FINAL PROJECT: REAL TIME DIGITAL STOPWATCH DESIGN AND FPGA IMPLEMENTATION

By: Vladjimir Nicolas

CS

12/8/23

Objectives:

1. Combining all the concepts learned throughout the semester to implement a Real Time Digital Stopwatch.
2. The stopwatch should have the following features: - a Start/Stop button (single button) that switches the watch on/off when depressed. - It should start by displaying 00:00 when powered on

GOAL

An opportunity to ‘connect the dots’ between theoretical knowledge and practical implementations. The project may require time commitment; however, I can assure you that the outcome will be extremely satisfying! After the successful completion of this project, you can proudly say that you have designed something “Cool”!

RCA and Half Adder Module Code:

```
module Half_Adder(X,Y,Sum, Carry);
input X,Y;
output Sum, Carry;
assign Carry=X&Y;
assign Sum = X^Y;
endmodule

module RCA(A,Cin, Sum);
input [3:0] A;
input Cin;
wire [2:0] C;
wire OV;
output [3:0]Sum;
Half_Adder Half0(A[0], Cin, Sum[0], C[0]);
Half_Adder Half1 (A[1], C[0], Sum[1], C[1]);
Half_Adder Half2(A[2], C[1], Sum[2], C[2]);
Half_Adder Half3(A[3], C[2], Sum[3], OV);
endmodule
```

BCD Display Code:

```
module BCD(B,D);
input [3:0]B;
output[6:0]D;
assign D[0] = (B[2]&~B[0])|(~B[3]&~B[2]&~B[1]&B[0]);
assign D[1] = B[2]&(B[1]^B[0]);
assign D[2] = ~B[2]&B[1]&~B[0];
assign D[3] = (B[2]&~B[1]&~B[0])|(~B[2]&~B[1]&B[0])|(B[2]&B[1]&B[0]);
assign D[4] = (B[2]&~B[1])|B[0];
assign D[5] = (B[1]&B[0])|(~B[2]&B[1])|(~B[3]&~B[2]&B[0]);
assign D[6] = (~B[3]&~B[2]&~B[1])|(B[2]&B[1]&B[0]);
endmodule
```

DFFO and Clock Divider:

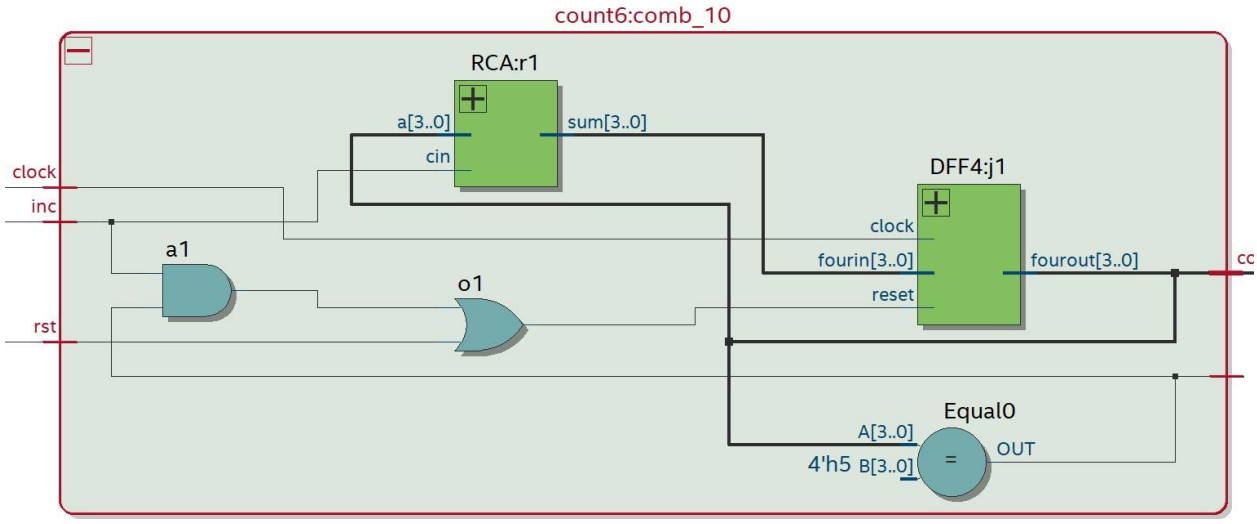
```
module DFF0(data_in,clock,reset, data_out);
input data_in;
input clock,reset;
output reg data_out;
always@(posedge clock)
begin
if(reset)
data_out<=1'b0;
else
data_out<=data_in;
end
endmodule

module clk_divider(clock, rst, clk_out);
input clock, rst;
output clk_out;
wire [18:0] din;
wire [18:0] clkdiv;
DFFO dff_inst0( .data_in(din[0]), .clock(clock), .reset(rst), .data_out(clkdiv[0])
);
genvar i;
generate
for (i = 1; i < 19; i=i+1)
begin: dff_gen_label DFFO dff_inst (
.data_in (din[i]),
.clock(clkdiv[i-1]), .reset(rst),
);
end
.data_out(clkdiv[i])
endgenerate
assign din = ~clkdiv;
assign clk_out = clkdiv[18];
endmodule
```

Count10 Module Code:

```
module count10(clock, inc, reset, Count, count_eq_9);
input clock, inc, reset;
output [3:0] Count;
reg [3:0] test;
wire [3:0] e;
output count_eq_9;
wire b,c;
DFFO Flip0(e[0], clock, c, Count[0]);
DFFO Flip1(e[1], clock, c, Count[1]);
DFFO Flip2(e[2], clock, c, Count[2]);
DFFO Flip3(e[3], clock, c, Count[3]);
RCA Ripple1(Count, inc, e);
assign count_eq_9 = (Count[3]&~Count[2]&~Count[1]&Count[0])?1:0; and g1 (b, count_eq_9, inc);
or g2(c, b, reset);
endmodule
```

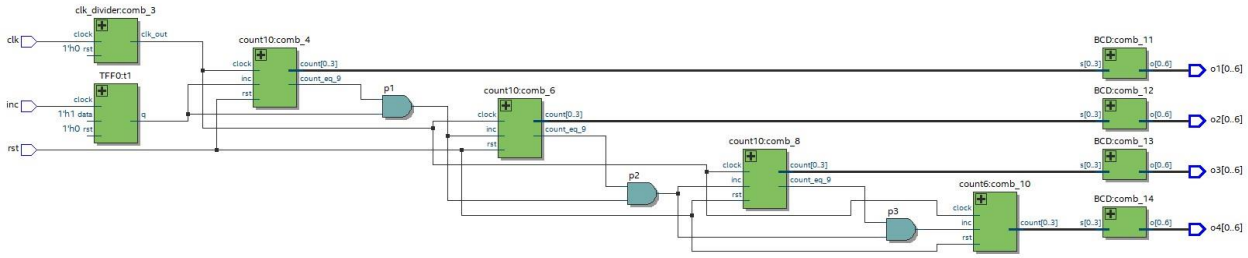
Count6 Module Code:

```
module count6(clock, inc, reset, Count, count_eq_5);
input clock, inc, reset;
output [3:0] Count;
wire [3:0] e;
output count_eq_5;
wire b,c;
DFFO Flip0(e[0], clock, c, Count[0]);
DFFO Flip1(e[1], clock, c, Count[1]);
DFFO Flip2(e[2], clock, c, Count[2]);
DFFO Flip3(e[3], clock, c, Count[3]);
RCA Ripple1(Count, inc, e);
assign count_eq_5 = (~Count[3]&Count[2]&~Count[1]&Count[0])?1:0;
and g1 (b,count_eq_5, inc);
or g2(c, b, reset);
endmodule
```


count6:comb_10
RCA:r1
a[3..0]
cin
sum[3..0]
DFF4:j1
clock
fourin[3..0]
reset
fourout[3..0]
clock
inc
rst
a1
o1
Equal0
A[3..0]
4'h5
B[3..0]
=
OUT

Final Project Code:

```
module stopwatch(clock, inc, reset, Cout, Out,sysOut,printOut, a, b,c,d,start); input clock, reset,inc;
input start;
wire clock_out,intReset;
wire w,x,y,z,l,m,n;
output [6:0]a,b,c,d;
output [3:0] Cout, Out, sysOut, printOut;
clk_divider clk(.clock(clock), .rst(reset), .clk_out(clock_out));
TFFO T1 (inc, start, reset, intReset);
count10 c1(clock_out, intReset, reset, Out,w);
BCD b1(Out,a);
and g1 (x, intReset, w);
count10 c2(clock_out, x, reset, Cout, y);
BCD b2(Cout, b);
and g2(z, x, y);
count10 c3(clock_out, z, reset, sysOut, l);
BCD b3(sysOut, c);
and g3(m,l,z);
count6 c4(clock_out, m, reset, printout, n);
BCD b4(printOut, d);
endmodule
```



Project Comment:

I've acquired a profound comprehension of digital logic design through my coursework, specifically delving into the realms of gates and flip flops. This lab assignment presented an opportunity for me to put into practice the concepts I absorbed in class, applying them to a tangible real-world project. From crafting a 4x1 multiplexer to building a full adder using multiplexers and other gates, I successfully applied the skills accumulated over the semester. The pinnacle of this endeavor was the development of a stopwatch, made feasible by the insights

gained into digital circuits, Verilog, and FPGA technology. This experience not only deepened my understanding of digital logic design but also honed my ability to employ this knowledge in practical scenarios. In essence, this project served as a valuable learning journey, significantly augmenting my proficiency and expertise in digital logic design.

Conclusion:

The Real-Time Digital Stopwatch project integrates a range of concepts assimilated throughout the semester, showcasing a holistic grasp of digital logic design. This undertaking involves devising a 4-bit Ripple Carry Adder, a BCD-to-Seven Segment Display Controller, and counter modules for module 10 and 6. Additionally, a stopwatch block diagram illustrates the interconnectedness of all modules to achieve the desired functionality. The project elevates the learner's proficiencies in various domains, including Verilog programming, digital logic design, and FPGA implementation. It also prompts the learner to bridge theoretical knowledge with practical applications, necessitating the application of concepts in a real-world context. Upon completing this project, learners can derive a sense of achievement and take pride in their ability to fashion a functional and valuable device.