A Hybrid Buffer Design with STT-MRAM for On-Chip Interconnects

Hyunjun Jang, Baik Song An, Nikhil Kulkarni, Ki Hwan Yum, and Eun Jung Kim

Dept. of Computer Science & Engineering
Texas A&M University
Outline

- Background of NoC
- Motivation of selecting STT-MRAM
- Challenges in using STT-MRAM
- Approaches
  - Hybrid Buffer Design
  - Simple & Lazy Migration Scheme
- Performance and Power Evaluation
- Conclusions
Networks-on-Chip (NoCs)

- NoCs for Large-Scale Chip Multi-Processors (CMPs)
- Packet-Switching Networks
  - Switch-based interconnects
    - Scalable
    - More suitable for large-scale Multi-Processor Systems

But, Power & Area Budgets in On-Chip Networks are very Limited
Why STT-MRAM in NoCs

- Near-zero leakage power compared to SRAM or DRAM
- Much higher density than SRAM (more than 4xs)
- Much higher endurance compared to other Non-volatile memories e.g., PCM, or Flash
  - Tolerate much more frequent write accesses

STT-MRAM bit storage (MTJ)

Hyunjun Jang - NOCS 2012
Weaknesses of STT-MRAM

- Long write latency compared to SRAM
  - More than 10 cycles
- High write power compared to SRAM
  - More than 8xs

To exploit the benefits of STT-MRAM, these challenges should be addressed first.
Approaches

- Hiding the Long Write Latency, while Maximizing Area Efficiency
  - SRAM + STT-MRAM Hybrid Buffer Design

- Sacrificing the Retention Time
  - From 10yrs to 10ms
  - Accordingly, latency also changes: 3.2 ns → 1.8ns, which is corresponding to 6 cycles in 3GHz clock frequency

- Reducing the Dynamic Write Power
  - Adaptive flit migration scheme in hybrid buffer considering current SRAM buffer occupancy
Hybrid Buffer Design

- Hiding the Long Write Latency (write lat = 6cycles)
Hybrid Buffer Design

Hiding the Long Write Latency (write lat = 6cycles)
Hybrid Buffer Design

- Hiding the Long Write Latency (write lat = 6cycles)

Hyunjun Jang - NOCS 2012
Hybrid Buffer Design

- Hiding the Long Write Latency (write lat = 6cycles)

But, in a low network load, migration energy is unnecessarily wasted.

This is a Simple Migration Scheme:
Read/Write can be done every cycle.

Hyunjun Jang - NOCS 2012
Reducing Dynamic Power Consumption

Lazy Migration Scheme

- IF (SRAM Buffer Occupancy >= Threshold)
  - Start migrating flits to STT-MRAM
- ELSE
  - Maintain flits in SRAM

- e.g. threshold in SRAM4 case: 0%, 25%, 50%, 75%

ref. Credit-based Flow Control
- Only considers SRAM buffer in credit management
Front-end SRAM Buffer Size

- In our experiment, Flits written into buffer stay at least 3 cycles in each on-chip router (Intra-router latency)
- It is possible to reduce front-end SRAM from 6 to 3
  - Thus, we can replace more SRAM with STT-MRAM

![Graph showing latency (cycles) vs. injection rate (flits/node/cycle)](image)

3 cycles
Various Hybrid Buffer Configurations

- STT-MRAM is 4xs denser than SRAM
- Therefore, under the same area budget, 1 SRAM space can be replaced with 4 STT-MRAM space
- So, under the baseline SRAM6 space,
  - SRAM5-STT4
  - SRAM4-STT8
  - SRAM3-STT12
  - SRAM2-STT16

All these 4 different hybrid configurations have same area budget (SRAM6)

Performed experiments to find best hybrid buffer configuration

Hyunjun Jang - NOCS 2012
## Performance/Power Evaluation

- **Performance Model**: Cycle-accurate on-chip network simulator
  - Models all router pipeline stages in detail
- **Power Model**: Orion for both dynamic and leakage power estimation

<table>
<thead>
<tr>
<th>Topology</th>
<th>8×8 Mesh, 2D-Torus, Flattened BFly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing</td>
<td>XY, O1TURN</td>
</tr>
<tr>
<td># of VC/Port</td>
<td>4</td>
</tr>
<tr>
<td>Buffer Depth/VC (Same area budget)</td>
<td>SRAM6(baseline), SRAM5-STT4, SRAM4-STT8, SRAM3-STT12, SRAM2-STT16</td>
</tr>
<tr>
<td>Packet Length</td>
<td>4 flits (128bits/flit)</td>
</tr>
<tr>
<td>Synthetic Traffic, Benchmark</td>
<td>UR, BC, NN, Splash-2</td>
</tr>
<tr>
<td><strong>SRAM</strong> Read, Write Energy</td>
<td>5.25 (pJ/flit), 5.25 (pJ/flit)</td>
</tr>
<tr>
<td><strong>SRAM</strong> Read, Write Latency</td>
<td>1 cycle for Read and Write</td>
</tr>
<tr>
<td><strong>STT</strong> Read, Write Energy</td>
<td>3.826 (pJ/flit), 40.0 (pJ/flit)</td>
</tr>
<tr>
<td><strong>STT</strong> Read, Write Latency</td>
<td>1 cycle for Read, 6 cycles for Write</td>
</tr>
</tbody>
</table>
Performance Analysis - Different Traffic

- Traffic (UR) - 18%
- Traffic (BC) - 28%

Injection Rate (flts/node/cycle) vs. Latency (cycles)

Hyunjun Jang - NOCS 2012
Performance Analysis
- Different Routing, Topology

- Routing (O1TURN)
- Topology (2D-Torus)

![Graphs showing latency vs. injection rate for different memory technologies in a network. The graphs compare SRAM memories with different STT values, indicating performance differences at 15% and 13%.]
Performance Analysis - Various STT Write latencies

- Write latencies (30, 10, 6 cycles)

![Bar chart showing normalized throughput for different write latencies and various STT-MRAM configurations.](chart.png)

- 30 cycles: 11% improvement
- 10 cycles: 13% improvement
- 6 cycles: 18% improvement

Hyunjun Jang - NOCS 2012
Performance Analysis - Benchmark Test

- SPLASH-2 parallel benchmarks

![Diagram showing latency comparison between SRAM4 and SRAM3_STT4]
Power Analysis

- Dynamic Power consumption of Input Buffers
- Dynamic + Leakage Power consumption of on-chip routers

Graphs showing normalized power consumption for different injection rates (flits/node/cycle), comparing SRAM, SIMPLE, LAZY (0.25), LAZY (0.5), and LAZY (0.75) with percentage changes in power reduction:

- Dynamic Power consumption: -53%
- Dynamic + Leakage Power consumption: -16%, +4%

Hyunjun Jang - NOCS 2012
Conclusions

- Hybrid Buffer Design with STT-MRAM
  - Provide more buffer space under the same area budget
  - Throughput-efficient

- Performance Improvement
  - 21% on average in synthetic workloads
  - 14% on average in SPLASH-2 parallel benchmarks

- Power Savings
  - Lazy migration scheme reduces power by 61% on average compared to simple migration scheme