Benha University
Faculty of Science
Dept. of Mathematics

Time: Two Hours
First Semester 2014-2015
Date : 21/12/2014

Computer Structure (MC421) for Fourth Level Students (Computer Science)

جامعة بنها – كلية العلوم – قسم الرياضيات

المستوي الرابع (علوم حاسب)

يوم الامتحان: الاحد

تاريخ الامتحان: ٢١ / ١٢ / ٢٠١٤ م

المادة : بنية الحاسب (٤٢١ رس)

الممتحن: د/ مصعب عبد الحميد محمد حسان

مدرس بقسم الرياضيات بكلية العلوم

الاسئلة و نموذج الإجابة

ورقة كاملة

Benha University
Faculty of Science
Dept. of Mathematics

Time: Two Hours
First Semester 2014-2015
Date : 21/12/2014

Computer Structure (MC421) for Fourth Level Students (Computer Science)

Answer the following question

Question 1. (12 marks)

A- Define microoperations, instruction code, common bus system, assembler? (4 marks)

B- Consider 128K X 16 memory. How many address lines does it have? How many data lines it have? What is its total capacity in Kbytes? What will be the total capacity if the address lines are changed to 20 lines without changing the word size? (3 marks)

C- Draw the bus system for two registers such that each register has eight bits? (5 marks)

Question 2. (14 marks)

A- List the registers in the basic computer and write a description of their function and the number of bits that they contain? (4 marks)

B- Write the instruction cycle steps in details? (6 marks)

C- What is the difference between a direct and indirect address instruction? How many references to memory are needed for each type of instruction to bring an operand into a processor register? (4 marks)

Question 3. (12 marks)

A- For each timing signal, show the contents in hexadecimal of registers PC, AR, DR, IR, and SC of the basic computer when an ISZ indirect instruction is fetched from memory and executed. The initial content of PC is 7FE. The content of memory at address 7FE is EA9F. The content of memory at address A9F is 0C35. The content of memory at address C35 is FFFF. (4 marks)

B- List the memory reference instructions and for each instruction write operation decoder and symbolic description? (5 marks)

C- List the rules of the assembly language? (3 marks)

<u>Question 4. (10 marks)</u>

A- List the categories of programs written for a computer? (3 marks)

B- Write assembly language program to calculate the expression a+b-c ? (4 marks)

C- Given the common bus system of the basic computer, the following register transfers are to be executed in the system? For each transfer, specify: (1) the binary value that must be applied to bus select inputs S_2, S_1, and S_0; (2) the register whose LD control input must be active (if any); (3) a memory read or write operation (if needed) (3 marks)

(i) AR ← PC (ii) IR ← M[AR]

(iii) M[AR] ← TR

Best Wishes
Dr. Mosab Abd El-Hameed

Benha University
Faculty of Science
Dept. of Mathematics

Time: Two Hours
First Semester 2014-2015
Date : 21/12/2014

Computer Structure (MC421) for Fourth Level Students (Computer Science)

Answer of Question 1

A- *microoperations* are operations executed on data stored in one or more registers

instruction code is a group of bits that instructs the computer to perform a specific operation (sequence of microoperations). It is divided into parts (basic part is the operation part)

common bus system is a scheme for transferring information between registers in a multiple-register configuration

The translation of symbolic program into binary is done by a special program called an *assembler*

B- This memory has

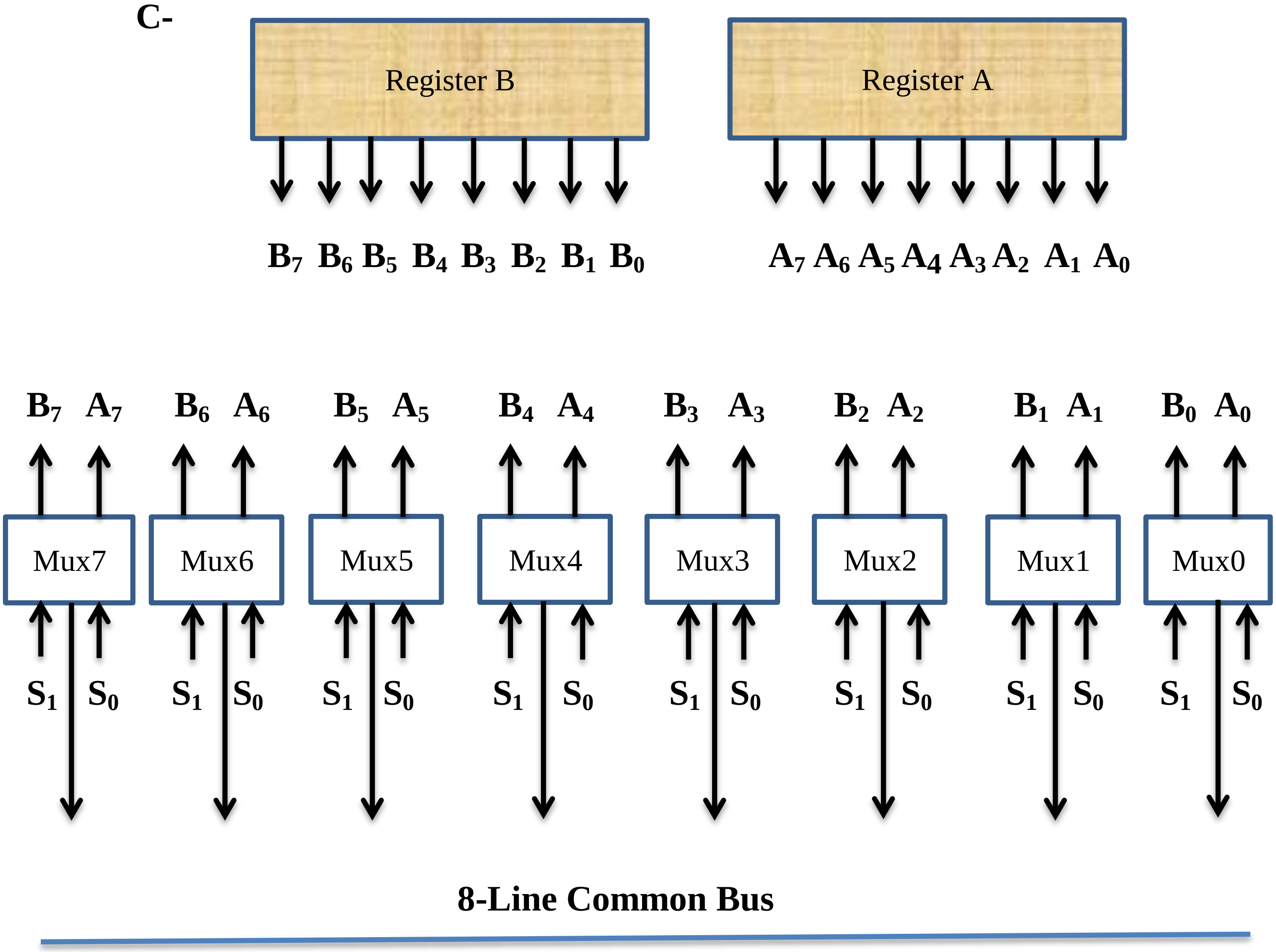
-17 address lines.

- 16 data lines.

- Total capacity of (128 x 1024 x 16) / (1024 x 8) = 256 Kbytes

- If the address lines are changed to 20 lines without changing the word size, the memory will be 1024K X 16. Then the total capacity will be (1024 x 1024 x 16) / (1024 x 8) = 2048 Kbytes

C-



Answer of Question 2

A- List of BC registers:

DR	**16**	**Data Register**	**Holds memory operand**
AR	**12**	**Address Register**	**Holds address for memory**
AC	**16**	**Accumulator**	**Processor register**
IR	**16**	**Instruction Register**	**Holds instruction code**
PC	**12**	**Program Counter**	**Holds address of instruction**
TR	**16**	**Temporary Register**	**Holds temporary data**
INPR	**8**	**Input Register**	**Holds input character**
OUTR	**8**	**Output Register**	**Holds output character**

B-

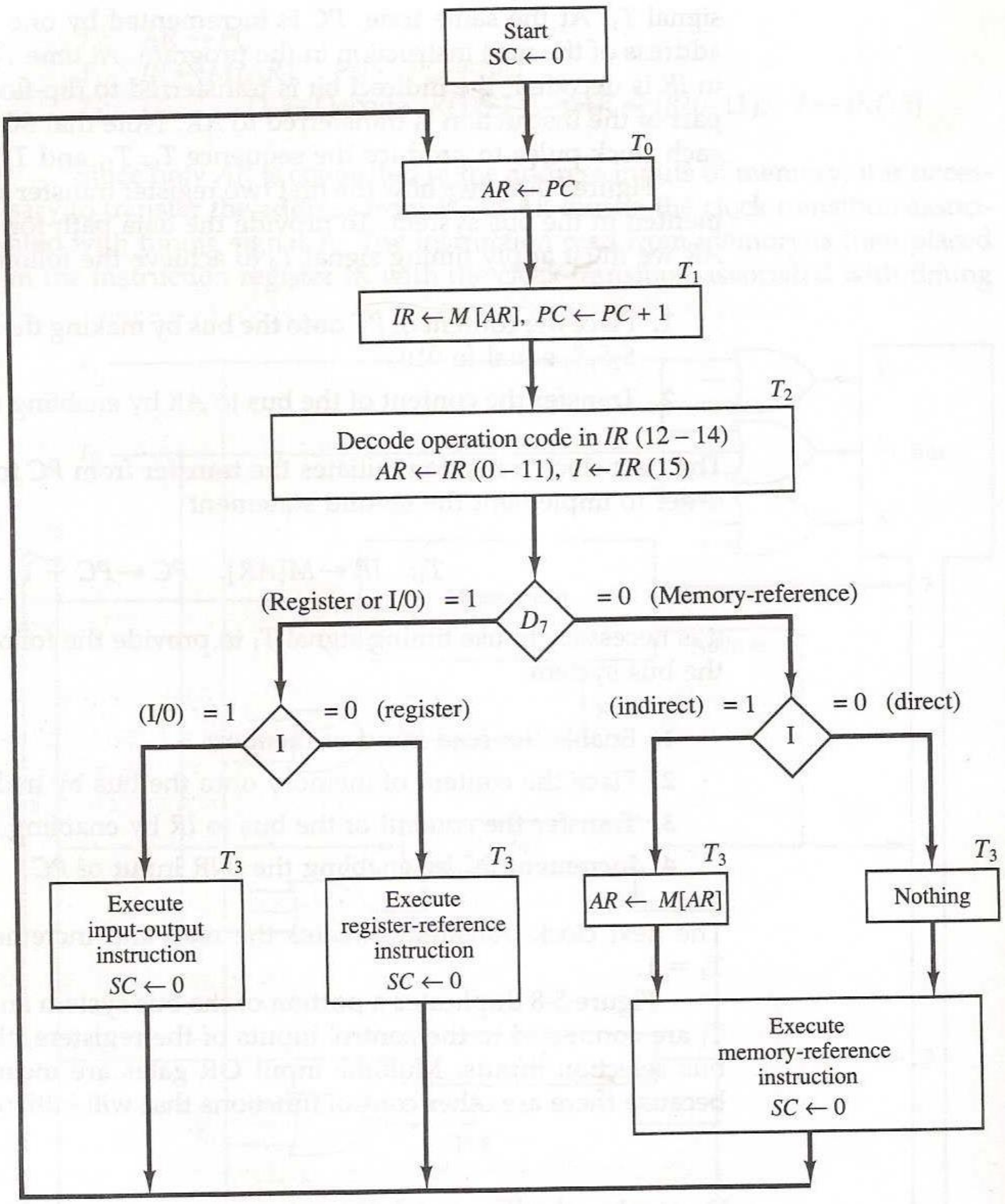
- Instruction Cycle Steps:
  1- Fetch an instruction from memory
  2- Decode the instruction
  3- Read the effective address from memory if the instruction has an indirect address
  4- Execute the instruction

This cycle repeats indefinitely unless a HALT instruction is encountered

- Instruction Cycle Fetch and Decode
  1- Initially, the Program Counter (PC) is loaded with the address of the first instruction in the program
  2- The sequence counter SC is cleared to 0, providing a decoded timing signal T_0
  3- After each clock pulse, SC is incremented by one, so that the timing signals go through a sequence T_0, T_1, T_2, and so on

T_0: AR ← PC (this is essential)
The address of the instruction is moved to AR.
T_1: IR ← M[AR], PC ← PC+1
The instruction is fetched from the memory to IR, and the PC is incremented.
T_2: $D_0, \ldots, D_7$ ← Decode IR(12-14), AR←IR(0-11), I←IR(15)

Next figure shows the instruction cycle flowchart. we can determine the type of instruction (see in the figure). For example:
D'7IT3: AR ← M[AR]
D'7I'T3: Nothing
D7I'T3: Execute a register-reference instr.
D7IT3: Execute an input-output instr.



C- Direct Address Instruction has its operand address specified directly in the instruction, but the Indirect Address Instruction has the address of the operand's address, i.e. not directly the address of the operand.

Direct Address Instruction: needs 2 visits to memory. The 1st one is to fetch the instruction, 2nd is to bring the operand.

Indirect Address Instruction: needs 3 visits to memory. The 1st one is to fetch the instruction, 2nd is to bring the operand address. And the 3rd is to bring the operand itself.

Answer of Question 3

A-

Given	7FE	X	X	X	0
T0	7FE	7FE	X	X	1
T1	7FF	7FE	X	EA9F	2
T2	7FF	A9F	X	EA9F	3
T3	7FF	C35	X	EA9F	4
T4	7FF	C35	FFFF	EA9F	5
T5	7FF	C35	0000	EA9F	6
T6	800	C35	0000	EA9F	0

B-

Symbol	Operation	Symbolic Description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], E \leftarrow C_{out}$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, PC \leftarrow AR + 1$
ISZ	D_6	$M[AR] \leftarrow M[AR] + 1$, if $M[AR] + 1 = 0$ then $PC \leftarrow PC+1$

C- Each line of an assembly language program is arranged in three columns called fields:

1-The **Label** field: May be empty or specify a symbolic address.

2- The **Instruction** field: Specifies a machine instruction or pseudo instruction.

3- The **Comment field:** May be

```
          ORG Lab
Lab,      ADD op1    / this is an add operation.
Label     Instruction   Comment
```

Answer of Question 4

A- The categories of programs written for a computer:

1- Binary code. This is a sequence of instructions and operands in binary that list the exact representation of instructions as they appear in computer memory.

2- Octal or hexadecimal code. This is an equivalent translation of the binary code or hexadecimal.

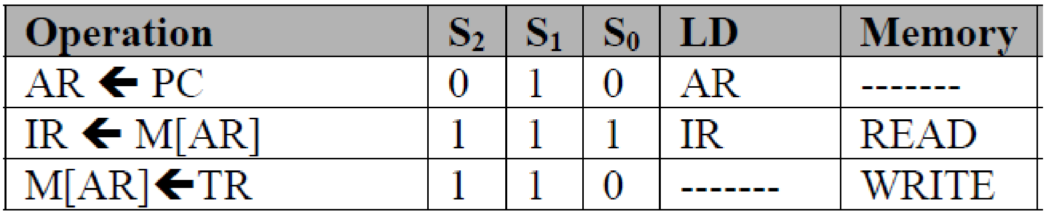
3- Symbolic code. The user employs symbols (letters, numerals, or special characters) for the operation part, the address part, and other parts of the instruction code. Each symbolic instruction can be translated into one binary coded instruction.

4- High-level programming languages. These are special languages developed to reflect the procedures used in the solution of a problem rather than be concerned with the computer hardware behavior. For example, C++, Java, C#.

B- Suppose we calculate 83 + 23 - 15

```
      ORG  0      /Origin of program is location 0
      LDA  A      /Load  operand from location A
      ADD  B      /Add  operand from location B
      STA  D      /Store sum in location D
      LDA  C      /Load  operand from location C
      CMA         /Complement AC
      INC         /Increment AC
      ADD  D      /Add  operand from location D
      STA  E      /Store the result in location E
      HLT         /Halt computer
   A, DEC  83     /Decimal operand
   B, DEC  23     /Decimal operand
   C, DEC  15     /Decimal operand
   D, DEC  0      /Sum stored in location D
   E, DEC  0      /The result stored in location E
     END          /End of symbolic program
```

C-



Operation	S_2	S_1	S_0	LD	Memory
AR ← PC	0	1	0	AR	-------
IR ← M[AR]	1	1	1	IR	READ
M[AR]←TR	1	1	0	-------	WRITE