

CMPS1134

Fundamentals of Computing

Data Manipulation 1

Computer Science: An Overview
Eleventh Edition
J. Glenn Brookshear
Chapter 2

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

Chapter 2: Data Manipulation

- ☐ **Computer Architecture**
- ☐ **Machine Language**

- ☐ Program Execution
- ☐ Arithmetic/Logic Instructions
- ☐ Communicating with Other Devices
- ☐ Other Architectures

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

2

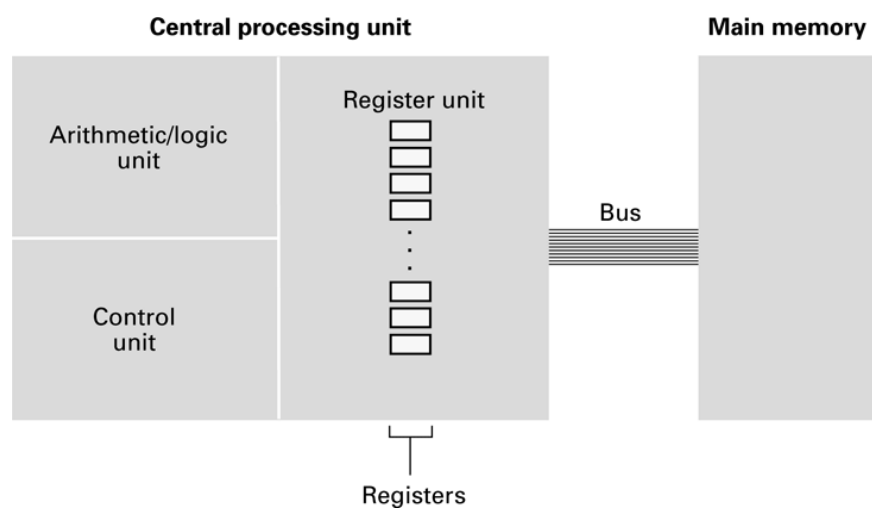
Computer Architecture

- ❑ **Central Processing Unit (CPU)** or processor consists of 3 parts:
 - 1. Arithmetic/Logic unit**
Circuitry that performs operations on data (e.g. addition and subtraction)
 - 2. Control unit**
Circuitry for coordinating the machines activities
 - 3. Registers**
Data storage cells that hold: data being manipulated by the CPU, the instruction being executed (instruction register), & address of the next instruction to be executed (program counter)
 - ❑ **General-purpose registers**
 - ❑ **Special-purpose registers**
- ❑ **Bus** – connects the CPU and Main Memory
- ❑ **Motherboard** – machine's main circuit board

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

3

Figure 2.1 CPU and main memory connected via a bus



Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

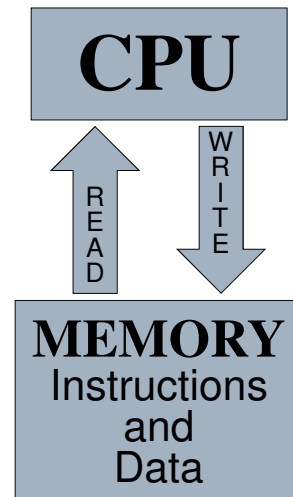
4

Stored Program Concept

A program can be encoded as bit patterns and stored in main memory.

From there, the CPU can then extract the instructions and execute them.

In turn, the program to be executed can be altered easily.



Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

5

Terminology

- ❑ **Machine instruction:** An instruction (or command) encoded as a bit pattern recognizable by the CPU
- ❑ **Machine language:** The set of all instructions recognized by a machine

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

6

Machine Language Philosophies

❑ **Reduced Instruction Set Computing (RISC)**

- Few, simple, efficient, and fast instructions
- Examples: PowerPC from Apple/ IBM/ Motorola and ARM

❑ **Complex Instruction Set Computing (CISC)**

- Many, convenient, and powerful instructions
- Example: Intel

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

7

Machine Instruction Types

- ❑ **Data Transfer:** copy data from one location to another
- ❑ **Arithmetic/Logic:** use existing bit patterns to compute a new bit patterns
- ❑ **Control:** direct the execution of the program

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

8

Figure 2.2 Adding values stored in memory

Step 1. Get one of the values to be added from memory and place it in a register.

2

Step 2. Get the other value to be added from memory and place it in another register.

3

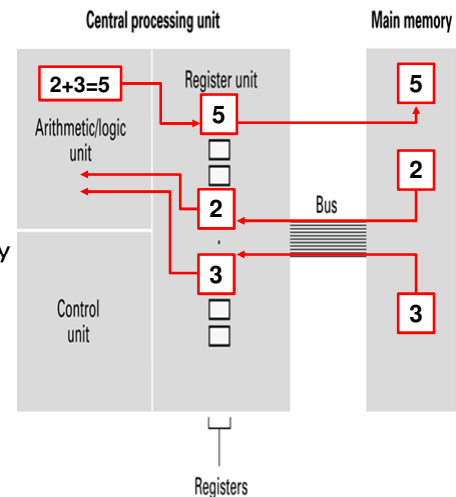
Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.

2+3=5

5

Step 4. Store the result in memory.

Step 5. Stop.



Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

9

Figure 2.3 Dividing values stored in memory

Step 1. LOAD a register with a value from memory.

8

Step 2. LOAD another register with another value from memory.

4

Step 3. If this second value is zero, JUMP to Step 6.

Step 4. Divide the contents of the first register by the second register and leave the result in a third register.

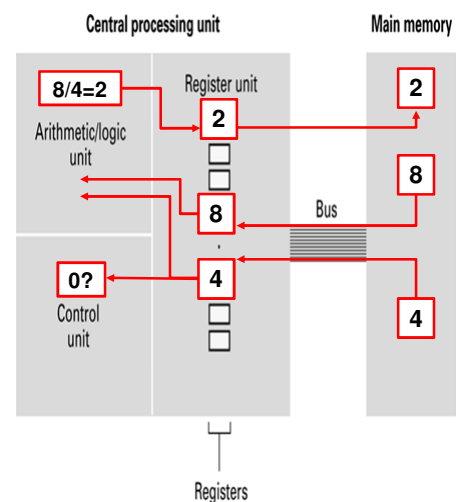
0?

8/4=2

Step 5. STORE the contents of the third register in memory.

2

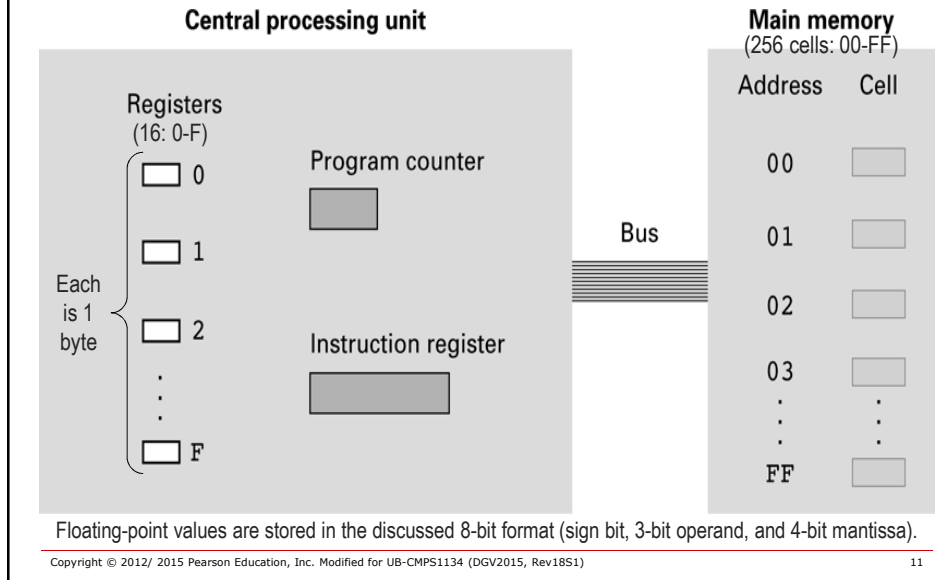
Step 6. STOP.



Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

10

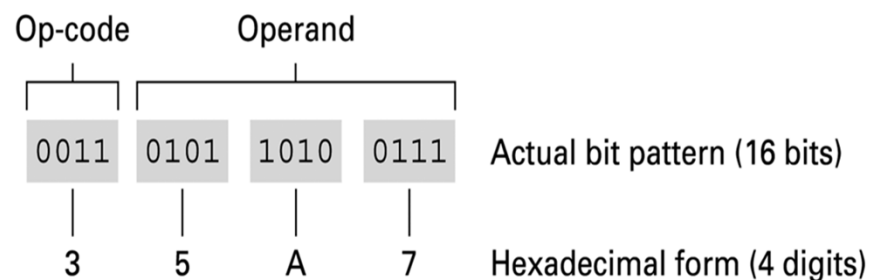
Figure 2.4 The architecture of the machine described in Appendix C



11

Parts of a Machine Instruction

- ❑ **Op-code:** Specifies which operation to execute
- ❑ **Operand:** Gives more detailed information about the operation
 - Interpretation of operand varies depending on op-code



Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

12

Appendix C: A Simple Machine Language

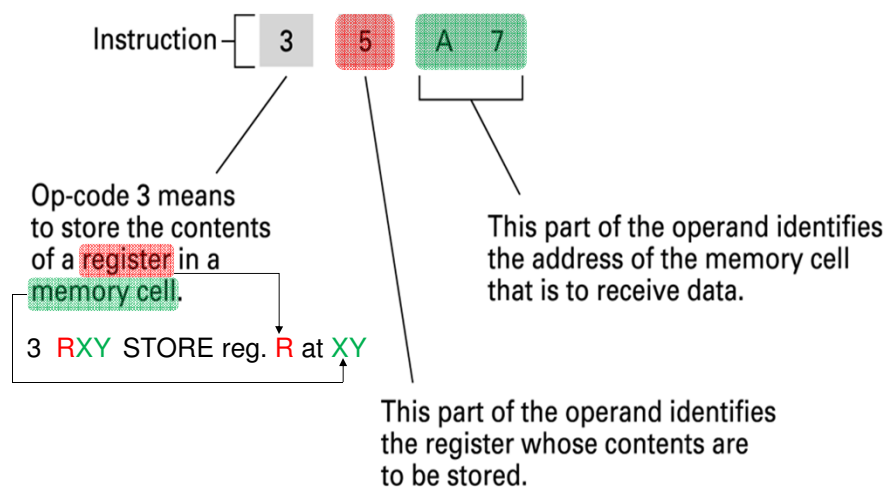
Op-code Operand Description

1	RXY	LOAD reg. R from cell XY.
2	RXY	LOAD reg. R with XY.
3	RXY	STORE reg. R at XY.
4	ORS	MOVE R to S.
5	RST	ADD S and T into R. (2's comp.)
6	RST	ADD S and T into R. (floating pt.)
7	RST	OR S and T into R.
8	RST	AND S and T into R.
9	RST	XOR S and T into R.
A	R0X	ROTATE reg. R X times.
B	RXY	JUMP to XY if R = reg. 0.
C	000	HALT

Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

13

Figure 2.6 Decoding the instruction 35A7



Copyright © 2012/ 2015 Pearson Education, Inc. Modified for UB-CMPS1134 (DGV2015, Rev18S1)

14

Figure 2.7 An encoded version of the instructions in Figure 2.2

