Mainstream Computer System Components

CPU Core
1 GHz - 3.8 GHz
4-way Superscaler
RISC or RISC-core (x86):
  Deep Instruction Pipelines
  Dynamic scheduling
  Multiple FP, integer FUs
  Dynamic branch prediction
  Hardware speculation

SDRAM
PC100/PC133
100-133MHz
64-128 bits wide
2-way interleaved
~ 900 MBYTES/SEC 64bit)

Current Standard
Double Date Rate (DDR) SDRAM
PC3200
200 MHz DDR
64-128 bits wide
4-way interleaved
~3.2 GBYTES/SEC
(one 64bit channel)
~6.4 GBYTES/SEC
(two 64bit channels)

RAMbus DRAM (RDRAM)
400 MHz DDR
16 bits wide (32 banks)
~ 1.6 GBYTES/SEC

System Bus = CPU-Memory Bus = Front Side Bus (FSB)

CPU

Caches

L1

L2

L3

System Bus

Off or On-chip

Memory

Memory Controller

Controllers

NICS

Disks
Displays
Keyboards

I/O Buses

I/O Devices:

Example: PCI, 33-66MHz
32-64 bits wide
133-528 MBYTES/SEC
PCI-X 133MHz 64 bit
1024 MBYTES/SEC

Networks

North Bridge

South Bridge

Chipset

I/O Subsystem (In Chapter 7)
The Memory Hierarchy

- Review of Memory Hierarchy & Cache Basics (from 550)
  - Cache Basics:
  - CPU Performance Evaluation with Cache

- Classification of Steady-State Cache Misses:
  - *The Three C’s of cache Misses*

- Cache Write Policies/Performance Evaluation

- Cache Write Miss Policies

- Multi-Level Caches & Performance

- Main Memory:
  - Performance Metrics: Latency & Bandwidth
    - Key DRAM Timing Parameters
  - DRAM System Memory Generations
  - Basic Memory Bandwidth Improvement/Miss Penalty Reduction Techniques

- Techniques To Improve Cache Performance:
  - Reduce Miss Rate
  - Reduce Cache Miss Penalty
  - Reduce Cache Hit Time

- Virtual Memory
  - Benefits, Issues/Strategies
  - Basic Virtual ➔ Physical Address Translation: Page Tables
  - Speeding Up Address Translation: Translation Lookaside Buffer (TLB)

Cache exploits access locality to:
- Lower AMAT by hiding long main memory access latency.
- Lower demands on main memory bandwidth.

(In Chapter 5.8 - 5.10)
A Typical Memory Hierarchy

Speed (ns):  
- Processor: < 1s
- Control: 1s
- Second Level Cache (SRAM) L2: 10s
- Main Memory (DRAM): 10,000,000s (10s ms)
- Virtual Memory, Secondary Storage (Disk): 10,000,000,000s (10s sec)
- Tertiary Storage (Tape): 10,000,000,000s (10s sec)

Size (bytes):  
- Processor: 100s
- Control: Ks
- Second Level Cache (SRAM) L2: Ms
- Main Memory (DRAM): Gs
- Virtual Memory, Secondary Storage (Disk): Ts
- Tertiary Storage (Tape): Ts

Faster
Larger Capacity

Processor
Control
Datapath
Registers
Level One Cache L1
Second Level Cache (SRAM) L2
Main Memory (DRAM)
Virtual Memory, Secondary Storage (Disk)
Tertiary Storage (Tape)
Main Memory

- Main memory generally utilizes Dynamic RAM (DRAM), which use a single transistor to store a bit, but require a periodic data refresh by reading every row increasing cycle time.
- Static RAM may be used for main memory if the added expense, low density, high power consumption, and complexity is feasible (e.g. Cray Vector Supercomputers).
- Main memory performance is affected by:
  - **Memory latency**: Affects cache miss penalty, M. Measured by:
    - **Memory Access time**: The time it takes between a memory access request is issued to main memory and the time the requested information is available to cache/CPU.
    - **Memory Cycle time**: The minimum time between requests to memory (greater than access time in DRAM to allow address lines to be stable)
  - **Peak Memory bandwidth**: The maximum sustained data transfer rate between main memory and cache/CPU.
    - In current memory technologies (e.g Double Data Rate SDRAM) published peak memory bandwidth does not take account most of the memory access latency.
    - This leads to achievable realistic memory bandwidth < peak memory bandwidth
Logical Dynamic RAM (DRAM) Chip Organization (16 Mbit)

Basic Steps:
- Control Signals:
  1 - Row Access Strobe (RAS): Low to latch row address
  2- Column Address Strobe (CAS): Low to latch column address
  3- Write Enable (WE) or Output Enable (OE)
  4- Wait for data to be ready

D, Q share the same pins (Single transistor per bit)

A periodic data refresh is required by reading every bit

EECC551 - Shaaban

#5 Lec # 10 Winter 2005 2-8-2006
Four Key DRAM Timing Parameters

- $t_{RAC}$: Minimum time from RAS (Row Access Strobe) line falling (activated) to the valid data output.
  - Used to be quoted as the nominal speed of a DRAM chip
  - For a typical 64Mb DRAM $t_{RAC} = 60$ ns

- $t_{RC}$: Minimum time from the start of one row access to the start of the next (memory cycle time).
  - $t_{RC} = t_{RAC} +$ RAS Precharge Time
  - $t_{RC} = 110$ ns for a 64Mbit DRAM with a $t_{RAC}$ of 60 ns

- $t_{CAC}$: Minimum time from CAS (Column Access Strobe) line falling to valid data output.
  - 12 ns for a 64Mbit DRAM with a $t_{RAC}$ of 60 ns

- $t_{PC}$: Minimum time from the start of one column access to the start of the next.
  - $t_{PC} = t_{CAC} +$ CAS Precharge Time
  - About 25 ns for a 64Mbit DRAM with a $t_{RAC}$ of 60 ns
Simplified DRAM Speed Parameters

- **Row Access Strobe (RAS) Time:** (similar to $t_{RAC}$):
  - Minimum time from RAS (Row Access Strobe) line falling (activated) to the first valid data output.
  - A major component of memory latency.
  - Only improves ~ 5% every year.

- **Column Access Strobe (CAS) Time/data transfer time:** (similar to $t_{CAC}$)
  - The minimum time required to read additional data by changing column address while keeping the same row address.
  - Along with memory bus width, determines peak memory bandwidth.

  - E.g. For SDRAM, Peak Memory Bandwidth = Bus Width / (0.5 x $t_{CAC}$)
    - For PC100 SDRAM Memory bus width = 8 bytes $t_{CAC} = 20$ns
    - Peak Bandwidth = $8 \times 100 \times 10^6 = 800 \times 10^6$ bytes/sec
## DRAM Generations

<table>
<thead>
<tr>
<th>Year</th>
<th>Size</th>
<th>RAS (ns)</th>
<th>CAS (ns)</th>
<th>Cycle Time</th>
<th>Memory Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>64 Kb</td>
<td>150-180</td>
<td>75</td>
<td>250 ns</td>
<td>Page Mode</td>
</tr>
<tr>
<td>1983</td>
<td>256 Kb</td>
<td>120-150</td>
<td>50</td>
<td>220 ns</td>
<td>Page Mode</td>
</tr>
<tr>
<td>1986</td>
<td>1 Mb</td>
<td>100-120</td>
<td>25</td>
<td>190 ns</td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>4 Mb</td>
<td>80-100</td>
<td>20</td>
<td>165 ns</td>
<td>Fast Page Mode</td>
</tr>
<tr>
<td>1992</td>
<td>16 Mb</td>
<td>60-80</td>
<td>15</td>
<td>120 ns</td>
<td>EDO</td>
</tr>
<tr>
<td>1996</td>
<td>64 Mb</td>
<td>50-70</td>
<td>12</td>
<td>110 ns</td>
<td>PC66 SDRAM</td>
</tr>
<tr>
<td>1998</td>
<td>128 Mb</td>
<td>50-70</td>
<td>10</td>
<td>100 ns</td>
<td>PC100 SDRAM</td>
</tr>
<tr>
<td>2000</td>
<td>256 Mb</td>
<td>45-65</td>
<td>7</td>
<td>90 ns</td>
<td>PC133 SDRAM</td>
</tr>
<tr>
<td>2002</td>
<td>512 Mb</td>
<td>40-60</td>
<td>5</td>
<td>80 ns</td>
<td>PC2700 DDR SDRAM</td>
</tr>
</tbody>
</table>

- **Capacity**: $8000:1$
- **Bandwidth**: $15:1$
- **Latency**: $3:1$

---

**Asynchronous DRAM**

- PC3200 DDR (2003)
- DDR2 SDRAM (2004)

---

A major factor in cache miss penalty $M$
Simplified Asynchronous DRAM Read Timing

Memory Cycle Time = \( t_{RC} = t_{RAC} + \text{RAS Precharge Time} \)

\( t_{RAC} \): Minimum time from RAS (Row Access Strobe) line falling to the valid data output.
\( t_{RC} \): Minimum time from the start of one row access to the start of the next (memory cycle time).
\( t_{CAC} \): Minimum time from CAS (Column Access Strobe) line falling to valid data output.
\( t_{PC} \): Minimum time from the start of one column access to the start of the next.

Peak Memory Bandwidth = Memory bus width / Memory cycle time

Example: Memory Bus Width = 8 Bytes  Memory Cycle time = 200 ns  
Peak Memory Bandwidth = \( 8 / 200 \times 10^{-9} = 40 \times 10^6 \) Bytes/sec

Source: http://arstechnica.com/paedia/r/ram_guide/ram_guide.part2-1.html
Page Mode DRAM (Early 80s)

- Regular DRAM Organization:
  - N rows x N column x M-bit
  - Read & Write M-bit at a time
  - Each M-bit access requires a RAS / CAS cycle

Asynchronous DRAM:

---

Memory Cycle Time

1st M-bit Access

2nd M-bit Access

RAS_L

CAS_L

A Row Address Col Address Junk Row Address Col Address Junk

M-bit Output

N rows

N cols

N M bits
Fast Page Mode DRAM (late 80s)

- Fast Page Mode DRAM
  - $N \times M$ “SRAM” to save a row

- After a row is read into the register
  - Only CAS is needed to access other $M$-bit blocks on that row
  - RAS_L remains asserted while CAS_L is toggled

- The first “burst mode” DRAM

A read burst of length 4 shown

Burst Mode Memory Access
Simplified Asynchronous Fast Page Mode (FPM) DRAM Read Timing

Fast Page Mode Read

FPM DRAM speed rated using tRAC ~ 50-70ns

Typical timing at 66 MHz: 5-3-3-3 (burst of length 4)
For bus width = 64 bits = 8 bytes cache block size = 32 bytes
It takes = 5+3+3+3 = 14 memory cycles or 15 ns x 14 = 210 ns to read 32 byte block
Miss penalty for CPU running at 1 GHz = M = 15 x 14 = 210 CPU cycles

One memory cycle at 66 MHz = 1000/66 = 15 CPU cycles at 1 GHz
Extended Data Out DRAM operates in a similar fashion to Fast Page Mode DRAM except putting data from one read on the output pins at the same time the column address for the next read is being latched in.

**Typical timing at 66 MHz:** 5-2-2-2 (burst of length 4)

For bus width = 64 bits = 8 bytes  
Max. Bandwidth = 8 x 66 / 2 = 264 Mbytes/sec

It takes = $5+2+2+2 = 11$ memory cycles or $15 \text{ ns} \times 11 = 165 \text{ ns}$ to read 32 byte cache block

Minimum Read Miss penalty for CPU running at 1 GHz = $M = 11 \times 15 = 165$ CPU cycles

One memory cycle at 66 MHz = $1000/66 = 15$ CPU cycles at 1 GHz

Basic Memory Bandwidth Improvement/Miss Penalty (M) Reduction Techniques

• **Wider Main Memory (CPU-Memory Bus):**
  
  Memory bus width is increased to a number of words (usually up to the size of a cache block).
  
  – Memory bandwidth is proportional to memory bus width.
    
    • e.g. Doubling the width of cache and memory doubles potential memory bandwidth available to the CPU.
    
  – The miss penalty is reduced since fewer memory bus accesses are needed to fill a cache block on a miss.

• **Interleaved (Multi-Bank) Memory:**
  
  Memory is organized as a number of independent banks.
  
  – Multiple interleaved memory reads or writes are accomplished by sending memory addresses to several memory banks at once or pipeline access to the banks.
  
  – **Interleaving factor:** Refers to the mapping of memory addresses to memory banks. **Goal reduce bank conflicts.**

  e.g. using 4 banks (width one word), bank 0 has all words whose address is:
  
  \[(\text{word address mod} \ 4) = 0\]
Three examples of bus width, memory width, and memory interleaving to achieve higher memory bandwidth

Simplest design:
Everything is the width of one word (lowest performance)

Wider memory, bus and cache (highest performance)

Narrow bus and cache with interleaved memory banks

Front Side Bus (FSB) = System Bus = CPU-memory Bus

EECC551 - Shaaban
Four Way (Four Banks) Interleaved Memory

Memory Bank Number

<table>
<thead>
<tr>
<th>Bank 0</th>
<th>Bank 1</th>
<th>Bank 2</th>
<th>Bank 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
</tbody>
</table>

Bank Width = One Word
Bank Number = (Word Address) Mod (4)
Memory Bank Interleaving

Can be applied at: 1- DRAM chip level (e.g. SDRAM, DDR) 2- DRAM module level 3- DRAM channel level

Access Pattern without Interleaving: (One Memory Bank)

Memory Bank Cycle Time

D1 available
Start Access for D1

Start Access for D2

(4 banks similar to the organization of DDR SDRAM memory chips)

Pipeline access to different memory banks to increase effective bandwidth

Access Pattern with 4-way Interleaving:

We can Access Bank 0 again

Number of banks ≥ Number of cycles to access word in a bank

Bank interleaving does not reduce latency of accesses to the same bank
## Synchronous DRAM Characteristics Summary

<table>
<thead>
<tr>
<th></th>
<th>SDRAM</th>
<th>DDR (Double Data Rate) SDRAM</th>
<th>RAMbus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PC100</td>
<td>DDR266 (PC2100)</td>
<td></td>
</tr>
<tr>
<td>Potential Bandwidth</td>
<td>0.8 GB/s</td>
<td>2.133 GB/s</td>
<td>1.6 GB/s</td>
</tr>
<tr>
<td></td>
<td>.1 x 8 = .8</td>
<td>.133 x 2 x 8 = 2.1</td>
<td>.4 x 2 x 2 = 1.6</td>
</tr>
<tr>
<td>Interface Signals</td>
<td>64(72) data</td>
<td>64(72) data</td>
<td>16(18) data</td>
</tr>
<tr>
<td></td>
<td>168 pins</td>
<td>168 pins</td>
<td>184 pins</td>
</tr>
<tr>
<td>Interface Frequency</td>
<td>100 MHz</td>
<td>133 MHz</td>
<td>400MHz</td>
</tr>
<tr>
<td>Latency Range</td>
<td>30-90 nS</td>
<td>18.8-64 nS</td>
<td>35-80 nS</td>
</tr>
<tr>
<td># of Banks per DRAM Chip</td>
<td>2</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>Bus Width Bytes</td>
<td>8</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

The latencies given only account for memory module latency and do not include memory controller latency or other address/data line delays. Thus realistic access latency is longer.
Synchronous Dynamic RAM, (SDRAM) (mid 90s)

Organization

SDRAM speed is rated at max. clock speed supported:
100MHZ = PC100
133MHZ = PC133

DDR SDRAM (late 90s - current)

organization is similar but four banks are used in each DDR SDRAM chip instead of two.

Data transfer on both rising and falling edges of the clock

DDR SDRAM rated by maximum or peak memory bandwidth
PC3200 = 8 bytes x 200 MHz x 2
= 3200 Mbytes/sec

SDRAM Peak Memory Bandwidth =
= Bus Width / (0.5 x tCAC)
= Bus Width x Clock rate

DDR SDRAM Peak Memory Bandwidth =
= Bus Width / (0.25 x tCAC)
= Bus Width x Clock rate x 2

SDRAM Peak Memory Bandwidth =
= Bus Width / (0.5 x tCAC)
= Bus Width x Clock rate

Data transfer on both rising and falling edges of the clock

DDR SDRAM rated by maximum or peak memory bandwidth
PC3200 = 8 bytes x 200 MHz x 2
= 3200 Mbytes/sec

EECC551 - Shaaban

#19 Lec # 10 Winter 2005 2-8-2006
**Simplified SDRAM/DDDR SDRAM Read Timing**

**SDRAM**

Typical timing at 133 MHz (PC133 SDRAM): 5-1-1-1

For bus width = 64 bits = 8 bytes

Max. Bandwidth = 133 x 8 = 1064 Mbytes/sec

It takes = 5+1+1+1 = 8 memory cycles or 7.5 ns x 8 = 60 ns to read 32 byte cache block

Minimum Read Miss penalty for CPU running at 1 GHz = 

\[ M = 7.5 \times 8 = 60 \] CPU cycles

**DDR SDRAM**

Possible timing at 133 MHz (DDR x2) (PC2100 DDR SDRAM): 5- .5-.5-.5

For bus width = 64 bits = 8 bytes

Max. Bandwidth = 133 x 2 x 8 = 2128 Mbytes/sec

It takes = 5+.5+.5+.5 = 6.5 memory cycles or 7.5 ns x 8 = 45 ns to read 32 byte cache block

Minimum Read Miss penalty for CPU running at 1 GHz = 

\[ M = 7.5 \times 6 = 49 \] CPU cycles

In this example for SDRAM: \( M = 60 \) cycles for DDR SDRAM: \( M = 49 \) cycles

Thus accounting for access latency DDR is \( 60/49 = 1.22 \) times faster

Not twice as fast \((2128/1064 = 2)\) as indicated by peak bandwidth!
The Impact of Larger Cache Block Size on Miss Rate

- A larger cache block size improves cache performance by taking better advantage of spatial locality. However, for a fixed cache size, larger block sizes mean fewer cache block frames.

- Performance keeps improving to a limit when the fewer number of cache block frames increases conflicts and thus overall cache miss rate.

For SPEC92

Improves spatial locality reducing compulsory misses

Block Size (bytes)
Memory Width, Interleaving: Performance Example

Given the following system parameters with single unified cache level L1 (ignoring write policy):

Block size = 1 word  Memory bus width = 1 word  Miss rate = 3%  M = Miss penalty = 32 cycles
(4 cycles to send address  24 cycles access time,  4 cycles to send a word to CPU)

<table>
<thead>
<tr>
<th>Block Size</th>
<th>CPI Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 word</td>
<td>2 + (1.2 x 0.03 x 32) = 3.15</td>
</tr>
<tr>
<td>2 words</td>
<td>2 + (1.2 x 0.02 x 64) = 3.54</td>
</tr>
<tr>
<td>4 words</td>
<td>2 + (1.2 x 0.01 x 128) = 3.54</td>
</tr>
</tbody>
</table>

Memory access/instruction = 1.2  \( \text{CPI}_{\text{execution}} \) (ignoring cache misses) = 2

Miss rate (block size = 2 word = 8 bytes) = 2%  Miss rate (block size = 4 words = 16 bytes) = 1%

- The CPI of the base machine with 1-word blocks = 2 + (1.2 x 0.03 x 32) = 3.15  (For Base system)

Increasing the block size to two words (64 bits) gives the following CPI:  (miss rate = 2%)

- 32-bit bus and memory, no interleaving,  \( M = 2 \times 32 = 64 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.02 \times 64) = 3.54 \)
- 32-bit bus and memory, interleaved,  \( M = 4 + 24 + 8 = 36 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.02 \times 36) = 2.86 \)
- 64-bit bus and memory, no interleaving,  \( M = 32 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.02 \times 32) = 2.77 \)

Increasing the block size to four words (128 bits); resulting CPI:  (miss rate = 1%)

- 32-bit bus and memory, no interleaving,  \( M = 4 \times 32 = 128 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.01 \times 128) = 3.54 \)
- 32-bit bus and memory, interleaved,  \( M = 4 + 24 + 16 = 44 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.01 \times 44) = 2.53 \)
- 64-bit bus and memory, no interleaving,  \( M = 2 \times 32 = 64 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.01 \times 64) = 2.77 \)
- 64-bit bus and memory, interleaved,  \( M = 4 + 24 + 8 = 36 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.01 \times 36) = 2.43 \)
- 128-bit bus and memory, no interleaving,  \( M = 32 \text{ cycles} \)  \( \text{CPI} = 2 + (1.2 \times 0.01 \times 32) = 2.38 \)

Miss Penalty = \( M = \) Number of CPU stall cycles for an access missed in cache and satisfied by main memory
Three-Level Cache Example

- CPU with $\text{CPI}_{\text{execution}} = 1.1$ running at clock rate = 500 MHz
- 1.3 memory accesses per instruction.
- $L_1$ cache operates at 500 MHz (no stalls on a hit in $L_1$) with a miss rate of 5%
- $L_2$ hit access time = 3 cycles ($T_2=2$ stall cycles per hit), local miss rate 40%
- $L_3$ hit access time = 6 cycles ($T_3=5$ stall cycles per hit), local miss rate 50%,
- Memory access penalty, $M=100$ cycles (stall cycles per access). Find CPI.

With No Cache, $\text{CPI} = 1.1 + 1.3 \times 100 = 131.1$

With single $L_1$, $\text{CPI} = 1.1 + 1.3 \times 0.05 \times 100 = 7.6$

With $L_1$, $L_2$ $\text{CPI} = 1.1 + 1.3 \times (0.05 \times 0.6 \times 2 + 0.05 \times 0.4 \times 100) = 3.778$

$\text{CPI} = \text{CPI}_{\text{execution}} + \text{Mem Stall cycles per instruction}$

Mem Stall cycles per instruction = Mem accesses per instruction $\times$ Stall cycles per access

Stall cycles per memory access = $(1-H_1) \times H_2 \times T_2 + (1-H_1) \times (1-H_2) \times H_3 \times T_3 + (1-H_1)(1-H_2)(1-H_3) \times M$

$= 0.05 \times 0.6 \times 2 + 0.05 \times 0.4 \times 0.5 \times 5 + 0.05 \times 0.4 \times 0.5 \times 100$

$= 0.06 + 0.05 + 1 = 1.11$

AMAT = 1.11 + 1 = 2.11 cycles (vs. AMAT = 3.06 with $L_1$, $L_2$, vs. 5 with $L_1$ only)

$\text{CPI} = 1.1 + 1.3 \times 1.11 = 2.54$

Speedup compared to $L_1$ only $= 7.6/2.54 = 3$

Speedup compared to $L_1$, $L_2$ $= 3.778/2.54 = 1.49$
3-Level (All Unified) Cache Performance

Memory Access Tree (Ignoring Write Policy)

CPU Stall Cycles Per Memory Access

**CPU Memory Access** (100%)

- **L1 Hit:**
  - Hit Access Time = 1
  - Stalls Per access = 0
  - Stalls = H1 x 0 = 0
  - (No Stall)

- **L1 Miss:**
  - % = (1-H1) = .05 or 5%

- **L1 Miss, L2 Hit:**
  - Hit Access Time = T2 + 1 = 3
  - Stalls per L2 Hit = T2 = 2
  - Stalls = (1-H1) x H2 x T2 = .05 x .6 x 2 = .06

- **L1 Miss, L2 Miss:**
  - % = (1-H1)(1-H2) = .05 x .4 = .02 or 2%

- **L1 Miss, L2 Miss, L3 Hit:**
  - Hit Access Time = T3 + 1 = 6
  - Stalls per L2 Hit = T3 = 5
  - Stalls = (1-H1) x (1-H2) x H3 x T3 = .01 x 5 = .05 cycles

- **L1 Miss, L2 Miss, L3 Miss:**
  - Miss Penalty = M = 100
  - Stalls = (1-H1)(1-H2)(1-H3)x M = .01 x 100 = 1 cycle

**AMAT**

- 1 + Stall cycles per memory access

**T2 = 2 cycles = Stalls per hit access for Level 2**

**T3 = 5 cycles = Stalls per hit access for Level 3**

**M = Memory Miss Penalty = M = 100 cycles**

**CPI Calculation**

- CPI = CPI\text{execution} + (1 + \text{fraction of loads and stores}) \times \text{stalls per access}

- CPI = 1.1 + 1.3 \times 1.11 = 2.54

**Example Calculation**

- H1 = 95%  
  - T1 = 0 cycles
- H2 = 60%  
  - T2 = 2 cycles
- H3 = 50%  
  - T3 = 5 cycles
- M = 100 cycles

Stalls on a hit

Repeated here from lecture 8 with values for example added
Program Steady-State Bandwidth-Usage Example

- In the previous example with three levels of cache (all unified, ignore write policy)
- CPU with $CPI_{\text{execution}} = 1.1$ running at clock rate = 500 MHz
- 1.3 memory accesses per instruction.
- $L_1$ cache operates at 500 MHz (no stalls on a hit in $L_1$) with a miss rate of 5%
- $L_2$ hit access time = 3 cycles ($T_2=2$ stall cycles per hit), local miss rate 40%
- $L_3$ hit access time = 6 cycles ($T_3=5$ stall cycles per hit), local miss rate 50%
- Memory access penalty, $M=100$ cycles (stall cycles per access to deliver 32 bytes from main memory to CPU)

We found the CPI:
- With No Cache, $CPI = 1.1 + 1.3 \times 100 = 131.1$
- With single $L_1$, $CPI = 1.1 + 1.3 \times 0.05 \times 100 = 7.6$
- With $L_1$, $L_2$ $CPI = 1.1 + 1.3 \times (0.05 \times 0.6 \times 2 + 0.05 \times 0.4 \times 100) = 3.778$
- With $L_1$, $L_2$, $L_3$ $CPI = 1.1 + 1.3 \times 1.11 = 2.54$

Assuming:
- All cache blocks are 32 bytes

For each of the three cases with cache:
A. What is the peak (or maximum) number of memory accesses and effective peak bandwidth for each cache level and main memory?
B. What is the total number of memory accesses generated by the CPU per second?
C. What is the percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?
Program Steady-State Bandwidth-Usage Example

A. What is the peak (or maximum) number of memory accesses and effective peak bandwidth for each cache level and main memory?

- L1 cache requires 1 CPU cycle to deliver 32 bytes, thus:
  Maximum L1 accesses per second = \(500 \times 10^6\) accesses/second
  Maximum effective L1 bandwidth = \(32 \times 500 \times 10^6 = 16,000 \times 10^6 = 16 \times 10^9\) byes/sec

- L2 cache requires 3 CPU cycles to deliver 32 bytes, thus:
  Maximum L2 accesses per second = \(500/3 \times 10^6 = 166.67 \times 10^6\) accesses/second
  Maximum effective L2 bandwidth = \(32 \times 166.67 \times 10^6 = 5,333.33 \times 10^6 = 5.33 \times 10^9\) byes/sec

- L3 cache requires 6 CPU cycles to deliver 32 bytes, thus:
  Maximum L3 accesses per second = \(500/6 \times 10^6 = 83.33 \times 10^6\) accesses/second
  Maximum effective L3 bandwidth = \(32 \times 166.67 \times 10^6 = 2,666.67 \times 10^6 = 2.67 \times 10^9\) byes/sec

- Memory requires 101 CPU cycles (\(101 = M+1 = 100+1\)) to deliver 32 bytes, thus:
  Maximum main memory accesses per second = \(500/101 \times 10^6 = 4.95 \times 10^6\) accesses/second
  Maximum effective main memory bandwidth = \(32 \times 4.95 \times 10^6 = 158.42 \times 10^6\) byes/sec
Program Steady-State Bandwidth-Usage Example

• For CPU with L1 Cache:

B. What is the total number of memory accesses generated by the CPU per second?

• The total number of memory accesses generated by the CPU per second = (memory access/instruction) x clock rate / CPI = 1.3 x 500 x 10^6 / CPI = 650 x 10^6 / CPI

• With single L1 cache CPI was found = 7.6
– CPU memory accesses = 650 x 10^6 / 7.6 = 85 x 10^6 accesses/sec

C. What is the percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

• For L1:
  The percentage of CPU memory accesses that reach L1 = 100%
  L1 Cache bandwidth usage = 32 x 85 x 10^6 = 2,720 x 10^6 = 2.7 x 10^9 byes/sec
  Percentage of L1 bandwidth used = 2,720 / 16,000 = 0.17 or 17%
  (or by just dividing CPU accesses / peak L1 accesses = 85/500 = 0.17 = 17%)

• For Main Memory:
  The percentage of CPU memory accesses that reach main memory = (1-H1) = 0.05 or 5%
  Main memory bandwidth usage = 0.05 x 32 x 85 x 10^6 = 136 x 10^6 byes/sec
  Percentage of main memory bandwidth used = 136 / 158.42 = 0.8585 or 85.85%
For CPU with L1, L2 Cache:

B. What is the total number of memory accesses generated by the CPU per second?

The total number of memory accesses generated by the CPU per second =
(memory access/instruction) \times \text{clock rate} / \text{CPI} = 1.3 \times 500 \times 10^6 / \text{CPI} = 650 \times 10^6 / \text{CPI}

With L1, L2 cache CPI was found = 3.778

\[ \text{CPU memory accesses} = 650 \times 10^6 / 3.778 = 172 \times 10^6 \text{ accesses/sec} \]

C. What is the percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

For L1:
The percentage of CPU memory accesses that reach L1 = 100%
L1 Cache bandwidth usage = 32 \times 172 \times 10^6 = 5,505 \times 10^6 = 5.505 \times 10^9 \text{ byes/sec}
Percentage of L1 bandwidth used = 5,505 / 16,000 = 0.344 or 34.4%

(or by just dividing CPU accesses / peak L1 accesses = 172/500 = 0.344 = 34.4%)

For L2:
The percentage of CPU memory accesses that reach L2 = (1-H1) = 0.05 or 5%
L2 Cache bandwidth usage = 0.05 \times 32 \times 172 \times 10^6 = 275.28 \times 10^6 \text{ byes/sec}
Percentage of L1 bandwidth used = 275.28 / 5,333.33 = 0.0516 or 5.16%

(or by just dividing CPU accesses that reach L2 / peak L2 accesses = 0.05 \times 172 / 166.67 = 8.6/166.67 = 0.0516 = 5.16%)

For Main Memory:
The percentage of CPU memory accesses that reach main memory = (1-H1) \times (1-H2) = 0.05 \times 0.4 = 0.02 or 2%
Main memory bandwidth usage = 0.02 \times 32 \times 172 \times 10^6 = 110.11 \times 10^6 \text{ byes/sec}
Percentage of main memory bandwidth used = 110.11 / 158.42 = 0.695 or 69.5%

Exercises: What if Level 1 (L1) is split?
What if Level 2 (L2) is write back with write allocate?
Program Steady-State Bandwidth-Usage Example

B. What is the total number of memory accesses generated by the CPU per second?

- For CPU with L1, L2, L3 Cache:
  - The total number of memory accesses generated by the CPU per second = (memory access/instruction) x clock rate / CPI = 1.3 x 500 x 10^6 / CPI = 650 x 10^6 / CPI
  - With L1, L2, L3 cache CPI was found = 2.54
    - CPU memory accesses = 650 x 10^6 / 2.54 = 255.9 x 10^6 accesses/sec

C. What is the percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

- For L1:
  - The percentage of CPU memory accesses that reach L1 = 100%
  - L1 Cache bandwidth usage = 32 x 255.9 x 10^6 = 8,188 x 10^6 = 8.188 x 10^9 byes/sec
  - Percentage of L1 bandwidth used = 5,505 / 16,000 = 0.5118 or 51.18%
    (or by just dividing CPU accesses / peak L1 accesses = 172/500 = 0.344 = 34.4%)

- For L2:
  - The percentage of CPU memory accesses that reach L2 = (1-H1) = 0.05 or 5%
  - L2 Cache bandwidth usage = 0.05x 32 x 255.9 x 10^6 = 409.45 x 10^6 byes/sec
  - Percentage of L1 bandwidth used = 409.45 / 5,333.33 = 0.077 or 7.7%
    (or by just dividing CPU accesses that reach L2 / peak L2 accesses = 0.05 x 255.9/ 166.67 = 12.795/ 166.67= 0.077= 7.7%)

- For L3:
  - The percentage of CPU memory accesses that reach L2 = (1-H1)x (1-H2) = 0.02 or 2%
  - L2 Cache bandwidth usage = 0.02x 32 x 255.9 x 10^6 = 163.78 x 10^6 byes/sec
  - Percentage of L1 bandwidth used = 163.78 / 2,666.67 = 0.061 or 6.1%
    (or by just dividing CPU accesses that reach L2 / peak L2 accesses = 0.02 x 255.9/83.33 = 5.118/83.33= 0.061= 6.1%)

- For Main Memory:
  - The percentage of CPU memory accesses that reach main memory = (1-H1) x (1-H2) x (1-H3) = .05 x .4 x.5 = 0.01 or 1%
  - Main memory bandwidth usage = 0.01 x 32 x 255.9 x 10^6 = 81.89 x 10^6 byes/sec
  - Percentage of main memory bandwidth used = 110.11 / 158.42 = 0.517 or 51.7%

Exercises: What if Level 1 (L1) is split?
What if Level 3 (L3) is write back with write allocate?
Dual (64-bit) Channel PC3200 DDR SDRAM has a theoretical peak bandwidth of

400 MHz x 8 bytes x 2 = 6400 MB/s

Is memory bandwidth still an issue?

Source: The Tech Report 1-21-2004
### X86 CPU Dual Channel PC3200 DDR SDRAM

Sample (Realistic?) Latency Data

<table>
<thead>
<tr>
<th>CPU Model</th>
<th>Latency (104 CPU Cycles)</th>
<th>Latency (256 CPU Cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athlon 64 3400+</td>
<td>47</td>
<td>80</td>
</tr>
<tr>
<td>Athlon 64 3200+</td>
<td>47</td>
<td>84</td>
</tr>
<tr>
<td>Athlon 64 3000+</td>
<td>48</td>
<td>85</td>
</tr>
<tr>
<td>Opteron 146</td>
<td>56</td>
<td>92</td>
</tr>
<tr>
<td>Athlon 64 FX-51</td>
<td>56</td>
<td>112</td>
</tr>
<tr>
<td>P4 3.2GHz Extreme Edition</td>
<td>68</td>
<td>113</td>
</tr>
<tr>
<td>Pentium 4 3.2GHz</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Pentium 4 'C' 2.4GHz</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>Pentium 4 'C' 2.8GHz</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>Athlon XP 3200+</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>Athlon XP 2500+</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>Athlon XP 2800+</td>
<td>113</td>
<td></td>
</tr>
</tbody>
</table>

**Source:** The Tech Report (1-21-2004)


**PC3200 DDR SDRAM** has a theoretical latency range of 18-40 ns
(not accounting for memory controller latency or other address/data line delays).

**On-Chip Memory Controller Lowers Effective Memory Latency**

**Is memory latency still an issue?**
X86 CPU Cache/Memory Performance Example:
AMD Athlon XP/64/FX Vs. Intel P4/Extreme Edition

Main Memory: Dual (64-bit) Channel PC3200 DDR SDRAM
peak bandwidth of 6400 MB/s

Source: The Tech Report 1-21-2004