ESE534: Computer Organization

Day 25: April 18, 2012

Interconnect 6: Dynamically Switched Interconnect

Previously

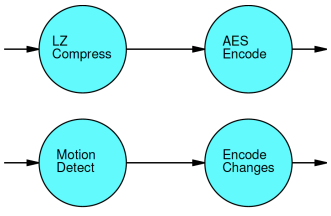
- Configured Interconnect
  - Lock down route between source and sink
- Multicontext Interconnect
  - Switched cycle-by-cycle from Instr. Mem.
- Interconnect Topology
- Data-dependent control for computation

Today

- Data-dependent control/sharing of interconnect
- Dynamic Sharing (Packet Switching)
  - Motivation
  - Formulation
  - Design
  - Assessment

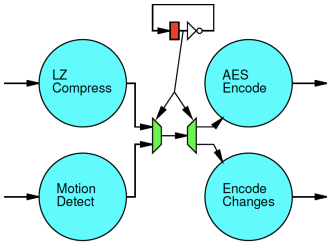
Motivation

Consider: Preclass 1



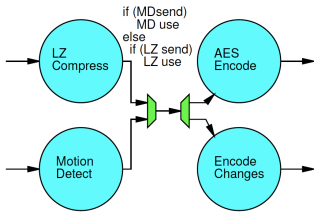
- How might this be inefficient with configured interconnect?

Consider: Preclass 1



- How might this be more efficient?
- What might be inefficient about it?

Consider: Preclass 1



- How might this be more efficient?
- What undesirable behavior might this have?

Opportunity

- Interconnect major area, energy, delay
- **Instantaneous** communication << potential communication
- **Question:** can we reduce interconnect requirements by only routing instantaneous communication needs?

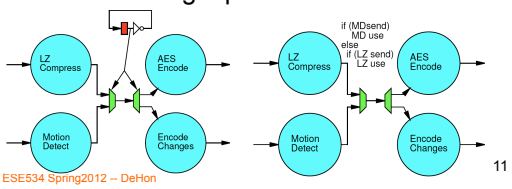
Examples

- Data dependent, unpredictable results
  - Search filter, compression
- Slowly changing values
  - Surveillance, converging values

Formulation

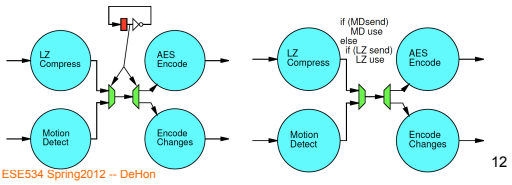
Dynamic Sharing

- Don't reserve resources
  - Hold a resource for a single source/sink pair
  - Allocate cycles for a particular route
- Request as needed
- Share amongst potential users



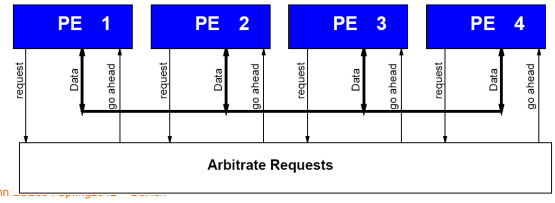
Bus Example

- Time Multiplexed version
  - Allocate time slot on bus for each communication
- Dynamic version
  - Arbitrate for bus on each cycle



Bus Example

- Time Multiplexed version
  - Allocate time slot on bus for each communication
- Dynamic version
  - Arbitrate for bus on each cycle



Dynamic Bus Example

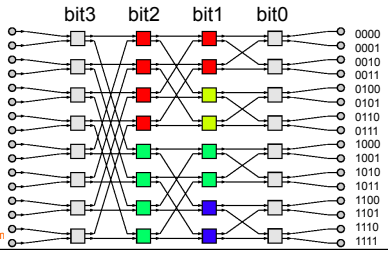
- 4 PEs
  - Potentially each send out result on change
  - Value only changes with probability 0.1 on each "cycle"
  - TM: Slot for each
    - PE0 PE1 PE2 PE3 PE0 PE1 PE2 PE3
  - Dynamic: arbitrate based on need
    - None PE0 none PE1 PE1 none PE3 ….
- TM either runs slower (4 cycles/compute) or needs 4 busses
- Dynamic single bus seldom bottleneck
  - Probability two need bus on a cycle?

Network Example

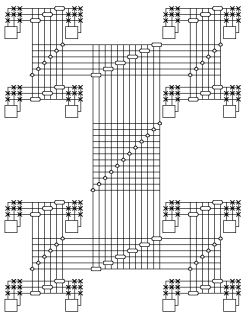
- Time Multiplexed
  - As assumed so far in class
  - Memory says how to set switches on each cycle
- Dynamic
  - Attach address or route designation
  - Switches forward data toward destination

Butterfly

- Log stages
- Resolve one bit per stage



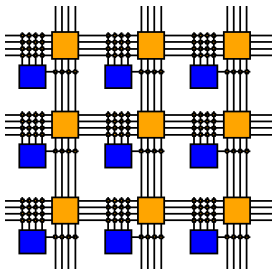
Tree Route



- Downpath resolves one bit per stage

Mesh Route

- Destination (dx,dy)
- Current location (cx,cy)
- Route up/down left/ right based on (dx-cx,dy-cy)



Prospect

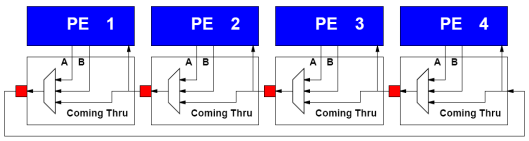
- Use less interconnect

<OR> get higher throughput across fixed interconnect

- By
  - Dynamically sharing limited interconnect
- Bottleneck on instantaneous data needs
  - Not worst-case data needs

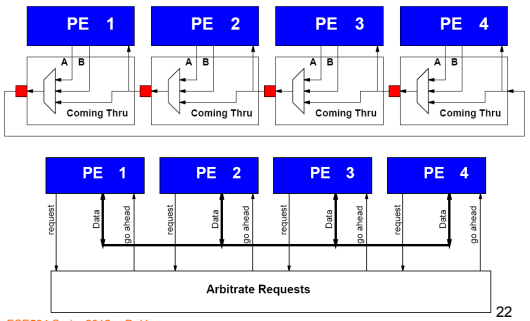
Design

Ring

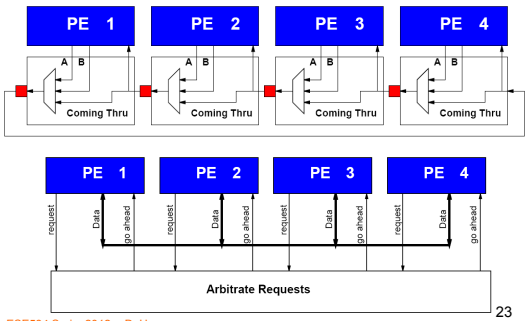


- Ring: 1D Mesh

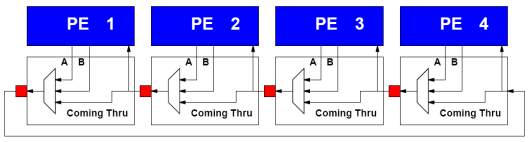
How like a bus?



Advantage over bus?



Preclass 2



- Cycles from simulation?
- Max delay of single link?
- Best case possible?

Issue 1: Local online vs Global Offline

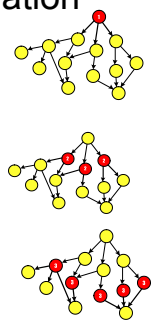
- Dynamic must make local decision
  - Often lower quality than offline, global decision
- Quality of decision will reduce potential benefits of reducing instantaneous requiements

Experiment

- Send-on-Change for spreading activation task
- Run on Linear-Population Tree network
- Same topology both cases
- Fixed size graph
- Vary physical tree size
  - Smaller trees → more serial
    - Many "messages" local to cluster, no routing
  - Large trees → more parallel

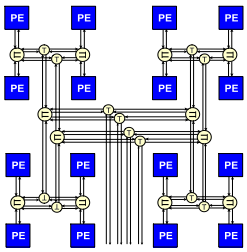
Spreading Activation

- Start with few nodes active
- Propagate changes along edges

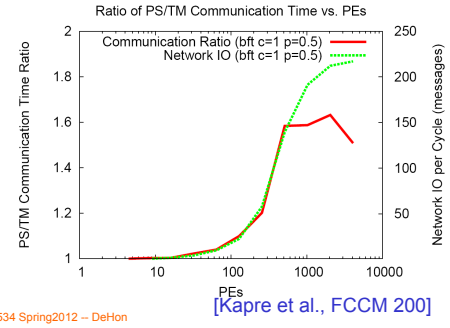


Butterfly Fat Trees (BFTs)

- Familiar from Day 17
- Similar phenomena with other topologies
- Directional version

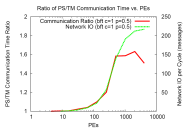


Iso-PEs



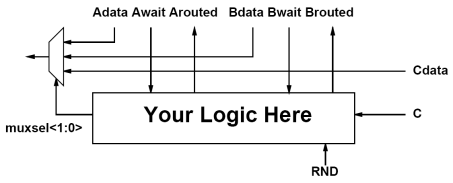
[Kapre et al., FCCM 200]

Iso-PEs



- PS vs. TM ratio at same PE counts
  - Small number of PEs little difference
    - Dominated by serialization (self-messages)
    - Not stressing the network
  - Larger PE counts
    - TM ~60% better
    - TM uses global congestion knowledge while scheduling

Preclass 3



- Logic for muxsel<0>?
- Logic for Arouted? Logic for Brouted?
- Gates?
- Gate Delay?

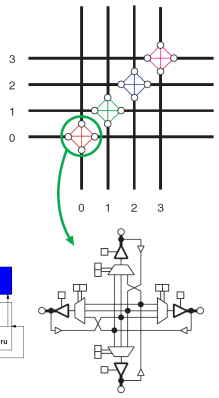
Issue 2: Switch Complexity

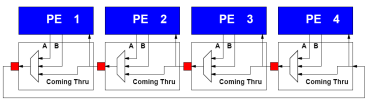
- Requires area/delay/energy to make decisions
- Also requires storage area
- Avoids instruction memory

- High cost of switch complexity may diminish benefit.

Mesh Congest

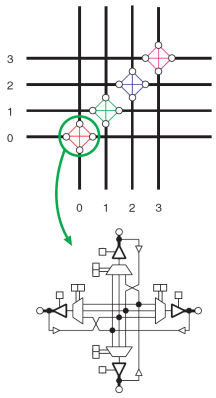
- Preclass 2 ring similar to slice through mesh
- A,B – corner turns
- May not be able to route on a cycle





Mesh Congest

- What happens when inputs from 2 (or 3) sides want to travel out same output?

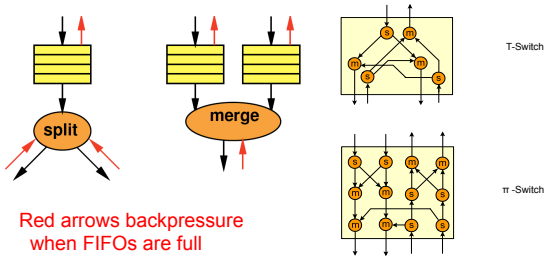


FIFO Buffering

- Store inputs that must wait until path available
  - Typically store in FIFO buffer

- How big do we make the FIFO?

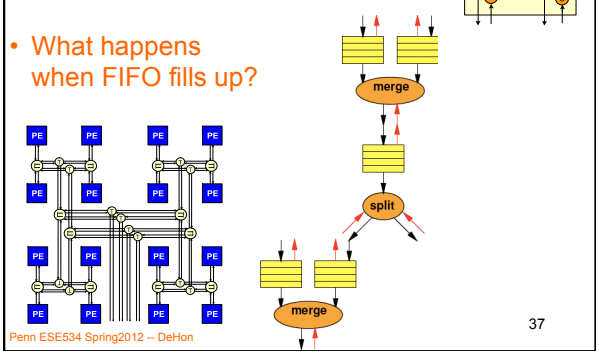
PS Hardware Primitives



Red arrows backpressure when FIFOs are full

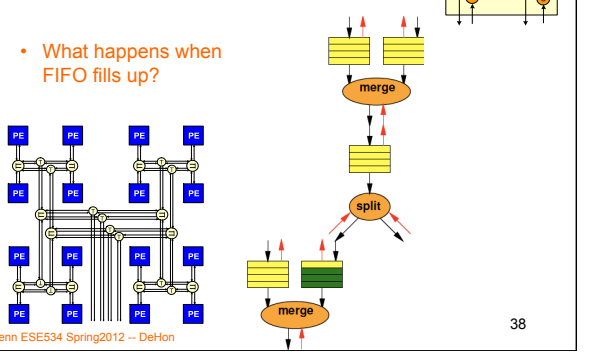
FIFO Buffer Full?

- What happens when FIFO fills up?



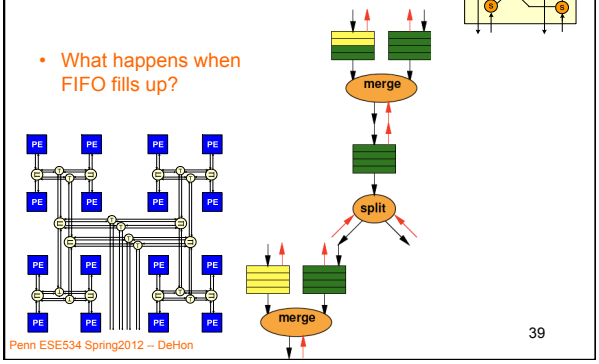
FIFO Buffer Full?

- What happens when FIFO fills up?



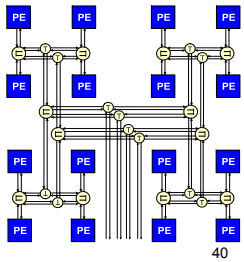
FIFO Buffer Full?

- What happens when FIFO fills up?

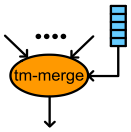


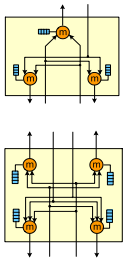
FIFO Buffer Full?

- What happens when FIFO fills up?
- Maybe backup network
- Prevent other routes from using
  - If not careful, can create deadlock



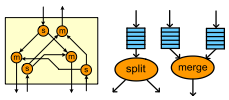
TM Hardware Primitives

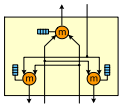




Area in PS/TM Switches

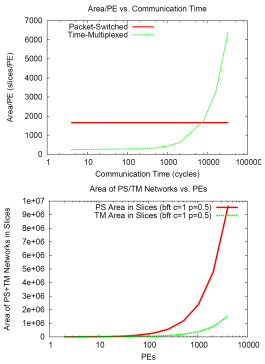
- Packet (32 wide, 16 deep)
  - 3split + 3 merge
  - Split 79
    - 30 ctrl, 33 fifo buffer
  - Merge 165
    - 60 ctrl, 66 fifo buffer
  - Total: 244
- Time Multiplexed (16b)
  - 9+(contexts/16)
  - E.g. 41 at 1024 contexts
- Both use SRL16s for memory (16b/4-LUT)
- Area in FPGA slice counts





Area Effects

- Based on FPGA overlay model
- *i.e.* build PS or TM on top of FPGA



Preclass 4

- Gates in static design: 8
- Gates in dynamic design: 8+?
- Which energy best?
  - P_d=1
  - P_d=0.1
  - P_d=0.5

PS vs TM Switches

- PS switches can be larger/slower/more energy
- Larger:
  - May compete with PEs for area on limited capacity chip

Assessment

Following from
Kapre et al. / FCCM 2006

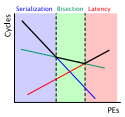
Analysis

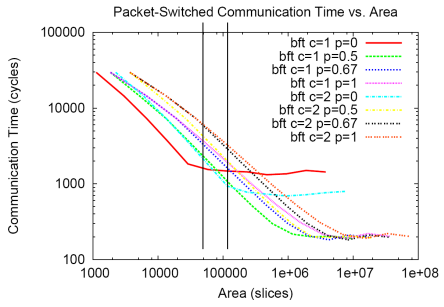
- PS v/s TM for same area
  - Understand area tradeoffs (PEs v/s Interconnect)
- PS v/s TM for dynamic traffic
  - PS routes limited traffic, TM has to route all traffic

Area Analysis

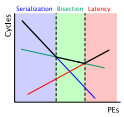
- Evaluate PS and TM for multiple BFTs
  - Tradeoff Logic Area for Interconnect
  - Fixed Area of 130K slices
    - p=0, BFT => 128 PS PEs => 1476 cycles
    - p=0.5, BFT => 64 PS PEs => 943 cycles
- Extract best topologies for PS and TM at each area point
  - BFT of different p best at different area points
- Compare performance achieved at these bests at each area point

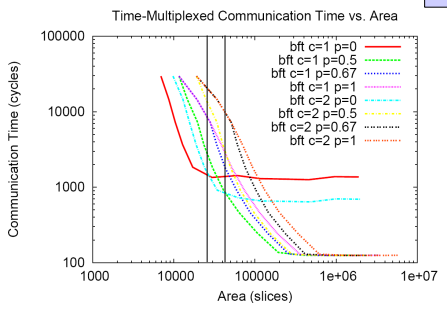
PS Iso-Area: Topology Selection



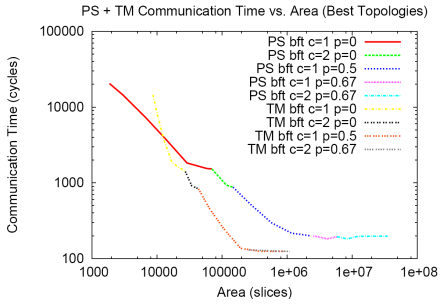


TM Iso-Area

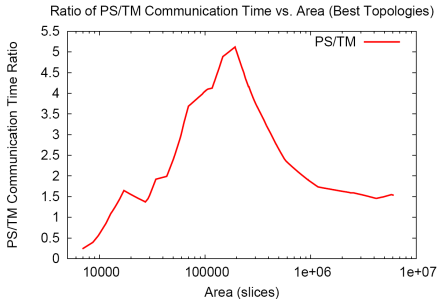




Iso-Area

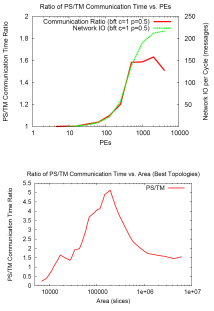


Iso-Area Ratio



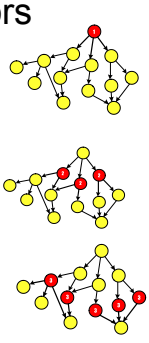
Iso-Area

- Iso-PEs = TM 1~2x better
- With Area
  - PS 2x better at small areas
  - TM 4-5x better at large areas
  - PS catches up at the end

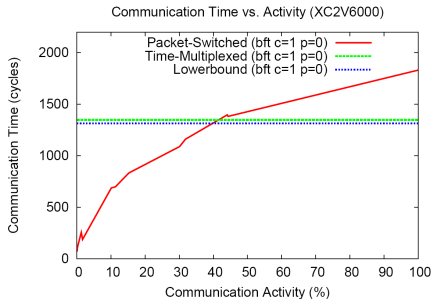


Activity Factors

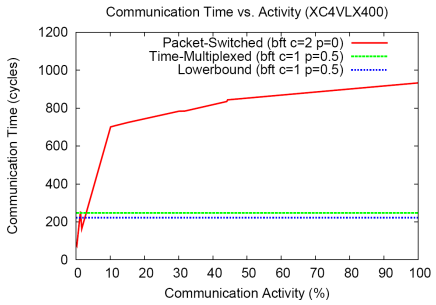
- Activity = Fraction of traffic to be routed
- TM needs to route all
- PS can route fraction
- Variable activity queries in ConceptNet
  - Simple queries ~1% edges
  - Complex queries ~40% edges



Activity Factors



Crossover could be less



Lessons

- Latency
  - PS could achieve same clock rate
  - But took more cycles
  - Didn't matter for this workload
- Quality of Route
  - PS could be 60% worse
- Area
  - PS larger, despite all the TM instrs
  - Big factor
  - Will be "technology" and PE-type dependent
    - Depends on relative ratio of PE to switches
    - Depends on relative ratio of memory and switching

Admin

- Final Exercise
  - Discussion period ends Tuesday
  - André traveling after lecture next week
- Reading for Monday on Blackboard

Big Ideas [MSB Ideas]

- Communication often data dependent
- When unpredictable, unlikely to use potential connection
  - May be more efficient to share dynamically
- Dynamic may cost more per communication
  - More logic, buffer area, more latency
  - Less efficient due to local view
- Net win if sufficiently unpredictable