ENEE 759H, Spring 2005
Memory Systems: Architecture and Performance Analysis

Fully Buffered DIMM Memory System

Credit where credit is due:
Slides contain original artwork (© Jacob, Wang 2005)
DRAM Datarate Trends

Commodity DRAM Devices Datarate:
~Doubling Every 3 Years (log scale)

- SDRAM (min burst:1)
- DDR SDRAM (min burst:2)
- DDR2 SDRAM (min burst:4)
- XDR (min burst:16)
  3.2 Gb/s (differential pair)
Recall - Multidrop Bus

Higher datarate, but loss of capacity

SDRAM - Max 8/6 ranks (loads); (133 MHz)
DDR SDRAM - Max 6 ranks; (400 Mbps)
DDR2 - Max 4 ranks; (667/800 Mbps)
DDR3 - Max 2 ranks; (800+ Mbps)
DDR II Memory System

How to keep commodity DRAM devices, keep high datarate, but maintain or increase capacity? (large servers)
FB-DIMM Solution

Point to point chaining of sub-memory systems

Advanced Memory Buffer (AMB)

up to 8 FB DIMMS

Use commodity DRAM devices
AMB Block Diagram:

- Pass-through Logic
- PLL
- LAI Controller
- SMBus Controller
- Thermal Sensor
- De-serializer & Decode
- Data Bus Interface
- Serializer
- Pass-through and Merging Logic
- Command & Addr
- Data Bus

From Controller: SMBus
To Controller: SMBus Interface
Salient Points

- ASIC-to-ASIC signalling. High datarate. Differential pair signalling, 6X data rate of DRAM devices. (i.e. DRAM: 800 Mbps, FBD link datarate: 4.8 Gbps)

- Asymmetric configuration. Higher inbound bandwidth (14 inbound pin-pair, 10 outbound pin-pair)

- Deskewing in ASIC, not on board

- Longer idle system latency, but much higher pin-bandwidth

- Keep increasing datarate and solves capacity problem

- Uses commodity DRAM devices.
## Technology Roadmap (ITRS)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi Generation (nm)</td>
<td>90</td>
<td>65</td>
<td>45</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>CPU MHz</td>
<td>3990</td>
<td>6740</td>
<td>12000</td>
<td>19000</td>
<td>29000</td>
</tr>
<tr>
<td>MLogicTransistors/ cm^2</td>
<td>77.2</td>
<td>154.3</td>
<td>309</td>
<td>617</td>
<td>1235</td>
</tr>
<tr>
<td>High Perf chip pin count</td>
<td>2263</td>
<td>3012</td>
<td>4009</td>
<td>5335</td>
<td>7100</td>
</tr>
<tr>
<td>High Performance chip cost (cents/pin)</td>
<td>1.88</td>
<td>1.61</td>
<td>1.68</td>
<td>1.44</td>
<td>1.22</td>
</tr>
<tr>
<td>Memory pin cost (cents/pin)</td>
<td>0.34 - 1.39</td>
<td>0.27 - 0.84</td>
<td>0.22 - 0.34</td>
<td>0.19 - 0.39</td>
<td>0.19 - 0.33</td>
</tr>
<tr>
<td>Memory pin count</td>
<td>48-160</td>
<td>48-160</td>
<td>62-208</td>
<td>81-270</td>
<td>105-351</td>
</tr>
</tbody>
</table>

**Trend:** Free Transistors & Costly Interconnects
Choices for Future

Direct Connect Commodity DRAM
Low Bandwidth + Low Latency

Indirect Connection
Highest Bandwidth

Inexpensive DRAM

Highest Latency
## Pin Count Comparison

<table>
<thead>
<tr>
<th></th>
<th>DDR2</th>
<th>FBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datarate (Mbps)</td>
<td>667</td>
<td>4000</td>
</tr>
<tr>
<td>Pin Count (Data Bus)</td>
<td>108</td>
<td>-</td>
</tr>
<tr>
<td>Channel pin count</td>
<td>141</td>
<td>59</td>
</tr>
<tr>
<td>(without pwr and gnd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel pin count</td>
<td>~200</td>
<td>~70</td>
</tr>
<tr>
<td>(with pwr and gnd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak bandwidth (GB/s)</td>
<td>5.3</td>
<td>8.0</td>
</tr>
<tr>
<td>Theoretical Efficiency</td>
<td>213</td>
<td>914</td>
</tr>
<tr>
<td>(bandwidth/pin) (Mbps)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

72 data pins + 18 diff pairs of DQS strobes

~ 4.5x increase in pin-bandwidth (counting power and ground)
Routing Comparison:

1 Channel of registered DDR2 SDRAM.
2 routing layer + power plane
path length matched traces
~200 traces per channel

2 Channels of FB-DIMM
2 routing layer including power delivery
no need to match path lengths (deskewing on ASIC)
~70 traces per channel
FB-DIMM Protocol:

- Pseudo network-like packet protocol
- Fixed packet (frame) size
- Everything is 12 “beats” (or 2 DRAM beats = 1 clk)
- $12 \times 14 = 168$ bits/frame upstream (to controller)
- 144 bit data payload per frame upstream (read data)

- $12 \times 10 = 120$ bits/frame downstream (from controller)
- 72 bit payload per frame downstream (write data)
- Write bandwidth = 0.5x of read bandwidth
- Total bandwidth = 1.5x of single module bandwidth
Downstream (southbound) I:

- 10 bit lanes x 12 bits = 120 bits per frame
- Command + write data
- 72 bits of data per frame
- 24 bits of command (max of 3 commands per frame)
- 2 bits command type
- 22 bits of CRC, strong protection in normal mode
- 10 bits of CRC in failover mode (reduced protection)

Transparent failover: lose a bitlane, keep on going.
Downstream (southbound) II:

- DRAM Commands
  - Activate (row-bank)
  - Write (column)
  - Read (column)
  - Precharge (all banks)
  - Precharge (bank)
  - Auto Refresh
  - Enter Self Refresh
  - Enter Power Down
  - Exit Self Refresh and Exit Power Down

- Channel Commands
  - Channel NOP
  - Sync
  - Soft Channel Reset
  - Write Config Register
  - Read Config Register
  - DRAM CKE per DIMM
  - DRAM CKE per Rank
  - Debug
  - ensure transmission density
  - transient bit failure recovery
  - Allows tuning of refresh mechanism. Per DIMM or per rank
Write FIFO

Southbound Frames (12 FBD beats per frame)

Northbound Frames

Burst-of-eight
(12 FBD beats per frame)

Burst-of-eight
(2 DRAM beats per frame)

Write data
(1 beat per frame)
buffered/queued
need not be
contiguous frames.
May be separated
by arbitrary number
of intervening frames.
Upstream:

- 14 bit lanes
- \(12 \times 14 = 168 \text{ bits/frame upstream (to controller)}\)
- 2 (72 or 64 bit) data payload per frame upstream
- Last AMB on channel initiates northbound frames.
- Idle frames contain permuting data pattern (LFSR)

- 14 bit channel with 12 bits CRC per payload. Failover to 13 bit channel with 6 bits of CRC per payload.
- 13 bit channel with 6 bits CRC per payload. Failover to 12 bit channel with ECC coverage only.
- 12 bit channel with 6 bits CRC per payload. No failover.
Command Scheduling I:

- Short channel: variable read latency capability

Oops

Delay read request until ctl knows that empty Northbound frame exists

FBD-merging, no rank-to-rank turnarounds
Command Scheduling II:

- Long channel: fixed latency
  longer channel = longer read latency for all requests

buffer/delay

“token passing”
Latency:

From Controller: ~2 ns

Point to point interface:

Pass-through Logic

PLL

LAI Controller

SMBus Controller

Thermal Sensor

De-serializer & Decode

Serializer

Data Bus Interface

Data Bus

Command & Addr

SMBus ~8.1 ns @ 667 Mbps

Pass through and Merging Logic

~5.0 ns @ 667 Mbps
Hypothetical Latency Distrib.

Longer Idle system latency, but higher pin-bandwidth. Lower avg latency?
## Power Impact

<table>
<thead>
<tr>
<th></th>
<th>9 x8</th>
<th>18 x8 or 18 x4</th>
<th>36 x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM power (assume ~0.3W per device)</td>
<td>2.7W</td>
<td>5.4W</td>
<td>10.8W</td>
</tr>
<tr>
<td>AMB Power</td>
<td>4W</td>
<td>4W</td>
<td>4W</td>
</tr>
<tr>
<td>Total FBD Power</td>
<td>~6.7W</td>
<td>~9.4W</td>
<td>~14.8W</td>
</tr>
<tr>
<td>Power overhead</td>
<td>~148%</td>
<td>~74%</td>
<td>~37%</td>
</tr>
</tbody>
</table>
Summary

- Can merge read data frames and achieve 100% read bandwidth utilization
- FBD Channel relies on deep channel to obtain “full bandwidth efficiency”
- Longer latency, particularly with long channel configurations.
- Asymmetric configuration. Peak write bandwidth = 0.5x peak read bandwidth. peak read + write bandwidth = 1.5x of single channel DIMM bandwidth