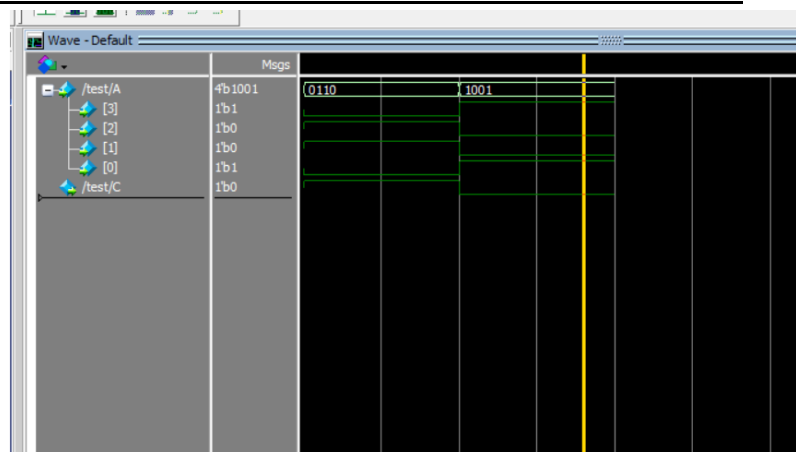


Assignment_1_extra

Mohammed Mogahed

Question_1:

Ln#	
1	<code>module test(</code>
2	<code> input [3:0] A,</code>
3	<code> output C</code>
4	<code>);</code>
5	
6	<code>assign C = (A >= 4'b0010 && A <= 4'b1000) ? 1'b1 : 1'b0;</code>
7	
8	<code>endmodule</code>
9	



Qusetion_2:

Date: Subject:

A

A 0 1 0 1 0 1

B 1 0 0 1

F

1 0 1 0 1 0 1 \Rightarrow out put

A

B

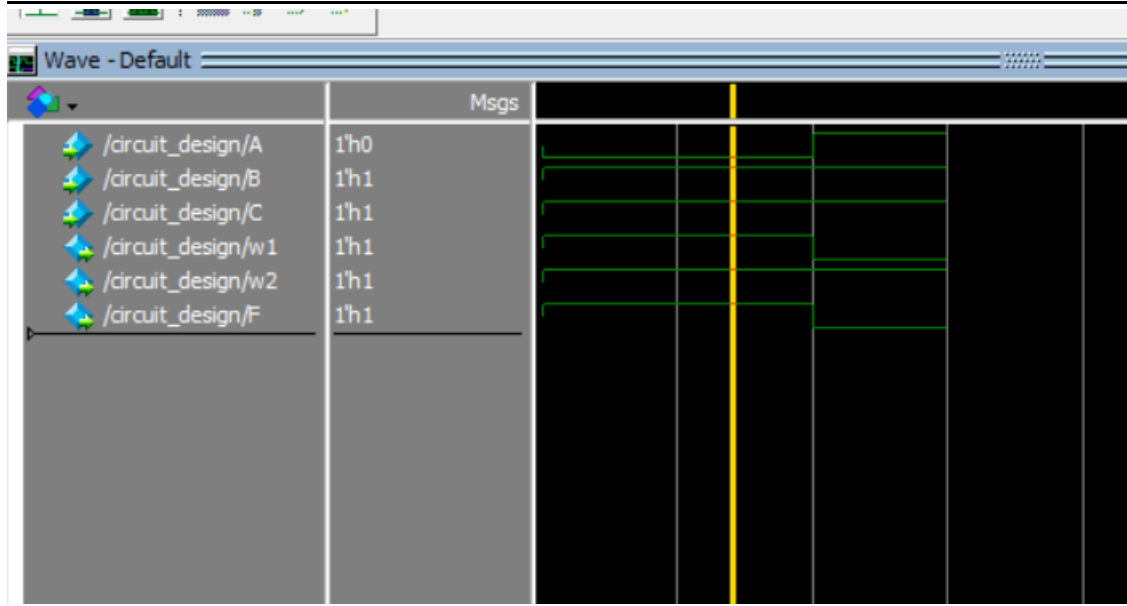
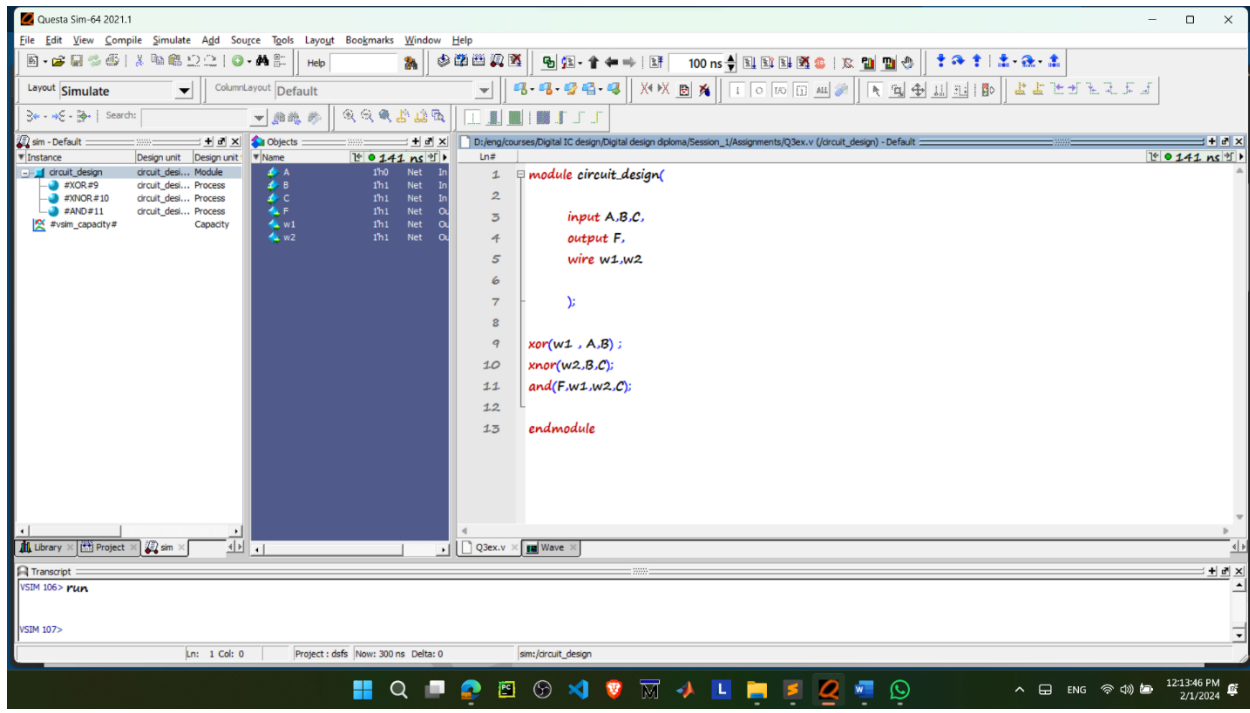
F

$A \oplus 0 = A$

B=1

$A \oplus 1 = \bar{A}$

Question_3:



The output = 1 when A = 0 , B =1 , and C= 1

$$C(A \oplus B)(\overline{B \oplus C})$$

A	B	C	$A \oplus B$	$B \oplus C$	$C(A \oplus B)(\overline{B \oplus C})$
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	1	0	0	0

$$\overline{A}Bc$$

Question_4:

```
Ln#
1  module prime_checker(
2      input [2:0] A,
3      output is_prime
4  );
5
6
7      assign is_prime =
8          (A == 3'b001) ||
9          (A == 3'b011) ||
10         (A == 3'b101) ||
11         (A == 3'b111);
12
13  endmodule
14
```

Wave - Default			
		Msgs	
/prime_checker/A	3'b111	110	111
	[2]		
	[1]		
	[0]		
/prime_checker/is_p...	1'b1		

sim:/prime_checker/A @ 22
No_Data

Question_5:

```
Ln#
1  module alu_1bit(
2      input a, b, Ainvert, Binvert, carryin, operation,
3      output result, carryout
4  );
5
6      wire w1 = (Ainvert == 1'b0) ? a : ~a;
7      wire w2 = (Binvert == 1'b0) ? b : ~b;
8
9      assign result = (operation == 3'b000) ? (w1 & w2) :
10      (operation == 3'b001) ? (w1 | w2) :
11      (w1 + w2 + carryin);
12
13      assign carryout = (operation == 3'b010) ? (w1 + w2 + carryin > 1'b1) : 1'b0;
14
15  endmodule
16
```

Wave - Default		Msgs						
/alu_1bit/a	1'b1							
/alu_1bit/b	1'b1							
/alu_1bit/Ainvert	1'b0							
/alu_1bit/Binvert	1'b0							
/alu_1bit/w1	1'b1							
/alu_1bit/w2	1'b1							
/alu_1bit/carryin	1'b1							
/alu_1bit/operation	1'b1							
/alu_1bit/result	1'b1							
/alu_1bit/carryout	1'b0							