CSE460

Lab Assignment 1

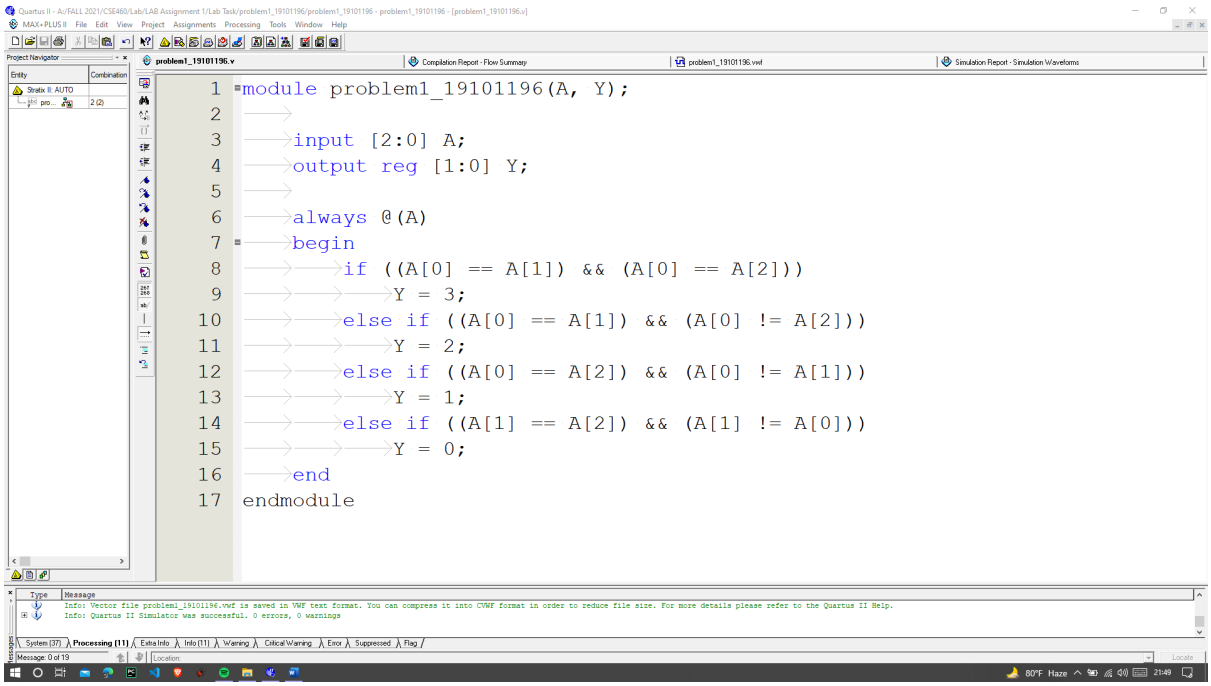
AHMAD AL ASAD

ID: 19101196

Section: 06

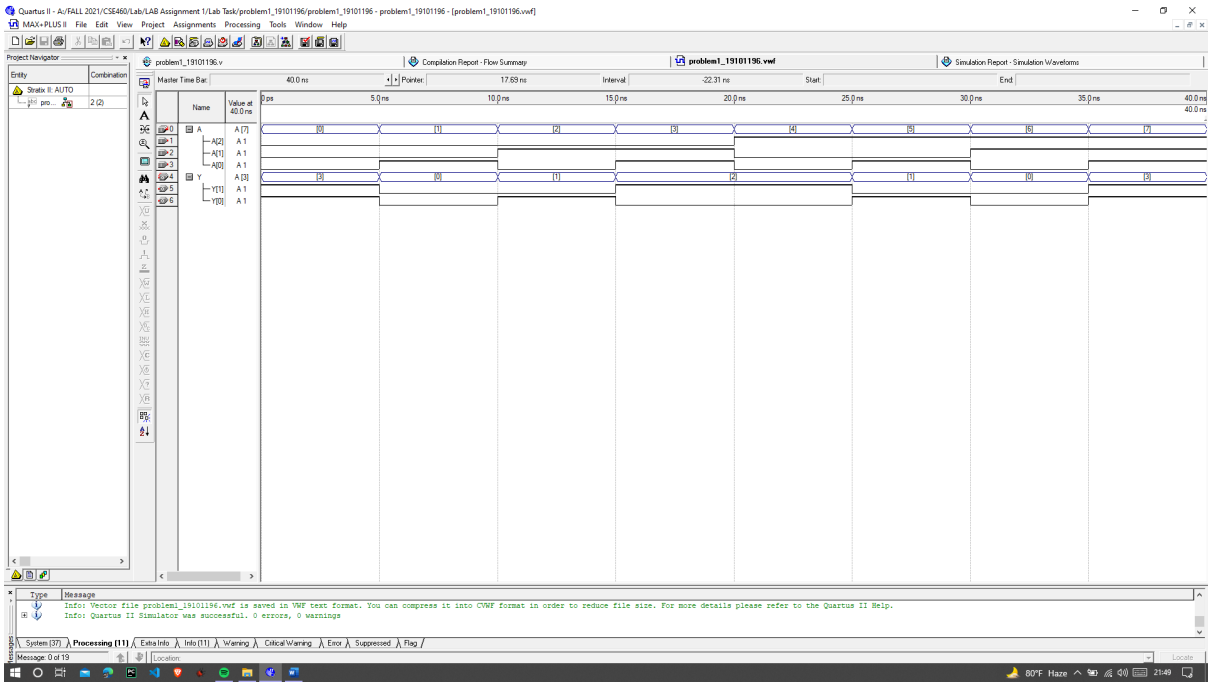
PROBLEM 1

Code:

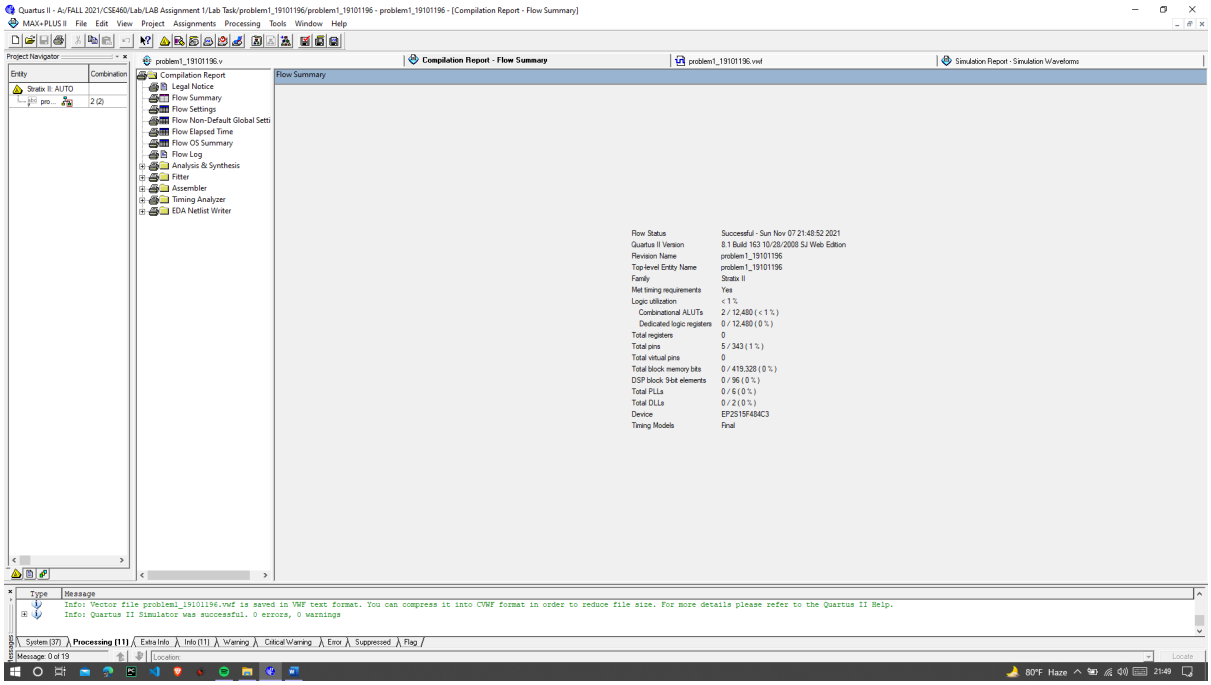


```
module problem1_19101196(A, Y);

    input [2:0] A;
    output reg [1:0] Y;

    always @(A)
    begin
        if ((A[0] == A[1]) && (A[0] == A[2]))
            Y = 3;
        else if ((A[0] == A[1]) && (A[0] != A[2]))
            Y = 2;
        else if ((A[0] == A[2]) && (A[0] != A[1]))
            Y = 1;
        else if ((A[1] == A[2]) && (A[1] != A[0]))
            Y = 0;
    end
endmodule
```

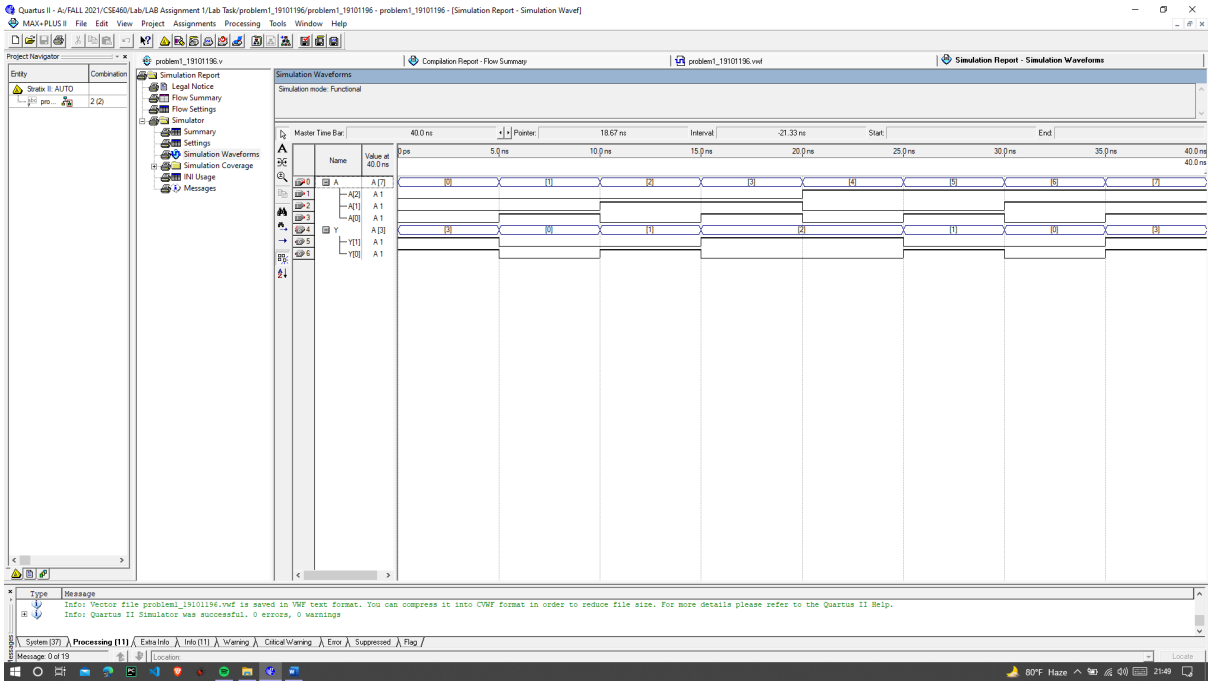
Waveform:



Compilation Report:



Simulation Report:



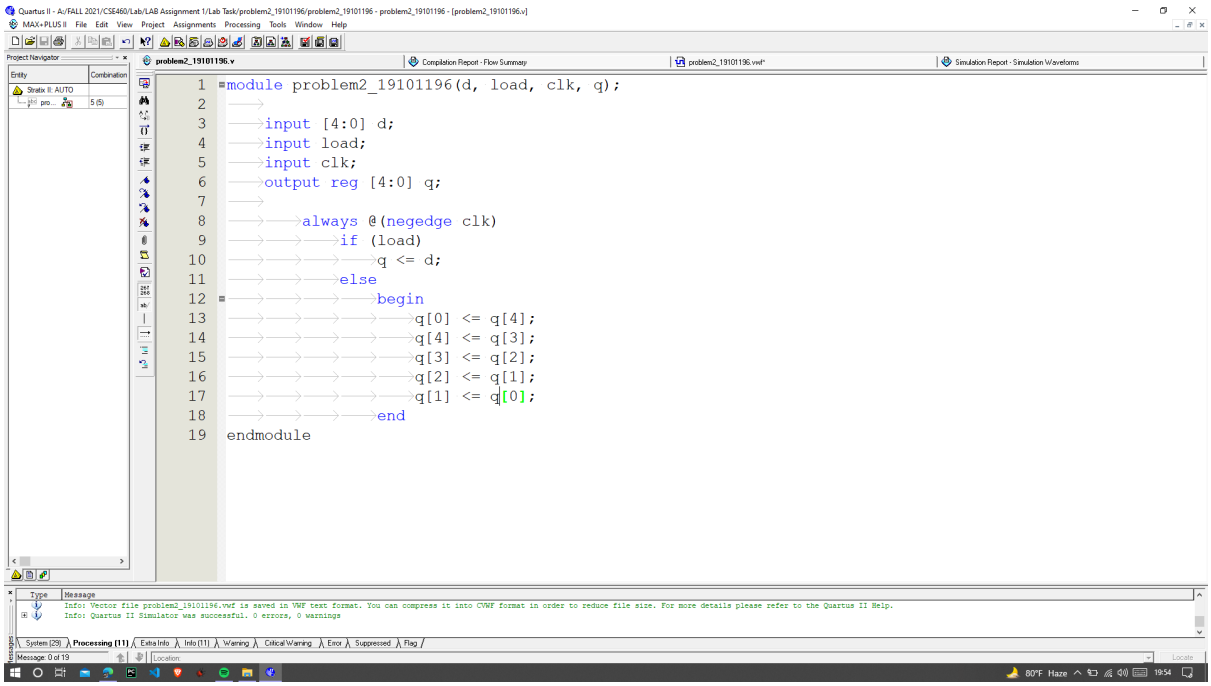
Explanation:

The input A contains all possible combinations for a 3 bits input.
From the above Waveform, the output is Y = 3 during 0ns – 5ns when input A = 000 and during 35ns – 40ns when input A = 111. So, when all bits are equal, output gives 3.
The output is Y = 0 during 5ns – 10ns when input A = 001 and during 30ns – 35ns when input A = 110. So, when there is a unique bit at position 0, output gives 0.

The output is Y = 1 during 10ns – 15ns when input A = 010 and during 25ns – 30ns when input A = 101. So, when there is a unique bit at position 1, output gives 1.
The output is Y = 2 during 15ns – 20ns when input A = 011 and during 20ns – 25ns when input A = 100. So, when there is a unique bit at position 1, output gives 2.

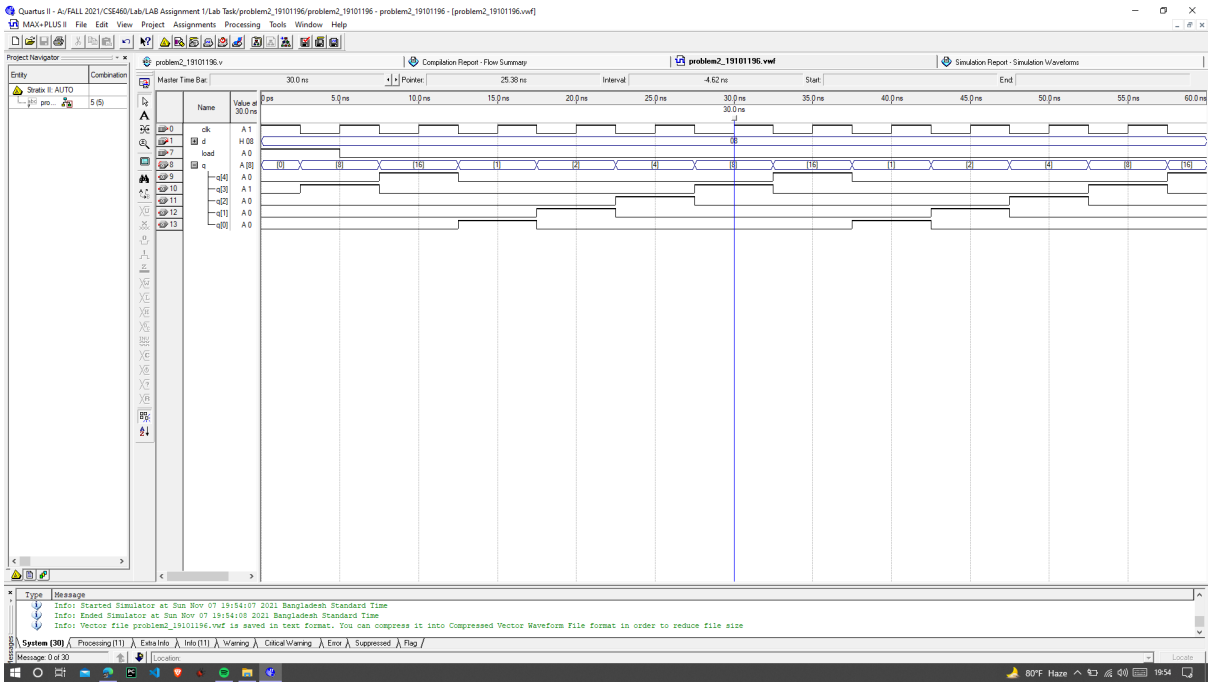
PROBLEM 2

Code:

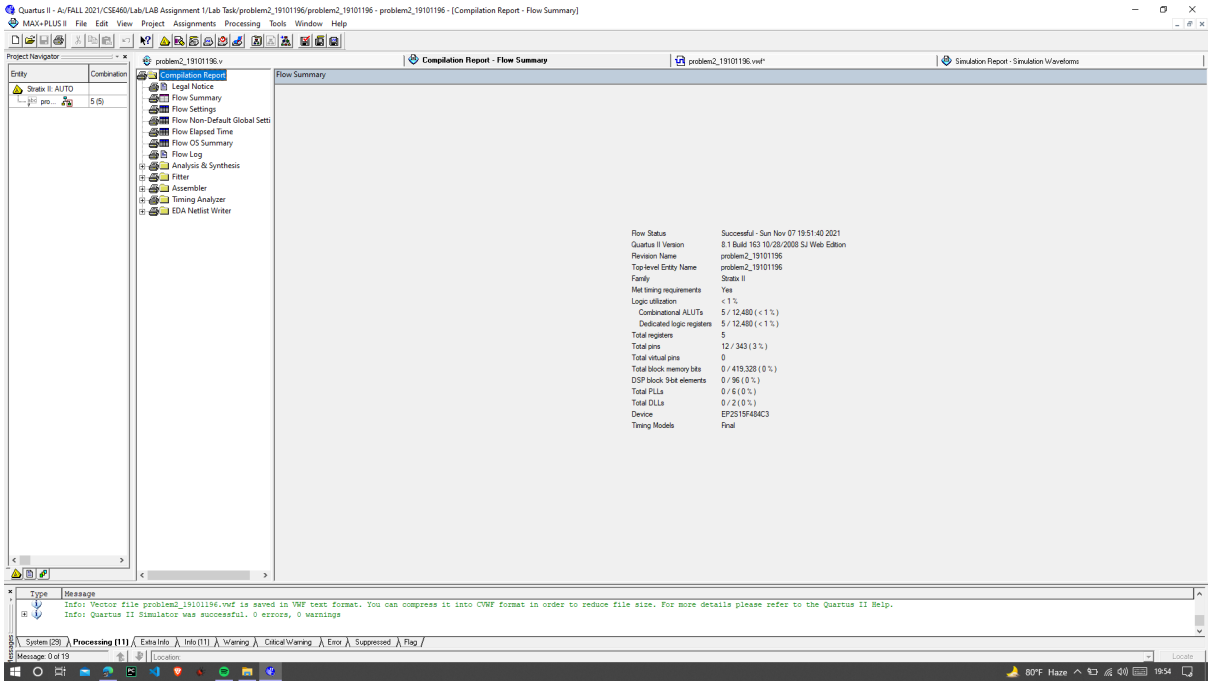


```
module problem2_19101196(d, load, clk, q);

    input [4:0] d;
    input load;
    input clk;
    output reg [4:0] q;

        always @(negedge clk)
            if (load)
                q <= d;
            else
                begin
                    q[0] <= q[4];
                    q[4] <= q[3];
                    q[3] <= q[2];
                    q[2] <= q[1];
                    q[1] <= q[0];
                end
endmodule
```

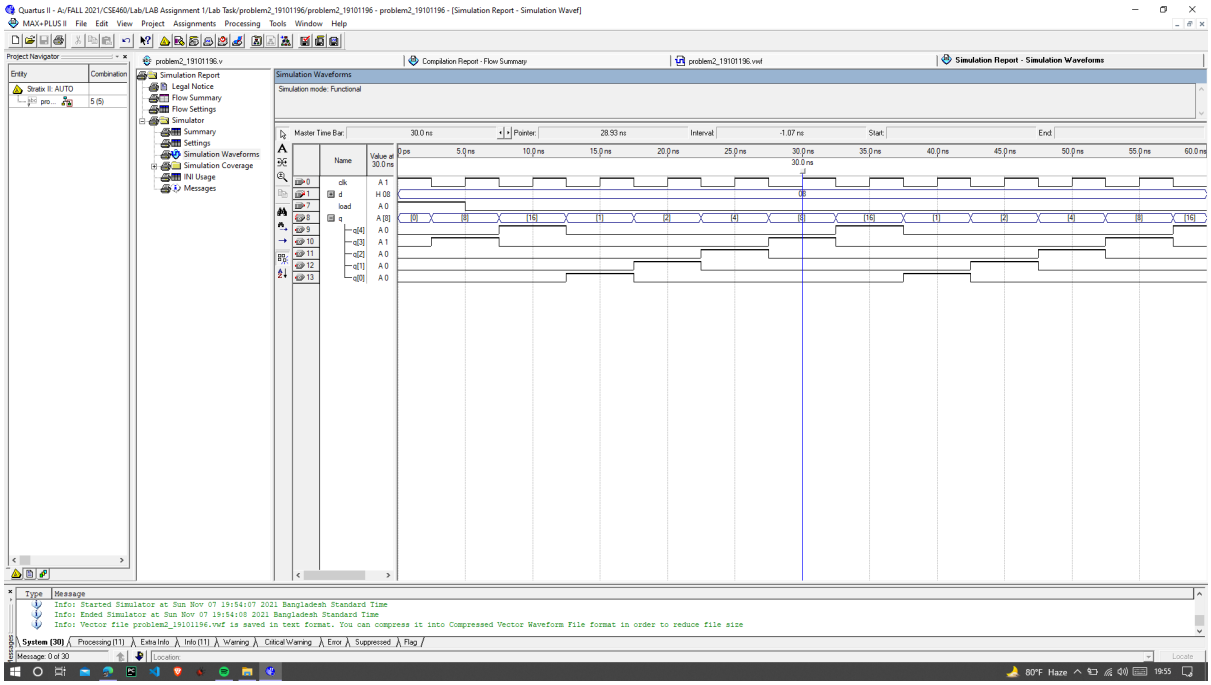
Waveform:



Compilation Report:



Simulation Report:



Explanation:

The input d contains an arbitrary value 8. In 5 bits binary, d = 01000.
The load is high (1) during 0ns – 5ns, and at 2.5ns the clock is at a negative edge. So, the input d gets copied to q. q = 01000.
After 5ns, the load is low (0).
At the next negative clock edge at 7.5ns, an end around left shift operation takes place. So, the new value for q = 10000, which is 16 in decimal.

At the next negative clock edge at 12.5ns, end around left shift operation takes place. So, the new value for q = 00001, which is 1 in decimal.
At the next negative clock edge at 17.5ns, end around left shift operation takes place. So, the new value for q = 00010, which is 2 in decimal.
At the next negative clock edge at 22.5ns, end around left shift operation takes place. So, the new value for q = 00100, which is 4 in decimal.
At the next negative clock edge at 27.5ns, end around left shift operation takes place. So, the new value for q = 01000, which is 8 in decimal. With this, the output q is back to the original input d = 01000.
Total time = 27.5 – 2.5 = 25ns
After 25ns more, the output cycles back to the original input, at 52.5ns.
The left shift operation takes place on 5 bits, and a 1 bit shift operation takes place on each negative edge of the clock. So, after 5 clock cycles, 5 bits will be shifted and one repetition will be completed.
Proof:
The clock period is 5ns. One repletion takes 25ns.
Total clock cycles = 25/5 = 5 cycles.