Final Project: Real Time Digital Stopwatch Design and FPGA Implementation

Kelsea Canaday

U0000010532

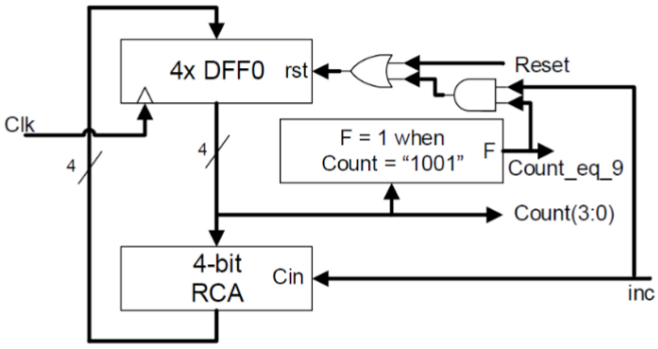
5/2/23

Objective:

The objective of this lab was to combine all of the concepts learned throughout the semester to implement a real time digital stopwatch.

Project Description:

This project involves making a stopwatch using Verilog code, the Quartus program, and an FPGA board. There is a start/stop button the switches the stopwatch on and off when pressed. There is also a switch that resets the stopwatch to 00:00 when switched.



This diagram explains how the count10 module works. If the Count variable does not equal 9, then the Count variable is incremented by one. When the Count variable is equal to 9, the value is reset back to 0. The count 6 module is similar in which it acts the same way, but the deciding number is 5 instead of 9.

RCA Code (with HA):

```
module HA(A,B,sum,carry);

input A,B;
output sum, carry;

assign sum = A^B;
assign carry = A&B;
endmodule



module RCA(A,cin,sum,carry);

input cin;
input [3:0] A;
output [3:0]sum;
output carry;
wire [2:0]g;

HA h1(A[0],cin, sum[0], g[0]);
HA h2(A[1],g[0],sum[1],g[1]);
HA h3(A[2],g[1],sum[2],g[2]);
HA h4(A[3],g[2],sum[3],carry);
endmodule
```

BCD_Display Code:

```
module BCD_Display(A,B,C,D,a,b,c,d,e,f,g);

input A,B,C,D;
output a,b,c,d,e,f,g;

assign a = (B&~D)|(~A&~B&~C&D);
assign b = (B&~C&D)|(B&C&~D);
assign c = (~B&C&~D);
assign d = (B&~C&~D)|(B&C&D)|(~B&~C&D);
assign e = D|(B&~C);
assign f = (~B&C)|(C&D)|(~A&~B&D);
assign g = (B&C&D)|(~A&~B&~C);
endmodule
```

count10 Code:

```
module count10(clk,inc,reset,count,count_eq_9);

input clk, inc, reset;
output [3:0]count;
output count_eq_9;
wire rst1, rst2;
wire [3:0] rca;

assign rst1 = (inc & count_eq_9);
assign rst2 = (rst1 | reset);

RCA r1(count, inc, rca);
DFF0_4 dff(rca, clk, rst2, count);
assign count_eq_9 = (count==4'b1001)?1:0;
endmodule
```

count6 Code:

```
module count6(clk,inc,reset,count,count_eq_6);
input clk, inc, reset;
output [3:0]count;
output count_eq_6;
wire rst1, rst2;
wire [3:0] rca;

assign rst1 = (inc & count_eq_6);
assign rst2 = (rst1 | reset);

RCA r1(count, inc, rca);
DFF0_4 dff(rca, clk, rst2, count);

assign count_eq_6 = (count==4'b0110)?1:0;

endmodule
```

stopwatch Code:

```
module stopwatch(start_stop, clock, reset, a1, b1, c1, d1, e1, f1, g1,
a2, b2, c2, d2, e2, f2, g2, a3, b3, c3, d3, e3, f3, g3, a4, b4, c4, d4,
e4, f4, g4);

input start_stop, clock, reset;
output a1, b1, c1, d1, e1, f1, g1, a2, b2, c2, d2, e2, f2, g2, a3, b3,
c3, d3, e3, f3, g3, a4, b4, c4, d4, e4, f4, g4;
wire [3:0] o1;
wire [3:0] o2;
wire [3:0] o3;
wire [3:0] o4;
wire i1, i2, i3, q, q1, q2, q3, q4, co;

assign q1 = (q&i1);
assign q2 = (q1&i2);
assign  q3 = (q2&i3);

clk_divider(clock, 1'b0, co);
count10 cc1(co, q, reset, o1, i1);
count10 cc2(co, q1, reset, o2, i2);
count10 cc3(co, q2, reset, o3, i3);
count6 cc6(co, q3, reset, o4, q4);
TFF0(1'b1, start_stop, 1'b0, q);

BCD_Display bb1(o1[3],o1[2],o1[1],o1[0],a1, b1, c1, d1, e1, f1, g1);
BCD_Display bb2(o2[3],o2[2],o2[1],o2[0],a2, b2, c2, d2, e2, f2, g2);
BCD_Display bb3(o3[3],o3[2],o3[1],o3[0],a3, b3, c3, d3, e3, f3, g3);
BCD_Display bb4(o4[3],o4[2],o4[1],o4[0],a4, b4, c4, d4, e4, f4, g4);
endmodule
```

Project Comment:

It was very fun to see how concepts covered in previous labs were all incorporated into this project. Specifically, it was interesting to see the count10 module made in the eighth lab being used significantly in the project. Beginning this class, I knew very little about using logic gates and didn't know what Verilog was. Now, I feel very proficient in using both.

Conclusion:

Creating a digital stopwatch using Verilog and an FPGA board was challenging but a very fun experience. The previous labs prepared me well for this final project.