Lab 5: Building an ALU and Displaying the Result

Tuesday, October 08, 2024 4:54 PM

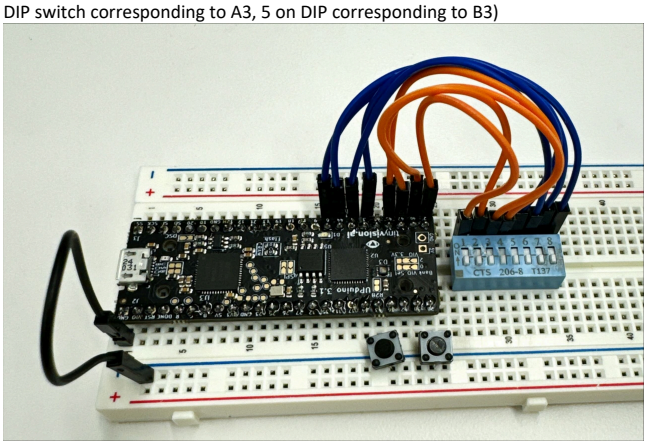
Jack Burton
TAs: Cory and Karen

Summary: Building a 4-bit ALU using an FPGA, and displaying the output on a seven-segment display.

Bench: 19 and 15

L1

Built the circuit with no issues—blue wires representing B0-B3, orange wires representing A0-A3 (1 on DIP switch corresponding to A3, 5 on DIP corresponding to B3)



L2/L3

Created Radiant project according to tutorial from course website. Pasted 4-bit ALU code from VHDLWeb into top.vhd in project.

Connected FPGA pins as follows:
A(0) = 45
A(1) = 47
A(2) = 46
A(3) = 2

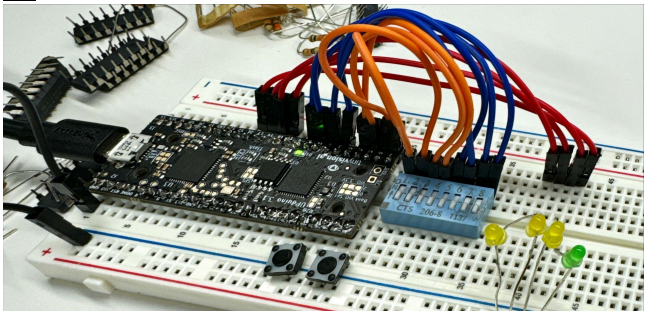
B(0) = 6
B(1) = 44
B(2) = 4
B(3) = 3

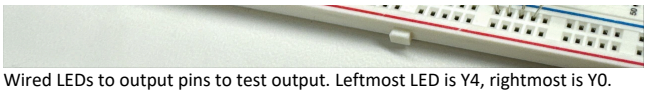
S(0) = 28
S(1) = 43

Y(0) = 21
Y(1) = 13
Y(2) = 19
Y(3) = 18

Exported ALU design and flashed to UPDuino successfully.

L4





Wired LEDs to output pins to test output. Leftmost LED is Y4, rightmost is Y0.

Tested various values of A and B with each operation:
Tested values to include overflow cases, negative values. All cases returned expected values:

A	B	OP	Exp	Actual
0001	0001	11	0000	0000
0001	0001	00	0001	0001
0001	0001	10	0010	0010
0001	0001	01	0001	0001
1010	0101	00	0000	0000
1010	0101	10	1111	1111
1010	0101	01	1111	1111
1010	0101	11	0101	0101
0000	1111	10	1111	1111
0000	1111	00	0000	0000
0000	1111	01	1111	1111
0000	1111	11	0001	0001
1111	1111	11	0000	0000
1111	1111	10	1110	1110
1111	1111	01	1111	1111
1111	1111	00	1111	1111

L5

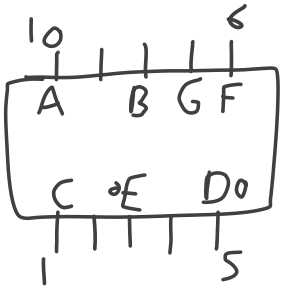
Reconfigured top.vhd to have submodules for alu and sevenseg. Encountered multiple syntax errors—used output section to find the lines for these errors and fixed them.

Encountered error with Radiant not finding my top module file—manually chose this as the top file and then it exported successfully.

Updated pin mappings:
Y(0) = 10, Y(1) = 12, Y(2) = 21, Y(3) = 13, Y(4) = 19, Y(5) = 18, Y(6) = 11

L6

Pin mappings:
A: 10
B: 8
C: 1
D: 5
E: 3
F: 6
G: 7



Leftmost is G, rightmost is A

Encountered issue with displaying digits because resistors for adjacent output bits were touching—debugged by connecting LEDs to each of 7 outputs and checking for expected values based on output digit.

L7

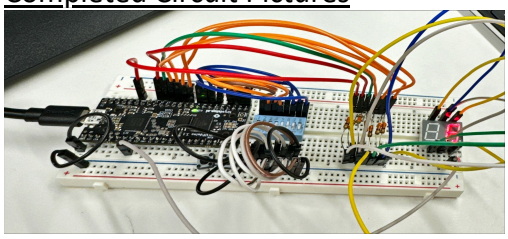
Tested by switching "A" bits to produce each number (keeping B as 0000, so subtracting zero).
Successfully produced 0-F on the 7-segment display:

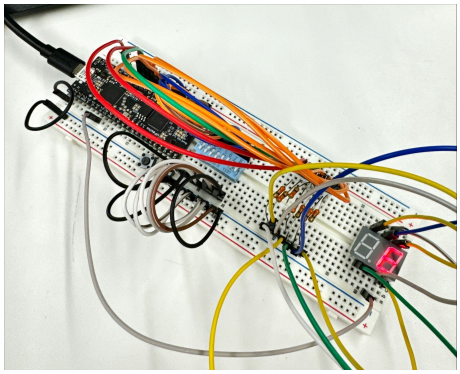
A	Exp	Actual
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4

0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	A
1011	b	b
1100	C	C
1101	d	d
1110	E	E
1111	F	F

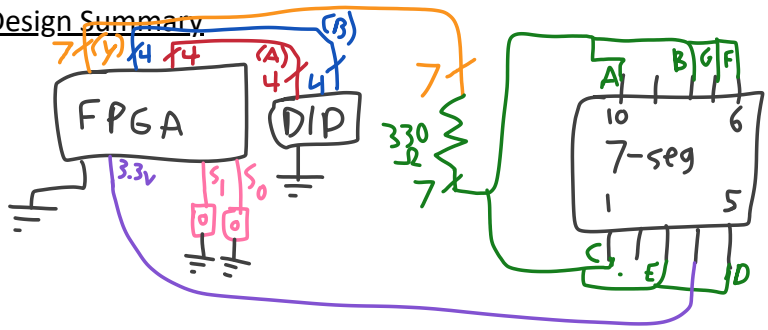
Further confirmed by performing various operations and checking the display output versus expected (ALU was fully tested in previous step, but just for confirmation that it is still working).

Completed Circuit Pictures



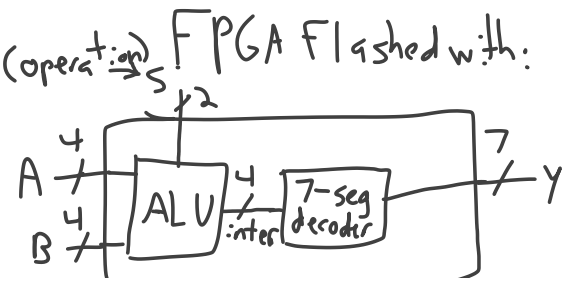


Design Summary



FPGA Pin Mappings:
A(0) = 45
A(1) = 47
A(2) = 46
A(3) = 2

B(0) = 6
B(1) = 44
B(2) = 4



B(3) = 3

S(0) = 28
S(1) = 43
Y(0) = 10, Y(1) = 12, Y(2) = 21, Y(3) = 13, Y(4) = 19, Y(5) = 18, Y(6) = 11

"ALU" submodule takes input 4-bit numbers A, B and does operation based on 2 "S" operation bits: logical and if S = 00, logical or if S = 01, + when S = 10, - when S = 11.
"7-seg decoder" submodule takes input of 4-bit number output from ALU and turns on appropriate digit segments for a 7-segment digit display to show this 4-bit binary number in decimal representation (except for 10-15 it shows A-F).

Questions

What was the most valuable thing you learned, and why? I learned that Radiant can be glitchy and there are often multiple ways to go about doing the same thing on it. This is valuable in case I have more issues with Radiant in the future when trying to export my VHDL code.

What skills or concepts are you still struggling with? What will you do to learn or practice these? The concept of modularity/hierarchy for VHDL modules is hard for me to wrap my head around coming from software land, since the formatting and compilation is very different. I will practice more VHDLWeb problems to improve on this.

How long did it take you to complete each part of this lab (prelab, the "lab" itself, and the report)? Prelab took ~1 hour, lab took about 4 hours, and the report took about 1 hour.

VHDL Code

Top.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity top is
  port(
  a : in unsigned(3 downto 0);
  b : in unsigned(3 downto 0);
  s : in std_logic_vector(1 downto 0);
  y : out std_logic_vector(6 downto 0)
  );
end top;



architecture synth of top is
component sevenseg is
port (
  S : in unsigned(3 downto 0);
  segments : out std_logic_vector(6 downto 0)
);
end component;

component alu is
port(
  a : in unsigned(3 downto 0);
  b : in unsigned(3 downto 0);
  s : in std_logic_vector(1 downto 0);
  y : out unsigned(3 downto 0)
);
```

```
end component;

signal inter : unsigned(3 downto 0);
begin
  alu_1 : alu port map (a, b, s, inter);

  seven_seg : sevenseg port map (inter, y);

end;
```

Alu.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity alu is
  port(
  a : in unsigned(3 downto 0);
  b : in unsigned(3 downto 0);
  s : in std_logic_vector(1 downto 0);
  y : out unsigned(3 downto 0)
  );
end alu;

architecture synth of alu is
begin
  with s select
  y <= (a and b) when "00",
       (a or b) when "01",
       (a + b) when "10",
       (a - b) when "11",
       a when others;
end;
```

Sevenseg.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity sevenseg is
  port(
  S : in unsigned(3 downto 0);
  segments : out std_logic_vector(6 downto 0)
  );
end sevenseg;

architecture synth of sevenseg is
begin
  with S select
    segments <= "0000001" when "0000",
                "1001111" when "0001",
                "0010010" when "0010",
                "0000110" when "0011",
                "1001100" when "0100",
                "0100100" when "0101",
                "0100000" when "0110",
                "0001111" when "0111",
                "0000000" when "1000",
                "0001100" when "1001",
                "0001000" when "1010",
                "1100000" when "1011",
```

```
                "0110001" when "1100",
                "1000010" when "1101",
                "0110000" when "1110",
                "0111000" when "1111",
                "1111111" when others;
end;
```