Segment Gating for Static Energy Reduction in Networks–on–Chip

Kyle C. Hale
Boris Grot
Stephen W. Keckler

December 12, 2009
Motivation

- Static Energy consumption due to leakage threatening to become dominant

- Network constitutes a substantial (up to 36%) portion of total chip energy [Kim et al., ISLPED ‘03]

- Abundant on-chip bandwidth often underutilized due to low injection rates
Outline

- Background
- Segment Gating
- Methodology
- Evaluation
  - Optimal Gating Scheme
  - Static Scheme with Random Segment Selection
  - Static Scheme with Intelligent Segment Selection
  - Dynamic Gating Scheme
- Future Work
Reducing Leakage in the Network

- Gated-\( V_{DD} \) (power gating) at various granularities
- Power-aware buffer designs
  [Chen & Peh, ISLPED ’03]
- Slow-Silent VCs (DVFS applied to links, power gate the buffers)
  [Matsutani et al., NOCS ‘08]
Additional Gating Targets

- Aggressive gating of idle resources
- Link
  - Driver, repeaters
- Router
  - All VC buffers and management logic
  - Xbar ports
    - line drivers & switching elements
  - Allocators
    - 2-level stateful allocators
Outline

- Background
- Segment Gating
- Methodology
- Evaluation
  - Optimal Gating Scheme
  - Static Scheme with Random Segment Selection
  - Static Scheme with Intelligent Segment Selection
  - Dynamic Gating Scheme
- Future Work
Three types of Gating Schemes
- Optimal Gating (oracle, no cost)
  - Upper bound on energy savings
- Static Segment Gating
  - Off-line decision on which segments to gate
- Dynamic Segment Gating
  - “On-line” decisions based on dynamic workload

Evaluated via analytical approaches
- Not simulation based (for now)
- Effects of contention ignored
Optimal Gating Scheme

- No power-down/wake-up overheads (latency, energy)
- Segment shuts down during any period of inactivity and instantly wakes up on-demand
- Used as a baseline measurement for static schemes
Static Segment Gating

- Objective: Turn off a certain number of segments before the run of the workload
- Measure impact of gating through effect on hop counts
- Static analysis tool:
  - Represents mesh as directed graph
  - Hop counts derived from shortest path lengths between communicating nodes
  - Shortest paths may be longer than min manhattan routes
- Segments turned off via stochastic process
  - Invariant: full connectivity maintained
  - Take multiple samples to generate a distribution
Select segments at random to power down

Limits of static segment gating:
  ◦ 161 segments (links) out of 224 gated in a 64-node mesh
  ◦ A gated segment remains in that state for rest of workload

For certain traffic patterns, a random decision could lead to a bad choice
Random selection ignores characteristics of traffic patterns
Instead, pick segments based on link utilization
  ◦ For applications with communication regularity
  ◦ Requires us to know communication pattern \textit{a priori}

Two–stage approach
  ◦ Stage 1: Pick segments (links) with utilization zero
    ◦ 92 for \textit{bit–complement}
    ◦ 100 for \textit{transpose}
  ◦ Stage 2 (iterative):
    ◦ Turn off least–utilized segment
    ◦ Recompute utilization based on new traffic flow
Dynamic Segment Gating

- Objective: Dynamically gate segments to accommodate a changing workload
  - PARSEC traces run through cycle-accurate network simulator
    - Log each link’s idle/active periods
- Off-line analysis of activity logs
  - Combine with power model
  - Gate idle links
  - Wake up links on demand
  - Ignore contention and wake-up delays
  - Segments must be gated long enough to amortize energy cost of wake-up & power-down
### Power Model

#### per-component energy values

<table>
<thead>
<tr>
<th>Component</th>
<th>Static (nJ/cycle)</th>
<th>Dynamic (nJ/flit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flit buffers</td>
<td>1.5</td>
<td>7.23</td>
</tr>
<tr>
<td>Crossbar</td>
<td>0.491</td>
<td>10.3</td>
</tr>
<tr>
<td>Allocators</td>
<td>0.215</td>
<td>0.7</td>
</tr>
<tr>
<td>Link</td>
<td>0.556</td>
<td>8.1</td>
</tr>
</tbody>
</table>

- Derived using ORION 2.0
- Allocator energy: energy from 1\textsuperscript{st} & 2\textsuperscript{nd} level switch and VC allocators combined
- Note: Buffers only account for 55% of leakage energy!
## Network Configuration

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topology</strong></td>
<td>64-node mesh</td>
</tr>
<tr>
<td><strong>Channels</strong></td>
<td>128 bits wide, 1 cycle/link</td>
</tr>
<tr>
<td><strong>Synthetic Workloads</strong></td>
<td>Uniform random, transpose, bit-complement. Each workload comprises 1,000 packets injected by each node</td>
</tr>
<tr>
<td><strong>PARSEC traces</strong></td>
<td>Blackscholes, bodytrack, fluidanimate, vips, x264. Sim-medium datasets</td>
</tr>
<tr>
<td><strong>Router details</strong></td>
<td>2-stage speculative pipeline, 5 ports, 4 VCs/port, 5 flits/VC</td>
</tr>
</tbody>
</table>
Outline

- Background
- Segment Gating
- Methodology
- Evaluation
  - Optimal Gating Scheme
  - Static Scheme with Random Segment Selection
  - Static Scheme with Intelligent Segment Selection
  - Dynamic Gating Scheme
- Future Work
Optimal Energy Consumption

- **Uniform Random**
- **Transpose**
- **Bit Complement**

Total Energy (µJ)

- Dynamic
- Leakage
~20 edges can be removed with a negligible hop count increase

2–4x increase in hop count with max # of segments gated
Total Energy With Static Segment Gating (Random Selection)

Department of Computer Science
The University of Texas at Austin
Dynamic Energy Incurred from Utilization-aware Segment Gating

![Dynamic Energy Graph](image)
Dynamic Policy Space

- Two policies for idle period and break-even point
  - Aggressive: 2 cycles idle, 10 cycles to break even
  - Conservative: 10 cycles idle, 50 cycles to break even
Energy Consumption for PARSEC Benchmark Traces with Dynamic Policies
Future Work

- Detailed simulation and analysis
- Consider performance impact and contention
- Other network configurations
- Clearly establish regimes that should use Segment Gating
- Explore application to fault tolerant systems
Conclusions

- Real applications show sparse communication so potential for static energy savings is high
- Using link utilization, we can minimize dynamic energy incurred from gating segments statically
- Aggressive dynamic policy gives us static energy savings for up to 99% of cycles