Why Is Memory So Important?
(Especially Today)
Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor.

Main memory system must scale (in size, technology, efficiency, cost, and management algorithms) to maintain performance growth and technology scaling benefits.
Memory System: A *Shared Resource View*

![Diagram of Memory System]

- Shared Memory
- Shared Memory Control
- Shared L3 Cache
- Shared L2 Cache
- Core 1, Core 2, Core 3, Core 4, Core 5, Core 6, Core 7, Core 8, Core 9
- Shared Memory Interconnect
- Storage

SAFARI
State of the Main Memory System

- Recent technology, architecture, and application trends
  - lead to new requirements
  - exacerbate old requirements

- DRAM and memory controllers, as we know them today, are (will be) unlikely to satisfy all requirements

- Some emerging non-volatile memory technologies (e.g., PCM) enable new opportunities: memory+storage merging

- We need to rethink/reinvent the main memory system
  - to fix DRAM issues and enable emerging technologies
  - to satisfy all requirements
Major Trends Affecting Main Memory (I)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending
Demand for Memory Capacity

- More cores ➔ More concurrency ➔ Larger working set

- Modern applications are (increasingly) data-intensive

- Many applications/virtual machines (will) share main memory
  - Cloud computing/servers: Consolidation to improve efficiency
  - GP-GPUs: Many threads from multiple parallel applications
  - Mobile: Interactive + non-interactive consolidation
  - …
Example: The Memory Capacity Gap

- Memory capacity per core expected to drop by 30% every two years
- Trends worse for memory bandwidth per core!
Major Trends Affecting Main Memory (II)

- Need for main memory capacity, bandwidth, QoS increasing
  - Multi-core: increasing number of cores/agents
  - Data-intensive applications: increasing demand/hunger for data
  - Consolidation: Cloud computing, GPUs, mobile, heterogeneity

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending
Major Trends Affecting Main Memory (III)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern
  - IBM servers: ~50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer 2003]
  - DRAM consumes power when idle and needs periodic refresh

- DRAM technology scaling is ending
Major Trends Affecting Main Memory (IV)

- Need for main memory capacity, bandwidth, QoS increasing

- Main memory energy/power is a key system design concern

- **DRAM technology scaling is ending**
  - ITRS projects **DRAM will not scale easily below X nm**
  - Scaling has provided many benefits:
    - higher capacity, higher density, lower cost, lower energy
The DRAM Scaling Problem

- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Access transistor should be large enough for low leakage and high retention time
  - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]

- DRAM capacity, cost, and energy/power hard to scale
Repeatedly opening and closing a row enough times within a refresh interval induces **disturbance errors** in adjacent rows in most real DRAM chips you can buy today.

loop:
    mov (X), %eax
    mov (Y), %ebx
    clflush (X)
    clflush (Y)
    mfence
    jmp loop
loop:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp loop
loop:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp loop
**loop:**
- `mov (X), %eax`
- `mov (Y), %ebx`
- `clflush (X)`
- `clflush (Y)`
- `mfence`
- `jmp _loop_`
# Observed Errors in Real Systems

<table>
<thead>
<tr>
<th>CPU Architecture</th>
<th>Errors</th>
<th>Access-Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Haswell (2013)</td>
<td>22.9K</td>
<td>12.3M/sec</td>
</tr>
<tr>
<td>Intel Ivy Bridge (2012)</td>
<td>20.7K</td>
<td>11.7M/sec</td>
</tr>
<tr>
<td>Intel Sandy Bridge (2011)</td>
<td>16.1K</td>
<td>11.6M/sec</td>
</tr>
<tr>
<td>AMD Piledriver (2012)</td>
<td>59</td>
<td>6.1M/sec</td>
</tr>
</tbody>
</table>

- **A real reliability & security issue**
- **In a more controlled environment, we can induce as many as ten million disturbance errors**

Main Memory
Main Memory in the System
Review: Memory Bank Organization

Read access sequence:

1. Decode row address & drive word-lines

2. Selected bits drive bit-lines
   - Entire row read

3. Amplify row data

4. Decode column address & select subset of row
   - Send to output

5. Precharge bit-lines
   - For next access
**Review: DRAM (Dynamic Random Access Memory)**

Bits stored as charges on node capacitance (non-restorative)
- bit cell loses charge when read
- bit cell loses charge over time

**Read Sequence**
1. address decode
2. drive row select
3. selected bit-cells drive bitlines (entire row is read together)
4. a “flip-flopping” sense amp amplifies and regenerates the bitline, data bit is mux’ed out
5. precharge all bitlines

**Refresh**: A DRAM controller must periodically read all rows within the allowed refresh time (10s of ms) such that charge is restored in cells.
The DRAM Subsystem
DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column
- Cell
Page Mode DRAM

- A DRAM bank is a 2D array of cells: rows x columns
- A “DRAM row” is also called a “DRAM page”
- “Sense amplifiers” also called “row buffer”

- Each address is a <row,column> pair
- Access to a “closed row” (Closed Page Policy)
  - Activate command opens row (placed into row buffer)
  - Read/write command reads/writes column in the row buffer
  - Precharge command closes the row and prepares the bank for next access
- Access to an “open row” (Open Page Policy)
  - No need to Activate
  - Keep row open until conflict
DRAM Bank Operation

Access Address:
(Row 0, Column 0)
(Row 0, Column 1)
(Row 0, Column 85)
(Row 1, Column 0)

Row address 0

Row decoder

Columns

Rows

Row 1

Row Buffer

CONFLICT!

Column address 05

Column mux

Data
The DRAM Chip

- Consists of multiple banks (8 is a common number today)
- Banks share command/address/data buses
- The chip itself has a narrow interface (4-16 bits per read)

- Changing the number of banks, size of the interface (pins), whether or not command/address/data buses are shared has significant impact on DRAM system cost
128M x 8-bit DRAM Chip
**DRAM Rank and Module**

- **Rank**: Multiple chips operated together to form a wide interface
- All chips comprising a rank are controlled at the same time
  - Respond to a single command
  - Share address and command buses, but provide different data

- A DRAM module consists of one or more ranks
  - E.g., DIMM (dual inline memory module)
  - This is what you plug into your motherboard

- If we have chips with 8-bit interface, to read 8 bytes in a single access, use 8 chips in a DIMM
A 64-bit Wide DIMM (One Rank)
A 64-bit Wide DIMM (One Rank)

- Advantages:
  - Acts like a high-capacity DRAM chip with a wide interface
  - Flexibility: memory controller does not need to deal with individual chips

- Disadvantages:
  - Granularity: Accesses cannot be smaller than the interface width
Multiple DIMMs

Advantages:
- Enables even higher capacity

Disadvantages:
- Interconnect complexity and energy consumption can be high
  → Scalability is limited by this
- 2 Independent Channels: 2 Memory Controllers (Above)
- 2 Dependent/Lockstep Channels: 1 Memory Controller with wide interface (Not Shown above)
Generalized Memory Structure
Generalized Memory Structure
The DRAM Subsystem
The Top Down View
DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column
- Cell
The DRAM subsystem

“Channel”

DIMM (Dual in-line memory module)

Processor

Memory channel

Memory channel
Breaking down a DIMM

DIMM (Dual in-line memory module)

Side view

Front of DIMM

Back of DIMM
Breaking down a DIMM

DIMM (Dual in-line memory module)

Front of DIMM

Rank 0: collection of 8 chips

Back of DIMM

Rank 1
Rank

Rank 0 (Front)

Rank 1 (Back)

Addr/Cmd

CS <0:1>

Data <0:63>

Memory channel
Breaking down a Rank

Rank 0

Chip 0

Chip 1

...  

Chip 7

<0:63>

<0:7>

<8:15>

<56:63>

Data <0:63>
Breaking down a Chip
Breaking down a Bank
DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column
- Cell
Example: Transferring a cache block

Physical memory space

0xFFFF...F
0x40
0x00

64B cache block

Mapped to

Channel 0

DIMM 0

Rank 0

Mapped to

Core™ i7
Example: Transferring a cache block

Physical memory space

0xFFFF...F

64B cache block

0x40

0x00

Chip 0

Chip 1

Rank 0

Chip 7

<0:7>

<8:15>

<56:63>

Data <0:63>
Example: Transferring a cache block

Physical memory space

0xFFFF...F

Chip 0

Row 0
Col 0

Chip 1

<0:7>
<8:15>
<56:63>

Data <0:63>

64B cache block

Chip 7

Rank 0

...
Example: Transferring a cache block

Physical memory space

Chip 0
Chip 1
Chip 7

Rank 0

Row 0
Col 0

64B cache block

0xFFFF...F

0x00

0x00

0x40

8B

0x40

8B

Data <0:63>

8B

<0:7>

<8:15>

<56:63>
Example: Transferring a cache block

Physical memory space

Chip 0

Chip 1

Chip 7

Row 0 Col 1

Data <0:63>

64B cache block

0xFFFF...F
Example: Transferring a cache block

Physical memory space

0xFFFF...F

32B cache block

Row 0 Col 1

Chip 0

Chip 1

Chip 7

Rank 0

Data <0:63>

<0:7> <8:15> <56:63>
Example: Transferring a cache block

A 64B cache block takes 8 I/O cycles to transfer.

During the process, 8 columns are read sequentially.