King Saud University
College of Computer and Information Sciences
Computer Science Department

Course Code	CSC 220		
Course Title	Computer Organization		
Semester	S1 – 1441 (Fall-2020-21)		
Exam	Final		
Date	29/12/2020	**Duration**	**3 Hours**
Student Name			
Student ID			
Section No.			

Course Learning Outcomes		**Relevant question**	**Full mark**	**Student mark**
Knowledge	**CLO 1, 2, 3**	**1**	**8**	
Knowledge	**CLO 4, 5, 6, 7**	**2**	**8**	
Cognitive Skills	**CLO 2, 3, 5**	**3**	**6**	
Cognitive Skills	**CLO 4, 5**	**4**	**6**	
Cognitive Skills	**CLO 3, 5**	**5**	**6**	
Cognitive Skills	**CLO 3, 6, 7**	**6**	**6**	

Feedback/Comments:

Question 1. Short Questions (8 Marks: 8×1)

i. Extend the sign of the 4-bit 2's complement binary number 1011 to convert it into a 8-bit number.

1111 1011

ii. What are the disadvantages of 1's complement representation of binary number?

Zero can be [0000, 1111]

iii. How many full adders are required to design a 32-bit parallel adder?

32 Full Adders

iv. Give the truth table for a 1-to-4 demultiplexer.

S1	S0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

v. A digital computer has a common bus system for 64 registers of 16 bits each. The bus is constructed with multiplexers. How many multiplexers are needed? How many selection bits are required?

i) Num of Mux = Num of bits = 16 Multiplexers

ii) Size of mux = 64x1 Mux ⟹ 6 Selection bits

vi. Give the transition table of J-K Flip Flop.

C	J	K	Q
0	X	X	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	Q(t+1)

vii. Write RTL instructions for the following operations: write the data contained in register R3 to the memory location contained in register AR.

M[AR] ← R3

viii. If a register containing data (11001100) is subjected to arithmetic shift right operation, then the content of the register after the operation.

11100110

Question 2. Short Questions (8 Marks: 2+2+2+2)

i. Suppose we have 1024 x 16 RAM chips. How many chips are needed to provide a memory capacity of 16KB? What will be the size of address base for the 16 KB RAM?

$$\text{chips} = \frac{2^4 \cdot 2^3 \cdot 2^{10}}{2^4 \cdot 2^{10}} = 2^3$$

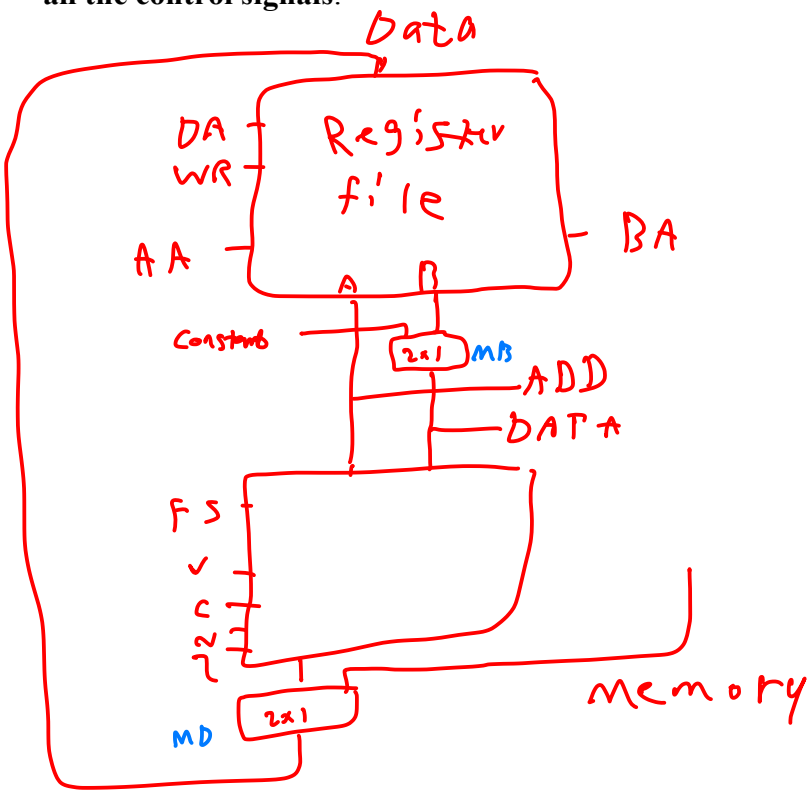
Number of chips	Size of address bus
$2^3 = 8$	$2^{10} \rightarrow 10$

ii. Assuming registers are 8-bits width, and R1 = BE, R2 = 6D, R3 = 9F, R4 = FF, where numbers are represented in HEX and in 2's complement. If the instruction R3 <-- R1 – R2 is executed? **(Show the steps of computation).** What is the content of R3 in Hex and what are the value of flags overflow "V", zero "Z", carry "C', and negative "N"?

R1 = 1011 1110
-R2 = 1001 0011
R3 = 0101 0001

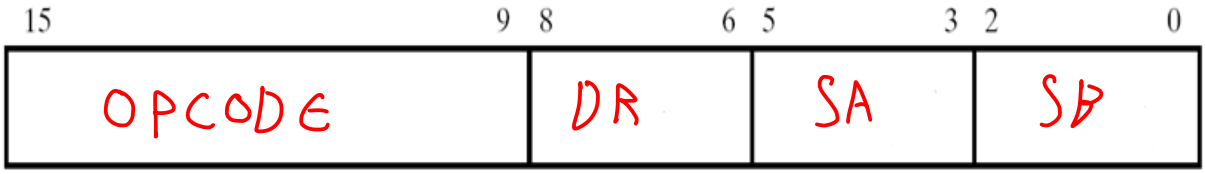
R3	V	Z	C	N
0101 0001 [5 1] Hex	1	0	1	0

iii. Give the block diagram of a datapath, that contains a register file (16 × *n*) and a function unit, showing **all the control signals**.

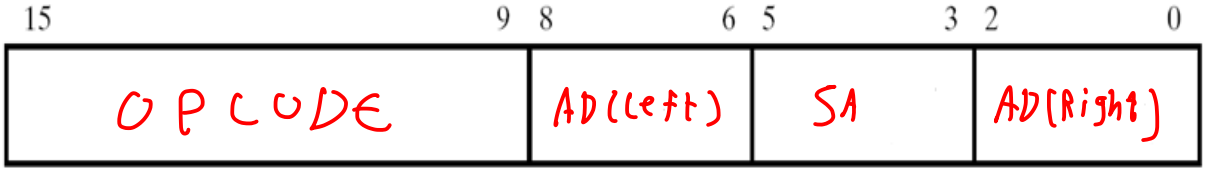


iv. Write the instruction format for

a) **Register** type instructions

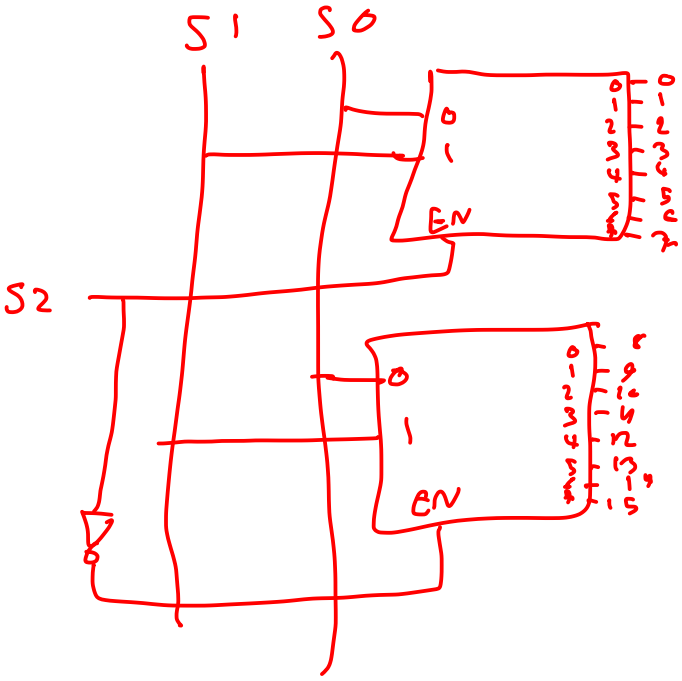


b) **Jump and branch** type instructions

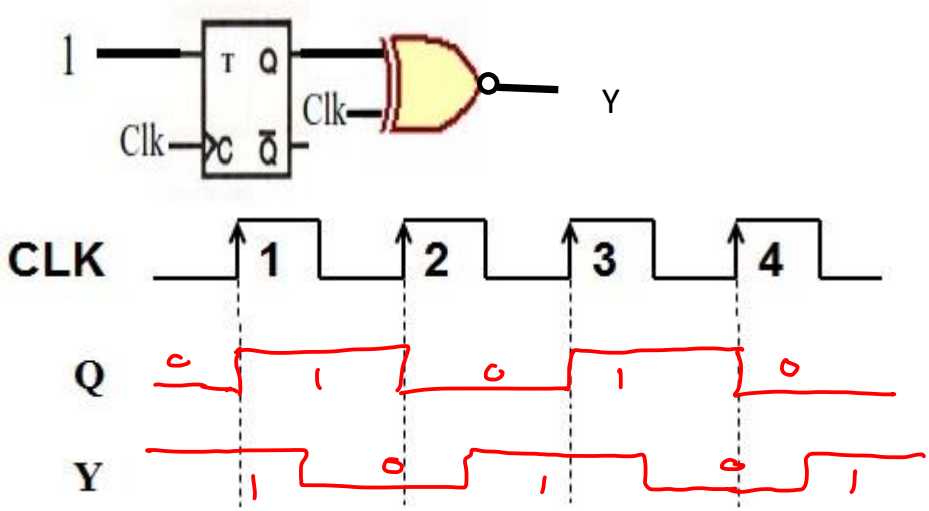


Question 3 (6 Marks: 2+2+2)

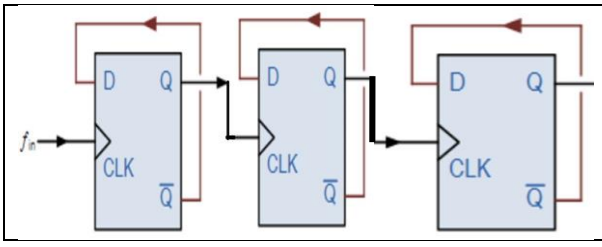
(a) Construct a **4x16** decoder from **two 3x8** decoders with 1-enable.



(b) For the given diagram below, what will be the output waveform at Q and Y (assuming that initially Q = 0)?



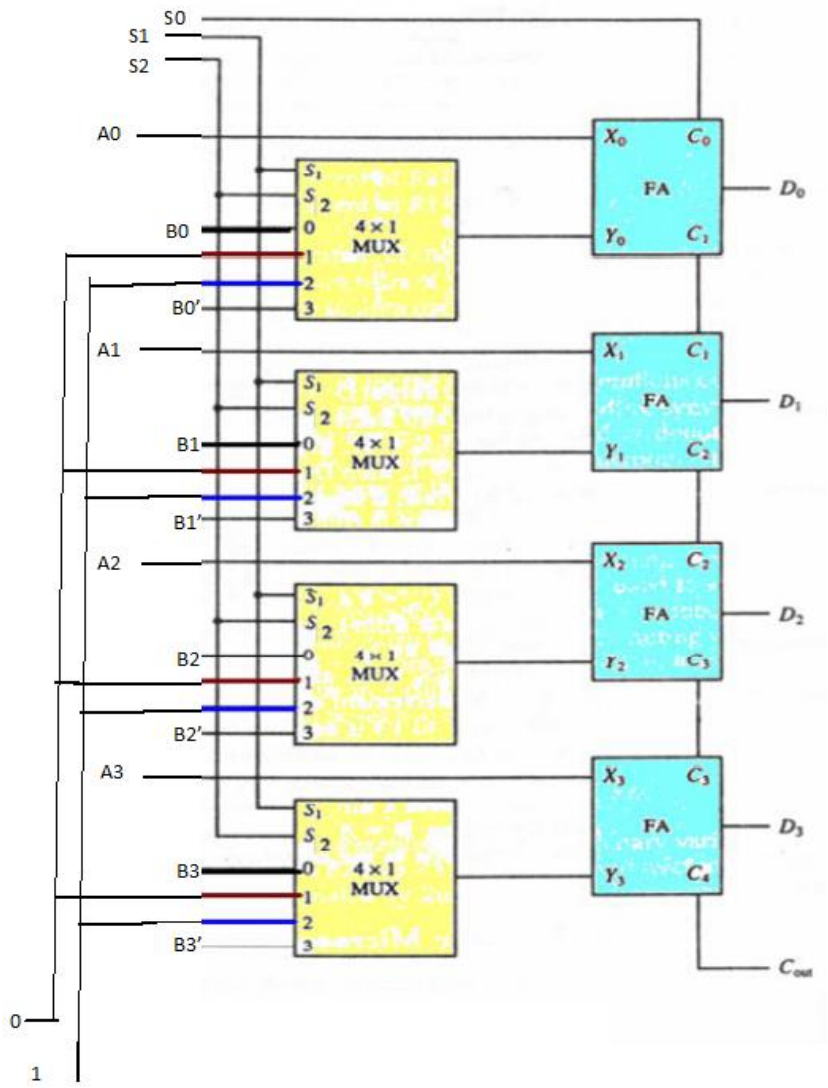
(c) Determine the output frequency (f_{out}) for a frequency division circuit that contains 3 flip-flops with an input clock frequency of 24.48 MHz



	Output Frequency
f_{out}	① $= \frac{24.48}{2} = 12.24$ ② $= \frac{12.24}{2} = 6.12$ ③ $= \frac{6.12}{2} = 3.06$ $f_{out} = 3.06$ MHz

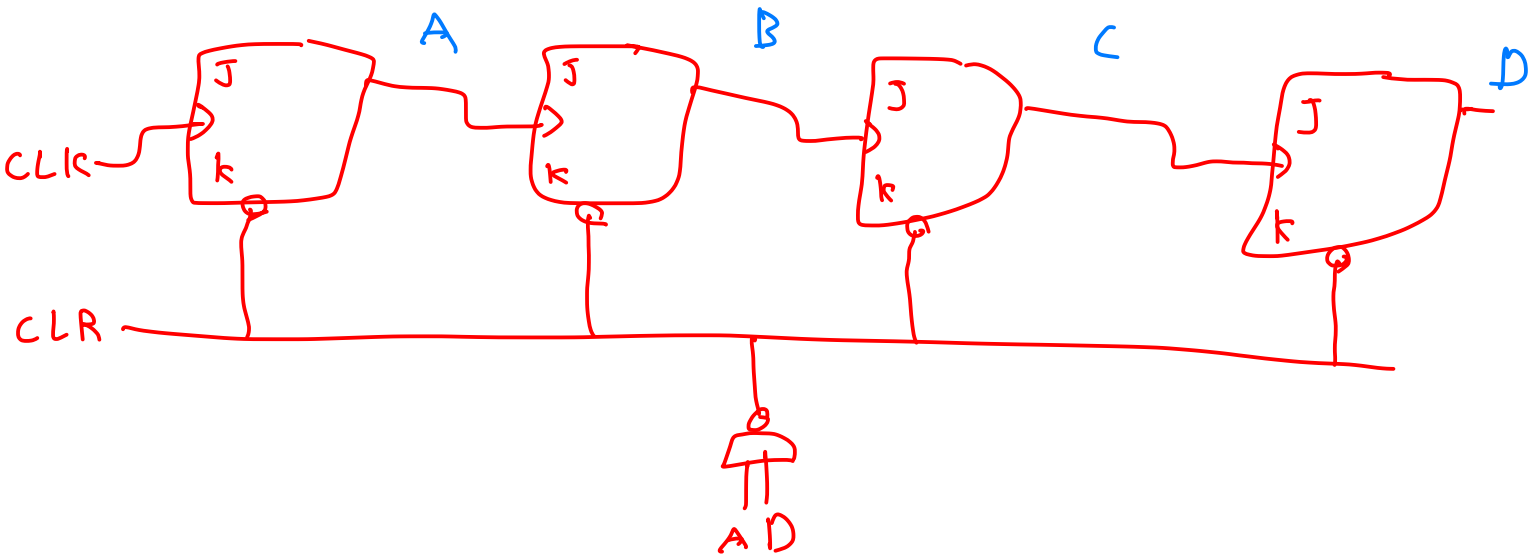
Question 4 (6 Marks: 4+2)

(a) Consider the following arithmetic unit. Write the function Y and D in terms of A and B for each of the three bit selections.



S2	S1	S0	Y =	D =
0	0	0	B	A + B
0	0	1	B	A + B + 1
0	1	0	0000	A
0	1	1	0000	A + 1
1	0	0	1111 (-1)	A - 1
1	0	1	1111 (-1)	A
1	1	0	B'	A + B'
1	1	1	B'	A + B' + 1

(b) Construct an asynchronous BCD counter.



Question 5 (6 Marks: 3+3)

Consider the datapath below described by table 8.5.

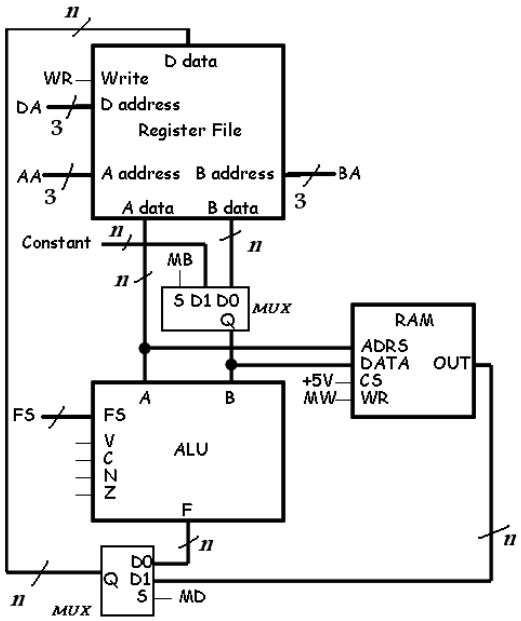


TABLE 8-5
Encoding of Control Word for the Datapath

DA, AA, BA		MB		FS		MD		RW	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code
*R*0	000	Register	0	$F = A$	0000	Function	0	No Write	0
*R*1	001	Constant	1	$F = A + 1$	0001	Data in	1	Write	1
*R*2	010			$F = A + B$	0010				
*R*3	011			$F = A + B + 1$	0011				
*R*4	100			$F = A + \overline{B}$	0100				
*R*5	101			$F = A + \overline{B} + 1$	0101				
*R*6	110			$F = A - 1$	0110				
*R*7	111			$F = A$	0111				
				$F = A \wedge B$	1000				
				$F = A \vee B$	1001				
				$F = A \oplus B$	1010				
				$F = \overline{A}$	1011				
				$F = B$	1100				
				$F = \text{sr } B$	1101				
				$F = \text{sl } B$	1110				

Fill the required information in the table below to perform the following instructions assuming that the registers are of 8 bits, and their initial signed 2's complement values (in HEX) were, R0= 0E, R1 = 09, R2 = 2E, R3 = 2F, data in memory (in HEX) are all shown in the table. The required information is:

a. The generated control signals (**DA, AA, BA, WR, MB, MD, MW, FS**) on the diagram to perform the instruction.

b. The **contents of memory and registers** after executing following 6 instructions.

R0 ← M[R1]
R0 ← R0 + 1
R1 ← M[R3]
R1 ← R0 + R1
M[R2] ← R1
M[R3] ← R0

Address	memory
09	**20**
0A	**A3**
0B	**21**
2E	**34**
2F	**A4**
30	**71**

Microoprations	**DA**	**BA**	**AA**	**WR**	**MB**	**MD**	**MW**	**FS**
R0 ← M[R1]	000	xxx	001	1	0	1	0	xxxx
R0 ← R0 + 1	000	xxx	000	1	0	0	0	0001
R1 ← M[R3]	001	xxx	011	1	0	1	0	xxxx
R1 ← R0 + R1	001	001	000	1	0	0	0	0001
M[R2] ← R1	xxx	001	010	0	0	x	1	xxxx
M[R3] ← R0	xxx	000	011	0	0	x	1	xxxx

The contents of Registers and Memory are as shown:

R0= 21

R1 = C5

R2 = 2E

R3 = 2F

address	memory
09	**20**
0A	**A3**
0B	**21**
2E	C5
2F	21
30	**71**

Question 6 (6 Marks: 3+3)

(a)

TABLE 8-8
Instruction Specifications for the Simple Computer

Instruction	Opcode	Mnemonic	Format	Description	Status Bits
Move A	0000000	MOVA	RD, RA	R[DR] ← R[SA]*	N, Z
Increment	0000001	INC	RD, RA	R[DR] ← R[SA] + 1*	N, Z
Add	0000010	ADD	RD, RA, RB	R[DR] ← R[SA] + R[SB]*	N, Z
Subtract	0000101	SUB	RD, RA, RB	R[DR] ← R[SA] − R[SB]*	N, Z
Decrement	0000110	DEC	RD, RA	R[DR] ← R[SA] − 1*	N, Z
AND	0001000	AND	RD, RA, RB	R[DR] ← R[SA] ∧ R[SB]*	N, Z
OR	0001001	OR	RD, RA, RB	R[DR] ← R[SA] ∨ R[SB]*	N, Z
Exclusive OR	0001010	XOR	RD, RA, RB	R[DR] ← R[SA] ⊕ R[SB]*	N, Z
NOT	0001011	NOT	RD, RA	R[DR] ← $\overline{R[SA]}$*	N, Z
Move B	0001100	MOVB	RD, RB	R[DR] ← R[SB]*	
Shift Right	0001101	SHR	RD, RB	R[DR] ← sr R[SB]*	
Shift Left	0001110	SHL	RD, RB	R[DR] ← sl R[SB]*	
Load Immediate	1001100	LDI	RD, OP	R[DR] ← zf OP*	
Add Immediate	1000010	ADI	RD, RA, OP	R[DR] ← R[SA] + zf OP*	N, Z
Load	0010000	LD	RD, RA	R[DR] ← M[SA]*	
Store	0100000	ST	RA, RB	M[SA] ← R[SB]*	
Branch on Zero	1100000	BRZ	RA, AD	if (R[SA] = 0) PC ← PC + se AD, if (R[SA] ≠ 0) PC ← PC + 1	N, Z
Branch on Negative	1100001	BRN	RA, AD	if (R[SA] < 0) PC ← PC + se AD, if (R[SA] ≥ 0) PC ← PC + 1	N, Z
Jump	1110000	JMP	RA	PC ← R[SA]*	

* For all of these instructions, PC ← PC + 1 is also executed to prepare for the next cycle.

Consider Table 8-8 containing instruction specification for a simple computer. Translate the following instructions into 16-bit binary machine codes. (**NB:** use 0 for don't care)

Instruction	**Binary machine code**			
	Opcode	**RD/AD(left)**	**RA**	**RB/OP/AD(Right)**
LDI R5, 5	1001100	101	000	101
SUB R3, R5, R1	0000101	011	101	001
ST (R6), R1	0100000	000	110	001
SHR R0, R2	0001101	000	000	010
BRZ R5 AD (AD = 101 011)	1100000	101	101	011
JMP R6	1110000	000	110	000

(b) Assume that an array A with 4 elements is located at the address 200 in data memory containing values shown in the figure. Write an assembly language program to calculate the summation of all elements in the array and store the average in variable S is located at the address 210. (**NB:** use only register R0 as address register).

address	memory
…	…
200	**2**
201	**83**
202	-21
203	98
…	…
210	…

RTL

```
R0 ← 200
R1 ← M[R0]
R0 ← R0+1
R2 ← M[R0]
R0 ← R0+1
R3 ← M[R0]
R0 ← R0+1
R4 ← M[R0]
R2 ← R1+R2
R3 ← R2+R3
R4 ← R3+R4
R0 ← R0+7
M[R0] ← R4
```

Assembly

```
LDI R0,200
LD R1,(R0)
Inc R0,R0
LD R2,(R0)
Inc R0,R0
LD R3,(R0)
Inc R0,R0
LD R4,(R0)
ADD R2,R1,R2
ADD R3,R2,R3
ADD R4,R3,R4
ADI R0,R0,7
ST (R0),R4
```

THE END