ESE532:
System-on-a-Chip Architecture

Day 27: December 6, 2017
Network-on-a-Chip (NoC)

Today

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

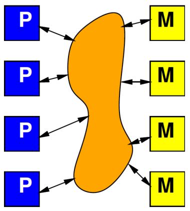
Message

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

Day 11

Interconnect

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

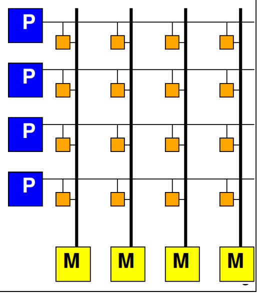


Interconnect Concerns

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

Crossbar

- Connect any I inputs, O outputs
- Area ~ I×O
- For N PEs
  scale as N^2



Today’s SoC Large

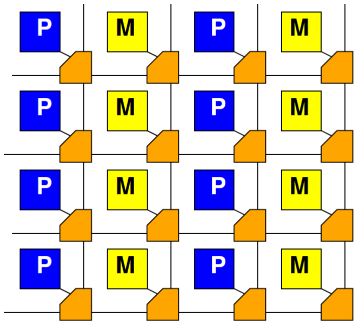
- At 1mm^2 per A9, can put 100 on 1cm^2 chip
- 120 core MIPS on Stratix V FPGA
  - – FPGA 2017
- 1680 core RISC-V on Xilinx Ultrascale
  - – http://fpga.org/2017/01/12/grvi-phalanx-joins-the-kilocore-club/
- Scaling to 100s and 1000s of processing elements (PEs) that need interconnect

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

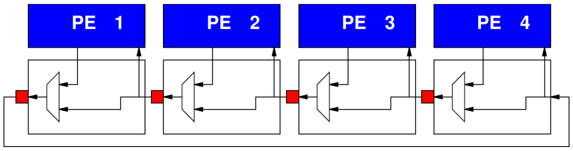
Day 11

Mesh



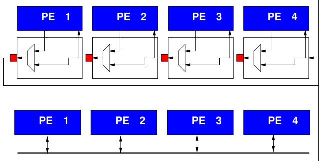
Bus to Ring

Ring

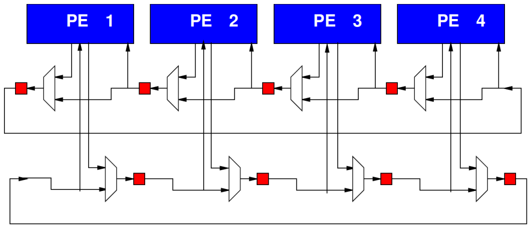


Preclass 1

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

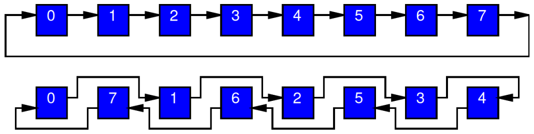


Bidirectional Ring

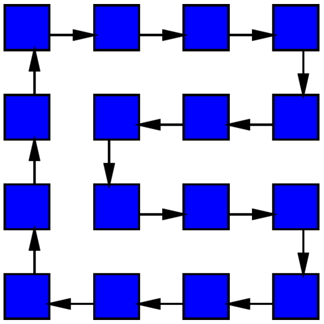


Interleaved Layout

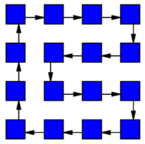
- What problem does this layout solve?



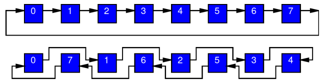
2D Layout



Scaling



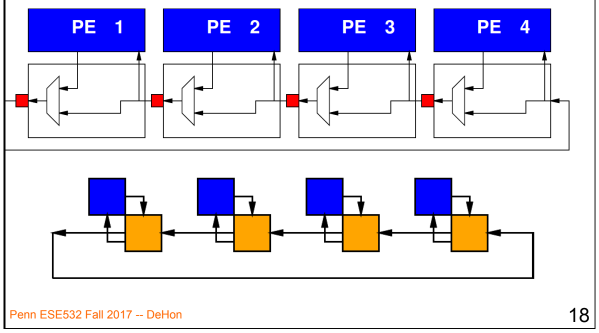
- How does area scale with N?
- How does neighbor distance scale with N?
  - Unidirectional
  - bidirectional

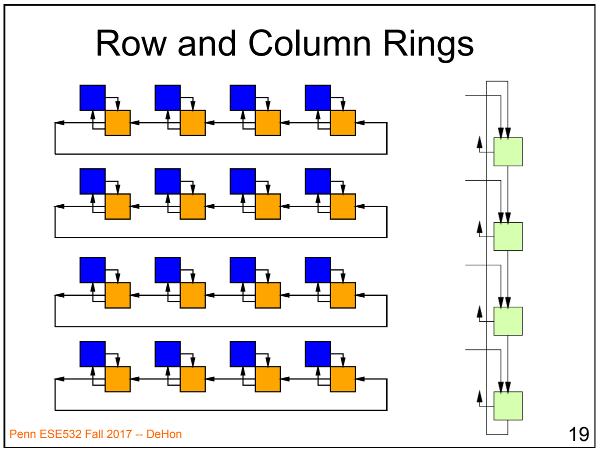


- How does worst-case distance in ring scale with N?
  - Unidirectional
  - bidirectional

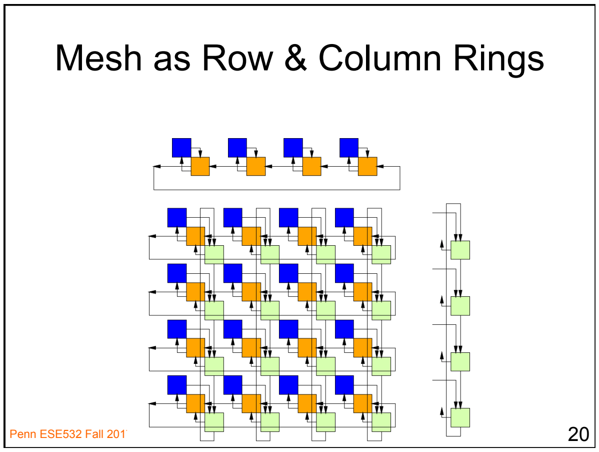
1D to 2D

Ring Abstract

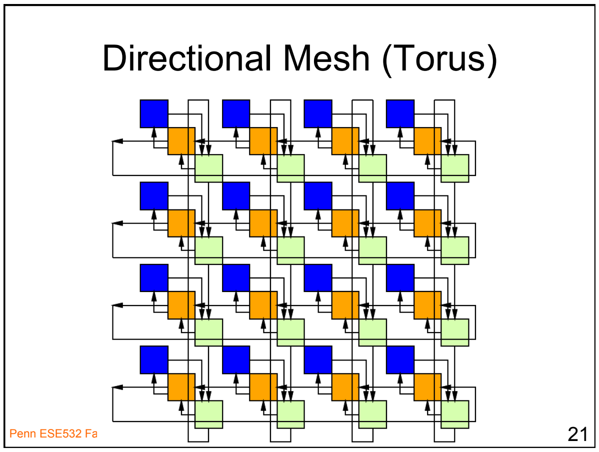




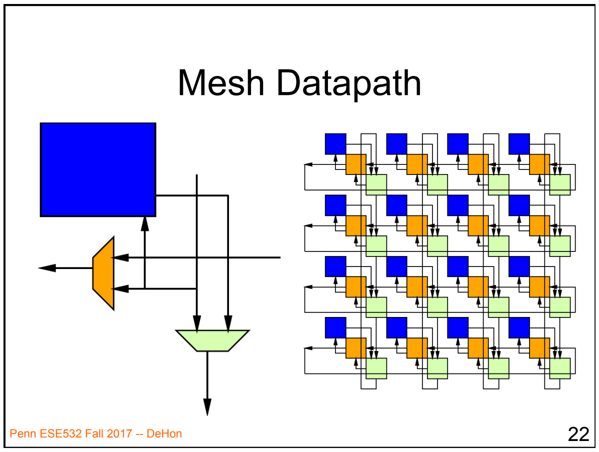
Row and Column Rings



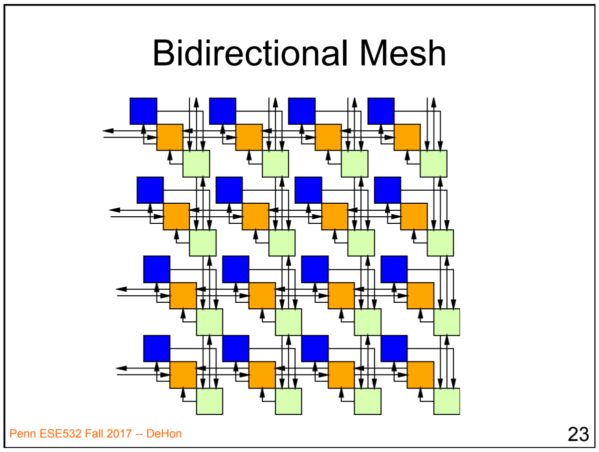
Mesh as Row & Column Rings



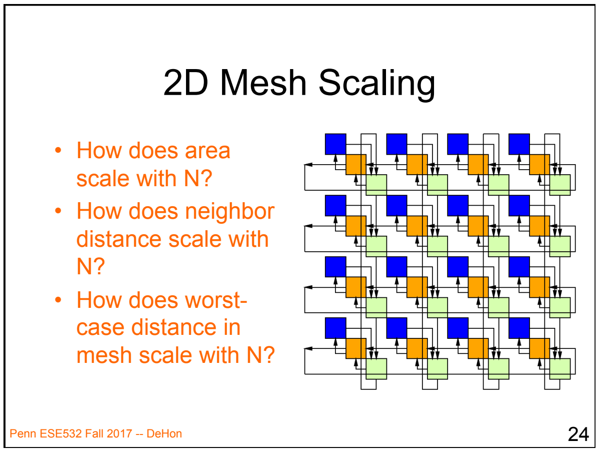
Directional Mesh (Torus)



Mesh Datapath



Bidirectional Mesh



2D Mesh Scaling

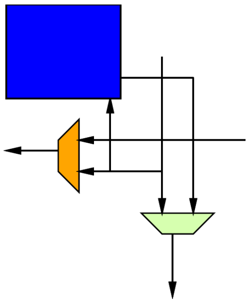
- How does area scale with N?
- How does neighbor distance scale with N?
- How does worst-case distance in mesh scale with N?

Specifying Destination

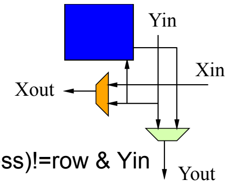
- Simple: add destination address
- Ring or Mesh wires carry:
  - Valid bit + Address + Payload (Data)

Mesh Routing

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



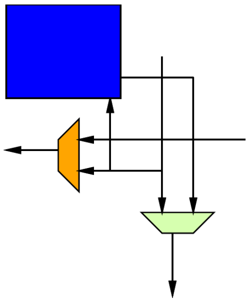
Mesh Routing



- Yout=Yin.valid & row(Yin.address)!=row & Yin + Pin.valid & P
- Xout=Xin.valid & column(Xin.address)!=column & Xin + Yin.valid & row(Yin.address==row)
- Not deal with congestion

Mesh Routing

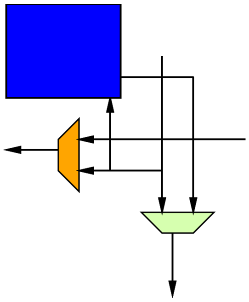
- Yout=Yin.valid & row(Yin.address)!=row & Yin + Pin.valid & P
- Xout=Xin.valid & column(Xin.address)!=column & Xin + Yin.valid & row(Yin.address==row)
- Complexity of route function can impact
  - Area, cycle time, route latency



Mesh Congestion

Mesh Congest

- What happens when inputs from 2 sides want to travel out same output?
  - (here Xin, Yin)



Dealing with Congestion

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

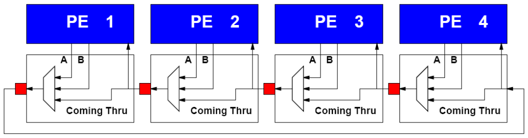
Congestion 1D

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

Preclass 2a

- Complete table – identify uncongested latencies

Preclass 2b

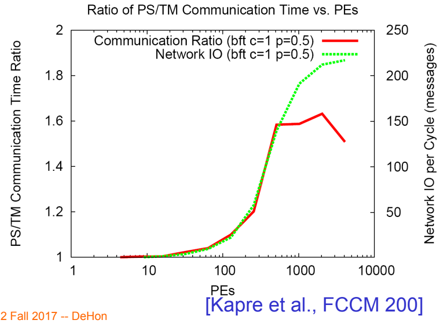


- Cycles from simulation?

Observe

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

Offline vs. Online



[Kapre et al., FCCM 200]

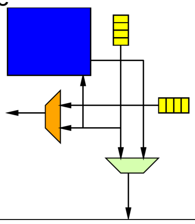
Dealing with Congestion

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

Congestion: Buffer

Store inputs that must wait until path available

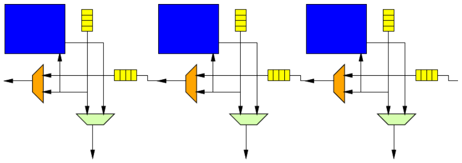
- Typically store in FIFO buffer
- How big do we make the FIFO?
- FIFO Buffers cost space
  - Often more than multiplexers



Congestion: Buffer

Store inputs that must wait until path available

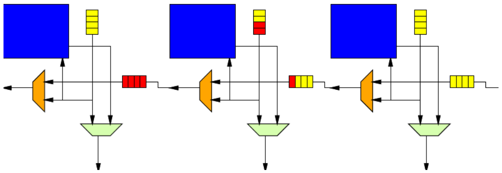
- Typically store in FIFO buffer
- How big do we make the FIFO?
- What if FIFO full?



Congestion: Buffer

Store inputs that must wait until path available

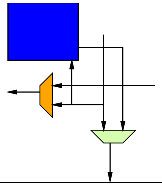
- Typically store in FIFO buffer
- How big do we make the FIFO?
- What if FIFO full?



Congestion: Deflect

Misroute: (deflection routing)

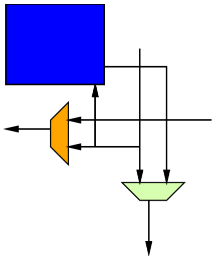
- Send in to an available (wrong) direction
- Avoid Buffer
- Requires balance of ins and outs
  - Can make work on mesh
- How much more traffic do we create misrouting?



Mesh Routing:

- Yout=Yin.valid & row(Yin.address)!=row & Yin + Pin.valid & P
- +row(Yin.address)==row & (column.Xin.address)!=column) & Y.in
- Xout=Xin.valid & column(Xin.address)!=column & Xin + Yin.valid & row(Yin.address==row)

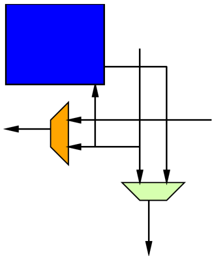
Gives Preference to X



Mesh Routing:

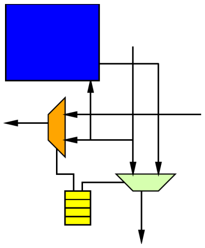
- Yout=Yin.valid & row(Yin.address)!=row & Yin + Pin.valid & P
- +row(Yin.address)==row & (column.Xin.address)!=column) & Y.in
- Xout=Xin.valid & column(Xin.address)!=column & Xin + Yin.valid & row(Yin.address==row)

Alternates:
random selection
preference based on aging
(keep track of # of times misrouted)



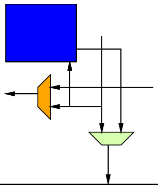
Static Schedule

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



Alternate Static Schedule

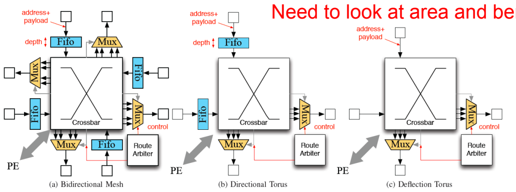
- Control injection cycle from processor so never have conflict
- Simple datapath logic to select available data
  - Needs address header on routed data



Mesh Packet Switched 32b

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

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

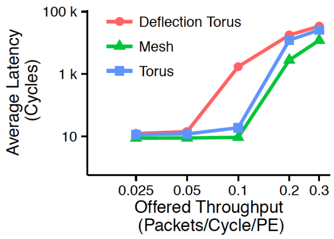


[Kapre+Gray, FPL 2015]

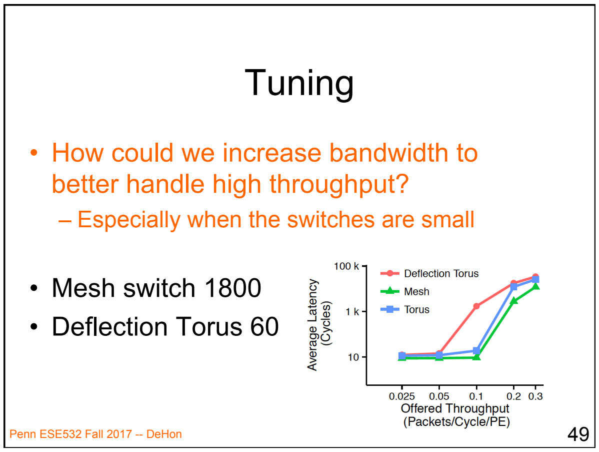
Deflection Route

- What concerns might we have about deflection route?

Buffer vs. Deflection



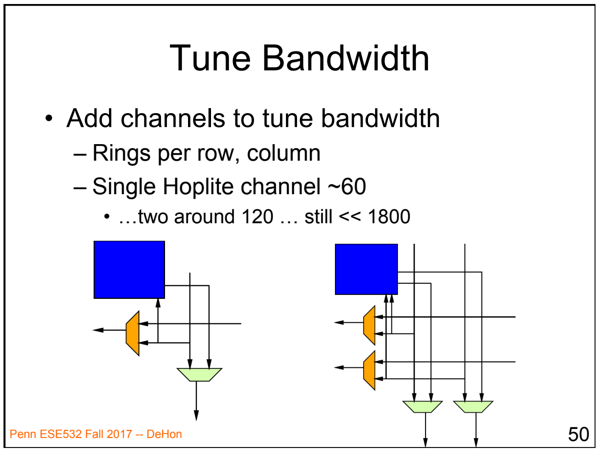
[Kapre+Gray, FPL 2015]



Tuning

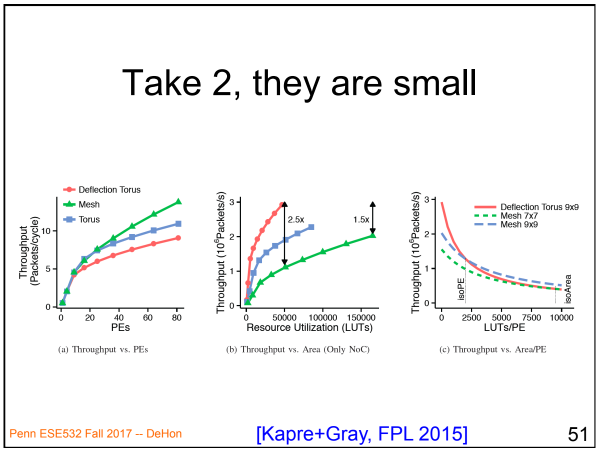
- How could we increase bandwidth to better handle high throughput?
  - Especially when the switches are small

- Mesh switch 1800
- Deflection Torus 60



Tune Bandwidth

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



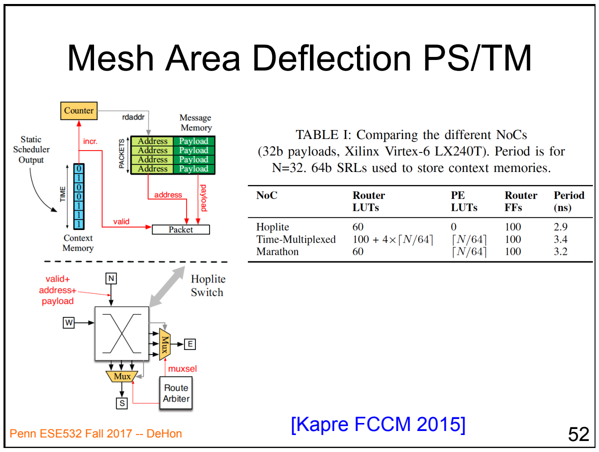
Take 2, they are small

(a) Throughput vs. PEs

(b) Throughput vs. Area (Only NoC)

(c) Throughput vs. Area/PE

[Kapre+Gray, FPL 2015]

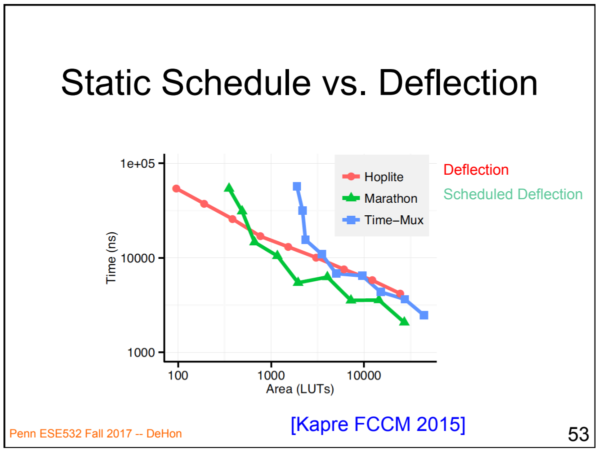


Mesh Area Deflection PS/TM

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

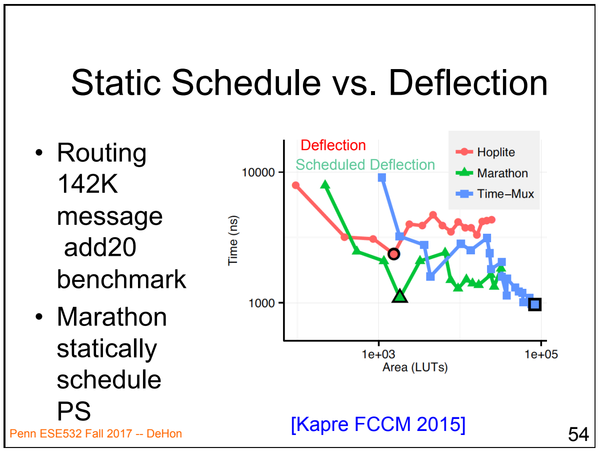
NoC	Router LUTs	PE LUTs	Router FFs	Period (ns)
Hoplite	60	0	100	2.9
Time-Multiplexed	$100 + 4\times\lceil N/64\rceil$	$\lceil N/64\rceil$	100	3.4
Marathon	60	$\lceil N/64\rceil$	100	3.2

[Kapre FCCM 2015]



Static Schedule vs. Deflection

[Kapre FCCM 2015]



Static Schedule vs. Deflection

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

[Kapre FCCM 2015]

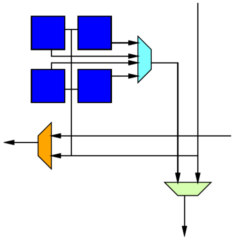
Mesh Customization

Tuning Down

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

Tuning Down Bandwidth

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



Simple Bandwidth/Area Control

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

Packets

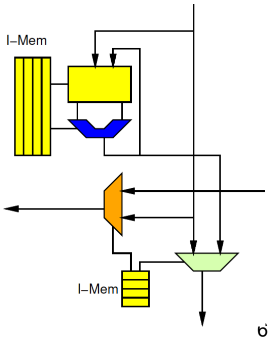
- Simple story is, each “word” routed on mesh is: address+payload
- Alternately:
  - Multiword packet with single address
  - Share “address” across larger payload
  - Control width of datapath separate from size of payload
- Additional control issues to route packet together and buffer

Customization

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

Large VLIW

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



Big Ideas

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

Admin

- Project Due Friday
- Wrapup lecture on Monday
  - Return Zed Boards