Module V: The Machine

Learning Goal [2+ classes]

you should be able to

• describe how a program (as text) is represented digitally and why high-level programs are translated into machine language programs
• describe the parts of fetch/decode/execute cycle
• simulate the operation of the CPU (central processing unit), given a diagram of its parts and a simple machine language program
• simulate the operation of small logic circuits, i.e. describe what output is produced by given inputs

Representing Programs Digitally

• programs are examples of text: letters, digits, punctuation marks, spaces and other symbols (e.g. +, tab, new-line)
• like all other data, text is stored in bits (0/1, on/off) in computer memory
• how might you represent text using bits?

Representing Programs Digitally

extended ASCII representation of text

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Extended ASCII representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>01100001</td>
</tr>
<tr>
<td>b</td>
<td>01100010</td>
</tr>
<tr>
<td>c</td>
<td>01100011</td>
</tr>
<tr>
<td>d</td>
<td>01100100</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>00110101</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>(</td>
<td>00101000</td>
</tr>
<tr>
<td>Symbol</td>
<td>Extended ASCII representation</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>a</td>
<td>01100001</td>
</tr>
<tr>
<td>b</td>
<td>01100010</td>
</tr>
<tr>
<td>c</td>
<td>01100011</td>
</tr>
<tr>
<td>d</td>
<td>01100100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text</th>
<th>Extended ASCII representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bad</td>
<td>01100010 01100001 01100100</td>
</tr>
<tr>
<td></td>
<td>01100011 01100001 01100010</td>
</tr>
</tbody>
</table>

Selected RQs

- A computer "thinks" in 0's and 1's. Voltages are translated into 0's and 1's. But, I am still confused about why or how the computer uses 0's and 1's. Is it actual 0's and 1's or is it current pulses (but then, why would they be translated to binary in the first place)?

  - Idalia

Representing Programs Digitally

- high-level programs are represented using extended ASCII
- extended ASCII is an example of a code which maps symbols over one alphabet to symbols over another
- codes are used in many non-computing contexts—can you think of examples?

Representing Programs Digitally

- high-level languages are English-language-like in appearance (e.g., Scratch, Python, JavaScript)
- programs written in high-level languages are translated into programs in low-level (simpler) machine language before they can be executed
- an interpreter or compiler is a program that does the translation
Representing Programs Digitally

high level vs. low level computer languages

- high-level languages are designed to support humans in specifying processes precisely
- low-level machine languages are designed to facilitate fast execution by computer hardware

Machine Language

machine language is linked to machine architecture

Machine Language

machine architecture abstracted

memory or RAM (random access memory): where program instructions and associated data are stored when running on the computer

I/O ports: communication with peripherals

processor or CPU: where program instructions are executed; contains ALU (arithmetic and logic unit) and control unit

Fetch/Decode/Execute Cycle

1. fetch the instruction specified by program counter from memory
2. decode the instruction's parts, particularly the type of operation and its arguments (register numbers, memory addresses, etc.)
3. execute the operation (which may send a value to registers, memory, or the program counter)
4. update the program counter to prepare for the next instruction
Machine Language

memory is a giant array of words
- each word stores 4/8 bytes
- each byte stores 8 bits
- the location of each byte has an address
- memory stores both data and machine language instructions

Selected RQs

• Does it make your computer slower if there is more information/programs stored in the RAM, because when the CPU needs to access a specific program instruction, it will take longer for the memory to find the right address and location of a specific program for the CPU, than if there were only a few addresses/locations to look through?

(submitted by Adrienne, 2011W1 Student)

A – Yes   B - No
Machine Language

**Machine Language instruction examples**

- **add, subtract, multiply** data stored in the processor (i.e., in registers)
- **load** data from memory to register
- **store** data to memory from register
- **branch** and **jump** instructions
- (instructions for communication with IO devices)

### Machine Language: Clicker Question

**Machine Language instruction examples**

The diagram indicates values *before* instruction **add 3, 4, 2** is executed. What value is in register 2 *after*?

A. 7  
B. 9  
C. 14  
D. 16

**Machine Language**

**Machine Language instruction examples**

**add 3, 4, 2:**
add the contents of registers 3 and 4 and store the answer in register 2

This **does not** mean add the numbers 3, 4 and 2

### Machine Language: Clicker Question

**Machine Language instruction examples**

The diagram indicates values *before* instruction **load 3, 7001** is executed. What value is in register 3 *after*?

A. 4  
B. 5  
C. 7  
D. 8
Machine Language

Machine Language instruction examples

- to explain jmp and branch, we need to introduce the program counter
- machine language instructions are stored sequentially in memory
- the program counter contains the address of the instruction which is currently being executed

Clicker Question

What value is in the program counter after the instruction referred to by the program counter is executed?

A. 1000
B. 1004
C. 1012
D. 7000
Machine Language

**high-level language to machine language**

- the translator assigns each variable a place in memory

```java
if (x < y) {
    z = y + x;
} else {
    z = y - x;
}
```

- what instruction should be in address 1012?

**Machine Language: Summary**

- machine language instructions strongly reflect machine architecture, referring to specific registers and memory locations
- machine language instructions have variable length, 1-15 bytes long
Selected RQs

- If every computer has its own machine language, what is used to translate different languages into every computer’s specific language? Does this mean that every computer functions in a different language? Why is that? Wouldn’t it be easier if they all spoke the same?
  - Camila

- Are Mac and PC (Windows) programs in different languages?
  - Sarah
Selected RQs

- If computers work by using the main memory, and the CPU fetches machine language instructions and then executes them, what is it precisely that is being effected when one gets a virus? Has the virus changed how the CPU is able to understand its functions, or has the virus inputted data into the memory that is causing malfunctions?

(submitted by Marya, 2011W1 Student)
• On a PC, when programs freeze I press Ctrl+Alt+Delete, and it opens up the task manager. It also displays something called “CPU usage” and graphs. What does this mean? The computer also gets very loud, why is this?

- Shauny