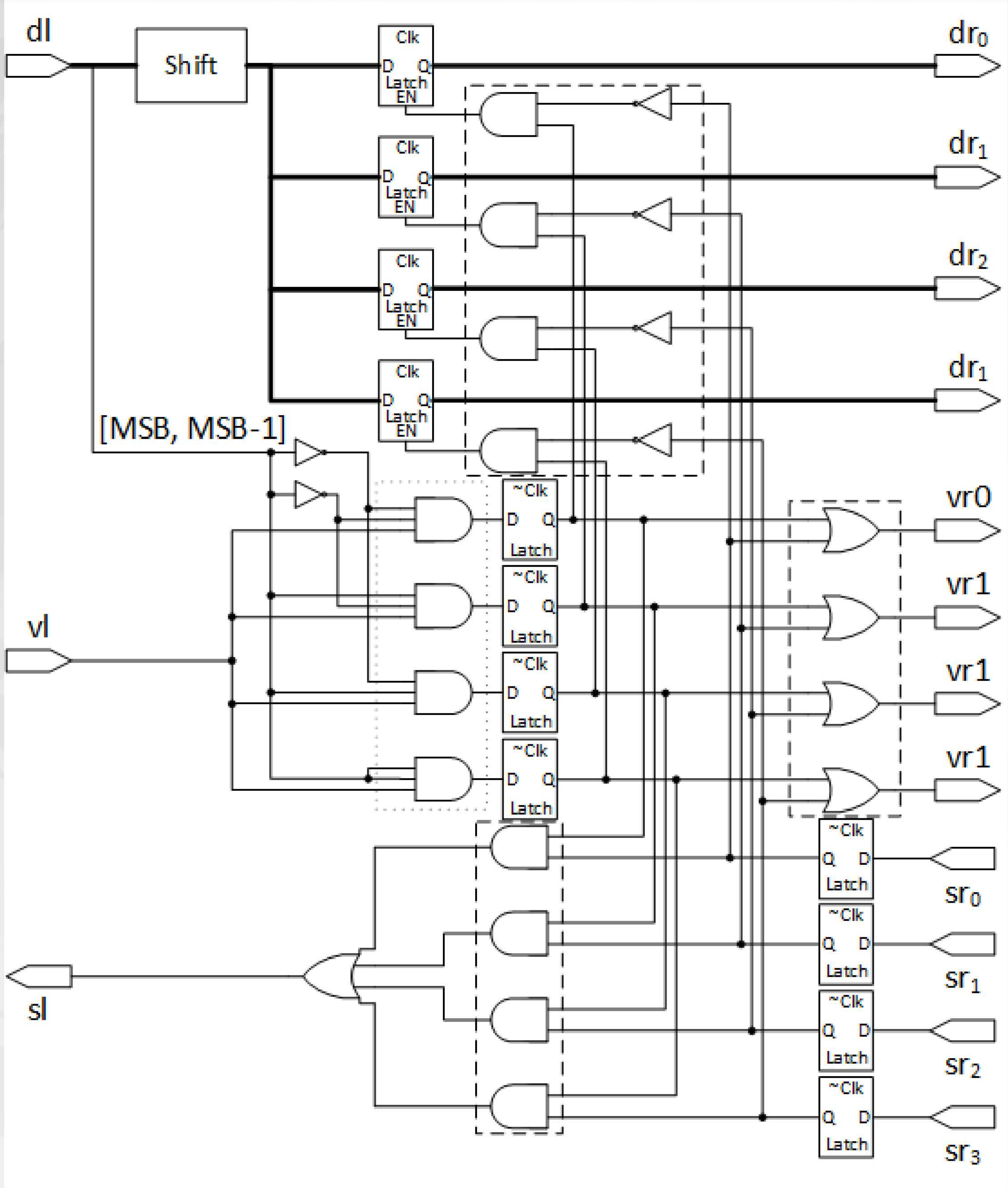
Address Routing Continued

- XY Routing Algorithm
- 5 Port Router
  - 2 bits required per hop
  - Direction
    - 01 North
    - 00 East
    - 10 South
    - 11 West
  - Address termination by repeating incoming port address
  - Return address provided after routing

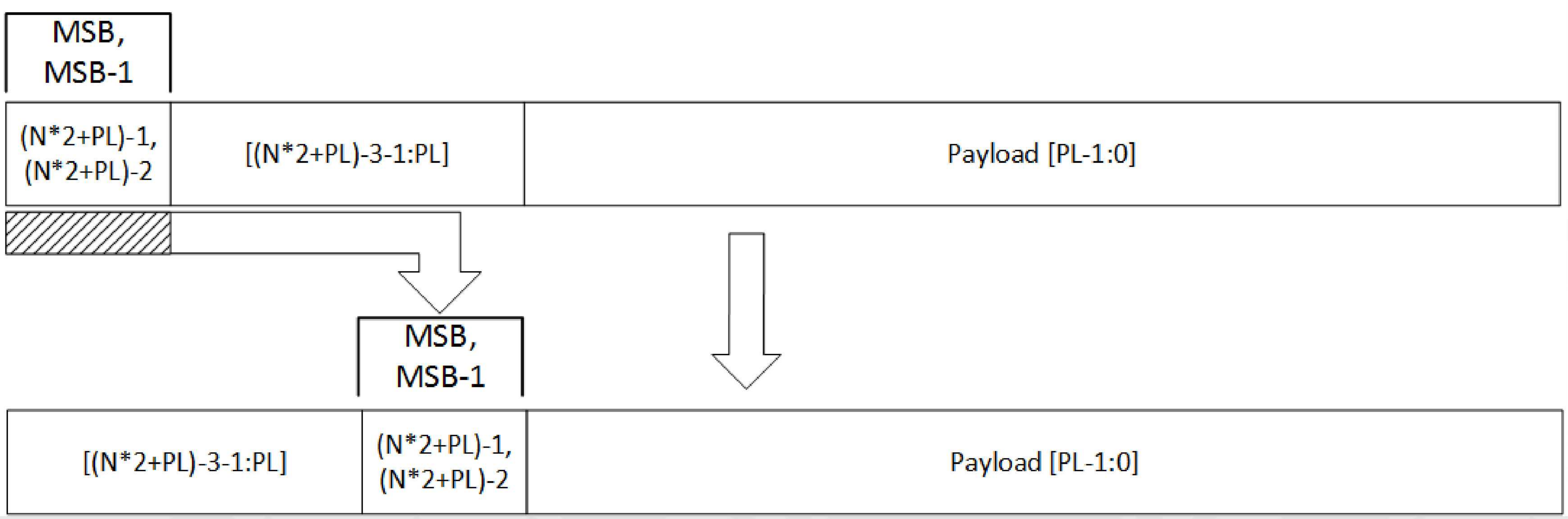


Switch

- Non blocking port
  - 4 data left (dl) latches
    - Latch bank size is dependent on packet size
  - 4 valid left (vl) latches
  - 4 stall right (sr) latches



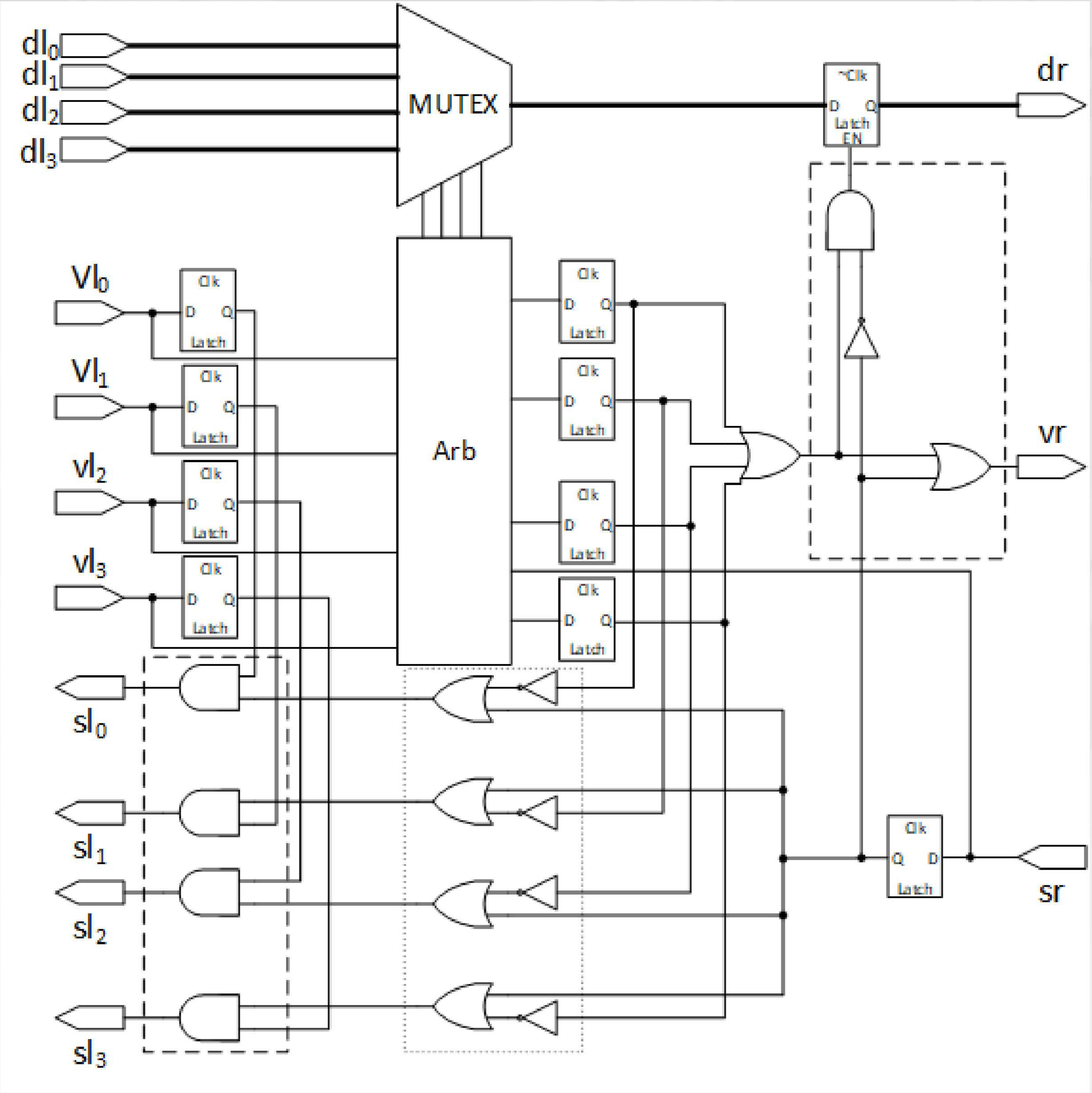
Shift Module



- MSB and MSB-1 are shifted to bits above payload
- Remaining address bits are shifted left by 2
- Payload is not shifted and left preserved

Merge Module

- Port
  - 1 data left latch
  - 4 valid left (vl) pre arbitration latches
  - 4 valid left (vl) post arbitration latches
  - 1 stall right (sr) latch
- Arbitration Module



Simulation

- CMOS research team developed accurate network simulator
  - Utilized for performing analysis and performance on the SELF NoC
  - 8x8 Network was designed and simulated for the planar mesh topology
  - Simulator configuration
    - XY Routing
    - Random traffic generator
    - First simulation
      - Injection Rate - 50ns
      - Total message injection – 500 messages
    - Last simulation
      - Injection Rate - 239ps
      - Total message injection - 33,623 messages

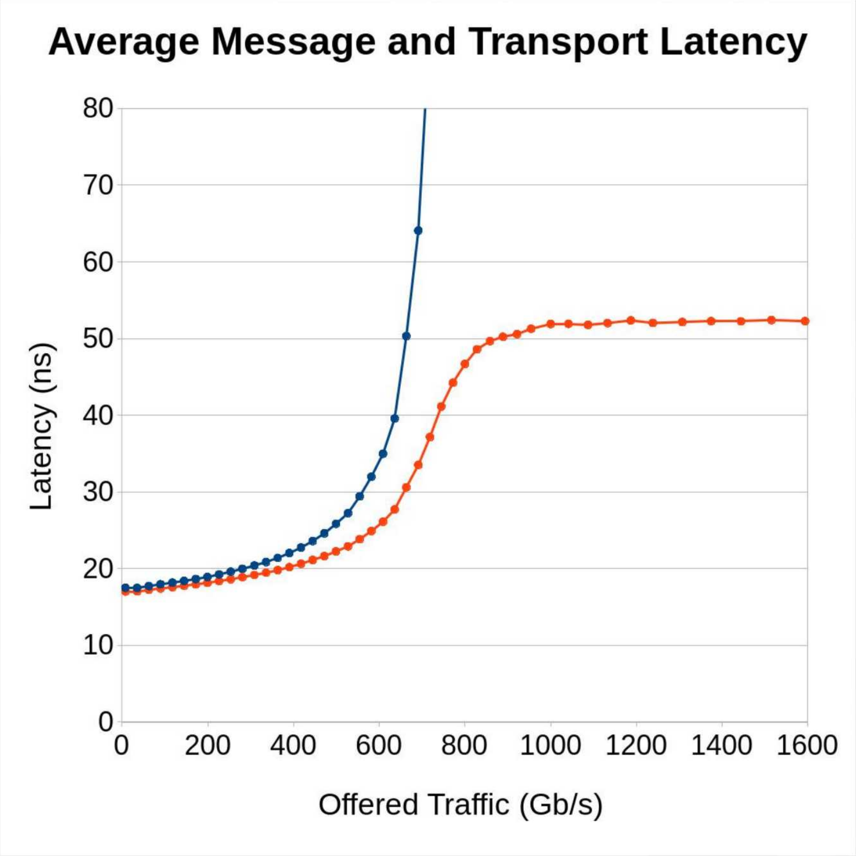
Results

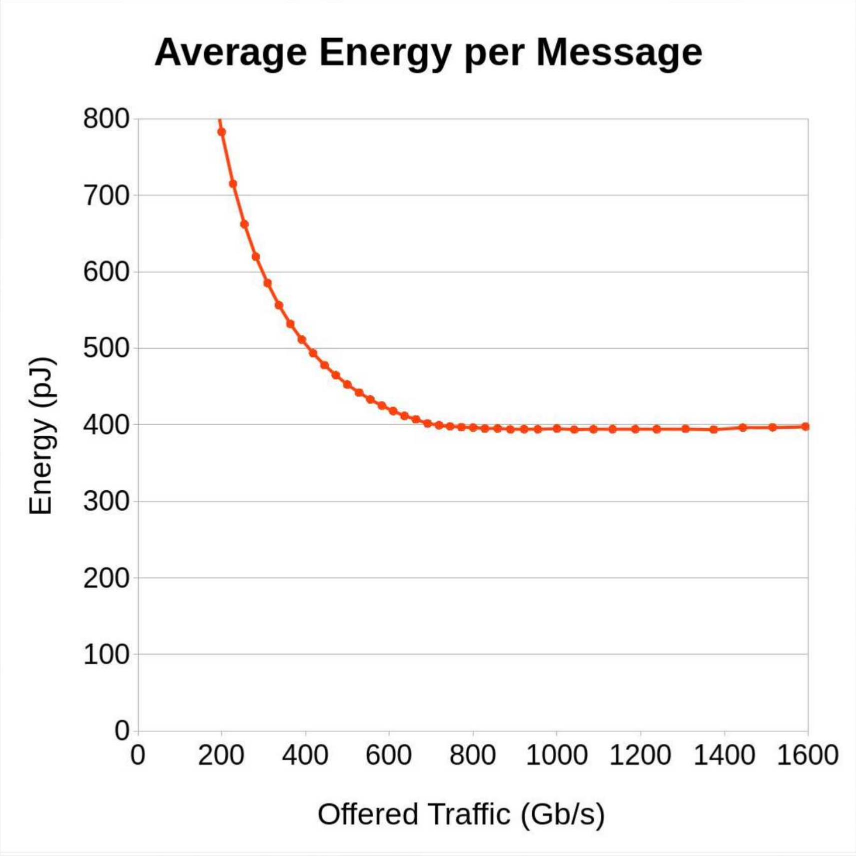
- 1.67GHz – Fastest clock synthesized for 65nm process
- 1.14GHz – Actual clock implemented with 1mm links

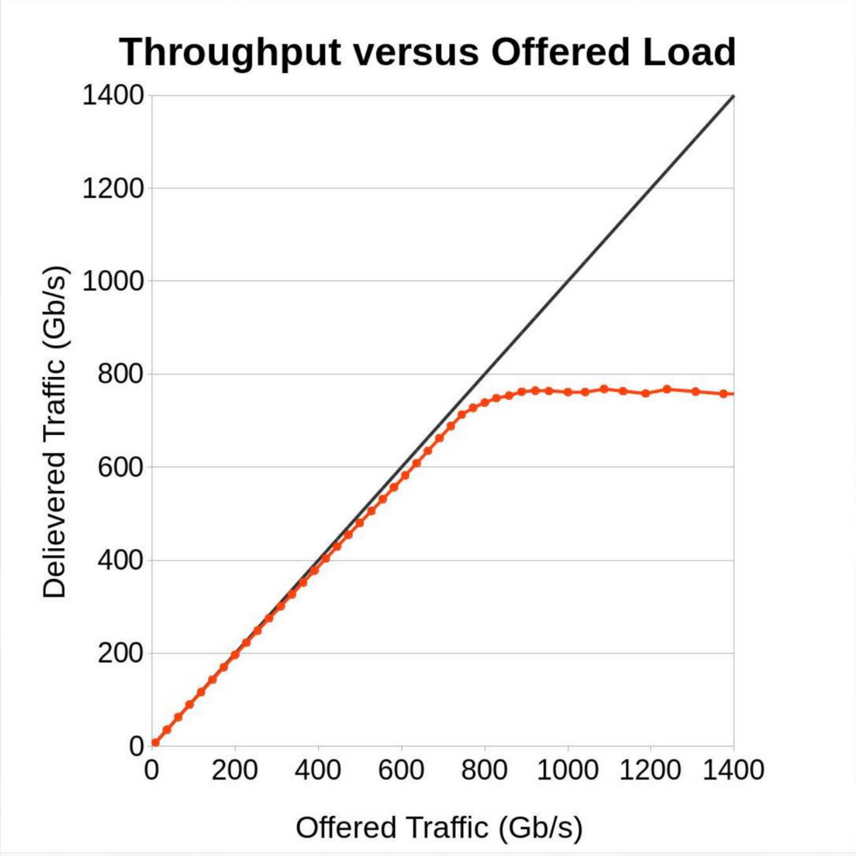
- Switch
  - Combinational cells: 37
  - Sequential Cells: 104
  - Buffers/Inverters: 19
  - Combinational Area: 108µm^2
  - Buffers/Inverters Area: 42µm^2
  - Non-combinational Area: 811.2µm^2
  - Total Cell Area: 919.2µm^2

- Merge
  - Combinational cells: 152
  - Sequential Cells: 44
  - Buffers/Inverters: 27
  - Combinational Area: 409µm^2
  - Buffers/Inverters Area: 55.2µm^2
  - Non-combinational Area: 352.2µm^2
  - Aribter4 Cell Area: 227.9µm^2
    - 30% of merge module
  - Total Cell Area: 762µm^2

Results Continued







- Network saturation point approximately 745Gb/s

Design Comparison

Design	Area	Power	Frequency
SELF	10,268μm^2	11.27mW	1.14GHz
FNoC	12,484μm^2	4.886mW	617MHz
EB	30,061μm^2	--------------	259MHz

Conclusions

- By implementing the SELF protocol into a synchronous router provides gains in the following:
  - Implementing point-to-point communication allows for high clock frequencies
  - Decrease the following:
    - Leakage power
    - Area
    - Wire lengths
    - Delay through the SoC
  - Increase throughput

Questions