

NAME: REIMARC G. CORPUZ

COURSE, YEAR & SECTION: BSCPE 3GF

7 segment decoder – case statement

output reg [1:7] leds;

always @(hex)

case (hex) //abcdefg

0: leds = 7'b1111110;

1: leds = 7'b0110000;

2: leds = 7'b1101101;

3: leds = 7'b1111001;

4: leds = 7'b0110011;

5: leds = 7'b1011011;

6: leds = 7'b1011111;

7: leds = 7'b1110000;

8: leds = 7'b1111111;

9: leds = 7'b1111011;

10: leds = 7'b1110111;

11: leds = 7'b0011111;

12: leds = 7'b1001110;

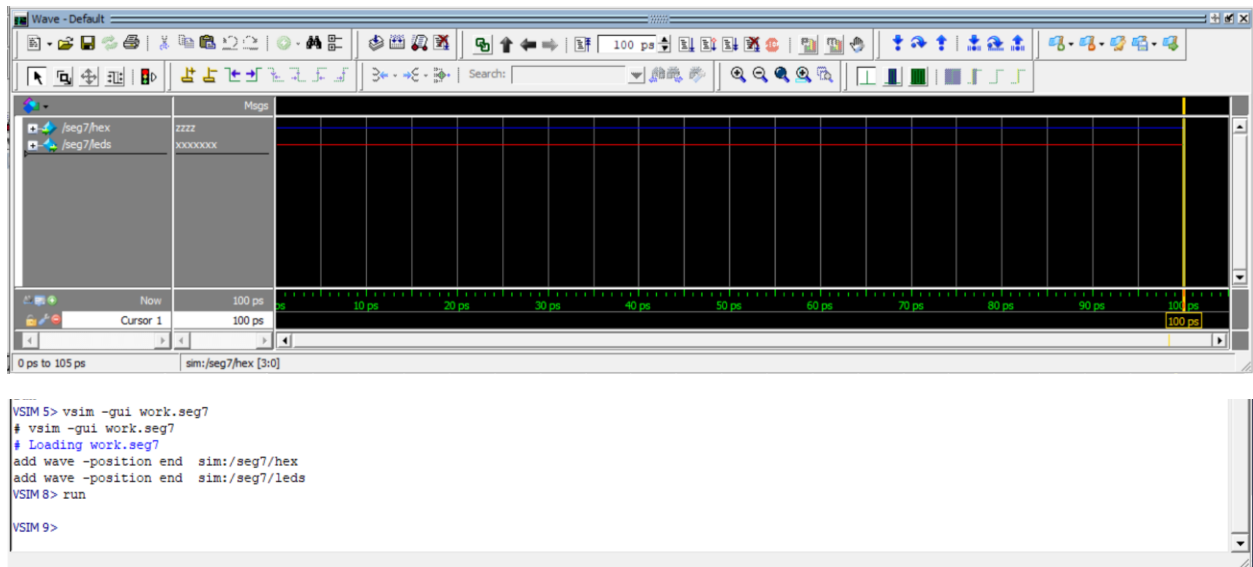
13: leds = 7'b0111101;

14: leds = 7'b1001111;

15: leds = 7'b1000111;

endcase

endmodule



74381 ALU – case statement

module alu (S, A, B, F);

input [2:0] S;

input [3:0] A, B;

output reg [3:0] F;

always @(S, A, B)

case (S)

0: F = 4'b0000;

1: F = B - A;

2: F = A - B;

3: F = A + B;

4: F = A ^ B;

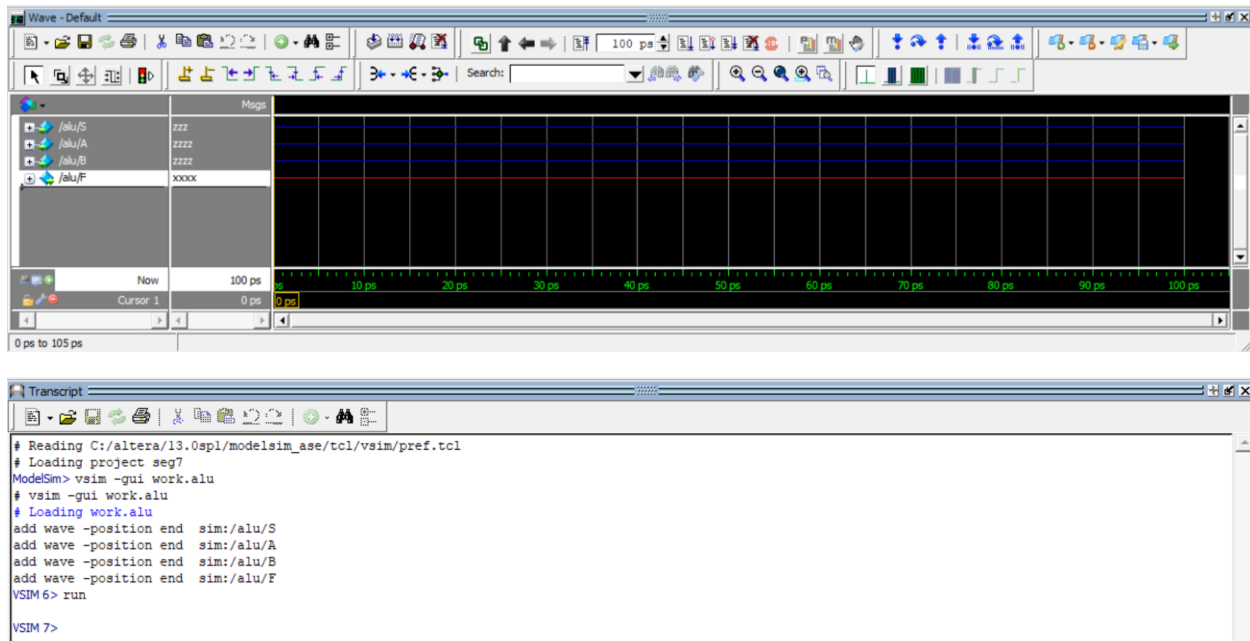
5: F = A | B;

6: F = A & B;

7: F = 4'b1111;

endcase

endmodule



Reflection:

A seven-segment display decoder that makes use of the case statement that is required to occur within an always/process statement is a better example of using the always/process statement for combinational logic. The case statement validates the accuracy of the data. The statement does the operation after the colon, setting segments to 111110 when the data is 0. Similar tests are made on other data values up to 15 in the case statement. Show the letters A through F from 10 to 15.

Depending on the value of its input, the case statement does various actions. A case statement suggests combinational logic if all potential input combinations are defined. If not, it implies sequential logic because the output will retain its previous value in unknown instances.