ESE534: Computer Organization

Day 24: April 21, 2010

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

- Dynamic Sharing (Packet Switching)
  - Motivation
  - Formulation
  - Design
  - Assessment

Motivation

Unused Links

- Shortest Path
- Each node computes:
  - Delay = min(input delay)
  - Send to successors
    - Delay+Successors.LinkDelay
- If store delay, only send on change
  - Delay = infinity
  - While ()
    - If (InputDelay<Delay)
      - Delay=InputDelay
      - Send to successors
        - » (Delay +Successor.LinkDelay)

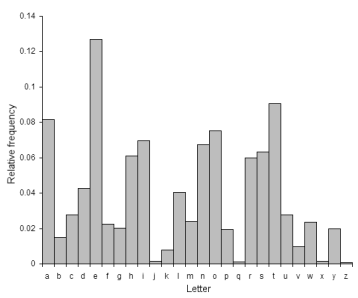
Unpredictable Results

- Searching/Filtering
  - Many PEs searching in parallel
    - pattern match in portion of an image
    - Better schedule or protein fold
  - When find result, report

Unpredictable Results

- Lossless compression
  - E.g. Huffman
  - Variable bit encoding for each input symbol

English Letter Frequency



http://en.wikipedia.org/wiki/File:English-slf.png

Encoding

Worst-case may produce output word per input symbol

Typical case, will be several input symbols per output word

Compare:

- encoding E, e with 2 or 3 bits
- encoding x with 9 bits

symbol	bits	encode	symbol	bits	encode
spc	3	111	a	4	0111
'	8	01101010	b	6	000101
,	6	101000	c	6	110001
-	9	101001110	d	5	10011
.	7	0110001	e	3	001
1	10	0001001111	f	6	110000
4	13	1010011110110	g	6	011011
7	11	01101001001	h	5	11001
9	13	0001001011101	i	4	0100
;	9	011010111	j	10	0110100110
E	10	0110000011	k	8	10100110
F	10	0001001001	l	5	00011
I	7	0001000	m	6	110101
J	12	011010011100	n	4	0101
M	10	0110100001	o	4	1000
P	10	0110100101	p	6	100100
R	11	10100111100	q	10	0110100010
T	9	011000011	r	4	0000
W	11	10100111110	s	5	11011
Y	12	011010011110	t	4	1011
			u	6	110100
			v	7	1010010
			w	6	100101
			x	9	011000000
			y	6	011001

Slow Changing Values

- Send values only on change
  - Or exceed threshold
- Simulation
  - Verilog timing – only on signal transition
- Constraint solver
  - Only send when constraints tighten
- Surveillance
  - Only when scene changes
    - Part that changes

Opportunity

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

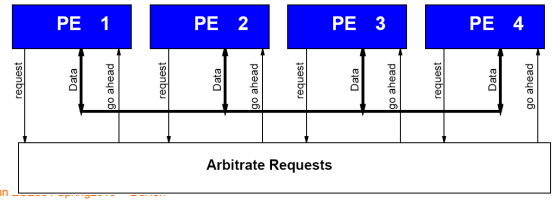
Formulation

Alternative

- 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



Dynamic Bus Example

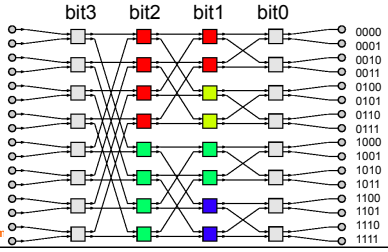
- 4 PEs
  - Potentially each send out result on change
  - Value only changes with probability 0.1 on each "cycle"
  - TM: Slot for each
    - PE0 PE1 PE2 PE3 PE0 PE1 PE2 PE3
  - Dynamic: arbitrate based on need
    - None PE0 none PE1 PE1 none PE3 ….
- TM either runs slower (4 cycles/compute) or needs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

Network Example

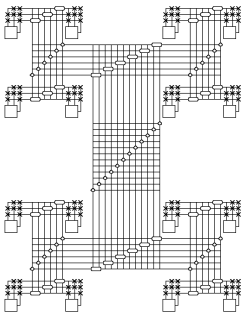
- Time Multiplexed
  - As assumed so far in class
  - Memory says how to set switches on each cycle
- Dynamic
  - Attach address or route designation
  - Switches forward data toward destination

Butterfly

- Log stages
- Resolve one bit per stage



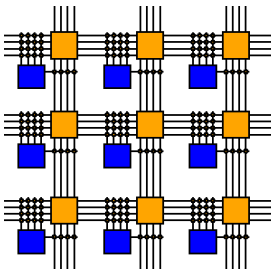
Tree Route



- Downpath resolves one bit per stage

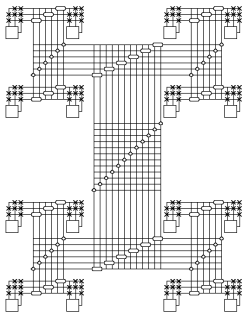
Mesh Route

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



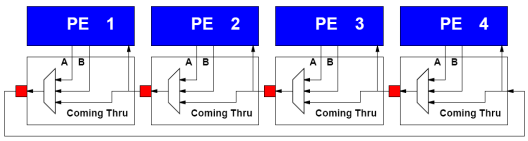
Dynamic Network Example

- Send to specific nodes on change
- E.g. shortest path
  - Send to successors



Design

Preclass 1



- Cycles from simulation?
- Best case possible?

Issue: Local online vs Global Offline

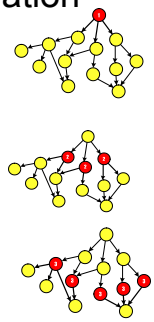
- Dynamic must make local decision
  - Often lower quality than offline, global decision

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

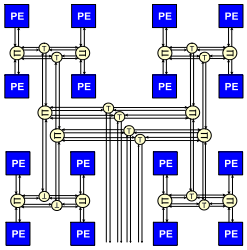
Spreading Activation

- Start with few nodes active
- Propagate changes along edges



Butterfly Fat Trees (BFTs)

- Familiar from Day 19
- Similar phenomena with other topologies
- Directional version



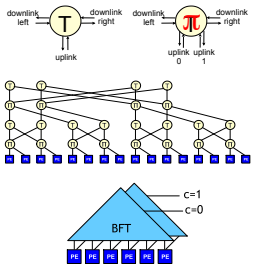
BFT Terminology

T = t-switch

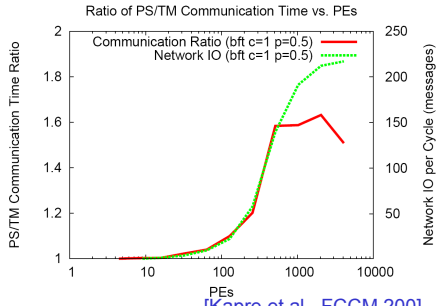
π = pi-switch

p = Rent Parameter
(defines sequence of T and π switches)

c = PE IO Ports
(parallel BFT planes)

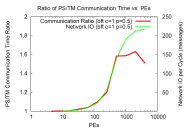


Iso-PEs



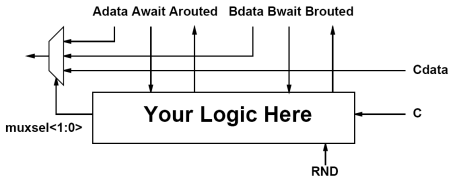
[Kapre et al., FCCM 200]

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 2



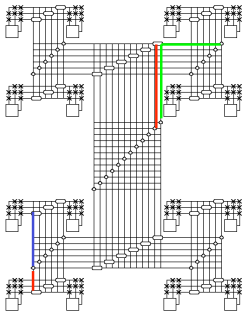
- Logic for muxsel<0>?
- Logic for Arouted?
- Gates?
- Gate Delay?

Issue 2: Switch Complexity

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

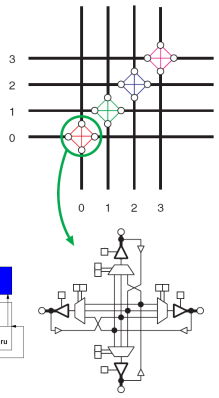
Congestion in Network

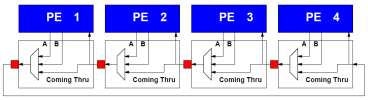
- What happens when contend for resources in network?



Mesh Congest

- Preclass 1 ring similar to slice through mesh
- A,B – corner turns
- May not be able to route on a cycle

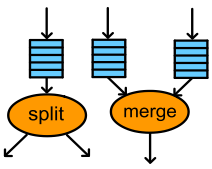


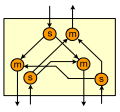


FIFO Buffering

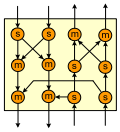
- Store inputs that must wait until path available
  - Typically store in FIFO buffer
- How big do we make the FIFO?

PS Hardware Primitives





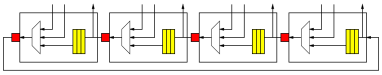
T-Switch



π -Switch

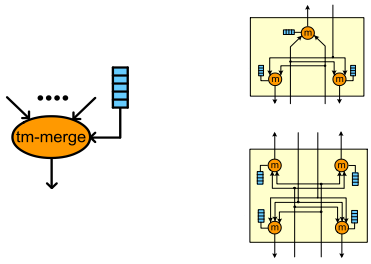
FIFO Buffer Full?

- What happens when FIFO fills up?



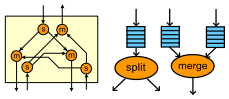
- Maybe backup network
- Prevent other routes from using
  - If not careful, can create deadlock

TM Hardware Primitives

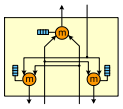


Area in PS/TM Switches

- Packet (32 wide, 16 deep)
  - 3split + 3 merge
  - Split 79
    - 30 ctrl, 33 fifo buffer
  - Merge 165
    - 60 ctrl, 66 fifo buffer
  - Total: 244



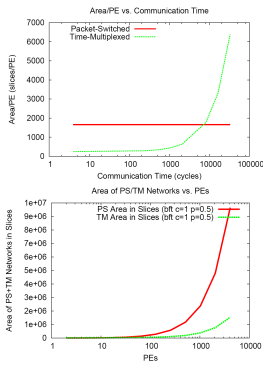
- Time Multiplexed (16b)
  - 9+(contexts/16)
  - E.g. 41 at 1024 contexts



- 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 3

- 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

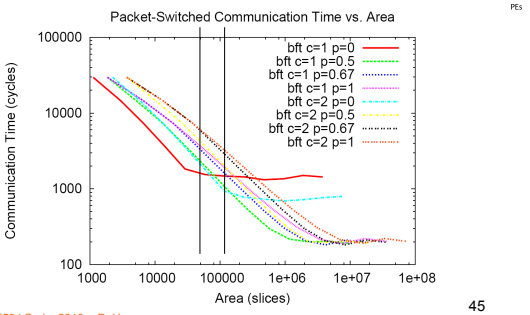
Analysis

- 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

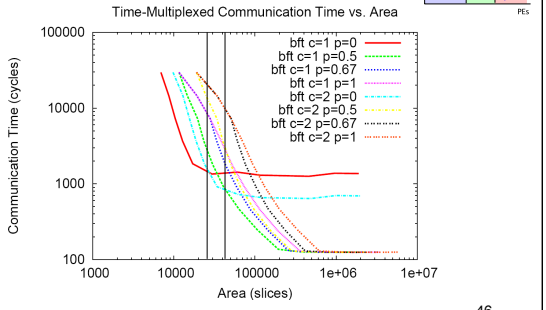
Area Analysis

- 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

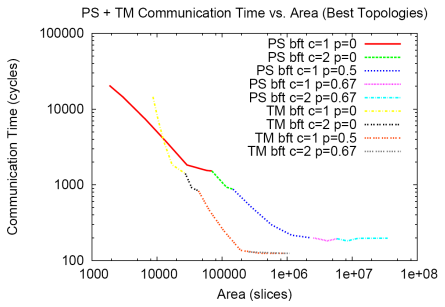
PS Iso-Area: Topology Selection



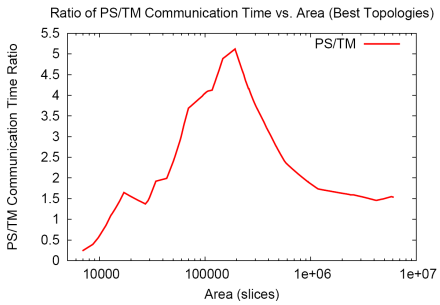
TM Iso-Area



Iso-Area

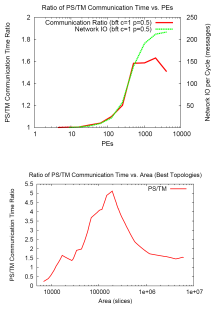


Iso-Area Ratio



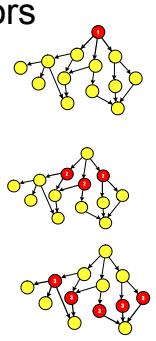
Iso-Area

- 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
- Iso-Area = TM ~5x better

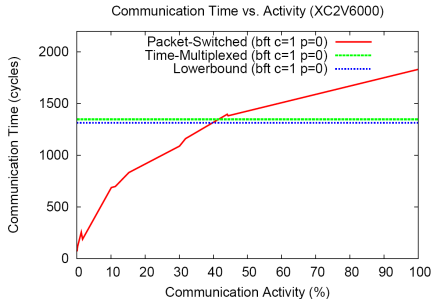


Activity Factors

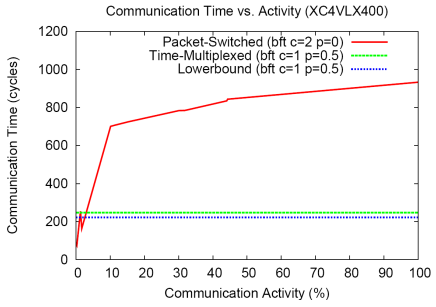
- Activity = Fraction of traffic to be routed
- TM needs to route all
- PS can route fraction
- Variable activity queries in ConceptNet
  - Simple queries ~1% edges
  - Complex queries ~40% edges



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 Monday
- Office Hours today
  - Last regularly scheduled
- Reading for Monday
  - None recommended (several suggested)
  - Read/ponder final exercise

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