CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UNIT 2.FUNCTIONAL ELEMENTS OF A COMPUTER

Computer Systems
CFGS DAW

Activity 2. Solution

Autores: Alfredo Oltra / Sergio Garcia
Adaptado: Vicent Bosch
vicent.bosch@ceedcv.es
2020/2021

Versión:201026.1825

Licencia

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

UD01. FUNCTIONAL ELEMENTS OF A COMPUTER
Activities 2. Solutions

1. EXERCISE 1

(Exercise 1) We have a hypothetical computer with this instruction format:

OP_CODE	OPERAND 1	OPERAND 2
4 BITS	4 BITS	4 BITS

0000	0xC2
0001	0x19
0010	0x5A
0011	0x2
…	

Figure 1. Memory (address and content)

SUM Rx, Ry 1001xxyy Add RX+RY and it is stored in RX.

Following the instruction sequence:

100100010010

a) What is the result after executing this instruction?

The UC decodes the instruction according to the instruction format:100100010010

1001	0001	0010
4 BITS	4 BITS	4 BITS

The instruction is *add* what it's stored in 0001 (address 1) and in 0010 (address 2)

If we check the *memory unit*, in address 0001 the value stored is 0x19 and in address 0010 the value stored is 0x5A. Since the values are represented using hexadecimal we must convert to binary and perform the *add* operation. In the CPU this operation is performed by the ALU.

19→ 0 0 0 ^{1} 1 ^{1} 1 0 0 1

5A→ 0 1 0 1 1 0 1 0

0 1 1 1 **0** 0 1 1 → **73** $_{16}$

b) Which will be the state of the memory after the execution of this instruction?

Since the instruction *add* says Add RX+RY and it is stored in RX. This means that the result of the add operation must be stored in address RX, which in this case is Address 0001. There the value is overwritten.

0000	0xC2
0001	***0x73***
0010	0x5A
0011	0x2
…	

c) Which is the addressing mode used in both operands?
Both operands store the address of the content to be used by the add instruction, therefore the addressing mode is absolute/direct (*page 11 Unit 2)*

d) What would be the result if operand 2 uses immediate addressing mode?

When immediate addressing mode is used, the content is stored in the operand field itself, therefore the add operation would be:

19→ 0 0 0 1 1 0 0 1
2→ 0 0 0 0 0 0 1 0
0 0 0 1 1 0 1 1→ **1B** $_{16}$

2. EXERCISE 2

Memory																		
0										16								
1										17	0	0	0	0	0	0	1	1
2										18								
3										19								
4										20								
5										21								
6										22								
7										23								
8										24								
9										25								
10										26								
11	0	0	0	0	0	0	0	1		27								
12	0	0	0	0	0	0	1	0		28	0	0	0	0	0	1	0	0
13	0	0	0	0	0	1	0	0		29								
14										30								
15										31								

Registers								
0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
3	0	0	0	0	0	1	0	0

00001011 Write 1 in memory position 11
00001100 Write 2 in memory position 12
00010001 Write 3 in memory position 17
00011100 Write 4 in memory position 28
01001011 Copy the data from memory position 11 to register 0
10000100 Copy the data from Register 0 to Register_1: 1 → in R1
01011100 Copy the data from memory position 28 to register 0
10001100 Copy the data from Register 0 to Register_3: 4 → in R3
01010001 Copy the data from memory position 17 to register 0
10001000 Copy the data from Register 0 to Register_2: 3 → in R2
10111110 Multiply the content of R3 and R2 and write the result in R3 [3*4] → 12 in R3
10101101 Subtract the content of R3 and R1 and write the result in R3 [12-1] → 11 in R3
01001100 Copy the data from memory position 12 to Register_0
10001000 Copy the data from Register 0 to Register_2: 2 → in R2
10011110 Add the content of R3 and R2 and write the result in R3 [2 + 11] → 13 ¡n R3
01010001 Copy the data from memory position 17 to Register_0
10001000 Copy the data from register_0 to register_2 → 3 in R2
11001110 Divide the content of R3 by R2 and write in R3 [13/3] 4 in R3
10000011 Copy the date from R3 to R0 → 4 in R0
01101101 Write in memory position 13 the content of Register 0
00101101 Show in the screen the content of memory position 13

2.1 Solution

a) Formula: ((D*C)-A+B)/C

b) 4 (Content of memory position 13)

c) The state shown in the solution

d) If the PC was initially at 258 and we have executed 21 instructions, the PC will contain the value 279

e) we have two bits, i. e. 4 registers.