Jefferson Bui & Kesti Koci

CMPEN 275

Prof. Yi Yang

18 November 2019

Lab 7

Code:

|  |
| --- |
| module SimpleALU(S,A,B,F,Sign,Overflow,Sevenseg); // Declared Inputs and outputs  input signed [3:0] A; // Declare necessary input, outputs and register's  input signed [3:0] B; // A and B are the 4 bits that we are using  input [1:0] S; // This is the case selector  output reg signed [4:0] F; // Temporary output (Binary) that is later converted into the 8 bits for the Seven Seg  output reg Sign; // Output for negative sign or not  output reg Overflow; // Overflow output to detect overflow  output [7:0] Sevenseg; // Output called from Binary2Eight file to be initiated later  always @(S,A,B)  begin  case(S)  0: F = 4'b0000; // We have 4 cases: Clear, Addition, Subtraction, and Comparison  1: F = A+B;  2: F = A-B;  3: if (A>=B)  F = A;  else  F = B;  endcase  end    always @(F) // Sign Indicator  begin  Sign = 1; // Sign is on by default  if (F[4] == 1) // If the fifth bit for the output F is equal to 1, then the sign is off, else it stays as on  Sign = 0;  else  Sign = 1;  end    always @(F) // Overflow  begin  Overflow = 0; // Overflow is off by default  if (F>7) // With output F signed, program detects numbers greater than 7 as overflow because our Binary2Eight file does not have anything bigger than 7.  Overflow = 1; //If the detected number is over 7, overflow will turn on, else it stays off.  else  Overflow = 0;  end      Binary2Eight a1(F,Sevenseg); // Initiates the seven segment decoder    endmodule // Ends module |