UNIVERSITE DE BOUMERDES UNIVERSITY OF BOUMERDES	***IGEE*** ***University M'Hamed*** ***Bougara of Boumerdès***	***Last Name:*** …… ***First Name:*** …… ***Group:*** ……	***Grade :***

EE321: Computer Architecture
Final Exam Solution (1h30)
2017 – 2018

Notes: Answer briefly and clearly using the provided space. No extra sheet will be accepted.

1. Interrupts (5 pts)

Consider a system with 3 I/O devices: D1, D2, and D3, with increasing priorities of 1 (low priority), 2 and 3 (high priority), respectively. At t=0, user program begins, device D2 interrupt occurs at t=10, device D1 interrupt occurs at t=15, device D3 interrupt occurs at t=25. We assume D1 ISR takes 30 time units, D2 ISR takes 20 units, and D3 ISR takes 10 units.

1.1 Explain interrupt process and its advantage.

Interrupt is a process by which other devices (peripherals) communicate with the CPU. Interrupts improve CPU processing efficiency.

1.2 Explain briefly the two techniques that can be used to deal with multiple interrupts.

The two techniques are:
a) Disable interrupts: CPU will ignore further interrupts while processing one interrupt and interrupts are handled in sequence as they occur.
b) Define priorities: Low priority interrupts can be interrupted by high priority interrupts.

1.3 Give the advantage and disadvatage of the first technique compared to the second one.

Disable interrupt approach is simple to implement but does not take into account relative priority.

1.4 Give time sequence (start and end time of each interrupt) considering the first technique.

Using disable interrupt approach, we have:
a) D1 ISR starts at t=30 and ends at t=60.
b) D2 ISR starts at t=10 and ends at t=30.
c) D3 ISR starts at t=60 and ends at t=70.

1.5 Give time sequence (start and end time of each interrupt) considering the second technique.

Using define priority approach, we have:
d) D1 ISR starts at t=40 and ends at t=70.
e) D2 ISR starts at t=10 and ends at t=40.
f) D3 ISR starts at t=25 and ends at t=35.

2. Instructions Set and Addressing Modes (8 pts)

Consider a simple 8-bit CPU with the following specifications: 8-bit address bus, memory is byte addressable, opcodes and addressing mode codes are given in Table 1, binary codes of the general purpose registers (GPR) are given in Table 3, and the instructions format is given in Table 4.

2.1 Give the size of registers R0, IR, and MBR. Justify.

R0: 8 bits (8-bit CPU).
IR: 16 bits (see instructions format in Table 4).
MBR: 8 bits (memory is byte addressable).

2.2 Give the two methods that are used to specify the addressing mode. Which one is used for this CPU. Justify.

1) Different opcodes.
2) Addressing mode field.
This CPU uses addressing mode field (see instructions format in Table 4).

2.3 Give the addressing mode of the following instructions: a)**ADD Rd, X**, b) **ADD Rd, Rs**, c)**ADD Rd, [Rs]**, d) **ADD Rd, [X]**, e)**ADD Rd, [[X]]**, and f) **ADD Rd, #X**.

a) Immediate mode.
b) Register mode.
c) Register-(in)direct mode.
d) Direct mode.
e) Indirect mode.
f) Displacement mode.

2.4 How many memory locations we can address using the following instructions: a)**ADD Rd, X**, b) **ADD Rd, Rs**, c)**ADD Rd, [Rs]**, d) **ADD Rd, [X]**, and e) **ADD Rd, #X**? Justify.

a) No memory access.
b) No memory access.
c) Since Rs is 8-bit register, the addressing space is $2^8 = 256$ locations.
d) Since we have 6 bits for the address operand, the addressing space is $2^6 = 64$ locations.
e) Since R0 is 8-bit register and X is 6-bit value, the addressing space is $2^8 = 256$ locations.

2.5 Give the hex code of the following instructions: a)**ADD R7,0FH** , b) **ADD R7, R0**, c)**ADD R7, [R0]**, d) **ADD R7, [0FH]**, e)**ADD R7, [[0FH]]**, and f) **ADD R7, #0FH**. Justify.

*a) The binary code of the instruction ADD R7, 0FH is: 0100 000 111 001111. The hex code of the instruction is **41CFH**.*
*b) The binary code of the instruction ADD R7, R0 is: 0100 001 111 000000. The hex code of the instruction is **43C0H**.*
*c) The binary code of the instruction ADD R7, [R0] is: 0100 010 111 000000. The hex code of the instruction is **45C0H**.*
*d) The binary code of the instruction ADD R7, [0FH] is: 0100 011 111 001111. The hex code of the instruction is **47CFH**.*
*e) The binary code of the instruction ADD R7, [[0FH]] is: 0100 100 111 001111. The hex code of the instruction is **49CFH**.*
*f) The binary code of the instruction ADD R7, #0FH is: 0100 101 111 001111. The hex code of the instruction is **4BCFH**.*

2.6 Consider the memory content given in Table 2. Assuming little endian memory, fill the following table with the contents of the registers after the execution of the instruction stored at address 10H. Put X if unknown.

The hex code of the instruction at address 10H is 400FH.
The binary code is 0100 000 000 00111, the instruction is: ADD R0, 0FH.
R0= R0 + 0FH = F0H + 0FH = FFH.

Register	PC	MBR	MAR	IR	R0	R1	R2	R3	R4
Before	10H	X	X	X	F0H	F1H	F2H	F3H	F4H
After	12H	40H	11H	400FH	FFH	-	-	-	-

2.7 Consider the memory content given in Table 2. Fill the following table with the contents of the registers after the execution of the instruction.

The hex code of the instruction is: 490AH.
The binary code is 0100 100 100 001010H, the instruction is: ADD R4, [[0AH]].
R4 = R4 + [[0AH]] = R4 + [05H] = F4H + 0CH = 00H.

Register	PC	MBR	MAR	IR	R0	R1	R2	R3	R4
After fetch	20H	49H	19H	490AH	F0H	F1H	F2H	F3H	F4H
After execution	-	0CH	05H	-	-	-	-	-	00H

3. Memory (7 pts)

Consider an 8-bit memory chip designed using one 1024x1024 memory array.

3.1 Give the size, width and depth of the memory.

Size = 1 Mbits = 1024 Kbits = 1048576 bits.
Width = 8 bits.
Depth = 131 072 locations = 128K locations.

3.2 How many address pins are required for the memory chip assuming: a) without address multiplexing, and b) with address multiplexing.

a) Without address multiplexing: 17 address pins.
b) With address multiplexing: 9 address pins.

3.3 How many address bits are required for: a) row decoder, and b) column decoder? Justify.

Since we have 1024 word-lines, we need 10 bits for the row decoder.
Since we have 1024 bit-lines and we select 8 bits for data bus, we need 7 bits for the column decoder.

3.4 If we assume the same memory chip is designed using 4 arrays (2 rows by 2 colums), how many address bits are required for: a) row decoder, and b) column decoder? Justify.

For each memory array, we have:
Number of word-lines is: 1024/2 = 512 word-lines, we need 9 bits for row decoder.
Number of bit-lines is: 1024/2 = 512 bit-lines and we select from each array 2 bits for data, we need 8 bits for column decoder.

3.5 Assume that a 64Kx32-bit main memory is built using this memory chip, find the number of the used memory chips to build the main memory. Justify.

The main memory depth is 64 K locations and the depth of the memory chip is 128 K.
The main memory data width is 32 bits and the data width for the memory chip is 8 bits, so we need 4 memory chips to built a 64Kx32-bit main memory.

3.6 Draw a transistor schematic of a ROM memory with the content given in Table 5.

Good Luck!

Instruction	Opcode (hex)	Description	Addressing mode code (binary)
MV Rd, X	1H	Rd = X	000
MV Rd, Rs	1H	Rd = Rs	001
LD Rd, [X]	2H	Rd = [X]	011
LD Rd, #X	2H	Rd = [R0+X]	101
LD Rd, Rs	2H	Rd = [Rs]	010
STR Rd, Rs	3H	[Rd] = Rs	010
ADD Rd,X	4H	Rd = Rd+X	000
ADD Rd, Rs	4H	Rd = Rd+Rs	001
ADD Rd, [Rs]	4H	Rd = Rd+[Rs]	010
ADD Rd, [X]	4H	Rd =Rd+ [X]	011
ADD Rd,[[X]]	4H	Rd=Rd+[[X]]	100
ADD Rd,#X	4H	Rd=Rd+[R0+X]	101
JPR X	5H	PC = PC + X	/
HLT	FH	Stop the execution of the program	/

Rd = Source/destination register (it can be any GPR).
Rs = Source register (it can be any GPR)
R0 = Register R0.
[X] = content of memory location X.

Table 1: Instructions description.

Address (hex)	Content (hex)
00H	0FH
01H	1EH
02H	2AH
03H	0AH
04H	0BH
05H	0CH
06H	0DH
07H	02H
08H	03H
09H	04H
0AH	05H
0BH	09H
0CH	0BH
0DH	0CH
0EH	0DH
0FH	0EH
10H	0FH
11H	40H
12H	2AH

Table 2: Memory content.

Register	R0	R1	R2	R3	R4	R5	R6	R7
Code	000	001	010	011	100	101	110	111

Table 3: Binary codes of GPRs.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode				Addressing mode			Destination register			Source operand					

Source operand can be register code, address, or value. If register code, bits 3 to 5 are equal to 0.
Destination register contains the binary code of source/destination register.

Table 4: Instructions format.

Address		Content		
A1	A0	D2	D1	D0
0	0	0	0	0
0	1	0	1	0
1	0	1	0	1
1	1	1	1	1

Table 5: ROM content.