Lab 6 – Arithmetic Logic Unit

Name: Von Kaukeano *Date*: 10/16/18

Course: Digital Circuit Design Lab – ECE 2613 *Section #*: 001

Summary/Abstract

An Arithmetic Logic Unit was designed which behaves like a calculator. The first eight switches were assigned as two separate four bit binaries. The switches from nine to eleven are logical switches that apply the arithmetic operation associated with the selector. Depending on the sizes of the bits when added or subtracted the LED light at switch eight will turn on when a carry out bit occurs.

Introduction

In this lab we designed an Arithmetic Logic Unit which behaves like a 4-bit calculator. An ALU is a central component of most processor designs. The background knowledge of this lab is basic binary addition and subtraction and understanding when a carry out occurs and when it does not. When applying arithmetic operations, carry outs occur and showing that is important to understand if a number is positive or negative.

Procedure

The procedure for this lab was to use the table to create a 4-bit ALU. We had to make sure that all of the switches, the LED, and the logical selectors were selected correctly and that the arithmetic operations were assigned to each logical combination. When logic was implemented we checked to make sure that the correct operation occurred.

Results

Design Code

```
//
// lab6 : version 10/08/2018
//
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////
module alu_1 (output logic [3:0] f, output logic c_out, input logic [2:0] sel,
    input logic [3:0] a, input logic [3:0] b, input logic c_in);

    // enter your code here
always @* begin
case(sel)
3'b000: {c_out, f} = {1'b0,a};
3'b001: {c_out, f} = a + b + c_in;
3'b010: {c_out, f} = a + b;
3'b011: {c_out, f} = a - b - c_in;
3'b100: {c_out, f} = a - b;
3'b101: {c_out, f} = a + 1;
3'b110: {c_out, f} = a - 1;
3'b111: {c_out, f} = {1'b0,a & b};
endcase
end

endmodule
```

```
//
// lab6 : version 10/08/2018
//
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////
module lab6_alu_1 (
    output logic c_out,
    output logic [6:0] cathode,
    input logic [11:0] sw
    );

    // enter your code here
    logic [3:0] calc;

alu_1 u_alu_1 ( .f(calc) , .c_out(c_out) , .sel(sw[11:9]) , .a(sw[3:0]) , .b(sw[7:4]), .c_in(sw[8])  );
svn_seg_decoder u_svn_seg(.seg_out(cathode), .display_on(1'b1), .bcd_in(calc));

endmodule
```

Simulation Results

```
****** xsim v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source xsim.dir/work.tb_alu_1/xsim_script.tcl
# xsim {work.tb_alu_1} -autoloadwcfg -tclbatch {tb_alu_1.tcl} -onerror quit
Vivado Simulator 2018.2
Time resolution is 1 ps
source tb_alu_1.tcl
## run 82us
Simulation complete - no mismatches!!!
$finish called at time : 81920 ns : File "/home/tuh42003/2613_2018f/lab6/tb_alu_1.sv" Line 67
## exit
INFO: [Common 17-206] Exiting xsim at Tue Oct  9 13:40:14 2018...
Compressing vcd file to lxt2 file. ...
```

```
****** xsim v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source xsim.dir/work.tb_lab6_alu_1/xsim_script.tcl
# xsim {work.tb_lab6_alu_1} -autoloadwcfg -tclbatch {tb_lab6_alu_1.tcl} -onerror quit
Vivado Simulator 2018.2
Time resolution is 1 ps
source tb_lab6_alu_1.tcl
## run 82us
Simulation complete - no mismatches!!!
$finish called at time : 81920 ns : File "/home/tuh42003/2613_2018f/lab6/tb_lab6_alu_1.sv" Line 117
## exit
INFO: [Common 17-206] Exiting xsim at Tue Oct  9 13:41:08 2018...
Compressing vcd file to lxt2 file. ...
```

Hardware Implementation

My design was demonstrated to Catherine on the afternoon of 10/9 during the lab period.

Conclusion

In conclusion, the ALU was designed successfully. The 4-bit addition and subtraction were successful with the carry out bit. A design like this is similar to how a regular calculator works, but with less complexity. The understanding of when a carry bit occurs and how to effects binary is important to understand in being successful to this lab.