

# ESE532: System-on-a-Chip Architecture

Day 25: November 28, 2018  
Network-on-a-Chip (NoC)



## Today

- Ring
- 2D Mesh Networks
- Design Issues
- Buffering and deflection
- Dynamic and static routing

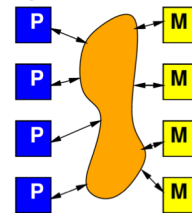
## Message

- Scalable interconnect for locality
  - has rich design space
- Customize to compute and application
- Support real-time with static scheduled communication

Day 12

## Interconnect

- Will need an infrastructure for programmable connections
- Rich design space to tune area-bandwidth-locality
  - Will explore more later in course

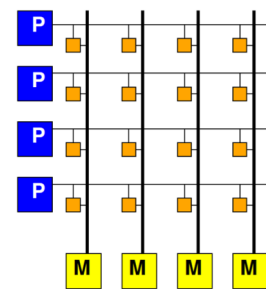


## Interconnect Concerns

- Avoid being a bottleneck
  - Bandwidth
  - Latency
- Competes for area and energy
  - against compute and memory

## Crossbar

- Connect any I inputs, O outputs
- Area  $\sim I \times O$
- For N PEs scale as  $N^2$



## Today's SoC Large

- At 0.4mm<sup>2</sup> small core on Apple A12
  - can put 250 on 1cm<sup>2</sup> chip
- Intel Knight's Bridge with 76 PEs
- 120 core MIPS on Stratix V FPGA
  - FPGA 2017
- 1680 core RISC-V on Xilinx Ultrascale
  - <http://fpga.org/2017/01/12/grvi-phalanx-joins-the-kilocore-club/>
- Scaling to 100s and 1000s of processing elements (PEs) that need interconnect

Penn ESE532 Fall 2018 -- DeHon

7

## Locality

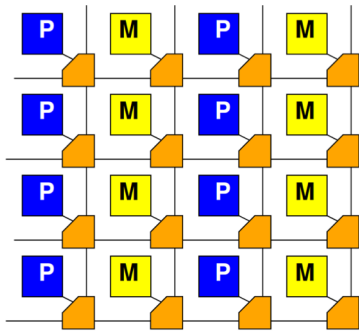
- Delay and energy proportional to distance
- Want to keep communications short
  - Data near compute
  - From compute block to compute block
- How build network?
  - Scalable (Area ~ N = things connected?)
  - Supports locality

Penn ESE532 Fall 2018 -- DeHon

8

Day 12

## Mesh



Penn ESE532 Fall 2018 -- DeHon

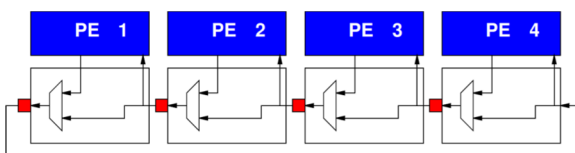
9

## Bus to Ring

Penn ESE532 Fall 2018 -- DeHon

10

## Ring

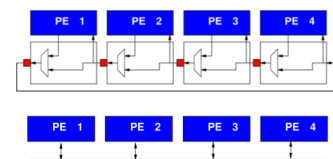


Penn ESE532 Fall 2018 -- DeHon

11

## Preclass 1

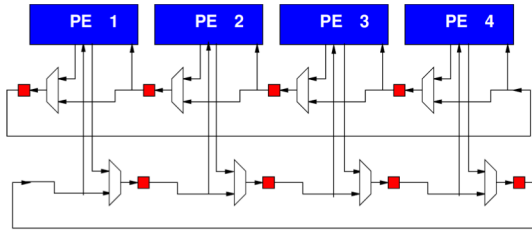
- Traffic pattern
  - Similar bandwidth?
  - One has higher bandwidth?



Penn ESE532 Fall 2018 -- DeHon

12

## Bidirectional Ring

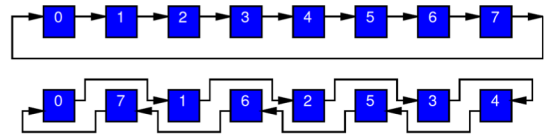


Penn ESE532 Fall 2018 -- DeHon

13

## Interleaved Layout

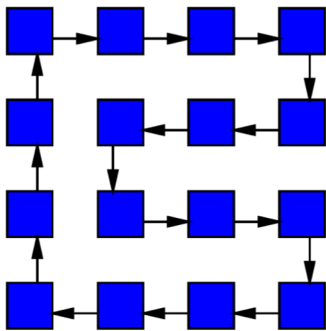
- What problem does the bottom layout solve (compared to top)?



Penn ESE532 Fall 2018 -- DeHon

14

## 2D Layout

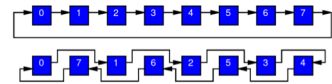


Penn ESE532 Fall 2018 -- DeHon

15

## Scaling

- How does area scale with  $N$ ?
- How does neighbor distance scale with  $N$ ?
  - Unidirectional
  - bidirectional
- How does worst-case distance in ring scale with  $N$ ?
  - Unidirectional
  - bidirectional



Penn ESE532 Fall 2018 -- DeHon

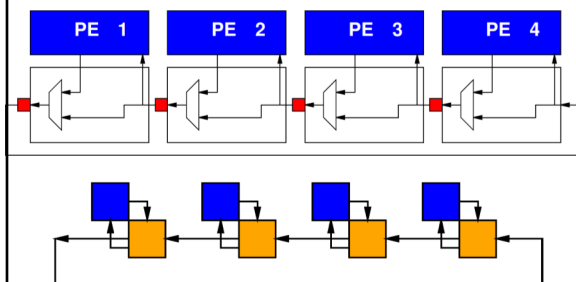
16

## 1D to 2D

Penn ESE532 Fall 2018 -- DeHon

17

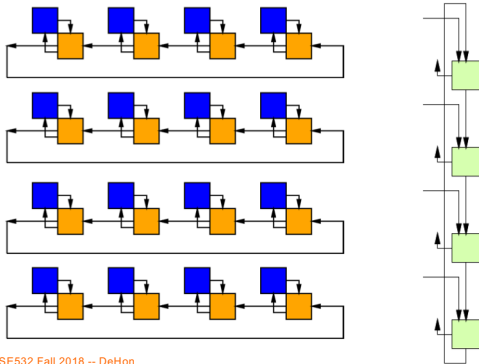
## Ring Abstract



Penn ESE532 Fall 2018 -- DeHon

18

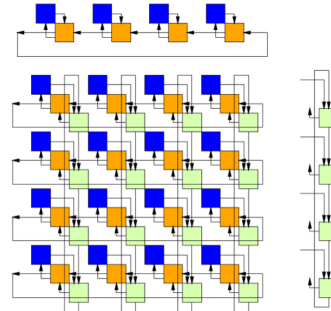
## Row and Column Rings



Penn ESE532 Fall 2018 -- DeHon

19

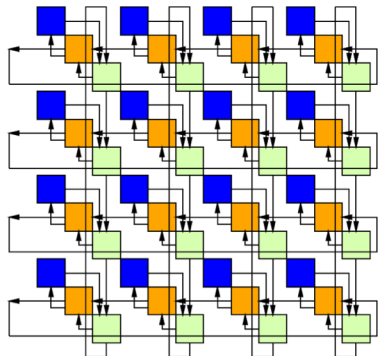
## Mesh as Row & Column Rings



Penn ESE532 Fall 2018

20

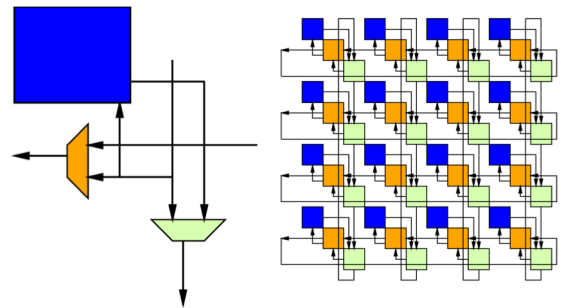
## Directional Mesh (Torus)



Penn ESE532 Fa

21

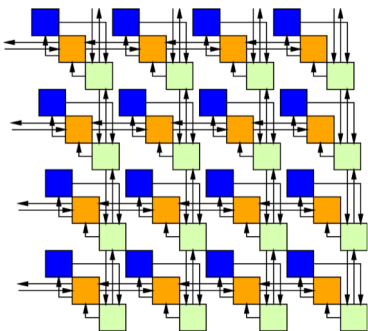
## Mesh Datapath



Penn ESE532 Fall 2018 -- DeHon

22

## Bidirectional Mesh

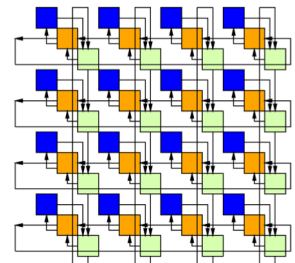


Penn ESE532 Fall 2018 -- DeHon

23

## 2D Mesh Scaling

- How does area scale with  $N$ ?
- How does neighbor distance scale with  $N$ ?
- How does worst-case distance in mesh scale with  $N$ ?



Penn ESE532 Fall 2018 -- DeHon

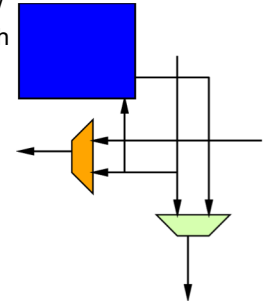
24

## Specifying Destination

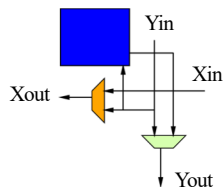
- Simple: add destination address
- Ring or Mesh wires carry:
  - Valid bit + Address + Payload (Data)

## Mesh Routing

- Route in Y until reach row
- Then route in X until reach column
- Consume from PE when arrives



## Mesh Routing

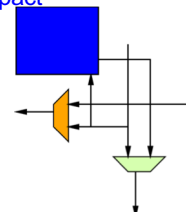


- $Y_{out} = Y_{in}.valid \ \& \ row(Y_{in}.address) \neq row \ \& \ Y_{in} \ + \ Pin.valid \ \& \ P$
- $X_{out} = X_{in}.valid \ \& \ column(X_{in}.address) \neq column \ \& \ X_{in} \ + \ Y_{in}.valid \ \& \ row(Y_{in}.address == row)$

- **Not deal with congestion**

## Mesh Routing

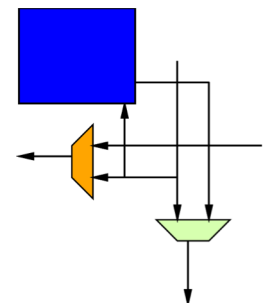
- $Y_{out} = Y_{in}.valid \ \& \ row(Y_{in}.address) \neq row \ \& \ Y_{in} \ + \ Pin.valid \ \& \ P$
- $X_{out} = X_{in}.valid \ \& \ column(X_{in}.address) \neq column \ \& \ X_{in} \ + \ Y_{in}.valid \ \& \ row(Y_{in}.address == row)$
- **Complexity of route function can impact**
  - Area, cycle time, route latency



## Mesh Congestion

## Mesh Congest

- What happens when inputs from 2 sides want to travel out same output?
  - (here X<sub>in</sub>, Y<sub>in</sub>)



## Dealing with Congestion

- Don't let it happen (offline/static)
  - Schedule to avoid
- Online/dynamic
  - Store in place -- Buffer
  - Misroute -- Deflect

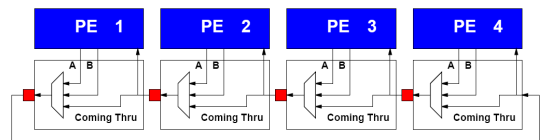
## Congestion 1D

- For simplicity, we look at congestion in 1D case (Preclass 2)

## Preclass 2ab

- Complete table – identify uncongested latencies
- Worst-case?

## Preclass 2c

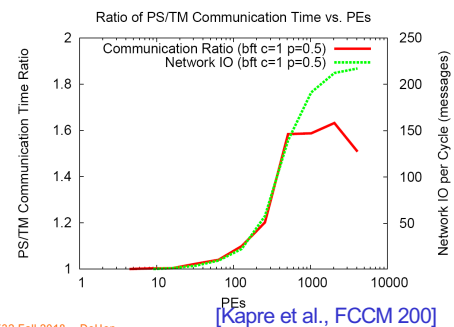


- Cycles from simulation?

## Observe

- Did have congestion
  - Ran slower than the single-link case (2b)
- How we make decisions matters
  - Who gets to route, which is stalled
- Best, global decision can be better than local decisions

## Offline vs. Online



## Dealing with Congestion

- Don't let it happen (offline/static)
  - Schedule to avoid
- Online/dynamic
  - Store in place -- Buffer
  - Misroute -- Deflect

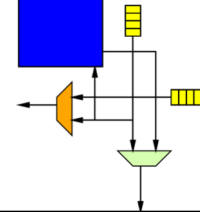
Penn ESE532 Fall 2018 -- DeHon

37

## Congestion: Buffer

Store inputs that must wait until path available

- Typically store in FIFO buffer
- How big do we make the FIFO?
- FIFO Buffers cost space
  - Often more than multiplexers



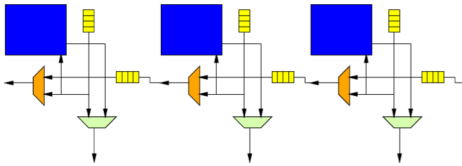
Penn ESE532 Fall 2018 -- DeHon

38

## Congestion: Buffer

Store inputs that must wait until path available

- Typically store in FIFO buffer
- How big do we make the FIFO?
- What if FIFO full?



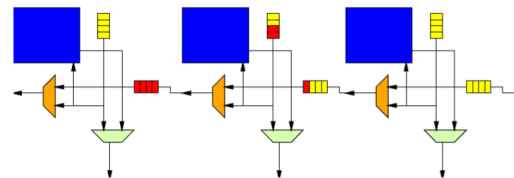
Penn ESE532 Fall 2018 -- DeHon

39

## Congestion: Buffer

Store inputs that must wait until path available

- Typically store in FIFO buffer
- How big do we make the FIFO?
- What if FIFO full?



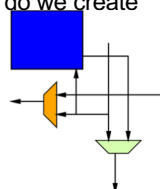
Penn ESE532 Fall 2018 -- DeHon

40

## Congestion: Deflect

Misroute: (deflection routing)

- Send in to an available (wrong) direction
- Avoid Buffer
- Requires balance of ins and outs
  - Can make work on mesh
- How much more traffic do we create misrouting?

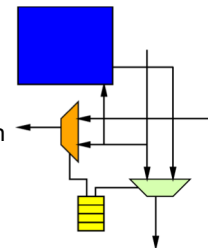


Penn ESE532 Fall 2018 -- DeHon

41

## Static Schedule

- Store per-cycle instruction for switch
  - Doesn't need address header on route
  - Local memories control destination

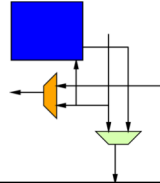


Penn ESE532 Fall 2018 -- DeHon

42

## Alternate Static Schedule

- Control injection cycle from processor so never have conflict
- Simple datapath logic to select available data
  - Needs address header on routed data



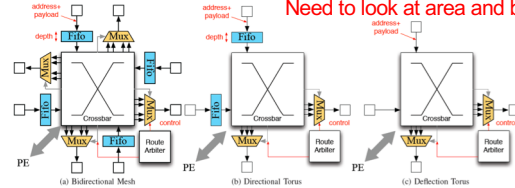
Penn ESE532 Fall 2018 -- DeHon

43

## Mesh Packet Switched 32b

- Bidirectional FIFO
  - bidirectional
  - 1800 LUTs
- Deflection (bufferless)
  - unidirectional
  - 60 LUTs

Big difference in area costs.  
Need to look at area and benefits.



Penn ESE532 Fall 2018 -- DeHon

[Kapr+Gray, FPL 2015]

44

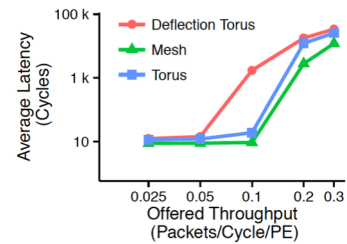
## Deflection Route

- What concerns might we have about deflection route?

Penn ESE532 Fall 2018 -- DeHon

45

## Buffer vs. Deflection



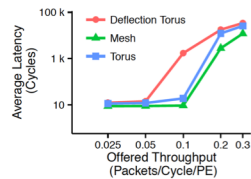
Penn ESE532 Fall 2018 -- DeHon

[Kapr+Gray, FPL 2015]

46

## Tuning

- How could we increase bandwidth to better handle high throughput?
  - Especially when the switches are small
- Mesh switch 1800
- Deflection Torus 60

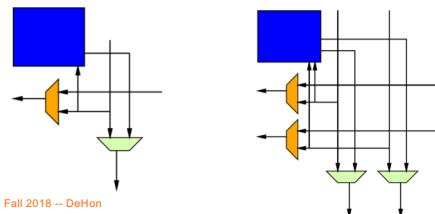


Penn ESE532 Fall 2018 -- DeHon

47

## Tune Bandwidth

- Add channels to tune bandwidth
  - Rings per row, column
  - Single deflection channel ~60
    - ...two around 120 ... still << 1800

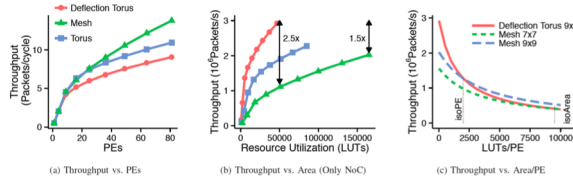


Penn ESE532 Fall 2018 -- DeHon

48



## Take 2, they are small



Penn ESE532 Fall 2018 -- DeHon

[Kapre+Gray, FPL 2015]

49

## Mesh Area Deflection PS/TM

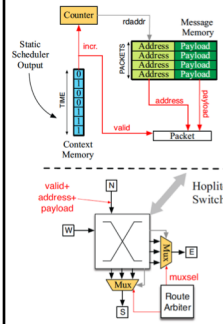


TABLE I: Comparing the different NoCs (32b payloads, Xilinx Virtex-6 LX240T). Period is for N=32. 64b SRLs used to store context memories.

NoC	Router LUTs	PE LUTs	Router FFs	Period (ns)
Hoplite	60	0	100	2.9
Time-Multiplexed	$100 + 4 \times [N/64]$	$[N/64]$	100	3.4
Marathon	60	$[N/64]$	100	3.2

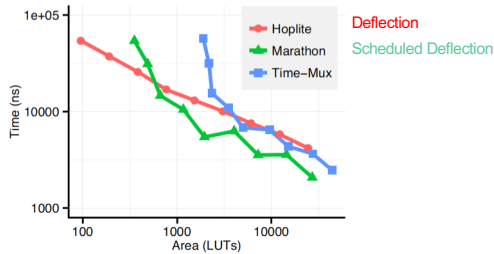
- Hoplite = Deflection
- Marathon = Deflection + static schedule at source

Penn ESE532 Fall 2018 -- DeHon

[Kapre FCCM 2015]

50

## Static Schedule vs. Deflection



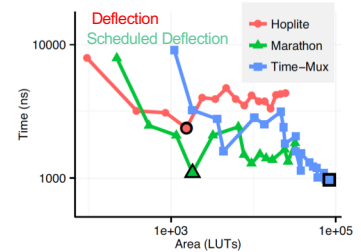
Penn ESE532 Fall 2018 -- DeHon

[Kapre FCCM 2015]

51

## Static Schedule vs. Deflection

- Routing 142K message add20 benchmark
- Marathon statically schedule PS



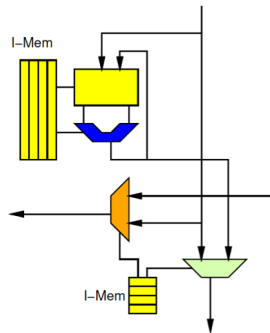
Penn ESE532 Fall 2018 -- DeHon

[Kapre FCCM 2015]

52

## Large VLIW

- Natural to use static network with VLIW clusters
  - Network routing becomes part of long instruction word
  - Schedule with computation
- Extreme one operator per mesh PE
- Tune bandwidth by clustering



Penn ESE532 Fall 2018 -- DeHon

53

## Mesh Customization

Penn ESE532 Fall 2018 -- DeHon

54

## Tuning Down

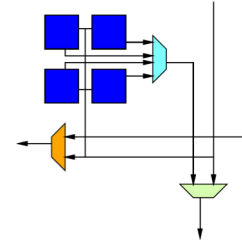
- What could we do to reduce area if we needed less bandwidth?

Penn ESE532 Fall 2018 -- DeHon

55

## Tuning Down Bandwidth

- If need less bandwidth, cluster multiple PEs to share a router.



Penn ESE532 Fall 2018 -- DeHon

56

## Simple Bandwidth/Area Control

- Width of channels
  - Like SIMD
  - All bits going to same destination

Penn ESE532 Fall 2018 -- DeHon

57

## Packets

- Simple story is, each “word” routed on mesh is: address+payload
- Alternately:
  - Multiword packet with single address
  - Share “address” across larger payload
  - Control width of datapath separate from size of payload
- Additional control issues to route packet together and buffer

Penn ESE532 Fall 2018 -- DeHon

58

## Customization

- Bandwidth
  - Width, clustering, channels
- Directional/Bidirectional
- Online dynamic/offline static
- Buffer/deflect
  - Buffer depth
- Route function sophistication

Penn ESE532 Fall 2018 -- DeHon

59

## Big Ideas

- Scalable interconnect for locality
  - Has rich design space
- Customize to compute and application
- Support real-time with static scheduled communication

Penn ESE532 Fall 2018 -- DeHon

60

## Admin

- P4 Due Friday