TERM PROJECT

Name: Servando Olvera ID# 1001909287

Date Submitted: 04/28/2023 Lab Section # 003

CSE 2441 – Digital Logic Design

Spring Semester 2023

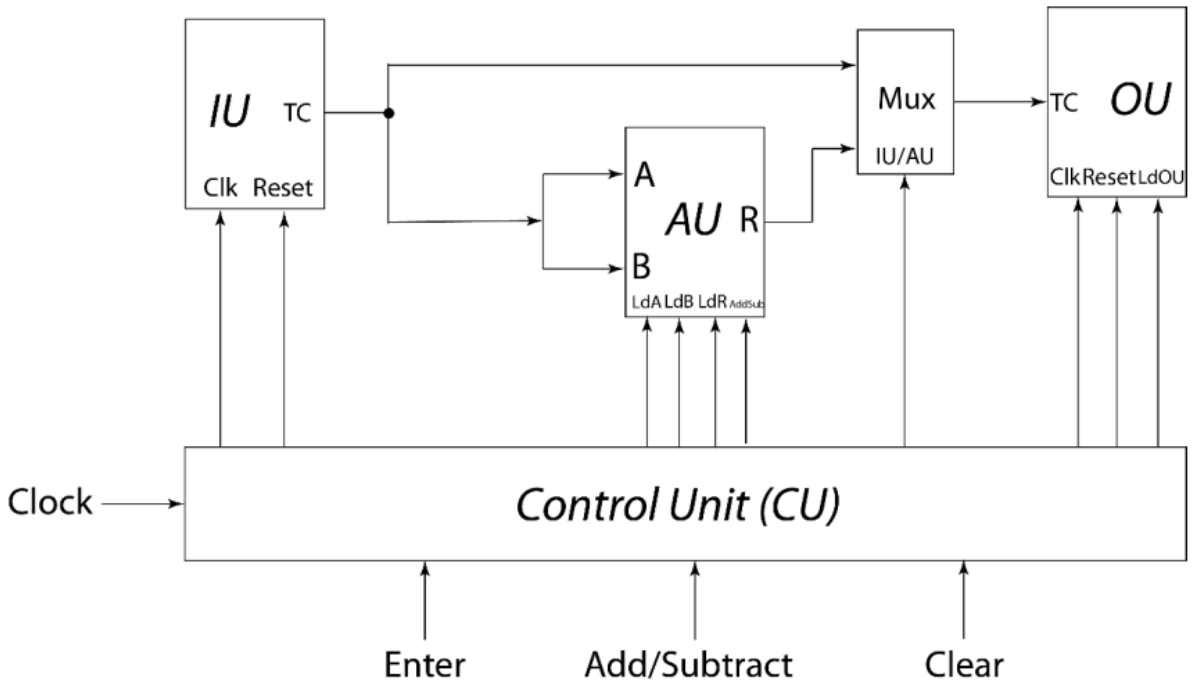
Term Project – Eight-Bit, Two-Function Calculator

(100+ Points)

Due by May 2, 2023, 11:59 pm

This lab is performed on the DE10-Lite

Calculator Top-Level Functional Diagram:



Added Features:

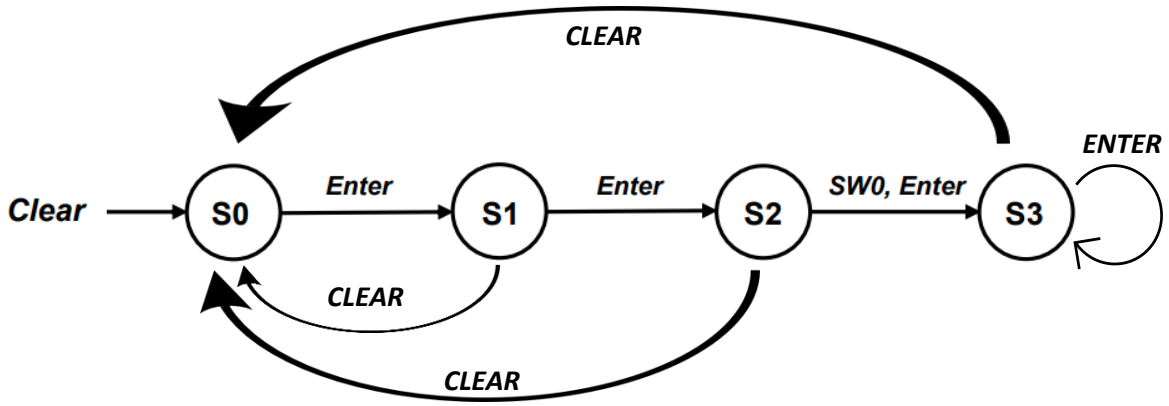
- Invalid Input Recognizer

Top-Level Verilog Code:

```
1   module TermProject (
2      input clock, enter, add_sub, clear,
3      input [3:0] row,
4      output [3:0] col,
5      output HEX3, inv,
6      output [0:6] HEX2, HEX1, HEX0,
7      output [1:0] LED
8      );
9
10     wire reset, LdA, LdB, LdR, LdOU, IUAU;
11
12     CU controlUnit (
13        .clock(clock),
14        .enter(enter),
15        .clear(clear),
16        .reset(reset),
17        .LdA(LdA),
18        .LdB(LdB),
19        .LdR(LdR),
20        .LdOU(LdOU),
21        .IUAU(IUAU),
22        .LED(LED)
23     );
24
25     wire [7:0] x;
26
27     Calc_IU IU (
28        .clock(clock),
29        .reset(clear),
30        .row(row),
31        .col(col),
32        .LEDR(x),
33        .invalid(inv)
34     );
35
36     wire [7:0] Rout;
37
38     EightBitRegister AU (
39        .X(x),
40        .inA(LdA),
41        .inB(LdB),
42        .Add_Sub(add_sub),
43        .Out(LdR),
44        .Clear(clear),
45        .Rout(Rout)
46     );
47
48     reg [7:0] out;
49
50     always @(IUAU) begin
51        if(IUAU == 1'b1)
52           out = x;
53        else
54           out = Rout;
55     end
56
57     CalculatorOU OU (
58        .x(out),
59        .HEX3(HEX3),
60        .HEX2(HEX2),
61        .HEX1(HEX1),
62        .HEX0(HEX0)
63     );
64
65  endmodule
```

Control Unit

State Diagram:



Verilog Code:

```
1   module CU (
2      input clock, enter, add_sub, clear,
3      output reg clck, reset, LdA, LdB, LdR, AddSub, LdOU, IUAU,
4      output reg [1:0] LED
5      );
6
7      reg [1:0] state, nextstate;
8      parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
9
10     wire enter_out;
11     EdgeDetect EdgeDetect_inst (
12        .in(enter),
13        .clock(clock),
14        .out(enter_out)
15     );
16
17     always @ (negedge enter_out, negedge clear)
18        if (clear == 0) state <= S0;
19           else state <= nextstate;
20        always @ (state)
21           case ({state})
22              S0: begin nextstate = S1; reset = 1'b1; LdA = 1'b0; LdB = 1'b0; LdR = 1'b0; LdOU
    = 1'b0; IUAU = 1'b1; LED = S0; end
23              S1: begin nextstate = S2; reset = 1'b1; LdA = 1'b1; LdB = 1'b0; LdR = 1'b0; LdOU
    = 1'b1; IUAU = 1'b1; LED = S1; end
24              S2: begin nextstate = S3; reset = 1'b1; LdA = 1'b0; LdB = 1'b1; LdR = 1'b0; LdOU
    = 1'b1; IUAU = 1'b1; LED = S2; end
25              S3: begin nextstate = S3; reset = 1'b1; LdA = 1'b0; LdB = 1'b0; LdR = 1'b1; LdOU
    = 1'b1; IUAU = 1'b0; LED = S3; end
26           endcase
27  endmodule
28
29
```

Test Demo:

Demoed in person by Andrew

Table of Results:

Test Input	Output
74 + 35	109
74 – 35	39
-74 + 35	-39
-74 – 35	-109
127 + 6	-123
127 – 6	121
9 + 10	19
9 – 10	-1
88 – 125	-37
88 + 125	-43

Unresolved Problems:

N/A

Lesson Learned:

Breaking down a rather large project into smaller components greatly facilitates the completion of said project. For instance, at the beginning of this course we learnt about the basics of Digital Logic, from there we started to build up into more complex topics. We went from basic Logic Gates to High-level Logic Devices, Synchronous Sequential Circuits, and Finite State Machines.

In a likely manner, with this Term Project its assembly was done quite easily since it had already been broken down for us into smaller chunks of work, which were in essence the foundation of the project. Labs 8, 9, and 10 took care of the Calculator's Arithmetic Unit, Output Unit, and Input Unit, respectively, and all this work amounted to about 80% of the Term Project. All there was left to do, was to merely implement the Control Unit, a MUX, and the Top-Level Module, where everything could connect. Having already the knowledge of how these components communicate with one another, the creation of Control Unit and the simple instantiations of the Arithmetic Unit, Output Unit, and Input Unit along with a simple if else statement for the MUX, the project is then completed.

Taking all of this in mind, if one carefully analyzing the requirements for a project, breaks it down into smaller chunks of work, and organizes how it all will fit together, it allows for a more efficient workflow.