SRAM vs. DRAM

- DRAM = Dynamic RAM

- SRAM: 6T per bit
  - built with normal high-speed CMOS technology

- DRAM: 1T per bit
  - built with special DRAM process optimized for density
Hardware Structures

SRAM

Bit lines

DRAM
DRAM Read/Write

• Write
  – Charge bitline HIGH or LOW and set wordline HIGH

• Read
  – Bitline is precharged
  – Wordline is set
  – Depending on the charge, bitline becomes slightly higher or lower
Destructive Read

After read of 0 or 1, cell contains something close to 1/2
DRAM Chip Organization

Row Decoder

Sense Amps

Row Buffer

Column Decoder

Memory Cell Array

Bit lines

Word lines

Row Address

Column Address

Data Bus

Row
Address

Column
Address
• Differences with SRAM
  • reads are *destructive*: contents are erased after reading

  – Row buffer/DRAM Page
    • Read lots of bits all at once, and then parcel them out based on different column addresses
    • Read from the same row buffer from different locations order

  – “Fast-Page Mode” FPM DRAM organizes the DRAM row to contain bits for a complete page
    • row address held constant, and then fast read from the consecutive locations from the same page
DRAM Read Operation

Row Decoder

Memory Cell Array

Sense Amps

Row Buffer

Column Decoder

0x1FE

0x000

Accesses need not be sequential

Data Bus
So after a read, the contents of the DRAM cell are gone.

The values are stored in the row buffer.

Write them back into the cells for the next read in the future.
Fairly gradually, the DRAM cell will lose its contents even if it’s not accessed

- This is why it’s called “dynamic”
- Contrast to SRAM which is “static” in that once written, it maintains its value forever (so long as power remains on)

All DRAM rows need to be regularly read and re-written
Accesses are asynchronous: triggered by RAS and CAS signals, which can in theory occur at arbitrary times (subject to DRAM timing constraints).
Double-Data Rate (DDR) DRAM transfers data on both rising and falling edge of the clock.

Timing figures taken from "A Performance Comparison of Contemporary DRAM Architectures" by Cuppu, Jacob, Davis and Mudge.
Burst Access

• One command access, multiple bytes are read/written.
• Hardware provides multiple burst length option and software can set.
Example Memory Latency Computation

- FSB freq = 200 MHz, SDRAM
- RAS delay = 2, CAS delay = 2
- Scheduling in memory controller
  
  A0, A1, B0, C0, D3, A2, D0, C1, A3, C3, C2, D1, B1, D2

- Think about hardware complexity…
REVIEW: VIRTUAL ADDR & CACHE
Virtual Memory/Physical Memory

- Programmer’s view: virtual memory space
- Actual hardware’s view: Physical memory space
- In hardware: translation from virtual address to physical address

<table>
<thead>
<tr>
<th>Virtual Address</th>
<th>Physical Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Page Number</td>
<td>Physical Frame Num</td>
</tr>
<tr>
<td>Page Offset</td>
<td>Page Offset</td>
</tr>
</tbody>
</table>

Translation

Protection check!
Read/write, kernel/user?
Need for Translation

Virtual Address

0xFC51908B

Virtual Page Number | Page Offset

0xFC519 | 0x00152

Page Table

Physical Address

0x0015208B

Main Memory
CPU Memory Access

• Program deals with virtual addresses
  – "Load R1 = 0[R2]"

• On memory instruction
  1. Compute virtual address (0[R2])
  2. Compute virtual page number
  3. Compute physical address of VPN’s page table entry
  4. Load* mapping
  5. Compute physical address
  6. Do the actual Load* from memory

Could be more depending on page table organization
Impact on Performance?

• Every time you load/store, the CPU must perform two (or more) accesses!

• Even worse, every *fetch* requires translation of the PC!

• Observation:
  – Once a virtual page is mapped into a physical page, it’ll likely stay put for quite some time
Idea: Caching!

- Not caching of data, but caching of translations

TLB also has protection bits, R/W, kernel/user information
Translation Cache: TLB

- TLB = Translation Look-aside Buffer

**Diagram:**

- If TLB hit, no need to do page table lookup from memory.
- Cache permissions as well.
- Note: data cache accessed by physical addresses now.

**Legend:**
- Virtual Address
- Physical Address
- TLB
- Cache Data
- Cache Tags
- Hit?
Multi-Level Page Tables

Virtual Page Number

Level 1  Level 2  Page Offset

Physical Page Number
TLB Miss?

• Software solution
  – Generate an exception
  – O/S

• Hardware solution
  – Hardware page walker
  – TLB miss handler
  – Needs to know TLB miss in advance
PAPT Cache

• So far we haven’t differentiate physical and virtual addresses so much

• Previous slide showed Physically-Addressed Physically-Tagged cache
  – Sometimes called PIPT (I=Indexed)

• Con: TLB lookup and cache access serialized
  – Caches already take > 1 cycle

• Pro: cache contents valid so long as page table not modified
Virtually Addressed Cache

- Pro: latency – no need to check TLB
- Con: Cache must be flushed on process change
• Pro: latency – TLB parallelized
• Pro: don’t need to flush $ on process swap
• Con: Limit on cache indexing (can only use bits *not* from the VPN/PPN)
Virtual Index Physical Tag

Virtual Address

Virtual Page Number | Page Offset

Physical Address

Physical Frame Num | Page Offset

TAG | Index | B. offset

Good

TAG | Index | B. offset

BAD
• Programming: Virtual or Physical?
• Data sharing in parallel programming
  – Virtual or Physical?
  – Different VAs need to mapped to the same PA

  – Virtual-index-physical-tag Cache
  – VA1 = PA1 = \{tag1, index1, offset1\}
  – VA2 = PA1 = \{tag1, index2, offset1\}
A computer has an 8KB write-through cache. Each cache block is 64 bits, the cache is 4-way set associative and uses the true LRU replacement policy. Assume a 24-bit address space and byte-addressable memory. How big (in bits) is the tag store?
# of LRU bits

- **Assume true-LRU**
  - 4-way: 2 bits
  - 8-way: 3 bits
  - 2-way: 0.5 bit or 1 bit

- **Pseudo LRU**
  - Have fewer bits than true LRU
  - Less accurate but less complex (storage, logic)
A computer has an 8KB write-through cache. Each cache block is 64 bits, the cache is 4-way set associative and uses the true LRU replacement policy. Assume a 24-bit address space and byte-addressable memory. How big (in bits) is the tag store?
Review of DRAM

- Main characteristics
  - 1T vs. 6T
  - Destructive read
  - DRAM page
  - Sense amplifier
  - Burst mode
DRAM Page/Buffer

Row Decoder

Row Address

Memory Cell Array

Sense Amps

Row Buffer

Column Decoder

Column Address

Word lines

Data Bus

Georgia Tech College of Computing
Like Write-Combining Buffer, Scheduler may coalesce multiple accesses together, or re-order to reduce number of row accesses.
Task of Memory Controller

- Manage all data movement between the processor and the memory modules
- Read/Write
- Refresh/Precharge
- Memory request scheduling
• Scheduling memory requests in the dram system to increase the DRAM utilization

• Suggested Reading
• Access to a "closed row"
  • 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"
  • No need for activate command
**DRAM Read Latency**

- CPU → controller transfer time
- Controller latency
  - Queuing & scheduling delay at the controller
  - Access converted to basic commands
- Controller → DRAM transfer time
- DRAM bank latency
  - Simple CAS is row is “open” OR
  - RAS + CAS if array precharged OR
  - PRE + RAS + CAS (worst case)
- DRAM → CPU transfer time (through controller)
• Open Page: Keep page open after read
  – Pros:
    • Temporal, spatial locality
    • Latency is limited by tcas only
  – Cons:
    • Energy consumption, pay the cost of closing a page.
    • Page close+page open + ras + cas +bus transfer time

• Closed Page: page close after read
  – Good for random access patterns
  – Page open+ras+cas+bus transfer time
• DRAM scheduler: FCFS/ FRFCFS
• DRAM memory system organization
Width/Speed varies depending on memory type

Significant wire delay just getting from the CPU to the memory controller

More wire delay getting to the memory chips

(plus the return trip…)

x16 DRAM

x16 DRAM

x16 DRAM

x16 DRAM

x16 DRAM

x16 DRAM

x16 DRAM
**DRAM Scheduling Policies-I**

- **FCFS** (first come first served)
  - Oldest request first
- **FR-FCFS** (first ready, first come first served)
  1. Row-hit first
  2. Oldest first

Goal: Maximize row buffer hit rate $\Rightarrow$ **maximize DRAM throughput**

- Actually, scheduling is done at the command level
  - Column commands (read/write) prioritized over row commands (activate/precharge)
  - Within each group, older commands prioritized over younger ones
A scheduling policy is essentially a prioritization order.

Prioritization can be based on:
- Request age
- Row buffer hit/miss status
- Request type (prefetch, read, write)
- Requestor type (load miss or store miss)
- Request criticality
  - Oldest miss in the core?
  - How many instructions in core are dependent on it?
Why are DRAM Controllers Difficult to Design?

• Need to obey **DRAM timing constraints** for correctness
  – There are many (50+) timing constraints in DRAM
  – tWTR: Minimum number of cycles to wait before issuing a read command after a write command is issued
  – tRC: Minimum number of cycles between the issuing of two consecutive activate commands to the same bank
  – …

• Need to **keep track of many resources** to prevent conflicts
  – Channels, banks, ranks, data bus, address bus, row buffers

• Need to handle **DRAM refresh**

• Need to optimize for **performance** (in the presence of constraints)
  – Reordering is not simple
  – Predicting the future?

www.ece.cmu.edu/~omutlu/ece741
Example Memory Latency Computation

- FSB freq = 200 MHz, SDRAM
- RAS delay = 2, CAS delay = 2, Precharge = 2
- Scheduling in memory controller
- Scheduler queue size = 6
  - A0, A1, B0, C0, D3, A2, D0, C1, A3, C3, C2, D1, B1, D2
- FCFS time?
- FRFCFS time?
  - A0, A1, A2, B0, C0, C1, C3, C2, D3, D0, D1, D2, A3, B1
• Bank, row, column → DRAM chip configuration
  – Banks: different banks can be operated independently
• Rank → a set of DRAM devices that operate in lockstep fashion to command in a memory (i.e. chips inside the same rank are accessed simultaneously)
• Channel → CPU and memory communication channel

Figure 3.5: Memory System with 2 ranks of DRAM devices.

Wang’s dissertation ‘05
From Prof. Hsien-sin Lee’s slides
**DRAM BANK: 512MB 4-bank DRAM**

- **BA[1:0]**
- **A[13:0]**
- **A[26:0]**
- **Address demultiplexer**
- **Row Decoder**
- **Memory Cell Array**
- **Sense Amps**
- **Row Buffer**
- **Column Decoder**
- **2k**
- **16k**
- **DRAM Page**

A DRAM Page = 2k*4B = 8KB

32 bits

From Prof. Hsien-sin Lee’s slides
Figure 3.6 shows an SDRAM device with 4 banks. Modern DRAM devices contain...
## Bank & Interleaving

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bank id</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>

- Interleaving: why?
Column Size

DRAM devices arranged in parallel in a given rank

SDRAM memory systems: width of data bus = column size

Figure 3.8: Classical DRAM system topology, width of data bus equals column size.
• One physical channel of 64 bit width

• Two physical channel of 64 bit wide busses

• One logical channel

• Two channels: 64 bit wide per channel
Mesh Topology

Figure 3.16: Topology of a generic DRAM memory system.

Figure 3.17: Topology of a generic Direct RDRAM memory system.
**FB-DIMM**

- AMB (advanced memory buffer)
- Each DIMM has their own DIMM memory controller
- Increase bandwidth
- ~ DDR2
<table>
<thead>
<tr>
<th>Memory Type</th>
<th>Features</th>
<th>Specifications</th>
</tr>
</thead>
</table>
| Patriot Gamer 2 Series 8GB (2 x 4GB)    | 240-Pin DDR3 SDRAM DDR3 1333 (PC3 10666) Desktop Memory                   | • DDR3 1333 (PC3 10666)  
• Timing 9-9-9-24  
• Voltage 1.65V                                                                 |
| Corsair Vengeance 8GB (2 x 4GB)         | 240-Pin DDR3 SDRAM DDR3 1866 (PC3 15000) Desktop Memory Model             | • DDR3 1866 (PC3 15000)  
• Timing 9-10-9-27  
• Cas Latency 9                                                                 |
| Crucial Ballistix Sport 4GB (2 x 2GB)   | 240-Pin DDR3 SDRAM DDR3 1600 Desktop Memory                               | • DDR3 1600  
• Timing 10-10-10-28  
• Cas Latency 10                                                                 |
| G.Skill Ripjaws Series 8GB (2 x 4GB)     | 240-Pin DDR3 SDRAM DDR3 1600 (PC3 12800) Desktop Memory                   | • DDR3 1600 (PC3 12800)  
• Timing 9-9-9-24-2N  
• Cas Latency 9                                                                 |
| **Free 4GB SDHC flash card w/ purchase** | **Limited Offer**                                                        |                                                                                       |
CL (CAS): ck cycles between sending a column address to the memory and the beginning of the data in response

tRCD: Clock cycles between RAS to CAS delay

TRP: Clock cycles between row precharge and activate (PRE)

TRC: from RAS to read&write

<table>
<thead>
<tr>
<th>Standard name</th>
<th>Memory clock (MHz)</th>
<th>Cycle time (ns)</th>
<th>I/O bus clock (MHz)</th>
<th>Data rate (MT/s)</th>
<th>Peak transfer rate (MB/s)</th>
<th>Timing (CL-tRCD-tRP)</th>
<th>CAS latency (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR3-1333</td>
<td>166.66</td>
<td>6</td>
<td>666.66</td>
<td>1333.33</td>
<td>10666.66</td>
<td>7-7-7-7</td>
<td>10.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8-8-8-8 ...</td>
<td>12 ...</td>
</tr>
</tbody>
</table>

DDr3-M transfer second

I/O frequency = ½ M transfer frequency

DIMM name = M transfer second *2 (dual) * 8B
e.g.) DDR3-1600 = PC12800 = 1600*2*8
# Table 5: Pin Assignments

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VREFDQ</td>
<td>31</td>
<td>DQ25</td>
<td>61</td>
<td>A2</td>
<td>91</td>
<td>DQ41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VSS</td>
<td>32</td>
<td>VSS</td>
<td>62</td>
<td>VDD</td>
<td>92</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DQ0</td>
<td>33</td>
<td>DQ3#</td>
<td>63</td>
<td>CK1</td>
<td>93</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DQ1</td>
<td>34</td>
<td>DQ3#</td>
<td>64</td>
<td>CK1#</td>
<td>94</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>VSS</td>
<td>35</td>
<td>VSS</td>
<td>65</td>
<td>VDD</td>
<td>95</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DQS0#</td>
<td>36</td>
<td>DQ26</td>
<td>66</td>
<td>VDD</td>
<td>96</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DQS0</td>
<td>37</td>
<td>DQ27</td>
<td>67</td>
<td>VREFCA</td>
<td>97</td>
<td>DQ43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>VSS</td>
<td>38</td>
<td>VSS</td>
<td>68</td>
<td>NC</td>
<td>98</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DQ2</td>
<td>39</td>
<td>NC</td>
<td>69</td>
<td>VDD</td>
<td>99</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DQ3</td>
<td>40</td>
<td>NC</td>
<td>70</td>
<td>A10</td>
<td>100</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>VSS</td>
<td>41</td>
<td>VSS</td>
<td>71</td>
<td>BA0</td>
<td>101</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DQ8</td>
<td>42</td>
<td>NC</td>
<td>72</td>
<td>VDD</td>
<td>102</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DQ9</td>
<td>43</td>
<td>NC</td>
<td>73</td>
<td>WE#</td>
<td>103</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>VSS</td>
<td>44</td>
<td>VSS</td>
<td>74</td>
<td>CAS#</td>
<td>104</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>DQS1#</td>
<td>45</td>
<td>NC</td>
<td>75</td>
<td>VDD</td>
<td>105</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>DQS1</td>
<td>46</td>
<td>NC</td>
<td>76</td>
<td>NC</td>
<td>106</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>VSS</td>
<td>47</td>
<td>VSS</td>
<td>77</td>
<td>NC</td>
<td>107</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>DQ10</td>
<td>48</td>
<td>NC</td>
<td>78</td>
<td>VDD</td>
<td>108</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>DQ11</td>
<td>49</td>
<td>NC</td>
<td>79</td>
<td>NC</td>
<td>109</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>VSS</td>
<td>50</td>
<td>CKE0</td>
<td>80</td>
<td>VSS</td>
<td>110</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>DQ16</td>
<td>51</td>
<td>VDD</td>
<td>81</td>
<td>DQS2</td>
<td>112</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>DQ17</td>
<td>52</td>
<td>BA2</td>
<td>82</td>
<td>DQS3</td>
<td>113</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>VSS</td>
<td>53</td>
<td>NC</td>
<td>83</td>
<td>VSS</td>
<td>113</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>DQS2#</td>
<td>54</td>
<td>VDD</td>
<td>84</td>
<td>DQS4#</td>
<td>114</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>DQS2</td>
<td>55</td>
<td>A11</td>
<td>85</td>
<td>DQS4</td>
<td>115</td>
<td>DQS#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>VSS</td>
<td>56</td>
<td>A7</td>
<td>86</td>
<td>VSS</td>
<td>116</td>
<td>VSS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>DQ18</td>
<td>57</td>
<td>VDD</td>
<td>87</td>
<td>DQS4</td>
<td>117</td>
<td>SA0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>DQ19</td>
<td>58</td>
<td>A5</td>
<td>88</td>
<td>DQS4</td>
<td>118</td>
<td>SCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>VSS</td>
<td>59</td>
<td>A4</td>
<td>89</td>
<td>VSS</td>
<td>119</td>
<td>SA2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>DQ24</td>
<td>60</td>
<td>VDD</td>
<td>90</td>
<td>DQ40</td>
<td>120</td>
<td>VTT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes: 1. Pin 172 is NC for 1GB and A14 for 2GB.
• L3 will be posted by tonight.
• Cache & DRAM (DRAM page) & MSHR
• Due (10/20)

• Exam & Lab 2 grade: will be posted by tonight.
• You can pick up your exam paper
  – Friday 4-5 pm (or send email)