California State University, Chico
Department of Electrical and Computer Engineering

EECE 343 Advanced Logic Design

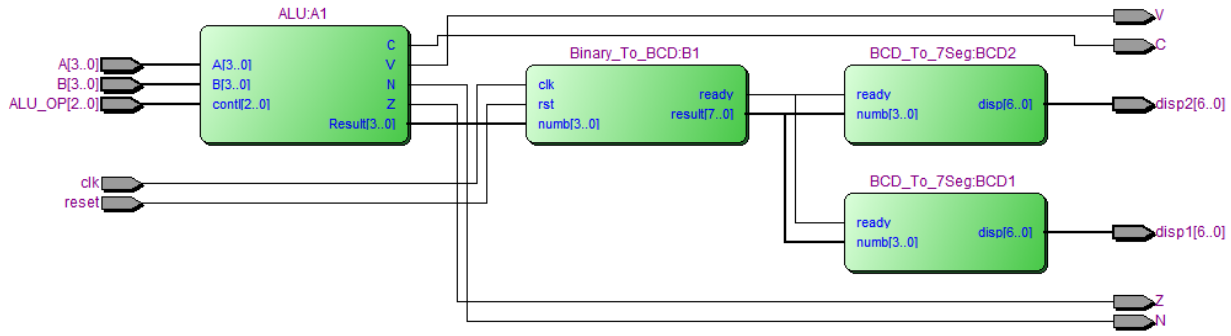
ALU to 7 Segments

by

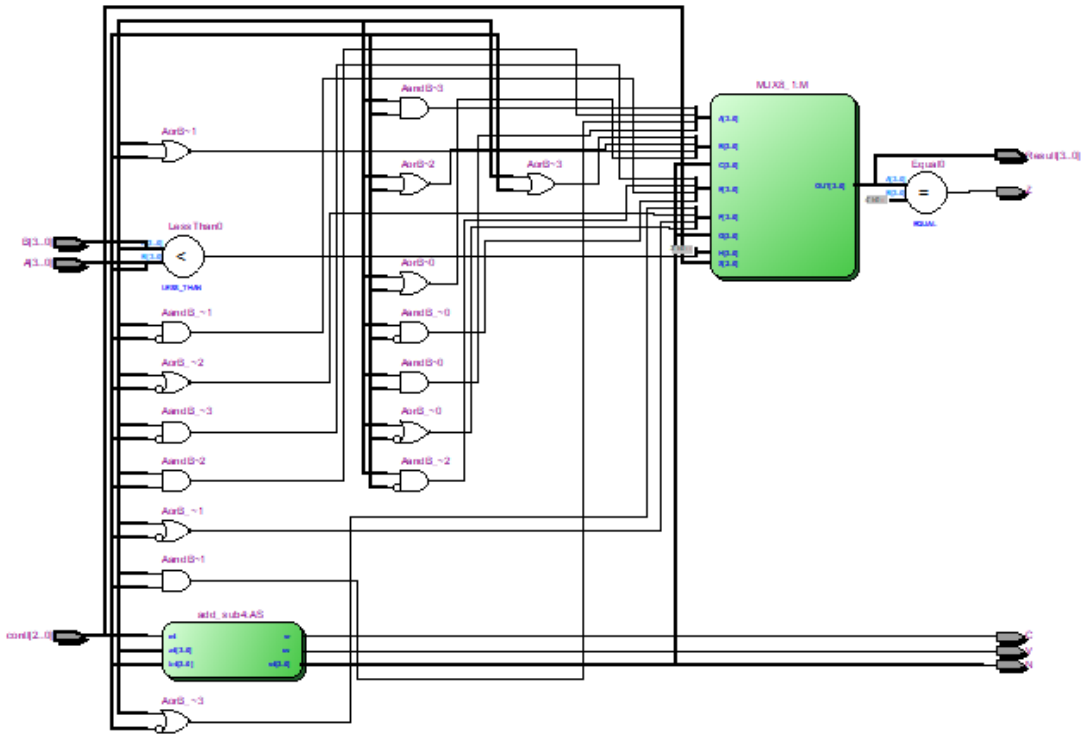
Moses P. Mccabe & Anthony Arevalo

April 16, 2019

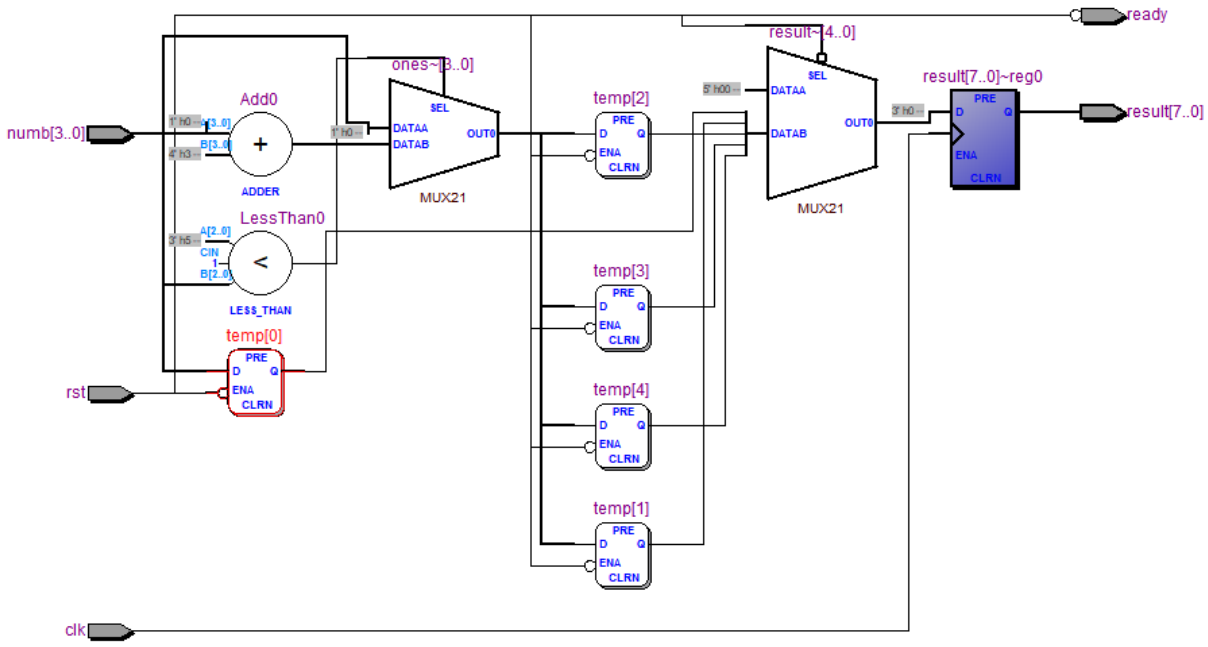
Alu to 7Seg



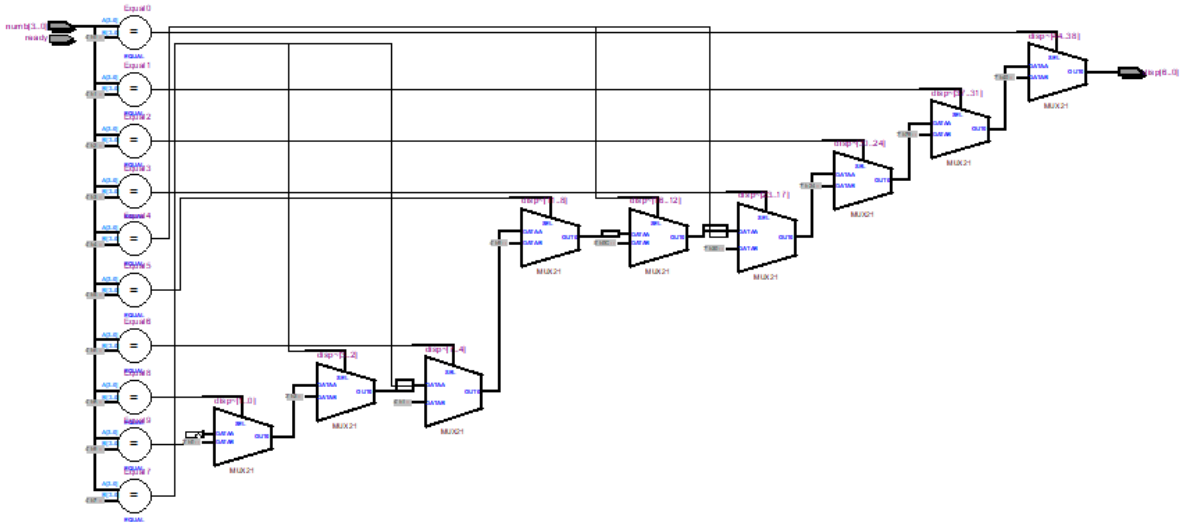
Alu



Binary To 7Seq



BCD To 7Seq



Date: April 16, 2019 BCD_To_7Seg.v Project: BCD_To_7Seg

```
1   // BCD to 7 segment Decoder
2   module BCD_To_7Seg(ready,numb, disp);
3   input ready;
4   input [3:0] numb;
5   output reg[6:0] disp;
6
7
8
9   always @(ready,numb) begin
10  if (numb == 4'b0000)      // 0
11     disp = 7'b0000001;
12  else if(numb == 4'b0001) // 1
13     disp = 7'b1001111;
14  else if(numb == 4'b0010) // 2
15     disp = 7'b0010010;
16  else if(numb == 4'b0011) // 3
17     disp = 7'b0000110;
18  else if(numb == 4'b0100) // 4
19     disp = 7'b1001100;
20  else if(numb == 4'b0101) // 5
21     disp = 7'b0100100;
22  else if(numb == 4'b0110) // 6
23     disp = 7'b0100000;
24  else if(numb == 4'b0111) // 7
25     disp = 7'b0001111;
26  else if(numb == 4'b1000) // 8
27     disp = 7'b0000000;
28  else if(numb == 4'b1001) // 9
29     disp = 7'b0000100;
30  else                      // everythg else
31     disp = 7'b0110000;
32  end
33  endmodule
34
```

```
1   module BCD_To_7Seg_test();
2   reg[3:0] A_in, B_in;
3   reg[2:0] ALU_OP_in;
4   reg clk_in, reset_in;
5
6   wire [6:0] disp1_out;
7   wire [6:0] disp2_out;
8   wire N_out,V_out,C_out,Z_out;
9
10  //module ALU_To_7Segment(A,B,ALU_OP,clk,reset,disp1,disp2,N,V,C,Z);
11  ALU_To_7Segment test(A_in,B_in,ALU_OP_in,clk_in,reset_in,disp1_out,disp2_out,N_out,V_out
    ,C_out,Z_out);
12
13  initial begin
14  clk_in = 0;
15  reset_in = 0;
16  ALU_OP_in = 3'b010; A_in = 4'b1000; B_in = 4'b0010;
17  #10
18  ALU_OP_in = 3'b001; A_in = 4'b1110; B_in = 4'b0001;
19  #10;
20  ALU_OP_in = 3'b011; A_in = 4'b1110; B_in = 4'b0011;
21  #10
22  ALU_OP_in = 3'b100; A_in = 4'b0001; B_in = 4'b0010;
23  #10
24  ALU_OP_in = 3'b011; A_in = 4'b0001; B_in = 4'b0010;
25  #10
26  ALU_OP_in = 3'b111; A_in = 4'b0001; B_in = 4'b0010;
27  end
28  always #5 clk_in = ~clk_in;
29  always #10 reset_in = ~reset_in;
30  endmodule
31
```

```
1   module Binary_To_BCD(numb, clk, rst, result, ready);
2   input [3:0] numb;
3   input clk, rst;
4   output reg [7:0] result;
5   output reg ready;
6   // temp wire
7   reg [3:0] ones, tens;
8   reg[7:0] temp;
9
10  integer i;
11
12  always @(rst, numb) begin
13  ones = 4'b0000;
14  tens = 4'b0000;
15
16  if (rst == 1) begin
17     ready = 1'b0;
18  end
19  else begin
20     for (i = 0; i < 4; i = i+1) begin
21        if(ones >= 4'b0101)
22           ones = ones + 4'b0011;
23        if(ones[3] == 1'b1 && !(tens >= 4'b0101))begin
24           tens = tens << 1;
25           tens[0] = ones[3];
26        end
27        if(tens >= 4'b0101) begin
28           tens = tens + 4'b0011;
29        end
30        ones = ones << 1;
31        ones[0] = numb[3-i];
32     end
33     ready = 1'b1;
34     temp = [tens,ones];
35  end
36  end
37
38  always @(posedge clk) begin
39     if(ready == 1)
40        result <= temp;
41     else
42        result <= 8'b00000000;
43
44  end
45  endmodule
46
```

```
1   module ALU_To_7Segment(A,B,ALU_OP,clk,reset,disp1,disp2,N,V,C,Z);
2   input[3:0] A,B;
3   input[2:0] ALU_OP;
4   input clk, reset;
5
6   output [6:0] disp1,disp2;
7   output N,V,C,Z;
8
9   wire ready;
10  wire [3:0]bin4;
11  wire [7:0]bin8;
12
13
14
15
16  //module ALU(A,B,cont1,C,V,N,Z,Result);
17  ALU A1(A,B,ALU_OP,C,V,N,Z,bin4);
18
19  //module Binary_To_BCD2(numb, clk, rst, result, ready);
20  Binary_To_BCD B1(bin4,clk,reset,bin8,ready);
21
22  //module BCD_To_7Seg(ready,numb, disp);
23  BCD_To_7Seg BCD1(ready, bin8[3:0],disp1);
24  BCD_To_7Seg BCD2(ready, bin8[7:4],disp2);
25
26  endmodule
27
```