Bandwidth Engine® Serial Memory Chip Breaks 2 Billion Accesses/sec

Michael J. Miller
VP Technology Innovation & Systems Applications, MoSys
Network Memory Access Requirements

Memory bandwidth (# of accesses / packet)

- DPI (Regular session)
- DPI (String)
- FIB (Algorithm)
- Layer 2 (Forwarding)
- Queuing/Scheduling
- Statistics/Counting
- What's needed for packet forwarding
- Commodity memory
- 4 off-chip DDR3-DRAM channels bandwidth and capacity

Optimized For Access Rate

- 100 GE Requires 150 Mpps x ~14 - 16 Access => 2 GA/s

Optimized For Density

Source: HotChips 2010 Huawei

©MoSys, Inc. 2011. All Rights Reserved.
Bandwidth Engine Design Challenge

- Networking memory characteristics
  - Synchronous interface
  - Modulo x9 accesses
  - Small quanta (36b to 72b per access)
  - High access rate

- Challenge: Create a 2+ GigaAccess networking memory device
  - High access availability (4x existing devices)
  - Minimize system power per access
  - Utilize existing electrical interfaces
  - Support 100G designs and scale to 400G
GA/s and tRC

- **GA/s is the number of billions of unique access to memory**
  - Access is a unique read or write “Transaction”
  - Depends on: the bandwidth, the cycle rate (tRC) and the transfer size...
  - Maximum GA/s = (I/O bandwidth: Gbps) / (Access size: bits)
  - Sustained GA/s = (# simultaneous bank accesses) x (memory cycle rate: 1/tRC)

- **tRC is the amount of time to cycle a memory bit for read or write**
  - Depends on: bitline RC & power/area allotted to Sense Amp
  - Bitline RC ≈ (# bit cells) x (RC per bit cell)
Keeping Up With Packet Rates

- **Memory Interface**
  - Parallel interfaces are becoming bottlenecks, scaling slow down
  - Serial interconnect is already everywhere except memory

- **Memory Core Performance**
  - Cycle the memory faster
    - Vs. Power/Density/Refresh/... tradeoffs
  - 400GE -> .8 ns or 1.2 GHz memory cells

- **Memory Architecture**
  - Run multiple banks in parallel
  - Use “Round Robin/Ping Pong” algorithms scale up effective access rate
    - tRC/n but also Mbits/n

- **Or a combination thereof**
  - There will have to be tradeoffs inevitably
Multi Core => Multi-Partition & Multi-bank

Multi-threaded Multi-Cores allow for high processing throughput.

Multi-bank Multi-partitions allow for high access availability.

Bandwidth Engine

Ingress and egress connections through Serial Links.

ALU for functional acceleration.
Breakthrough Performance

...4X Throughput of RLDRAM
- 2.75GA: >2 billion reads/sec (2 GA), 1B writes
- 72b words each access
- 15.9 ns roundtrip latency
- Up to 16, 10G CEI 11+ serial lanes
- Macro operations (RMW, Inc/Dec..)
- <7W worst case system power

High Density

...4-8X Density of QDR SRAM
- 576Mb 1T-SRAM
- 3.9ns bank tRC (1T-SRAM performance)

High Reliability

...70X better SER than 6T embedded SRAM
- Memory core: < 10 FIT/Mb native
- Interface: < 1 FIT

Array Manager & ALUs

Host

GCI Port

TX  RX

GCI Port

TX  RX

GCI Port A

RX  TX

GCI Port B

RX  TX

GCI Port

TX  RX
Bandwidth Engine Block Diagram

©MoSys, Inc. 2011. All Rights Reserved.
Conceptual Timing & Data Access Control

Read Latency of ~16ns
Chip Layout Balances Memory Access Paths
Package Layout Minimizes Rx/Tx Xtalk
BE Family Relative Performance

Giga-Accesses/sec (GA/s)

%Reads

0.0 0.5 1.0 1.5 2.0 2.5 3.0

Packet Buffer, State Mem. Zone

Control Plane Zone

BE 8888

BE 4488

BE 8844

RLDRAM3

QDR IIIe

Optimizing Read/Write Bandwidth

©MoSys, Inc. 2011. All Rights Reserved.
Bandwidth Engine On-chip Macro Operations

- 2X or Better I/O Performance, saves Power & Pins
- BE-1: 16b, 32b and 64b Add/Subtract
  - 4 SerDes lane BE-1 => 4M flows @ 2 counters per flow @ 250 Mpps,
- Possible Future Macro Operations
  - Data manipulation: Fully flexible increment/decrement, semaphore (R-M-W)
  - Pointer indirection for data structure walking
  - Data packing/unpacking
Four Stage Partition Macro Op Pipeline

GCI Port A

Addr ↓ 16b Imm

16

↓

16

↓

16b 32b or 64b

16b

32b or 64b

ECC

RA  WA  WD

2M x 72 Partition

RD

Note:
Conceptual pipeline

©MoSys, Inc. 2011. All Rights Reserved.
## Avoiding Collisions

258MHz => 129Mpps Per Partition x 4 => 516 Mpps Per BE

<table>
<thead>
<tr>
<th>Ld Addr Flow0</th>
<th>Op 16b Imm</th>
<th>Write Packet Count Flow0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ld Addr Flow0</td>
<td>Add “1”</td>
<td>Write Byte Count Flow0</td>
</tr>
<tr>
<td>Ld Addr Flow1</td>
<td>Add Byte # 0</td>
<td>Write Packet Count Flow1</td>
</tr>
<tr>
<td>Ld Addr Flow1</td>
<td>Add “1”</td>
<td>Write Byte Count Flow1</td>
</tr>
<tr>
<td>Ld Addr Flow2</td>
<td>Add Byte # 1</td>
<td>Write Packet Count Flow2</td>
</tr>
<tr>
<td>Ld Addr Flow2</td>
<td>Add Byte # 2</td>
<td>Write Byte Count Flow2</td>
</tr>
<tr>
<td>Ld Addr Flow3</td>
<td>Add “1”</td>
<td>Write Packet Count Flow3</td>
</tr>
<tr>
<td>Ld Addr Flow3</td>
<td>Add Byte # 3</td>
<td>Write Byte Count Flow3</td>
</tr>
<tr>
<td>Ld Addr Flow4</td>
<td>Add “1”</td>
<td></td>
</tr>
<tr>
<td>Ld Addr Flow4</td>
<td>Add “1”</td>
<td></td>
</tr>
<tr>
<td>Ld Addr Flow5</td>
<td>Add Byte # 4</td>
<td></td>
</tr>
<tr>
<td>Ld Addr Flow5</td>
<td>Add “1”</td>
<td></td>
</tr>
<tr>
<td>Ld Addr Flow6</td>
<td>Add Byte # 5</td>
<td></td>
</tr>
</tbody>
</table>

©MoSys, Inc. 2011. All Rights Reserved.
Tradeoff
“tRC” vs Density
Minimizing tRC can save power
Density vs tRC For Control Plane

- **tRC** is the amount of time to cycle a memory bit for read or write
  - Depends on: bitline RC & power/area allotted to Sense Amp
  - Bitline RC $\approx$ (# bit cells) x (RC per bit cell)
- **Density** is a function of:
  - Size of bit cell => Function of process node & drive
  - Ratio of #bit to Sense Amp => Length of bitlines
- **Proportional relation between density and tRC**
  - Sense Amp area $\sim$ 10x the area of 1 bit cell

- $150 \text{Mpps} \Rightarrow 6.7\text{ns}$
- Optimal tRC $\sim 3.3\text{ns}$
Speed Up Using Multi-Bank

Ping Pong Algorithm for 2x throughput

- Read and write in same 3.9ns $t_{RC}$ cycle -> 1.9ns effective $t_{RC}$
- Read gets priority in case of bank conflict
- Read from bank with most recent data according to bit map
- Write to the other bank and update bit map
- Bit map keeps record of most recent data
Case Study assumptions @ 100GE: Read, modify & write 288b every packet time (Packet count, Byte Count, Next schedule, etc.)

Packet arrival period in ns: 6.67
Record entry size bits: 288

<table>
<thead>
<tr>
<th>Device</th>
<th>Device Size in Mb</th>
<th>Rd/Wr t_{RC}</th>
<th>Required Speedup</th>
<th># Banks Per Entry</th>
<th># Updates per t_{RC}</th>
<th>Table Size in M</th>
</tr>
</thead>
<tbody>
<tr>
<td>QDR</td>
<td>144</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.50</td>
</tr>
<tr>
<td>BE</td>
<td>576</td>
<td>3.9</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>BE 2</td>
<td>576</td>
<td>3.1</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2.00</td>
</tr>
<tr>
<td>RLDAM II</td>
<td>576</td>
<td>15</td>
<td>5</td>
<td>9</td>
<td>4.5</td>
<td>0.22</td>
</tr>
<tr>
<td>RLDAM III</td>
<td>576</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>1.5</td>
<td>0.4</td>
</tr>
<tr>
<td>RLDAM III</td>
<td>1152</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>1.5</td>
<td>0.8</td>
</tr>
<tr>
<td>DRAM</td>
<td>2048</td>
<td>45</td>
<td>14</td>
<td>56</td>
<td>28</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Source for speed up ratio: PHd Dissertation: “Load Balancing & Parallelism for the Internet” Stanford University, Sundar Iyer
Effective Memory size vs Speed Up

Effective Memory Size

- Speed Up Factor
- Effective Memory Size
Comparing Performance and Power for 100G

State Memory 1M x 288b for 100 GE

Key Requirements:
1) 150Mpps x 288b Rd + Wr
2) 3.3ns effective tRC
3) 288 Mb effective density

Classification for 100 GE

Key Requirements:
1) 150Mpps x ~14 access
Summary Of Design Choices

- **Serial I/O**
  - Reuse the same electrical I/O as network interfaces
  - Scales same as network interfaces
  - Room for improvement on very short reach power

- **Multi-Bank Multi-Partition**
  - Increase access availability

- **Optimize for cycle time**
  - Achieves better system density for networking applications

- **Onchip ALU + Macro Operations**
  - Minimize I/O requirements for commands & data
  - Lower pin counts & system power
Thank You

Michael J. Miller
VP Technology Innovation & Systems Applications, MoSys