Chapter 7
Low-Level Programming Languages

Chapter Goals

• List the operations that a computer can perform
• Discuss the relationship between levels of abstraction and the determination of concrete algorithm steps
• Describe the important features of the Pep/7 virtual machine
• Distinguish between immediate mode addressing and direct addressing

Chapter Goals

• Convert a simple algorithm into a machine-language program
• Distinguish between machine language and assembly language
• Convert a simple algorithm into an assembly-language program
• Distinguish between instructions to the assembler and instruction to be translated
• Design and implement a test plan for a simple assemble-language program

Computer Operations

Computer
A programmable electronic device that can store, retrieve, and process data

Data and instructions to manipulate the data are logically the same and can be stored in the same place

What operations can a computer execute?

Levels of Abstraction

What do we mean by the expression "levels of abstraction"? Give several examples.

Machine Language

Machine language
The language made up of binary coded instructions built into the hardware of a particular computer and used directly by the computer

Why would anyone choose to use machine language? (Hint: they had no choice. Why?)
Machine Language

Characteristics of machine language:
- Every processor type has its own set of specific machine instructions
- The relationship between the processor and the instructions it can carry out is completely integrated
- Each machine-language instruction does only one very low-level task

Pep/7: A Virtual Computer

Virtual computer
A hypothetical machine designed to contain the important features of a real computer that we want to illustrate
Pep/7
A virtual computer designed by Stanley Warford that has 32 machine-language instructions

No, we are not going to cover all of them!

Features in Pep/7

Pep/7 Registers/Status Bits Covered
- The program counter (PC) (contains the address of the next instruction to be executed)
- The instruction register (IR) (contains a copy of the instruction being executed)
- The accumulator (A register)
- Status bit N (1 if A register is negative; 0 otherwise)
- Status bit Z (1 if the A register is 0; and 0 otherwise)
The memory unit is made up of 4,096 bytes

Instruction Format

Operation code
Specifies which instruction is to be carried out
Register specifier
Specifies which register is to be used (only use A in this chapter)
Addressing-mode specifier
Says how to interpret the operand part of the instruction
Instruction Format

Is there something we are not telling you about the addressing mode specifier? How can you tell?

Some Sample Instructions

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Meaning of Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>Step execution</td>
</tr>
<tr>
<td>00001</td>
<td>Load the operand into the A register</td>
</tr>
<tr>
<td>00010</td>
<td>Store the contents of the A register into operand</td>
</tr>
<tr>
<td>00011</td>
<td>Add the operand to the A register</td>
</tr>
<tr>
<td>00100</td>
<td>Subtract the operand from the A register</td>
</tr>
<tr>
<td>11011</td>
<td>Character input to the operand</td>
</tr>
<tr>
<td>11100</td>
<td>Character output from the operand</td>
</tr>
</tbody>
</table>

Sample Instructions

What do these instructions mean?

Instruction specifier: 00000100
Operand specifier: 0000000000000000

Instruction specifier: 00001011
Operand specifier: 0000000000000000

Instruction specifier: 00011010
Operand specifier: 0000000000000000

Instruction specifier: 00101010
Operand specifier: 0000000000000000

Sample Instructions

What do these instructions mean?

Instruction specifier: 00001011
Operand specifier: 0000000000000000

Instruction specifier: 00011010
Operand specifier: 0000000000000000

Instruction specifier: 00101010
Operand specifier: 0000000000000000
Sample Instructions

What do these instructions mean?

Instruction specifier: 00 10 01 10 00
Operand specifier: 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0

Instruction specifier: 11 0 0 0 0 0 0 1 0 0 0 0 0 8 0 0 0 0 0 0 1 0
Operand specifier: 0 0 0 0 0 0 0 0 1 0 0 0 0 3 0 1

Algorithms

Write “Hello” Are we concrete yet?

Write “Hell”
Write “H”
Write “e”
Write “l”
Write “l”
Write “o”

Are we concrete yet?

Write “Hello” Are we concrete yet?

Write “H”
Write “e”
Write 48 (hex)
Write 65 (hex)
Write “o”
Write 6F (hex)

Are we concrete yet?

A Program Example

<table>
<thead>
<tr>
<th>Module</th>
<th>Binary Instruction</th>
<th>Hex Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write “H”</td>
<td>11000000</td>
<td>00</td>
</tr>
<tr>
<td>Write “e”</td>
<td>11100000</td>
<td>0048</td>
</tr>
<tr>
<td>Write “l”</td>
<td>11110100</td>
<td>008B</td>
</tr>
<tr>
<td>Write “o”</td>
<td>11100000</td>
<td>0048</td>
</tr>
<tr>
<td>Stop</td>
<td>00000000</td>
<td>0</td>
</tr>
</tbody>
</table>

Hand Simulation

What is the fetch/execute cycle? How much is the PC incremented?

Program counter (PC)
Instruction Register (IR)

Hand Simulation

What is the fetch/execute cycle here?

Program counter (PC)
Instruction Register (IR)
Hand Simulation

What is the fetch/execute cycle here?

Pep/7 Simulator

A program that behaves just like the Pep/7 virtual machine behaves

To run a program

Enter the hexadecimal code, byte by byte with blanks between each

Pep/7 Simulator

What are the "zz"s for?

Pep/7 Simulator

What is a loader? What does it do?

Where does execution begin?

Assembly Language

Assembly language

A language that uses mnemonic codes to represent machine-language instructions

Assembler

A program that reads each of the instructions in mnemonic form and translates it into the machine-language equivalent
Pep/7 Assembly Language

<table>
<thead>
<tr>
<th>Pseudo-op</th>
<th>Operand</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDC</td>
<td>a</td>
<td>Load the characters between the (a) into memory.</td>
</tr>
<tr>
<td>BLOCK</td>
<td>#</td>
<td>Generate 8 bytes of storage and set each byte to zero.</td>
</tr>
<tr>
<td>PCL</td>
<td>a</td>
<td>Generate a word with the decimal value (a) stored in it.</td>
</tr>
<tr>
<td>PCLD</td>
<td>241256</td>
<td>Generate a word with the hexadecimal value (241256) stored in it.</td>
</tr>
</tbody>
</table>

What is the difference between operations and pseudo operations?

Assembly Process

A New Program

Reading and adding three numbers:
- Set sum to 0
- Read num1
- Add num1 to sum
- Read num2
- Add num2 to sum
- Read num3
- Add num3 to sum
- Write sum

Are we concrete yet?

Our Completed Program

BR Main: branch to location Main
sum: .WORD 0
num1: .BLOCK 4
num2: .BLOCK 4
num3: .BLOCK 4
Main:
LOADA sum.d
LOADA num1.d
ADD num1.d, sum.d
ADD num2.d, num1.d
ADD num3.d, num2.d
STORB sum.d
STOP: stop the processing
END: end of the program

Decision Making

Write "Error" if sum is negative.

Add num3 to sum
If sum is negative
Write "Error"
Else
Write sum

Decision Making

Remember the status bits A and Z?
Decision Making

Add num3 to sum
If status bit N is 1
  Go to NegMsg
Write sum
Quit: STOP
NegMsg: Write the message and go to Quit

Are we concrete yet?

Testing

Test plan
A document that specifies how many times and with what data the program must be run in order to thoroughly test it

Code coverage
An approach that designs test cases by looking at the code

Data coverage
An approach that designs test cases by looking at the allowable data values
Importthat Threads

Programming language
A set of grammar rules, symbols, and special words used to construct a program

Program
A sequence of instructions written to perform a specified task

Syntax
The formal grammar rules governing the construction of valid instructions

Semantics
The rules that give meaning to the instructions

Ethical Issues

Software Piracy and Copyrighting
Have you ever "borrowed" software from a friend?
Have you ever "lent" software to a friend?
Did you know that 107,000 jobs were lost in the US one year due to such "borrowing" and "lending?"

Who am I?

The architecture of the Pep? is named after me. Where was I born and from where did I escape? I consulted on which high-level language?

Do you know?

What can Platinum Blue's software predict?

How much money do software errors (bugs) cost the US economy annually?

To what did the Rosetta Stone provide the key to the translation?