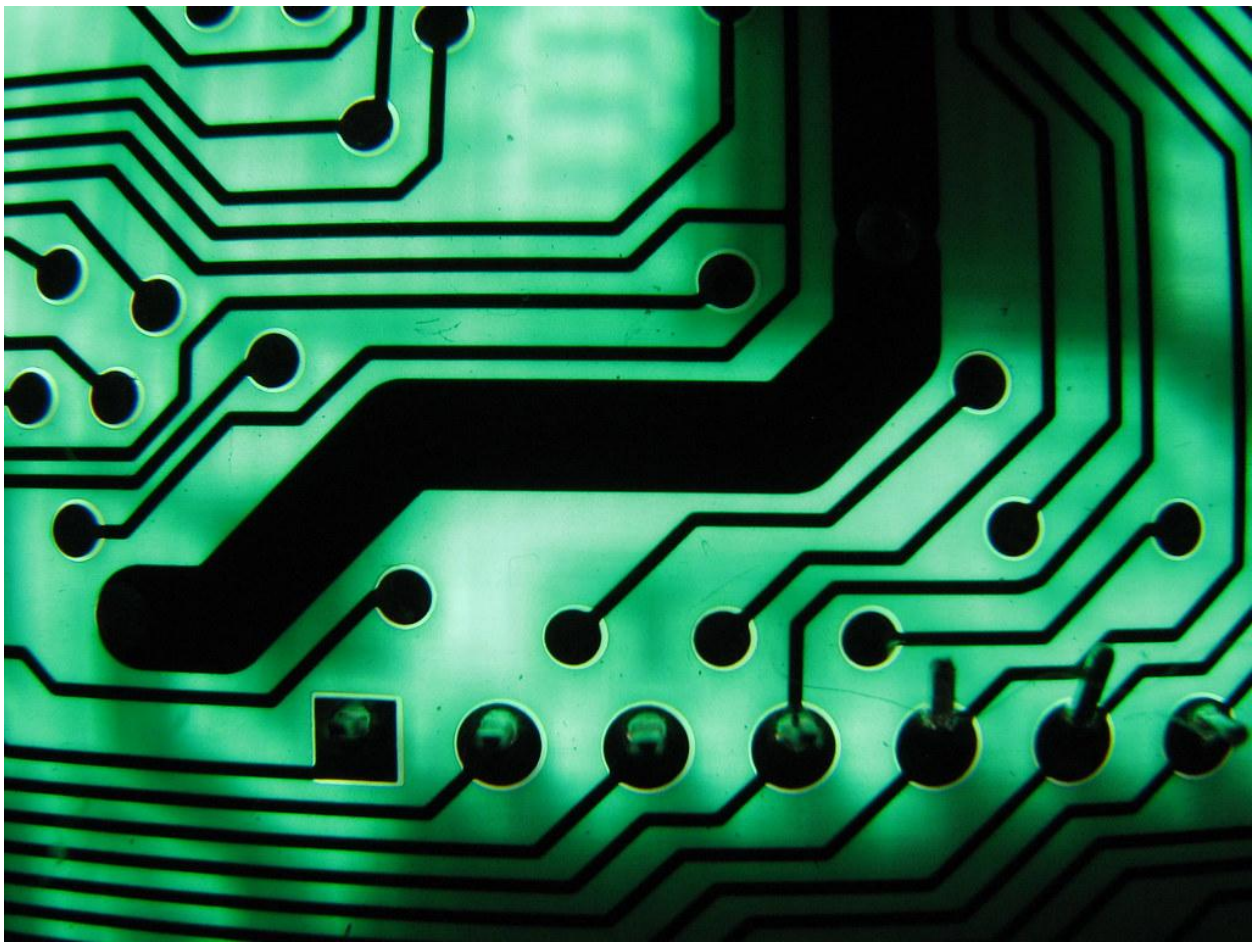
EEC 180 DIGITAL SYSTEMS II

Lab #3

Lab Report



Introduction

The purpose of this lab is to program an FPGA to do 3-bit adder arithmetic.

Tobin Joseph

Results

1 Prelab

Part 1.1 fa.v Draft

```
//Verilog fa.v module.
module fa (in1, in2, cin, sum, cout);

	//Declare inputs, outputs, and internal variables.
	input in1,
		in2,
		cin;
	output sum,
		Cout;

	// use Boolean expressions
	assign Sum  = (in1 ^ in2) ^ cin;
	assign Cout = (in1 & in2) | (in2 & cin) | (cin & in1);
endmodule
```

Figure 1.1.1 fa.v Draft Code.

Part 1.2 add3.v Draft

```
//Verilog add3.v module.
module add3(a,b,cin,sum);

//Declare inputs, outputs, and internal variables.
input [2:0]a,b;
input cin;
output wire [3:0]sum;
```

```
// instantiating four full-adder circuits - ripple carry
fa FA1(a[0],b[0],cin,sum[0],cout1);
fa FA2(a[1],b[1],cout1,sum[1],cout2);
fa FA3(a[2],b[2],cout2,sum[2],sum[3]);

endmodule
```

Figure 1.2.1 add3.v Draft Code.

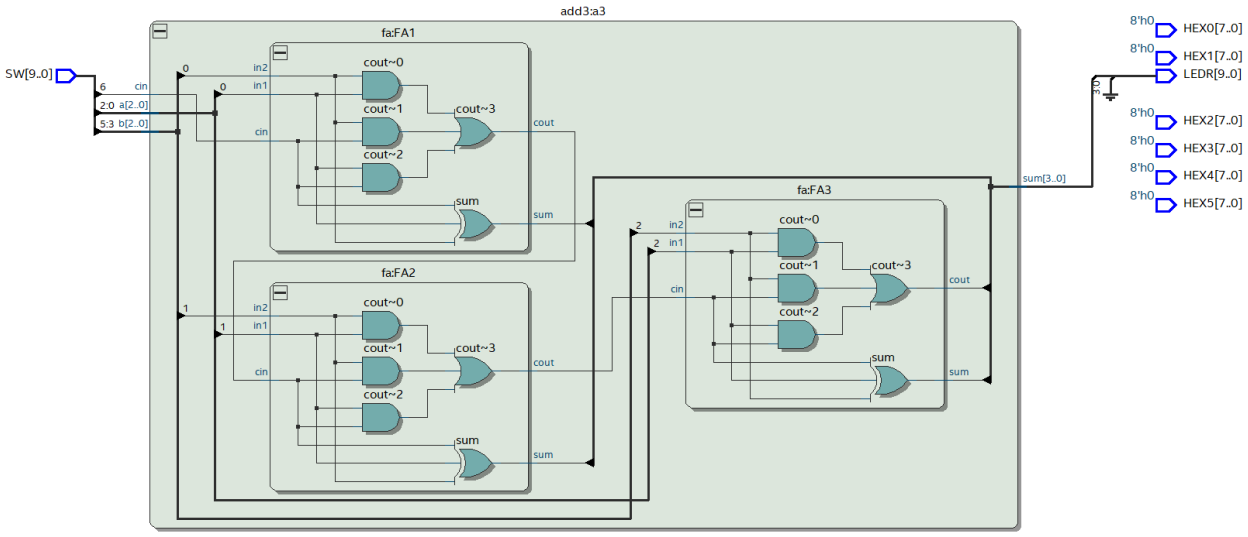


Figure 1.2.2 add3.v Draft Circuit.

1.3 add3_error.v Draft

```
//Verilog add3_error.v module.
module add3_error(a,b,cin,sum);

//Declare inputs, outputs, and internal variables.
```

```
input [2:0]a,b;
input cin;
output wire [3:0]sum;
wire [3:0] sumt;
reg [3:0] sumx;

fa FA1(a[0],b[0],cin,sumt[0],cout1);
fa FA2(a[1],b[1],cout1,sumt[1],cout2);
fa FA3(a[2],b[2],cout2,sumt[2],sumt[3]);

   always @(a)
begin
      case (a)
        4 :
            begin
                  case (b)
                        7 : sumx = 4'b1100;
                     default : sumx = sumt;
                  endcase
            end
        7 :
            begin
                  case (b)
                        4 : sumx = 4'b1100;
                     default : sumx = sumt;
                  endcase
            end
```

```
        default : sumx = sumt;
      endcase
end

assign sum = sumx;
endmodule
```

Figure 1.3.1 add3_error.v Draft Code.

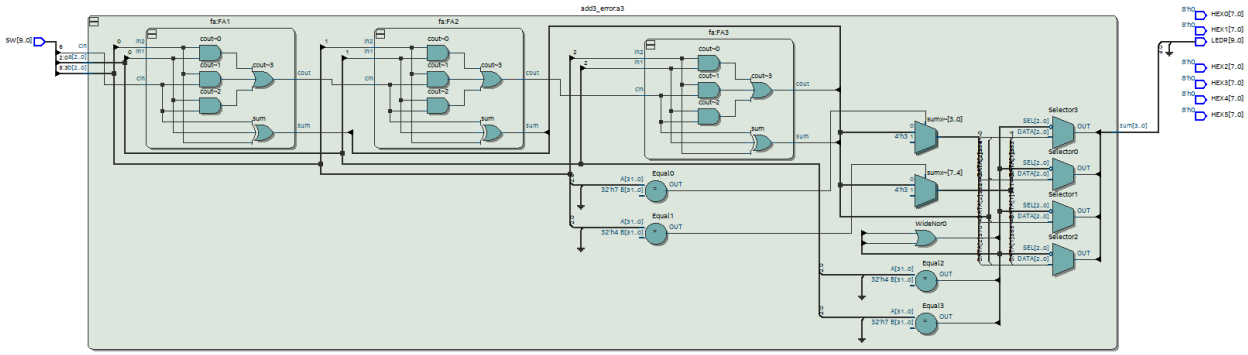


Figure 1.3.2 add3_error.v Draft Circuit.

1.4 add3_ref.v Draft

```
//Verilog add3_ref.v module.
module add3_ref(a,b,cin,sum);

//Declare inputs, outputs, and internal variables.
input [2:0]a,b;
input cin;
output wire [3:0]sum;


assign sum = a + b + cin;
```

```
endmodule
```

Figure 1.4.1 add3_ref.v Draft Code.

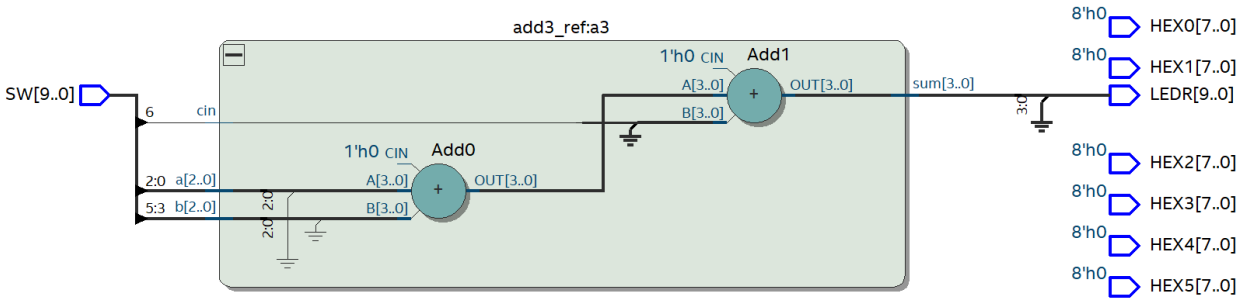


Figure 1.42 add3_ref.v Draft Circuit.

Discussion

When we ...

Conclusion

We can see ...