Solution

University of Engineering & Technology, Peshawar

Programme: BSc Computer Systems Engineering

Semester: 3rd Semester

Paper: Digital Logic Design **Date: 10th March, 2021**

Exam Type: Final Term

Allowed Time: 180 Minutes	**Max Marks: 50**
Student's Name:	
Student's Registration:	

Instructions:

1- This exam is OPEN books/notes/Internet.
2- Sharing of books, notes and other materials during this exam is not permitted.
3- Answer ALL questions on the question sheet.
4- There are 3 problems (and 9 questions) in total. Some questions are harder than others. Answer the easy ones first to maximize your score.
5- Questions will not be interpreted during the exam.

Problem 1: Decoders and Encoders ………………………………………………………………… (20 pts.)

1(a) ……………………………………………………………… (6 pts.)

The truth table shown below is for a 3-line to 8-line binary Decoder circuit:

Inputs			Outputs							
C	B	A	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

For each of the eight output lines, there is a Boolean expression describing its function. Write the Boolean expressions for output lines D_2, D_6, and D_7.

Solution:

D_2 = C’BA’

D_6 = CBA’

D_7 = CBA

1(b) ………………………………………………………………… (6 pts.)

Design a combinational circuit using a Decoder and OR gates. The circuit accepts a 2-bit number and generates an output binary number equal to the square of the input number.

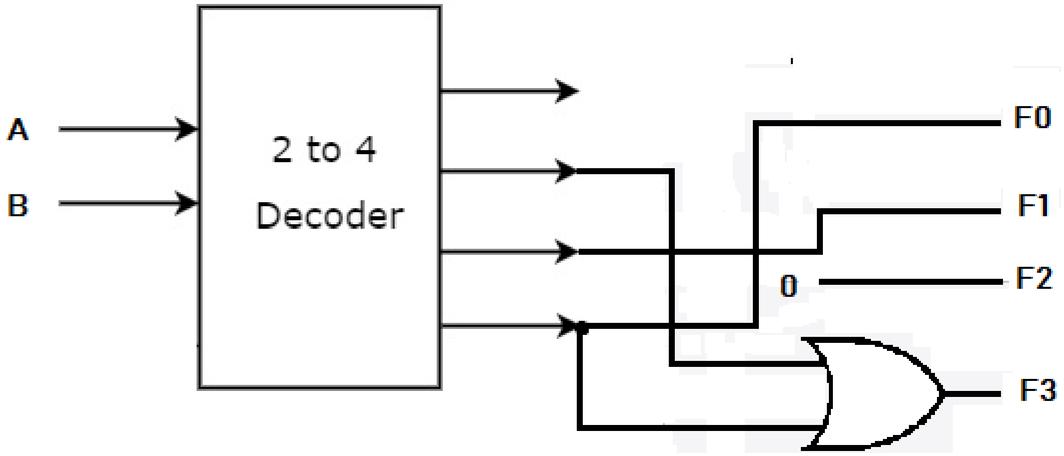
Solution:

Truth Table:

Inputs			Outputs				
Decimal	A	B	F_0	F_1	F_2	F_3	Decimal
0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	1
2	1	0	0	1	0	0	4
3	1	1	1	0	0	1	9

$F_0 = m_3$
$F_1 = m_2$
$F_2 = 0$
$F_3 = \sum_m (1, 3) = m_1 + m_3$

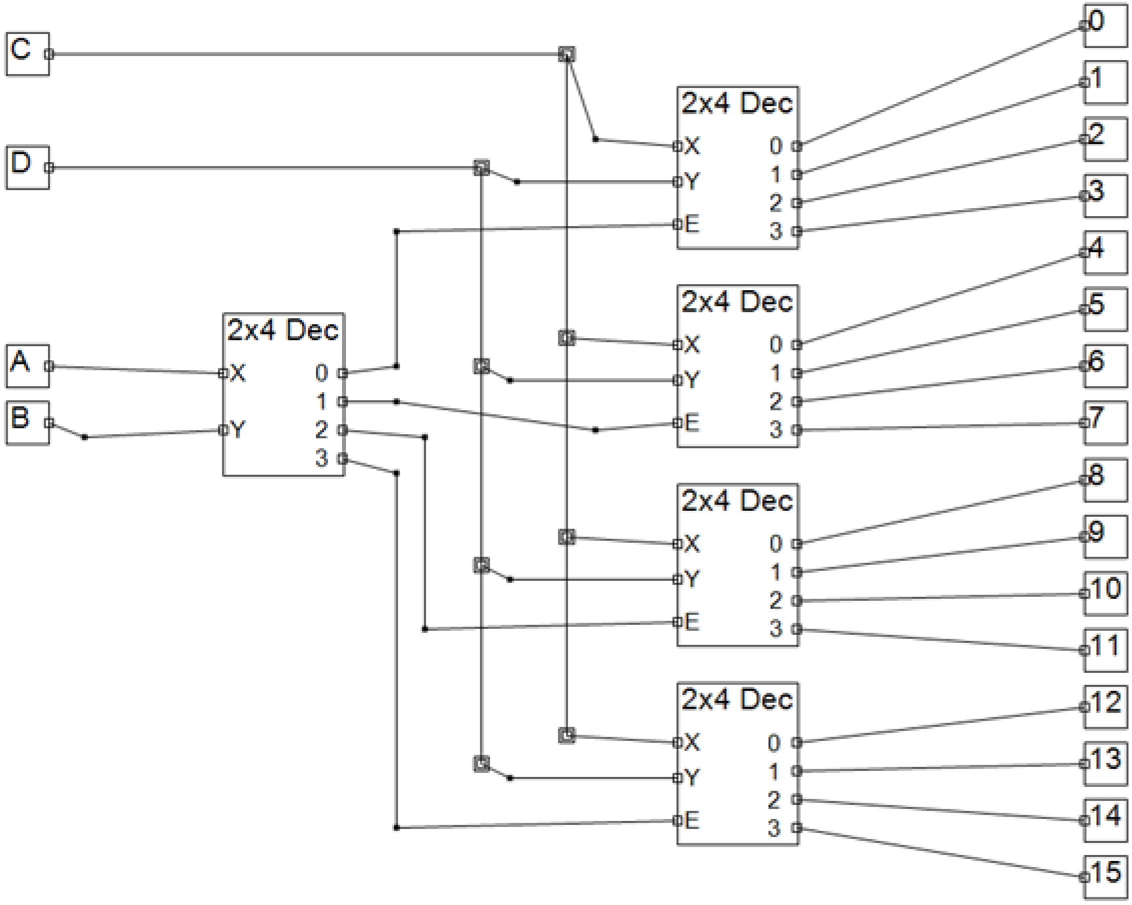
Design/Implementation:



1(c) .... (5 pts.)

Construct a 4x16 Decoder using 2x4 Decoders.

Solution:



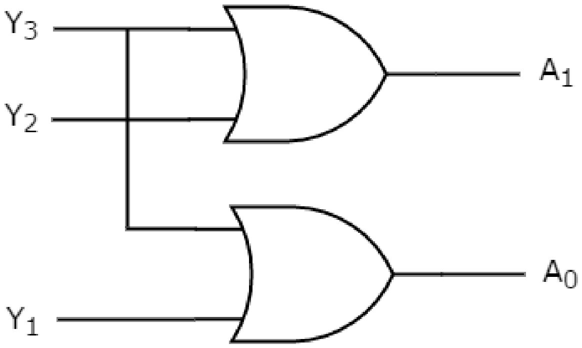
Here, D is the LSB, and A is the MSB. As an example, suppose ABCD = 1100, then the first decoder's output 3 would go high and others low, enabling only bottom-most decoder. The inputs to this decoder is CD = 00, thus its output 0 goes high. In the same manner other inputs can also be analyzed.

1(d) .... (3 pts.)

What is a priority Encoder, and how does it differ from a regular Encoder?

Solution:

The output lines of a digital Encoder generate the binary equivalent of the input line whose value is equal to 1. One of the main disadvantages of regular digital Encoders is that they can generate the wrong output code when there is more than one input present at logic level 1. For example, consider the 4x2 Encoder shown below. If we make inputs Y_1 and Y_2 HIGH at the same time, the resulting output is neither 01 or 10 but will be 11 which is an output binary number that is different to the actual input present. Also, an output code 00 can be generated when all of its inputs are 0 or when input Y_0 is equal to 1.



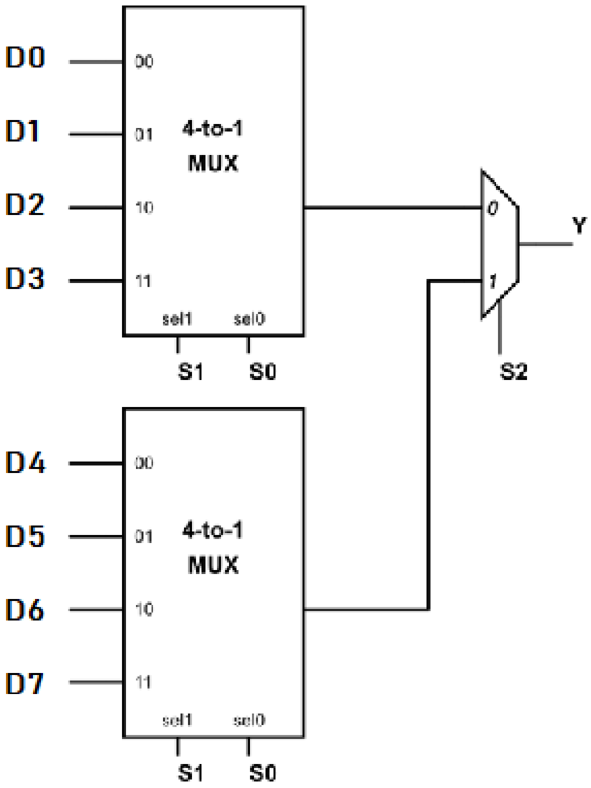
The **priority Encoder** solves the problems mentioned above by allocating a priority level to each input. The priority Encoders output corresponds to the currently active input which has the highest priority. So when an input with a higher priority is present, all other inputs with a lower priority are ignored.

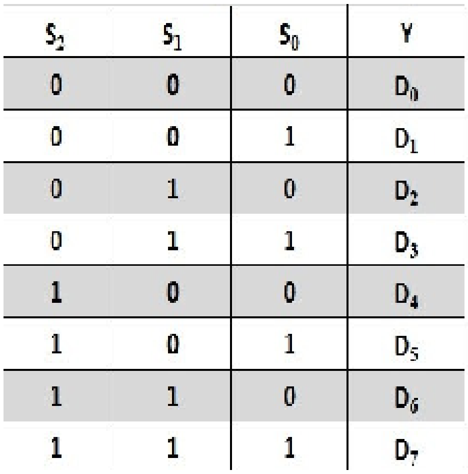
Problem 2: Multiplexers and De-Multiplexers .... (15 pts.)

2(a) .... (5 pts.)

Construct an 8x1 Multiplexer using two 4x1 Multiplexers and a 2x1 Multiplexer.

Solution:





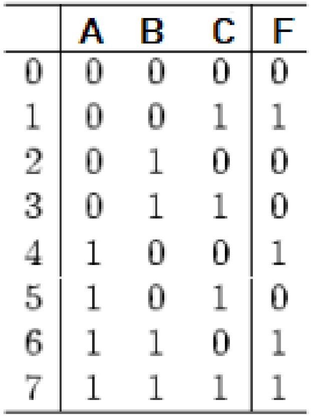
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

2(b) ………………………………………………………………… (5 pts.)

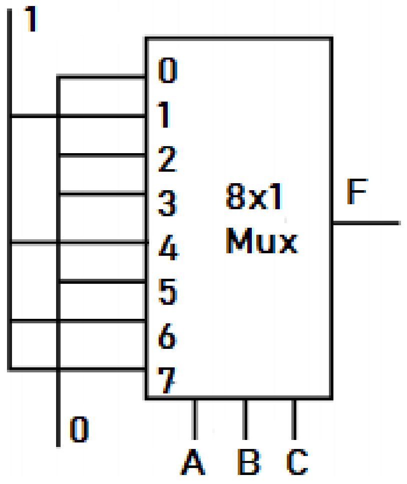
Implement the following function using a Multiplexer.

$$F(A, B, C) = \sum_m (1, 4, 6, 7)$$

Solution:



	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

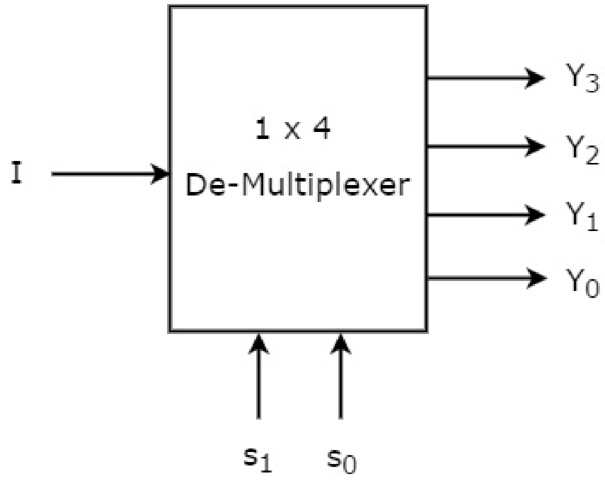


2(c) ……………………………………………………………… (5 pts.)

Design a 1x4 De-Multiplexer using gates. Draw its truth table and logic circuit.

Solution:

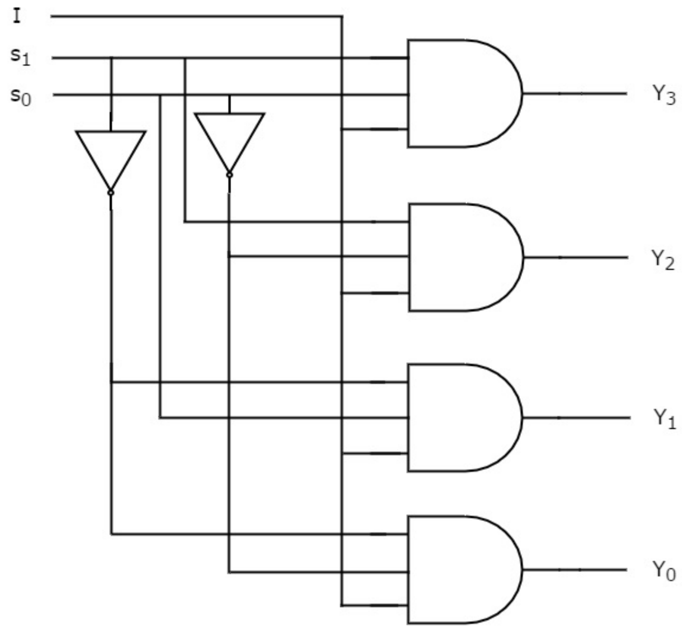
Truth Table:



Selection Inputs		Outputs			
S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

$Y_0 = s_1's_0'I$, $Y_1 = s_1's_0I$, $Y_2 = s_1s_0'I$, $Y_3 = s_1s_0I$

Logic Circuit:



Problem 3: Latches and Flip-Flops .... (15 pts.)

3(a) .... (5 pts.)

What is the difference between a Latch and a Flip-Flop?

Solution:

Latches and flip-flops are the basic elements for storing information. One latch or flip-flop can store one bit of information. The main difference between latches and flip-flops is that for latches, their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately when their inputs change. Flip-flops, on the other hand, have their content change only either at the rising or falling edge of the enable signal. This enable signal is usually the controlling clock signal. After the rising or falling edge of the clock, the flip-flop content remains constant even if the input changes. There are basically four main types of latches and flip-flops: SR, D, JK, and T. The major differences in these types are the number of inputs they have and how they change state.

3(b) ………………………………………………………………. (10 pts.)

The waveforms in Fig 1 are applied to the J, K, and CLK inputs of the positive edge-triggered JK flip-flop shown in Fig 2. Determine the Q output, starting in RESET state.

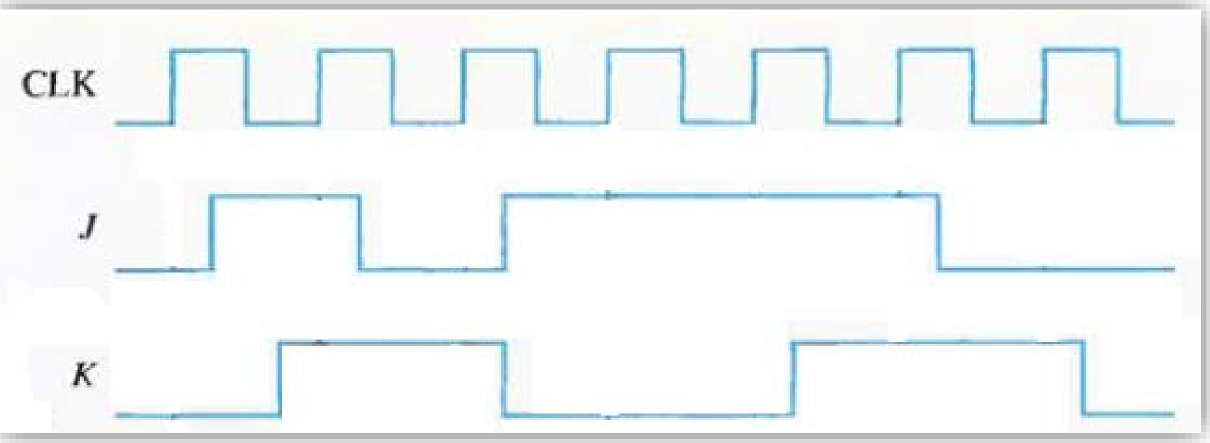
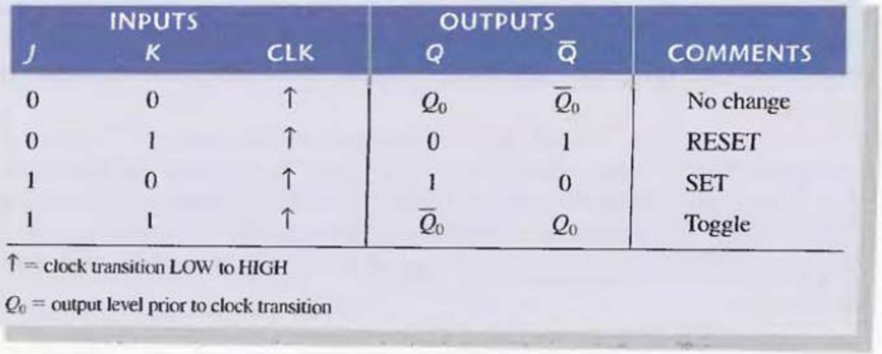


Fig 1. Input waveforms



INPUTS			OUTPUTS		COMMENTS
J	K	CLK	Q	$\overline{Q}$	
0	0	↑	Q_0	$\overline{Q}_0$	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	$\overline{Q}_0$	Q_0	Toggle

↑ = clock transition LOW to HIGH

Q_0 = output level prior to clock transition

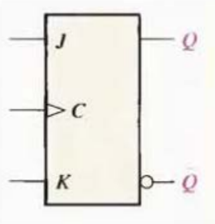
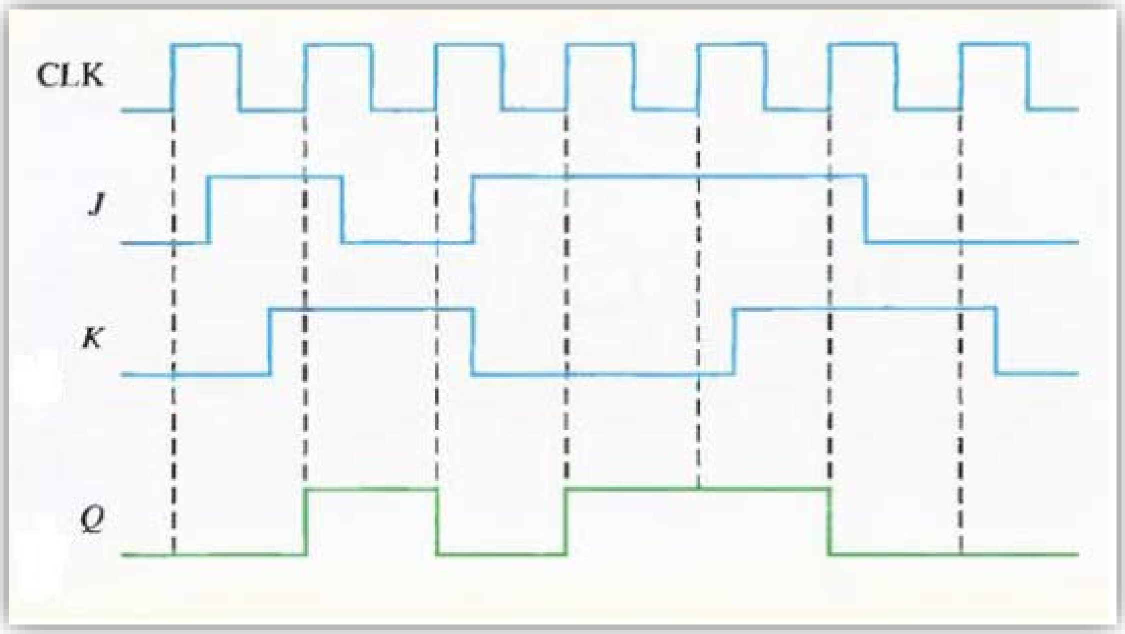


Fig 2. Truth table (left) and logic symbol (right) for a positive edge-triggered JK flip-flop

Solution:



The Q output assumes the state determined by the states of the J and K inputs at the positive-going edge (triggering edge) of the clock pulse. A change in J or K after the triggering edge of the clock has no effect on the output, as shown in Figure