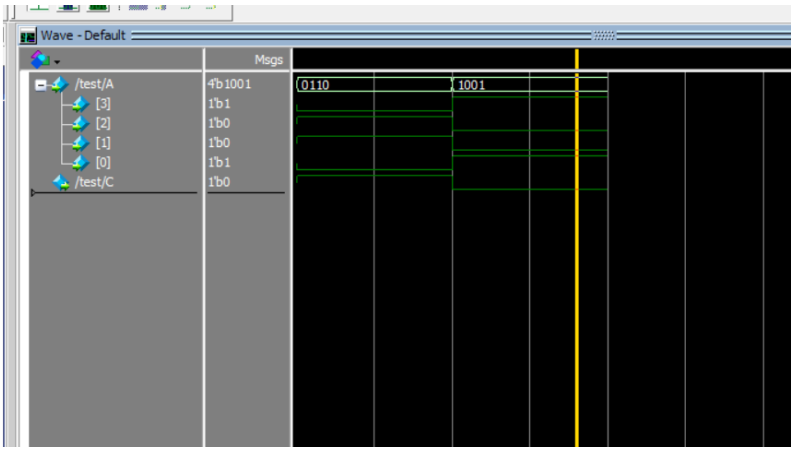
Assignment_1_extra

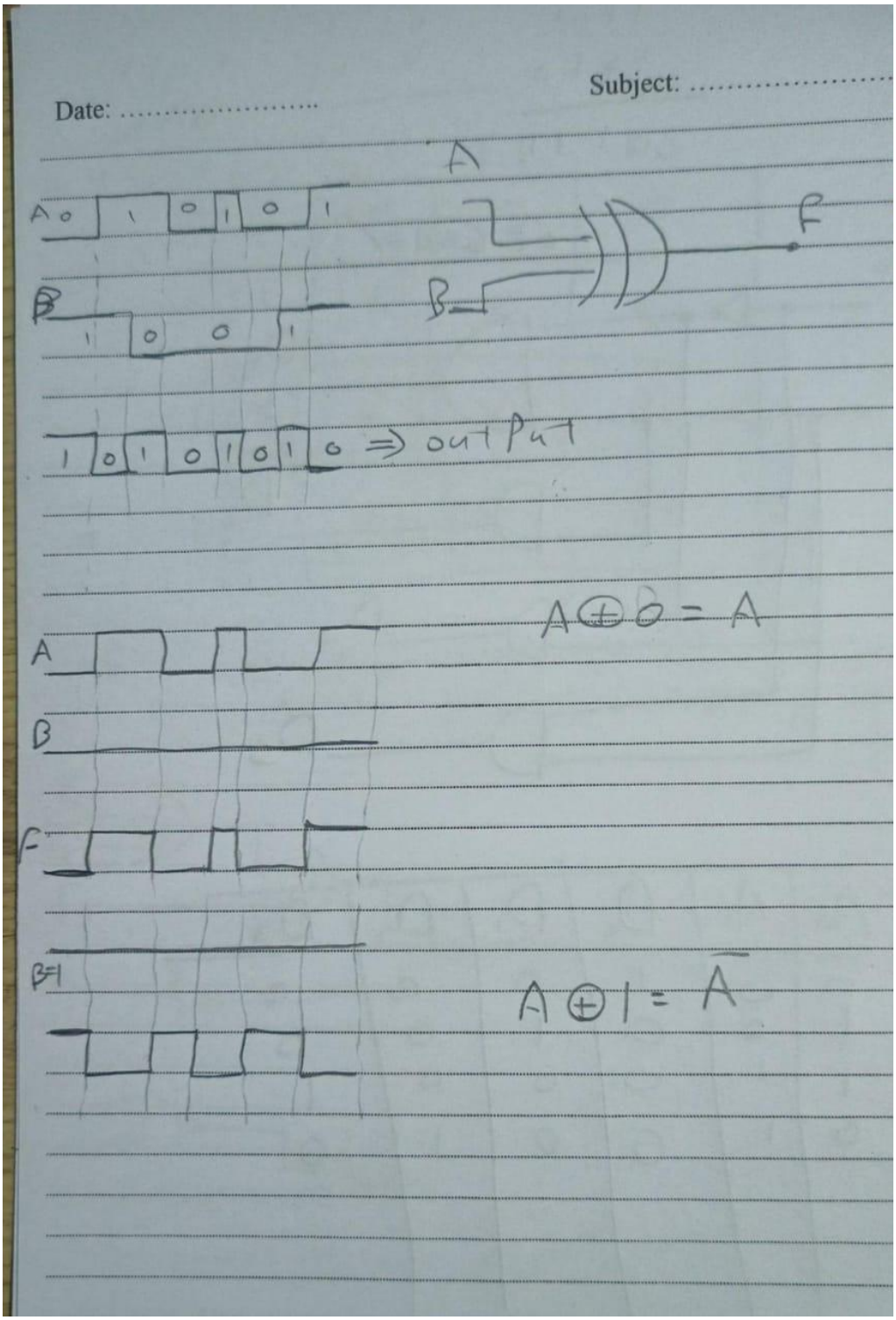
Mohammed Mogahed

Question_1:

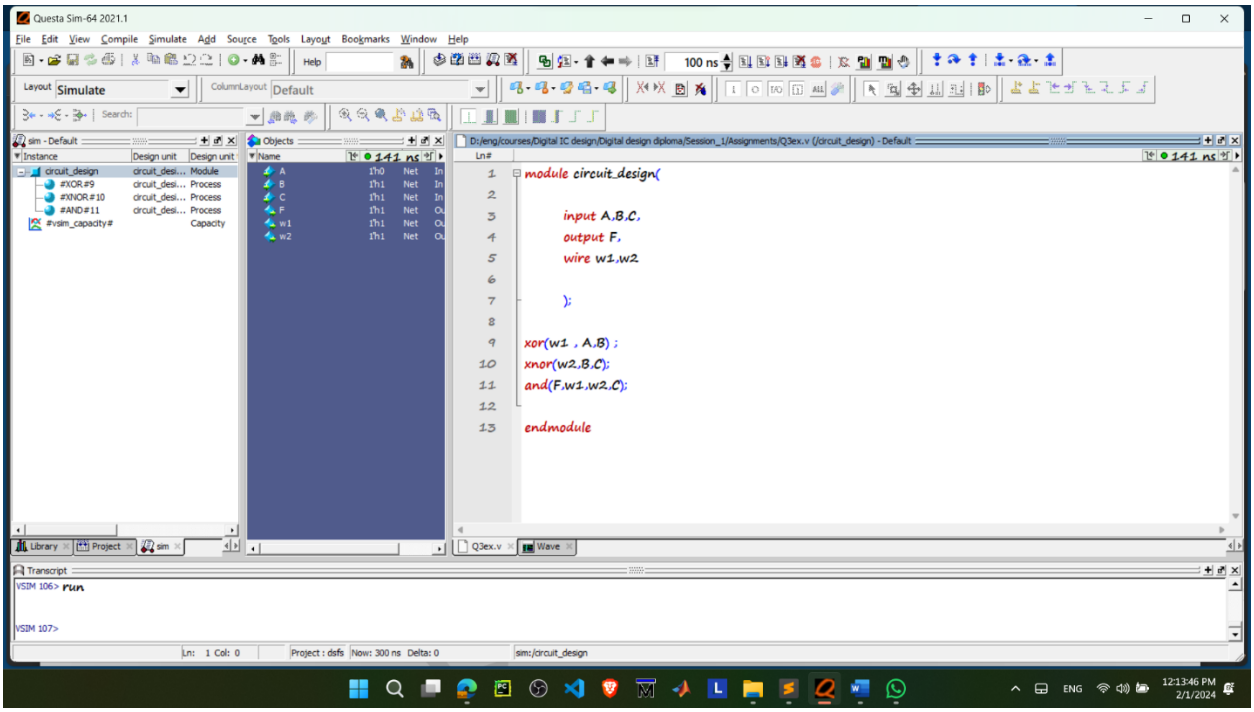
```
Ln#
 1  module test(
 2      input [3:0] A,
 3      output C
 4  );
 5
 6  assign C = (A >= 4'b0010 && A <= 4'b1000) ? 1'b1 : 1'b0;
 7
 8  endmodule
 9
```

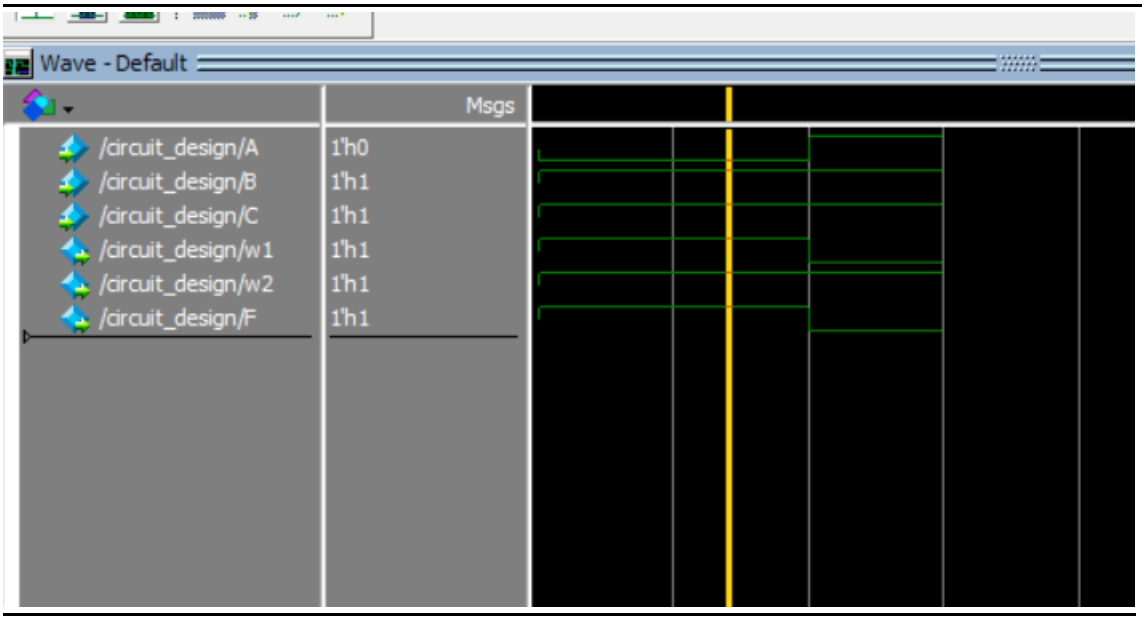


Qusetion_2:

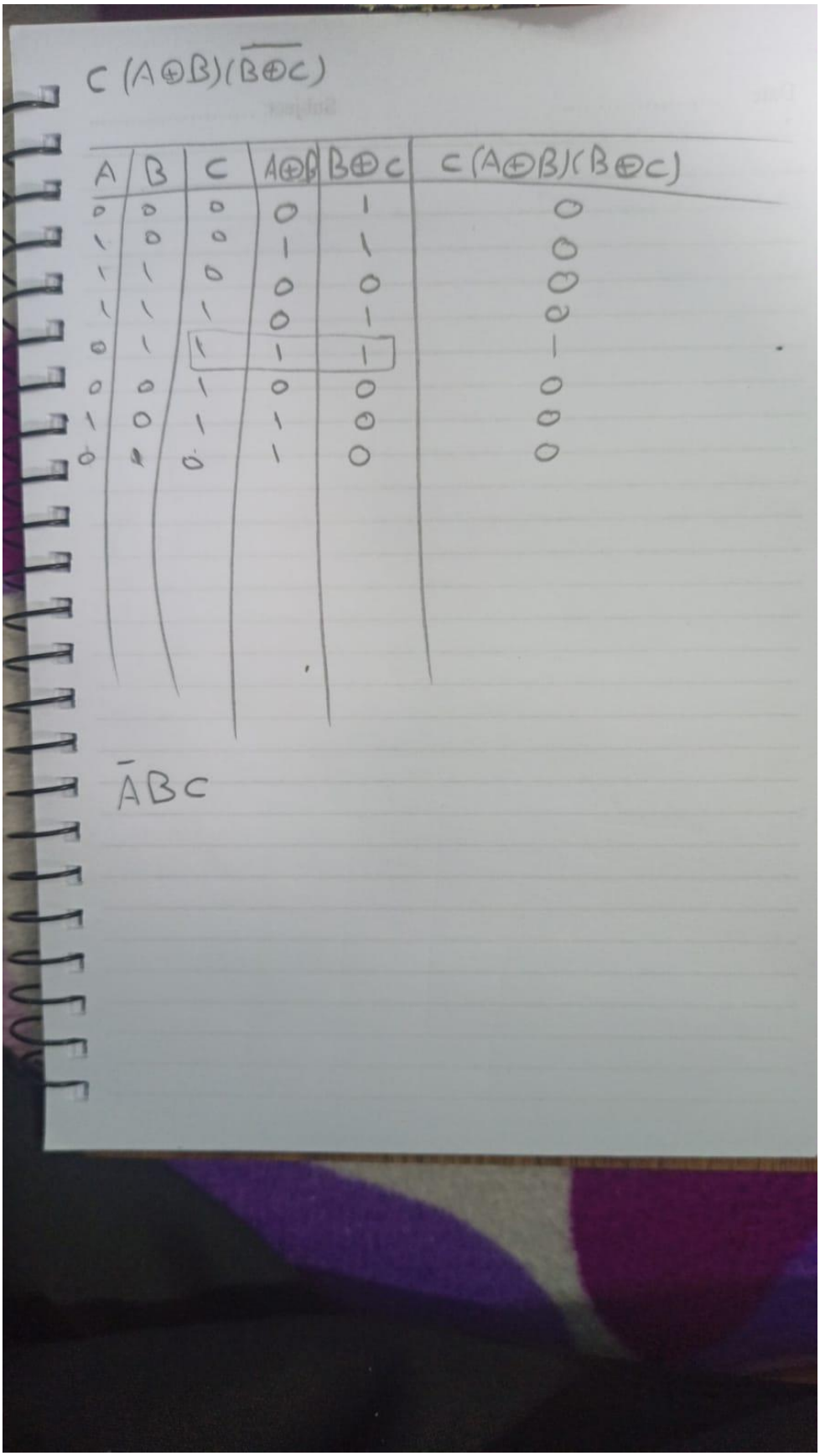

Date: .....
Subject: .....
A
A 0 1 0 1 0 1
B
1 0 0 1
B
F
1 0 1 0 1 0 1 0 ⇒ output
A ⊕ 0 = A
A
B
F
B=1
A ⊕ 1 = Ā

Question_3:





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



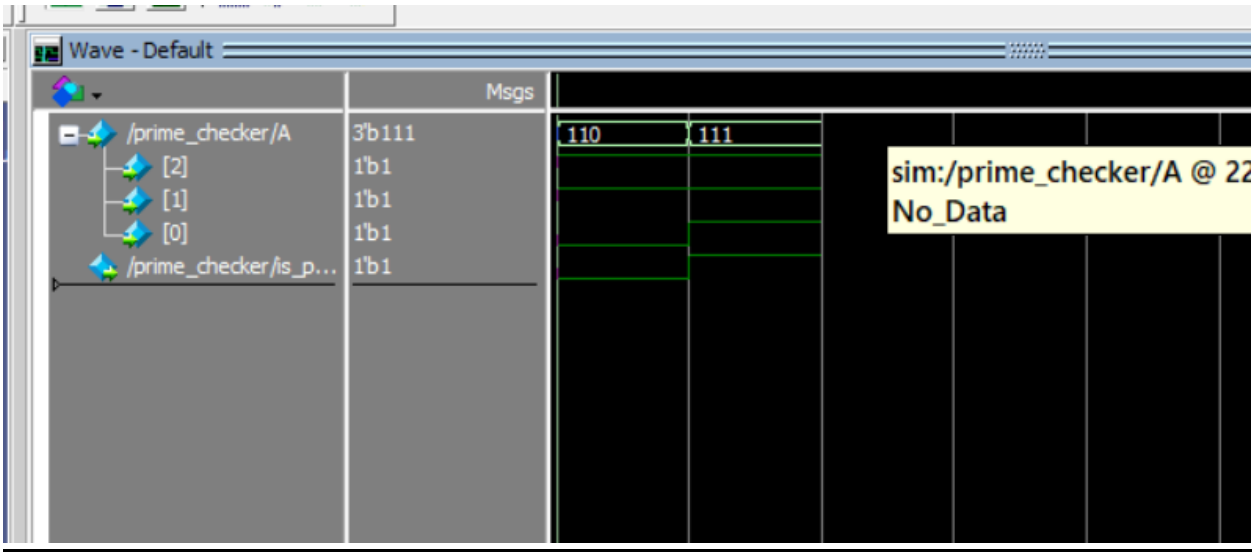
$C(A \oplus B)(\overline{B \oplus C})$

A	B	C	$A \oplus B$	$B \oplus C$	$C(A \oplus B)(B \oplus C)$
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	1	0

$\bar{A}BC$

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
```



Qustion_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
```