Computer Organization & Assembly Language Programming

CSE 2312
Lecture 6 Memory

Junzhou Huang, Ph.D.
Department of Computer Science and Engineering
Reviewing (1): CPU

The organization of a simple computer with one CPU and two I/O devices
Reviewing (2): Instruction Execution Steps

• Fetch-decode-execute
  – Fetch next instruction from memory into instruction register
  – Change the program counter to point out the following instruction
  – Determine type of instruction just fetched
  – If instructions uses a word in memory, determine where it is
  – Fetch the word, if needed, into a CPU register
  – Execute the instruction
  – Go to step 1 to begin executing following instruction

Central to the operation of all computers
Reviewing (3): Interpreting Instructions

• **Interpreter**
  – A program that fetches, examines and executes the instructions of other program
  – Can write a program to imitate the function of a CPU
  – Main advantage: the ability to design a simple processor with the complexity largely confined to the memory holding the interpreter

• **Benefits** (simple computer with interpreted instructions)
  – The ability to fix incorrectly implemented instructions or make up for design deficiencies in the basic hardware
  – The opportunity to add new instructions at minimal cost even after delivery of the machine
  – Structured design that permitted efficient development, testing and documenting of complex instructions
Reviewing (4): Design Principles

• **Instructions directly executed by hardware**
  – Eliminating a level of interpretation provides high speed for most instructions;
  – Less frequently occurring instructions are acceptable

• **Maximize rate at which instructions are issued**
  – Parallelism can play a major role in improving performance

• **Instructions should be easy to decode**
  – A critical limit on the rate of issue of instructions is decoding individual instructions to determine what resources they need;
  – Fewer different formats for instructions, the better

• **Only loads, stores should reference memory**
  – Access the memory can take a long time
  – All other instructions should operate only on registers

• **Provide plenty of registers**
  – Running out of registers leads to flush them back to memory
  – Memory access leads to slow speed
Memory

• **What is Memory**
  – Part of computer
  – Used for store data and program

• **Basic Unit of Memory: bit**
  – A bit contains a 0 or 1
  – Simplest possible unit

• **Other Units**
  – Byte *(8-bit)*
  – Word
Cell

- **Cell** (Memories consist of a number of cells)
  - Each cell can store a piece of information
  - Each cell has a number called its address
  - A cell with k bits can hold one of $2^k$ different bit combinations
  - A memory with n cells will have addresses 0 to n-1
  - Adjacent cells have consecutive addresses (by definition)
  - If an address has m bits, the maximum number of cells addressable is $2^m$
  - The number of bits in the address determines the maximum number of directly addressable cells in the memory and independent of the number of bits per cell

**Cell: the smallest addressable unit**
Memory Addresses (1)

Three ways of organizing a 96-bit memory

<table>
<thead>
<tr>
<th>Address</th>
<th>Address</th>
<th>1 Cell</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>12 bits</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>8 bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(a)</td>
</tr>
</tbody>
</table>

- **Question:**
  - How many bits are sufficient for an address to reference the memory of Fig (a), (b), (c)?
  - 4, 3, 3
Memory Addresses (2)

- **Recent:**
  - All computer manufactures have standardized on an 8-bit cell

- **Byte: 8-bit**

- **Word**
  - A group of bytes
  - Most instructions operate on entire words
  - 32-bit Word: 4 bytes per word
  - 64-bit Word: 8 bytes per word
  - 32-bit machines need 32-bit registers
  - 64-bit machines need 64-bit registers

<table>
<thead>
<tr>
<th>Computer</th>
<th>Bits/cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burroughs B1700</td>
<td>1</td>
</tr>
<tr>
<td>IBM PC</td>
<td>8</td>
</tr>
<tr>
<td>DEC PDP-8</td>
<td>12</td>
</tr>
<tr>
<td>IBM 1130</td>
<td>16</td>
</tr>
<tr>
<td>DEC PDP-15</td>
<td>18</td>
</tr>
<tr>
<td>XDS 940</td>
<td>24</td>
</tr>
<tr>
<td>Electrologica X8</td>
<td>27</td>
</tr>
<tr>
<td>XDS Sigma 9</td>
<td>32</td>
</tr>
<tr>
<td>Honeywell 6180</td>
<td>36</td>
</tr>
<tr>
<td>CDC 3600</td>
<td>48</td>
</tr>
<tr>
<td>CDC Cyber</td>
<td>60</td>
</tr>
</tbody>
</table>

*Number of bits per cell for some historically interesting commercial computers*
Byte Ordering (1)

(a) Big endian memory
SPARC, IBM mainframes

(b) Little endian memory
Inter family

- **Byte Order**
  - Left-to-right or right-to-left
  - In both cases, **32-bit integer** is represented starting from rightmost and zero-filling in the leftmost
  - For example, 6 will be represented as 110 in rightmost 3 bits. Other leftmost 29 bits will be filled as 0
  - Problem: mixture of integer, strings and others
Byte Ordering (2)

<table>
<thead>
<tr>
<th>Big endian</th>
<th>Little endian</th>
<th>Transfer from big endian to little endian</th>
<th>Transfer and swap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 J I M</td>
<td>M I J</td>
<td>M I J</td>
<td>J I M</td>
</tr>
<tr>
<td>4 S M I T</td>
<td>T I M S</td>
<td>T I M S</td>
<td>S M I T</td>
</tr>
<tr>
<td>8 H 0 0 0</td>
<td>0 0 0 H</td>
<td>0 0 0 H</td>
<td>H 0 0 0</td>
</tr>
<tr>
<td>12 0 0 0 21</td>
<td>0 0 0 21</td>
<td>21 0 0 0</td>
<td>0 0 0 21</td>
</tr>
<tr>
<td>16 0 0 1 4</td>
<td>0 0 1 4</td>
<td>4 1 0 0</td>
<td>0 0 1 4</td>
</tr>
</tbody>
</table>

(a) A personal record for a big endian machine; (b) The same record for a little endian machine; (c) The result of transferring from big endian to little endian; (d) The result of byte-swapping

No simple solution

**Possible Solution:** include a header in front of each data item indicating its data type and how long it is
Quiz 1

• Chapter 1
  – Slides
  – Textbook
  – Homework