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

---

<table>
<thead>
<tr>
<th>symbol</th>
<th>bits</th>
<th>encode</th>
<th>symbol</th>
<th>bits</th>
<th>encode</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>3</td>
<td>111</td>
<td>a</td>
<td>4</td>
<td>1111</td>
</tr>
<tr>
<td>b</td>
<td>6</td>
<td>100000</td>
<td>b</td>
<td>6</td>
<td>000001</td>
</tr>
<tr>
<td>c</td>
<td>5</td>
<td>10001</td>
<td>d</td>
<td>5</td>
<td>0001</td>
</tr>
<tr>
<td>e</td>
<td>3</td>
<td>001</td>
<td>f</td>
<td>6</td>
<td>100000</td>
</tr>
<tr>
<td>g</td>
<td>3</td>
<td>001</td>
<td>h</td>
<td>3</td>
<td>10000</td>
</tr>
<tr>
<td>i</td>
<td>4</td>
<td>001011</td>
<td>j</td>
<td>10</td>
<td>011000110</td>
</tr>
<tr>
<td>k</td>
<td>8</td>
<td>100100011</td>
<td>l</td>
<td>5</td>
<td>00011</td>
</tr>
<tr>
<td>m</td>
<td>6</td>
<td>1000100</td>
<td>n</td>
<td>6</td>
<td>100010</td>
</tr>
<tr>
<td>o</td>
<td>4</td>
<td>00100</td>
<td>p</td>
<td>6</td>
<td>1000100</td>
</tr>
<tr>
<td>q</td>
<td>10</td>
<td>011000010</td>
<td>r</td>
<td>4</td>
<td>00000</td>
</tr>
<tr>
<td>s</td>
<td>5</td>
<td>10011</td>
<td>t</td>
<td>4</td>
<td>1001</td>
</tr>
<tr>
<td>u</td>
<td>6</td>
<td>1001000</td>
<td>v</td>
<td>7</td>
<td>010000</td>
</tr>
<tr>
<td>w</td>
<td>6</td>
<td>1000100</td>
<td>x</td>
<td>7</td>
<td>01000000</td>
</tr>
<tr>
<td>y</td>
<td>5</td>
<td>010100</td>
<td>z</td>
<td>5</td>
<td>010010</td>
</tr>
</tbody>
</table>

---

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

---

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 PE0 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

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 \(\rightarrow\) more serial
    - Many "messages" local to cluster, no routing
  - Large trees \(\rightarrow\) 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

\[ T = t\text{-switch} \]
\[ \pi = \text{pi-switch} \]
\[ p = \text{Rent Parameter} \]
\[ (\text{defines sequence of } T \text{ and } \pi \text{ switches}) \]
\[ c = \text{PE IO Ports} \]
\[ (\text{parallel BFT planes}) \]

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

Iso-PEs

- 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 Congestion**

- 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**

**FIFO Buffer Full?**

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

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

PS vs TM Switches

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

Area in PS/TM Switches

- Packet (32 wide, 16 deep)
  - 3 split + 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 (16x4-LUT)
- Area in FPGA slice counts

Preclass 3

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

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

Communication Time vs. Activity (XC2V6000)

Packet-Switched (left 0:1 prob)
Time-Multiplexed (left 0:1 prob)
Lower-bound (left 0:1 prob)

Crossover could be less

Communication Time vs. Activity (XC4VLX400)

Packet-Switched (left 0:1 prob)
Time-Multiplexed (left 0:1 prob)
Lower-bound (left 0:1 prob)

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