

CSE460

Lab Assignment 2

AHMAD AL ASAD

ID: 19101196

Section: 06

PROBLEM 1

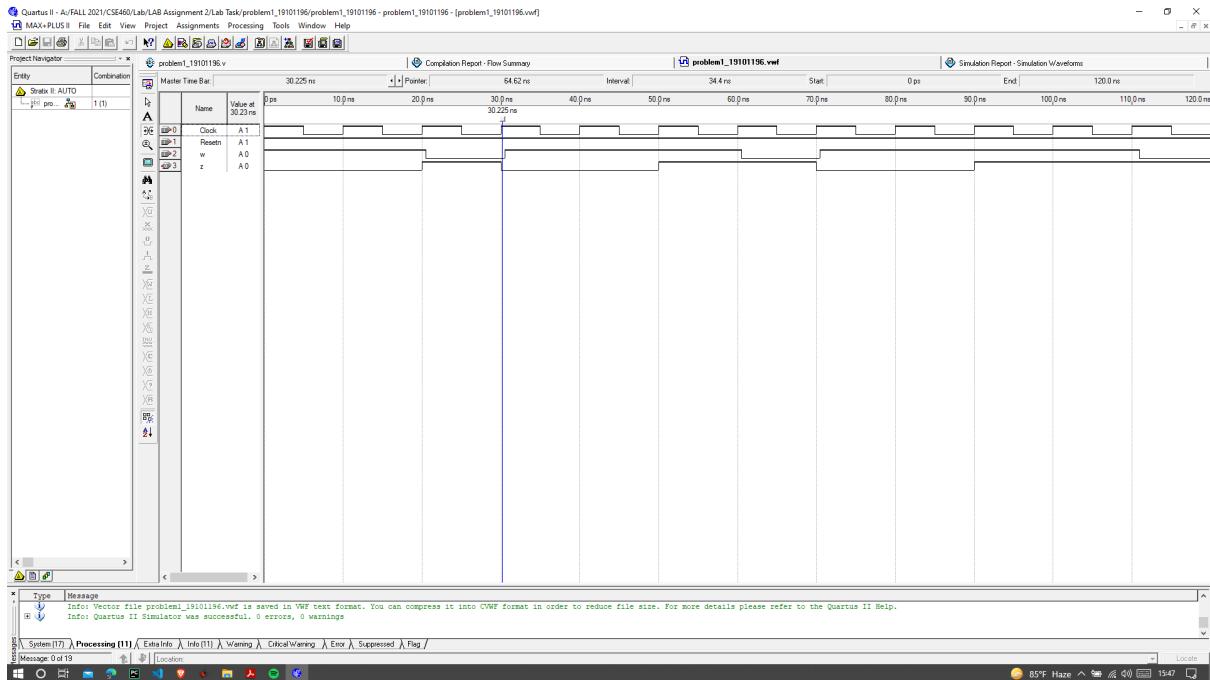
Code:

The screenshot shows the Quartus II MAX+PLUS II interface. The main window displays the Verilog code for 'problem1_19101196.v'.

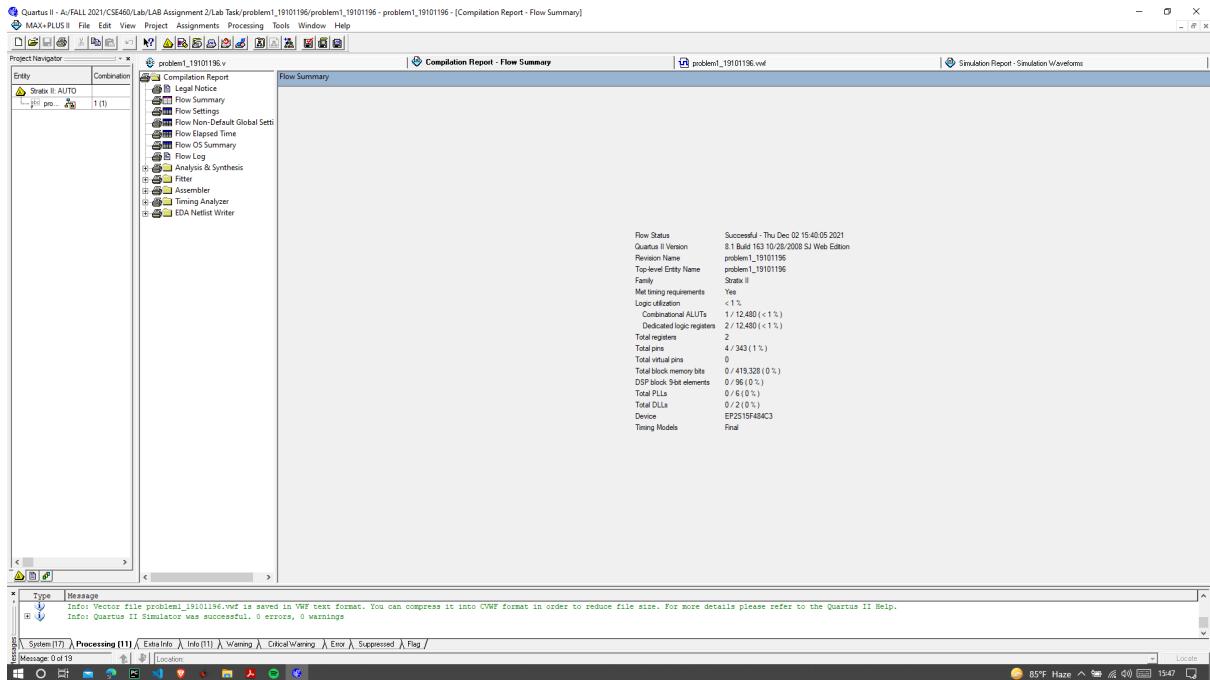
```
1 module problem1_19101196(Clock, Resetn, w, z);
2   input Clock, Resetn, w;
3   output z;
4   reg [2:1] y, Y;
5   parameter [2:1] A = 2'b00, B = 2'b01, C = 2'b11;
6
7   always @(w, y)
8     case (y)
9       A: if (w) Y = B;
10      else Y = A;
11      B: if (w) Y = C;
12      else Y = A;
13      C: if (w) Y = C;
14      else Y = A;
15      default: Y = 2'bxx;
16   endcase
17
18   always @ (posedge Clock)
19     if (Resetn == 0) y <= A;
20     else y <= Y;
21
22   assign z = (y == C);
23
24 endmodule
```

The message window at the bottom indicates that the vector file 'problem1_19101196.vwf' is saved in VWF text format, and the Quartus II Simulator was successful with 0 errors and 0 warnings.

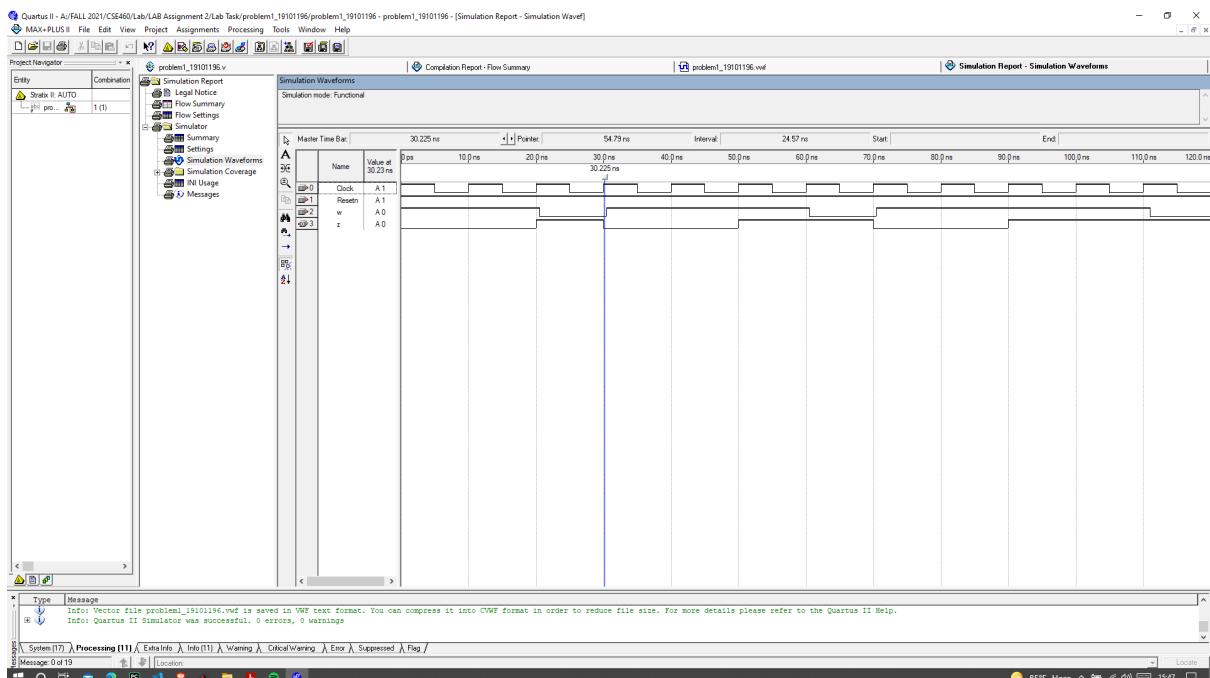
Waveform:



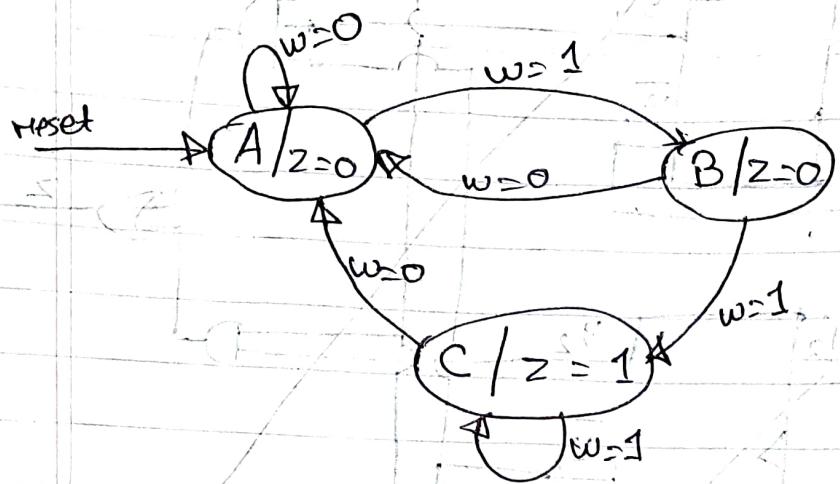
Compilation Report:



Simulation Report:



Problem 1



Let, $A \rightarrow 00$,

$B \rightarrow 01$

$C \rightarrow 11$

Current	Next		Output
$X_2 X_1$	$w=0$	$w=1$	Z
00	00	01	0
01	00	11	0
11	00	11	1

Input w is 1 for 2 clock cycles, 0ns to 10ns and 10ns to 20ns. So, the pattern 11 is seen, hence output z is 1 for the next clock cycle 20ns to 30ns, which is state C(1).

Similarly $w=1$ for 3 clock cycles, 30ns to 40ns, 40ns to 50ns and 50ns to 60ns. The pattern 111 is seen, hence output z is 1 for the next clock cycle 60ns to 70ns, which is in state C(11).

The input overlaps. So, from 70ns to 110ns, 4 clock cycles, there are 1111, which gives output $z=1$ in 90ns to 100ns and 100ns to 110ns and 110ns to 120ns. If input $w=0$, next clock cycle gives output $z=0$, not in state C(11), but in state A(00).

For 2 clock cycles 60ns to 80ns, the input pattern is 01. So, for next clock cycle 80ns to 90ns, output $z=0$.

PROBLEM 2

Code:

The screenshot shows two side-by-side windows of the Quartus II MAX+PLUS II software interface. Both windows display the same Verilog code for a module named `problem2_19101196`.

Top Window Content:

```
1  module problem2_19101196(Clock, Resetn, w, Q, c);
2    input Clock, Resetn;
3    input [2:1] w;
4    output reg [2:1] c;
5    reg [2:1] Y, X;
6    parameter [2:1] A = 2'b00, B = 2'b01, C = 2'b11, D = 2'b10;
7    parameter [2:1] Tk0 = 2'b00, Tk1 = 2'b01, Tk3 = 2'b10;
8
9    always @ (w, Y)
10   case (w)
11     A: case (Y)
12       Tk0:begin
13         Q = 0;
14         c = 00;
15         Y = A;
16       end
17     Tk1:begin
18       Q = 0;
19       c = 00;
20       Y = B;
21     end
22     Tk3:begin
23       Q = 0;
24       c = 00;
25       Y = D;
26     end
27   endcase
28   B: case (w)
29     Tk0:begin
30       Q = 0;
31       c = 00;
32       Y = B;
33     end
34   end
35   Tk1:begin
```

Bottom Window Content:

```
36   Q = 0;
37   c = 00;
38   Y = C;
39 end
40 Tk3:begin
41   Q = 1;
42   c = 00;
43   Y = A;
44 end
45 endcase
46 C: case (w)
47   Tk0:begin
48     Q = 0;
49     c = 00;
50     Y = C;
51   end
52   Tk1:begin
53     Q = 0;
54     c = 00;
55     Y = D;
56   end
57   Tk3:begin
58     Q = 1;
59     c = 10;
60     Y = A;
61   end
62 endcase
63 D: case (w)
64   Tk0:begin
65     Q = 0;
66     c = 00;
67     Y = D;
68   end
69   Tk1:begin
70     Q = 1;
```

Message Windows:

Both windows have a "Messages" panel at the bottom. The top window's message panel shows:

- Type: Message
- Info: Vector file problem2_19101196.vwf is saved in VWF text format. You can compress it into CWF format in order to reduce file size. For more details please refer to the Quartus II Help.
- Info: Quartus II Simulator was successful. 0 errors, 0 warnings

The bottom window's message panel shows:

- Type: Message
- Info: Vector file problem2_19101196.vwf is saved in VWF text format. You can compress it into CWF format in order to reduce file size. For more details please refer to the Quartus II Help.
- Info: Quartus II Simulator was successful. 0 errors, 0 warnings

Quartus II - A-FALL 2021 CS4540/Lab/LAB Assignment 2/Lab Task/problem2_19101196/problem2_19101196 - [problem2_19101196]

MAX+PLUS II File Edit View Project Assignments Processing Tools Windows Help

Project Navigator

Entry Combination

Stratix II AUTO 17(17)

problem2_19101196.v

Completion Report - Flow Summary

problem2_19101196.wf

Simulation Report - Simulation Waveforms

```
52     Tkl:begin
53         Q = 0;
54         C = 00;
55         Y = D;
56     end
57     Tk3:begin
58         Q = 1;
59         c = 10;
60         Y = A;
61     end
62 endcase
63 D: case (w)
64     Tk0:begin
65         Q = 0;
66         c = 00;
67         Y = D;
68     end
69     Tk1:begin
70         Q = 1;
71         c = 00;
72         Y = A;
73     end
74     Tk3:begin
75         Q = 1;
76         c = 11;
77         Y = A;
78     end
79 endcase
80 endmodule
81
82 always @ (posedge Resetn, posedge Clock)
83     if (Resetn == 0) y <= A;
84     else y <= Y;
85
86 endmodule
```

Type Message

Info: Vector file problem2_19101196.wvf is saved in WVF test format. You can compress it into CWVF format in order to reduce file size. For more details please refer to the Quartus II Help.

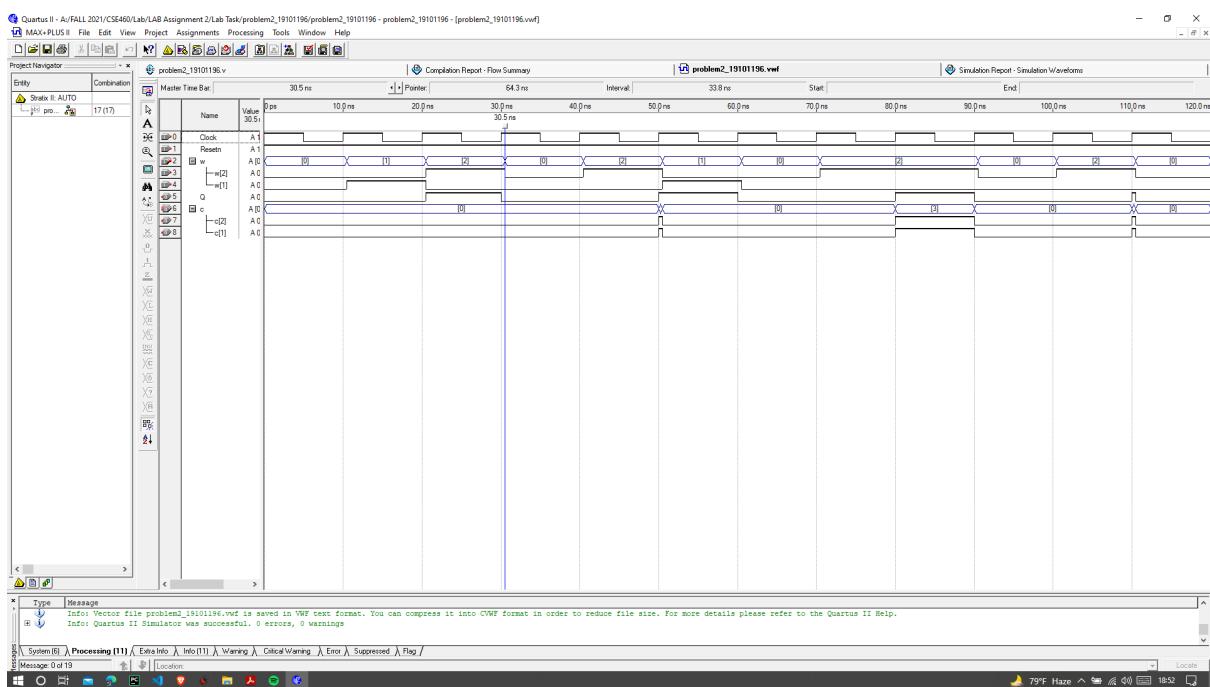
Info: Quartus II Simulator was successful. 0 errors, 0 warnings

System Processing (11) Exit Info (11) Warning (1) Critical Warning (1) Error (1) Suppressed (1) Log (1)

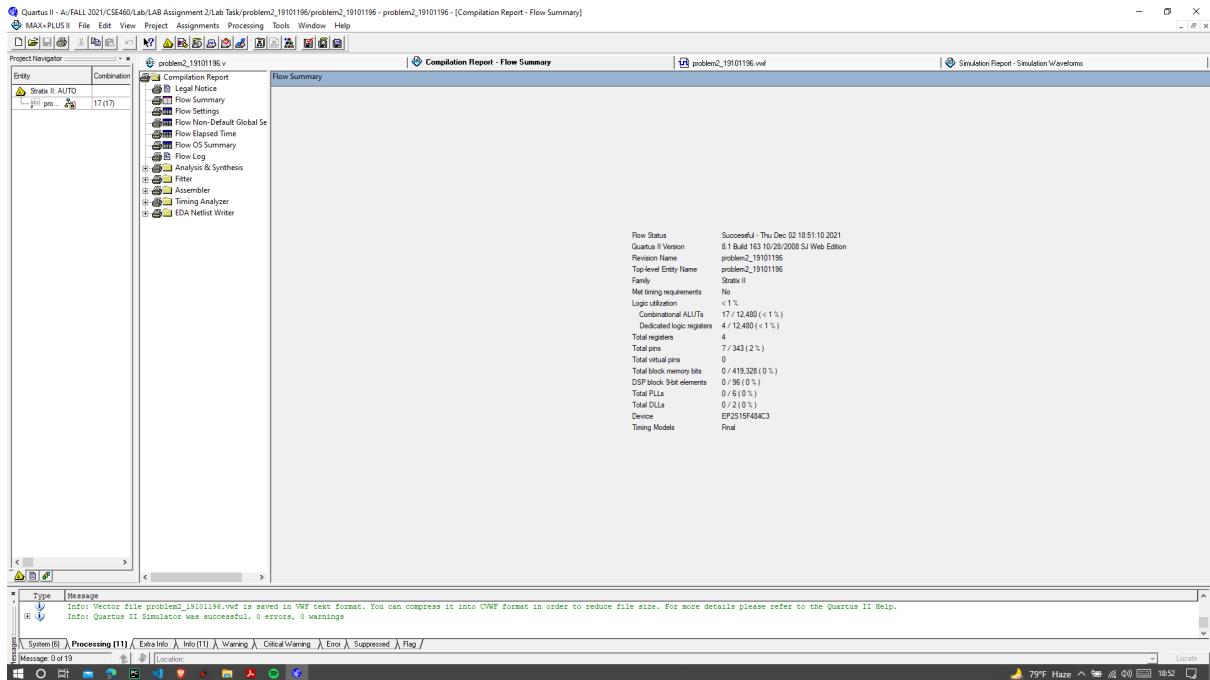
Message 0 of 19

Windows 79°F Haze Location 18:52

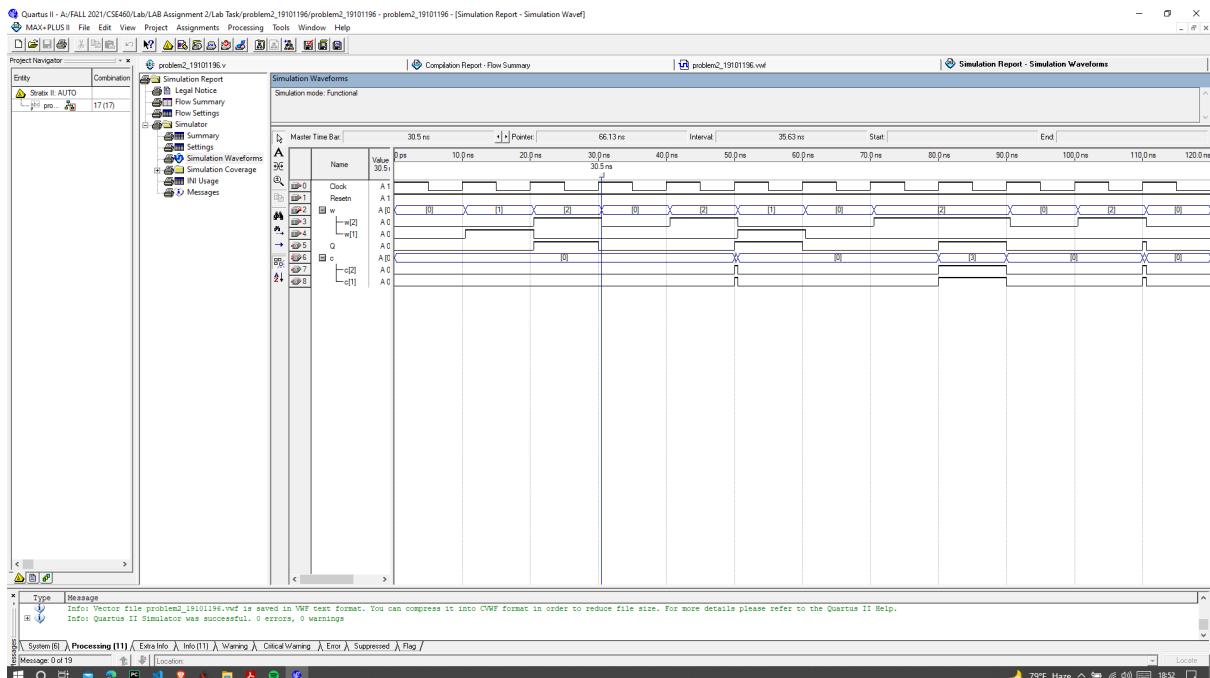
Waveform:



Compilation Report:



Simulation Report:



Problem 2

Final state of D flip flop = Q=000 at time

Let's see steps done from left to right = higher

<u>TK</u>	<u>w</u>	<u>TK(111)</u>	<u>c</u>	at time
000	00	0	000	initial
011	01	10	000	time 1
100	10	2D	011	time 2
111	11	(111) state in or idle, wait in an idle		

$$w=00/Q=0/c=00$$

$$w=01/Q=0/c=00$$

$$w=10/Q=1/c=00$$

$$w=00$$

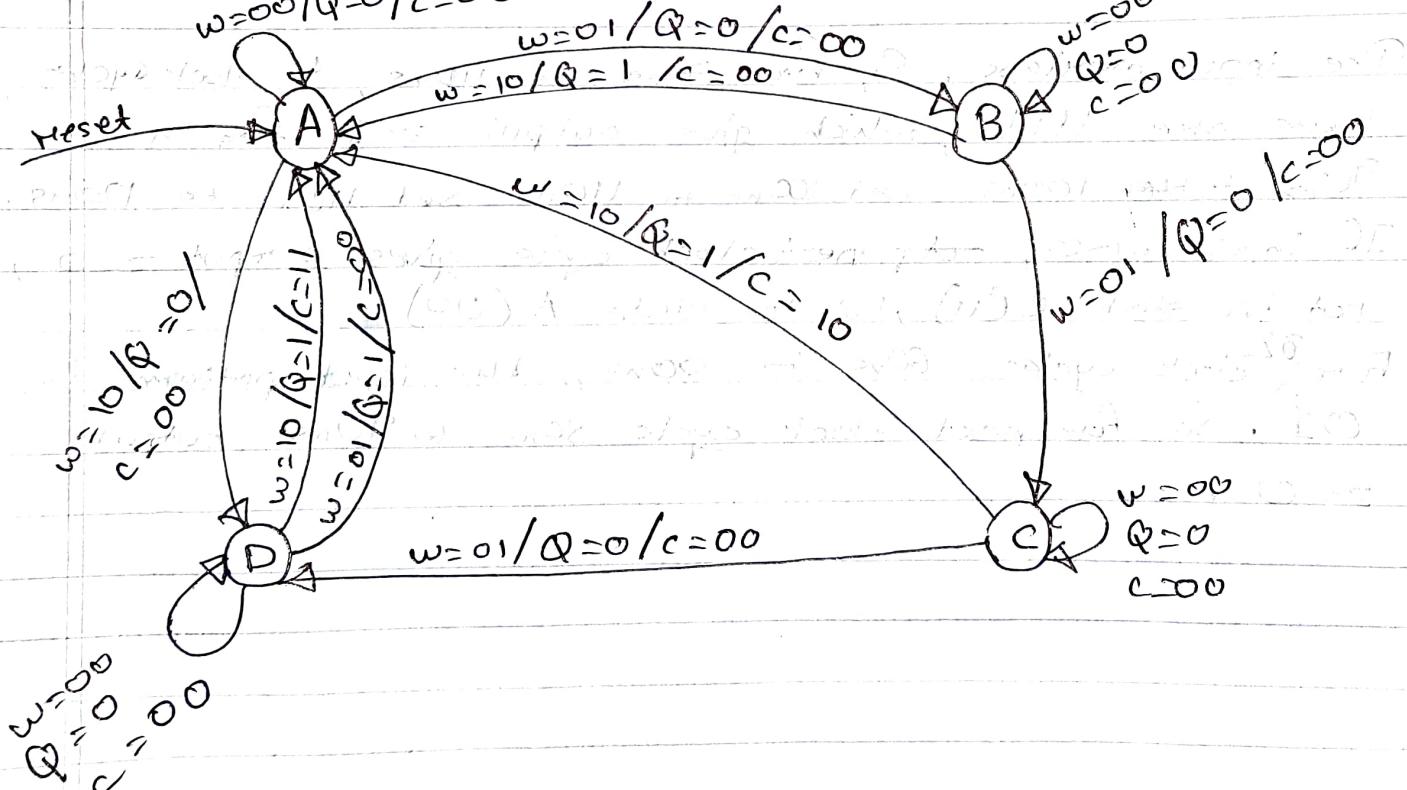
$$Q=0$$

$$c=0$$

$$w=00$$

$$Q=0$$

$$c=0$$



let, $A \rightarrow 00$ has higher bus logic priority, so

$B \Rightarrow 01$

$C \rightarrow 11$ and

$D \rightarrow 10$

Current $Y_2 Y_1$	S _{next} ($Y_2 Y_1$)			output / Q			change/c		
	w=00	w=01	w=10	w=00	w=01	w=10	w=00	w=01	w=10
00	00	01	10	0	0	0	00	00	00
01	01	11	00	0	0	1	00	00	00
10	11	11	00	01	01	1	00	00	10
11	10	00	00	0	1	1	00	00	11

initial state of SR flip-flop is 00

1. If current bus logic priority is 00 then output = 00

The possible input and output combinations are:

<u>Input</u>	<u>Q</u>	<u>Change</u>
1, 1, 1, 1	1	0
1, 1, 1, 3	1	2
1, 1, 3, 3	1	1
1, 3	1	0
3, 1 0 0 0	1	0
3, 3	1	2

From the waveform, we can see $w=00$ for 1st clock cycle, then $w=01$ and then $w=10$. Hence, total $1+3=4$ Tk has been input. So, Q in the 3rd clock cycle, 20ns to 30ns, output $Q=1$ and change is 0, hence $c=00$.

In 7th clock cycle, $w=00$. In 8th, $w=10$ and in 9th, $w=10$. So, total input is $3+3=6$ Tk. Hence, in 9th clock cycle 80ns to 90ns, output $Q=1$ and change is $6-4=2$, which is equivalent to $c=11$.

Similarly, 6th clock cycle, 80ns to 60ns also shows $Q=1$, meaning the vending machine generates output $Q=1$ and Tk 4 or more has been input.

E. normal

There are slight discrepancies in the output during times 50ns to 50.5ns and 110ns to 110.5ns.

The charge shows $c = 11$, although it should be 00.

This is because the input w has been offset by 500ps / 0.5ns to the right. Which is why, from 40.ns to 50.ns and 50.ns to 50.5ns, input is $w = 10$, so the charge shows $c = 11$, meaning GFK input with 2TK change.

edge detection

110

101

011

010 01 1 11

100 00 0

110 00 0

010 00 0

011 00 1

111 00 1

101 00 0

PROBLEM 3

Code:

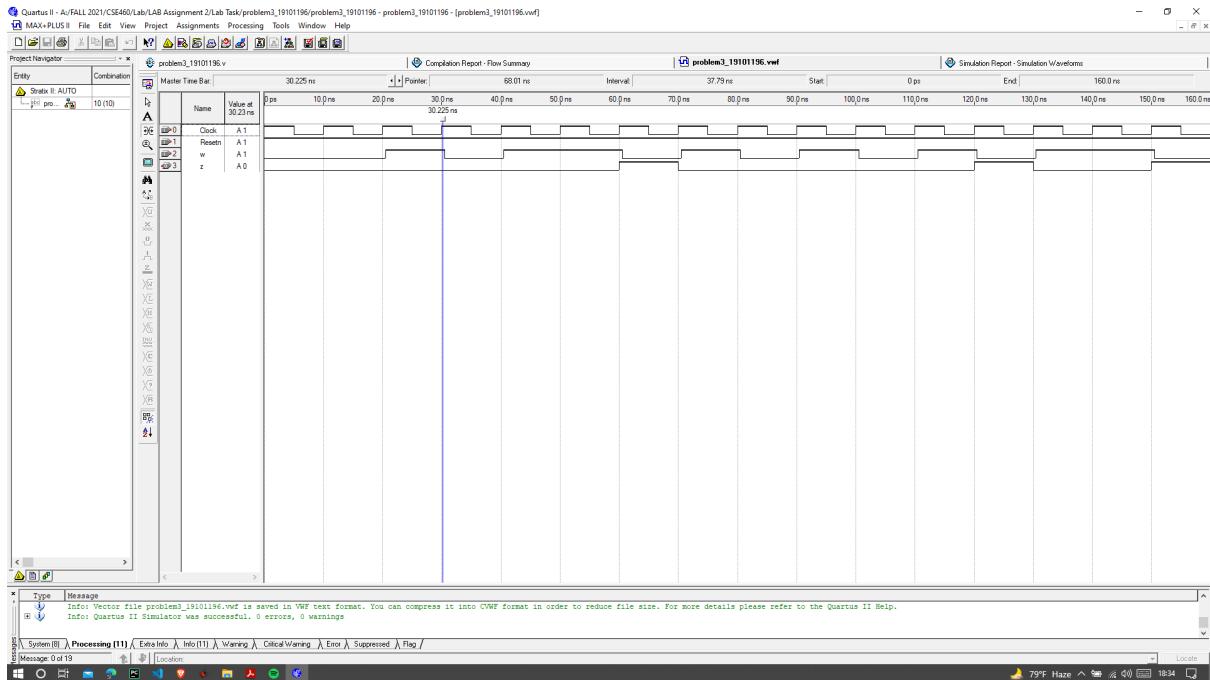
The screenshot shows the Quartus II MAX+PLUS II interface. The main window displays a Verilog code file named 'problem3_19101196.v' with the following content:

```
1 module problem3_19101196(Clock, Resetn, w, z);
2   input Clock, Resetn, w;
3   output z;
4   reg [3:1] y;
5   parameter [3:1] A = 3'b000, B = 3'b001, C = 3'b011, D = 3'b010, E = 3'b110, F = 3'b111, G = 3'b101;
6
7   always @(w, y)
8     case (y)
9       A: if (w) Y = B;
10      else Y = F;
11     B: if (w) Y = E;
12      else Y = C;
13     C: if (w) Y = D;
14      else Y = A;
15     D: if (w) Y = B;
16      else Y = F;
17     E: if (w) Y = A;
18      else Y = D;
19     F: if (w) Y = C;
20      else Y = G;
21     G: if (w) Y = A;
22      else Y = A;
23     default: Y = 2'bxx;
24   endcase
25
26   always @ (posedge Resetn, posedge Clock)
27     if (Resetn == 0) y <= A;
28     else y <= Y;
29
30   assign z = (y == D);
31
32 endmodule
```

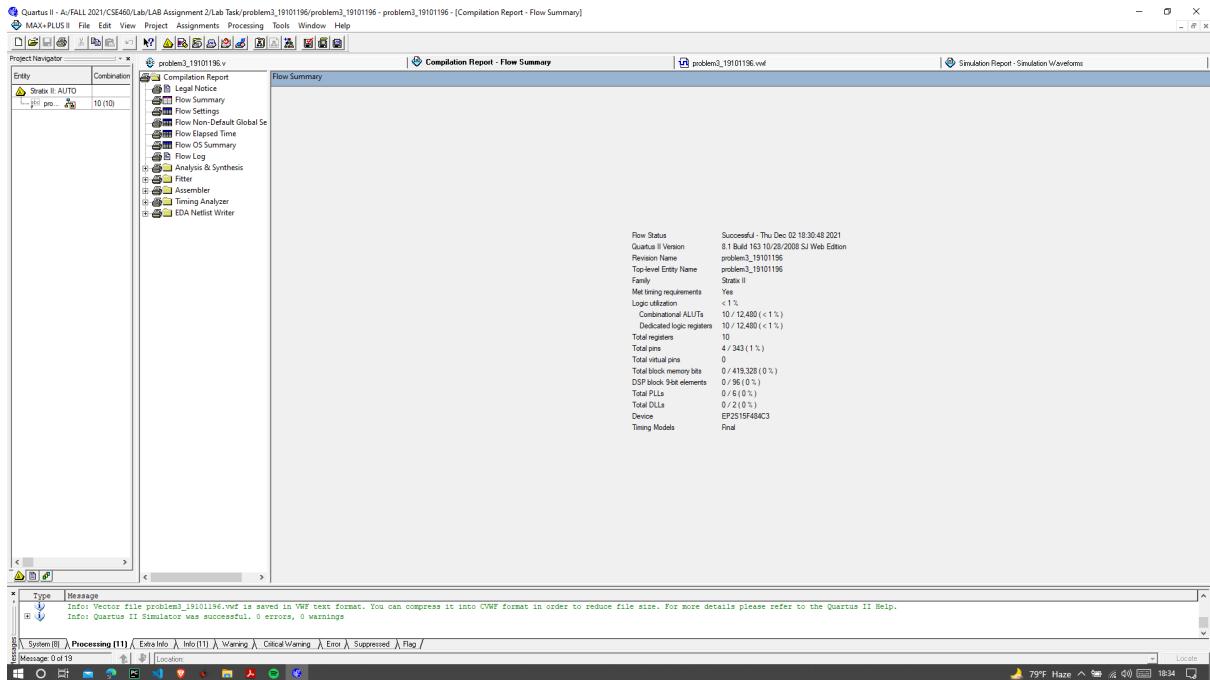
The message window at the bottom left shows:

- Info: Vector file problem3_19101196.vwf is saved in VWF text format. You can compress it into CWF format in order to reduce file size. For more details please refer to the Quartus II Help.
- Info: Quartus II Simulator was successful. 0 errors, 0 warnings

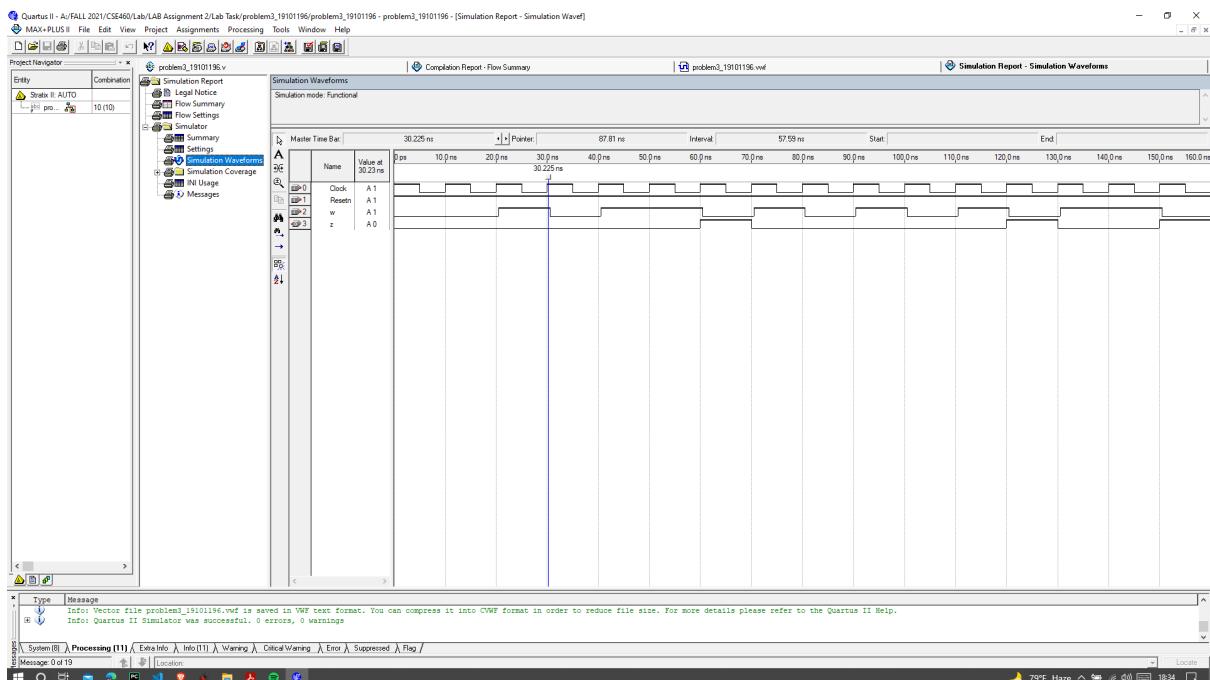
Waveform:



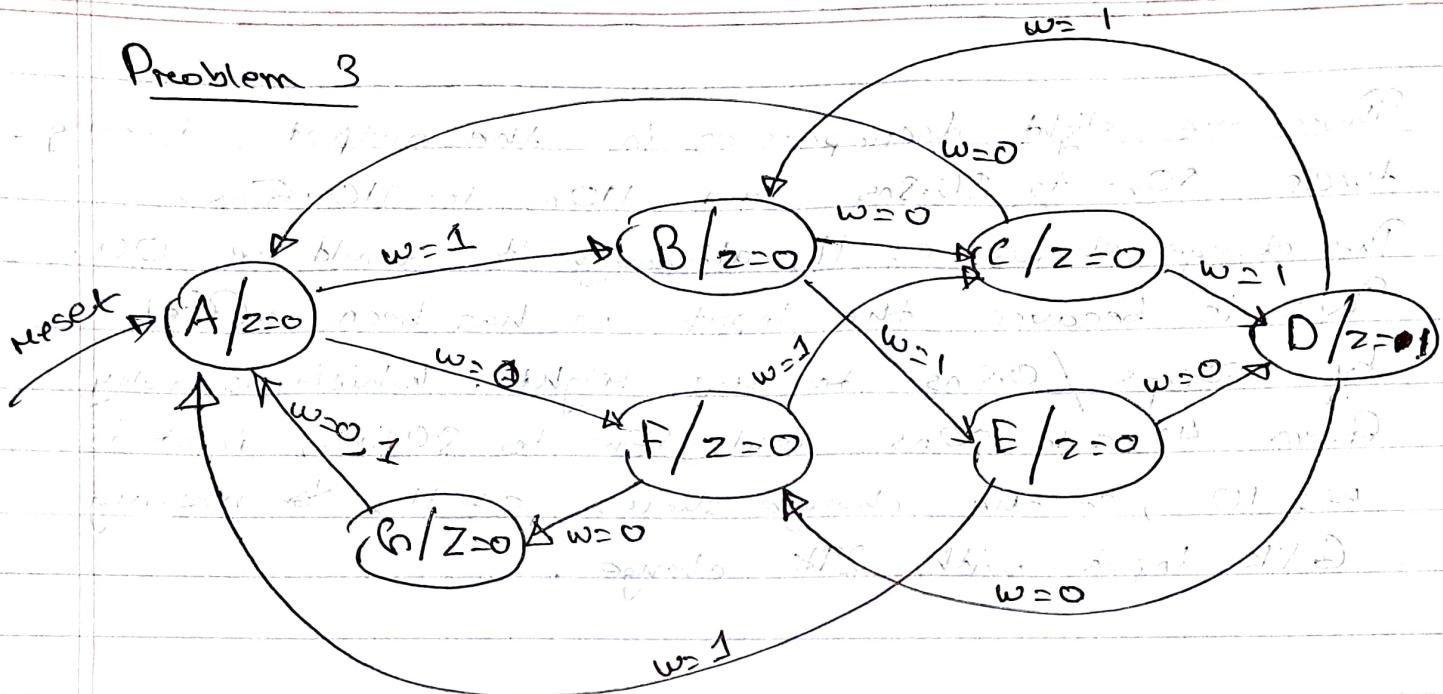
Compilation Report:



Simulation Report:



Problem 3



Let,

A \rightarrow	000
B \rightarrow	001
C \rightarrow	011
D \rightarrow	010
E \rightarrow	110
F \rightarrow	111
G \rightarrow	101

Possible ~~Inputs~~ Inputs

011

101

110

Current / y_2y_1

Next / y_2y_1

Output / z

$w=0$	$w=1$	z
000	001	0
001	001	0
011	000	0
010	111	1
110	010	0
111	101	0
101	000	0

For every 3 clock cycles, there has to be even 1s in the input. For clock cycles 1, 2 and 3 from 0ns to 30ns, the input sequence is 001. There is an odd number of 1s, so, in the 4th clock cycle 30ns to 40ns, output $z=0$.

From 30ns to 40ns, input $w=0$, 40ns to 50ns, $w=1$, 50ns to 60ns, $w=1$. So input sequence is 011, with an even number of 1s. So, in next clock cycle, 60ns to 70ns, it is in state D where output $z=1$.

3 bit sequences are not overlapping. In 8th, 9th and 10th clock cycle, the input sequence is 101, but the 11th clock cycle output is $z=0$. This is because, the 8th and 9th clock cycles are a part of the previous 3 consecutive

bits. So, for 7th, 8th and 9th clock cycles, 60ns to 90ns, input sequence is 0101. So, next clock cycle 90ns to 100ns, output $z=0$.

But for 10th, 11th and 12th clock cycles, 90ns to 120ns, input sequence is 101. So, next clock cycle 120ns to 130ns, output $z=1$.

0	0 1 0	0 0 0	1 1 0
1	1 0 0	1 1 1	0 1 0
0	0 0 0	0 1 0	0 1 1
1	1 0 0	1 0 1	1 1 0
0	0 0 0	0 0 0	1 0 1

The output bits will remain constant for 10ns and then change to 101 after 10ns.

For the next 10ns, it will remain at 101 and then change to 110 after 10ns.

After 10ns, it will remain at 110 for 10ns and then change to 0101 after 10ns.

It is after another 60ns at 110.

Now we have all 4 bits ready for our example. And it will take 10ns to change from 110 to 0101 together.

With a 50ns delay of 50ns, 0101 together with 110 will change to 101 after 50ns.

With another 50ns delay of 50ns, 101 together with 110 will change to 110 after 50ns.