CS M152A Lab 2

Ryan Riahi

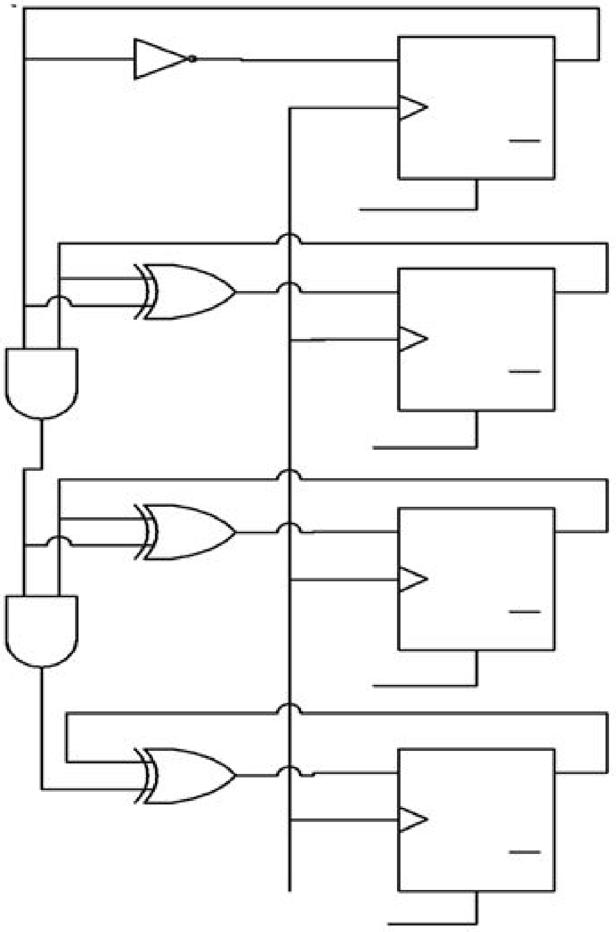
Introduction

The purpose of this laboratory was to to design and test various clock waveforms on a digital system. I used the system clock on the machine to generate various clocking systems of different frequencies and duty cycles. The modules to design will take the system clock and a reset signal as the input and output the derived clock.

Design Description

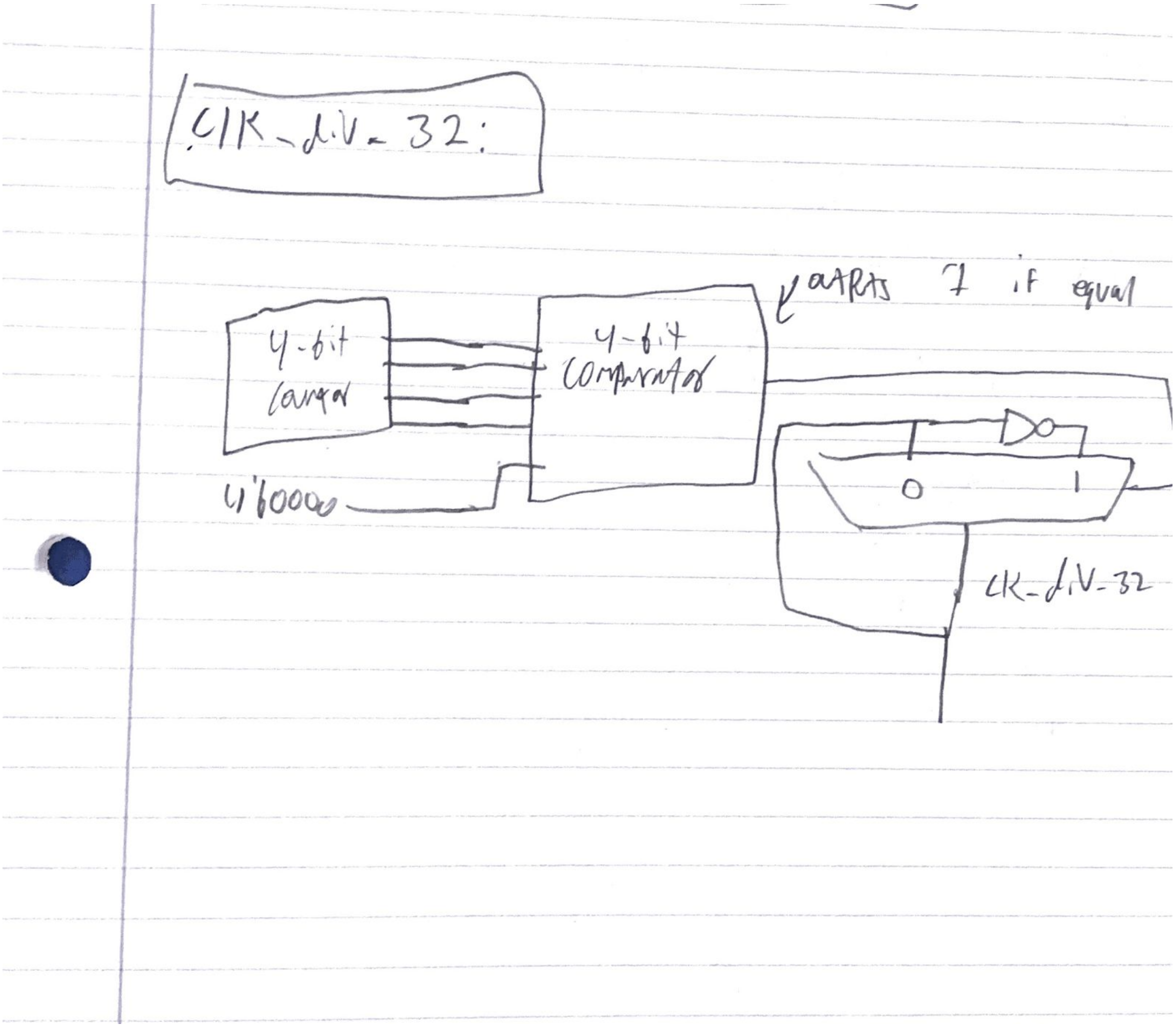
The overall design of each clock is described below.

4 bit counter schematic:



Clk_div_2, Clk_div_4, Clk_div_8, Clk_div_16 are all obtain by setting the to the least significant to most significant bits respectively of the counter.

The next schematic is for clk_div_32:



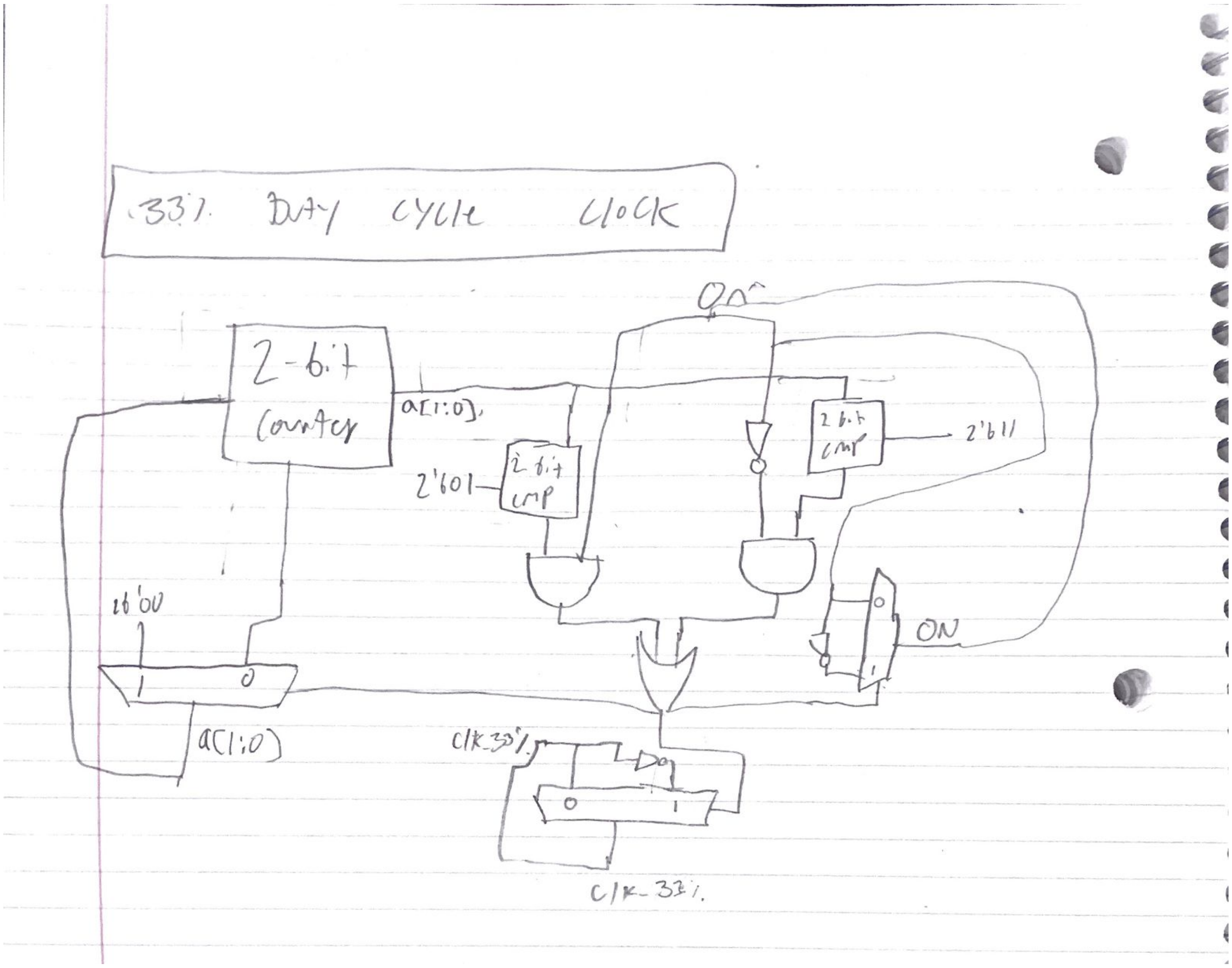
Essentially we just flip clk_div_32 whenever the counter overflows to 0.

For clk_div_28, we perform essentially the same thing except we compare the counter with the value 4’b1101 instead of 4’b0000. Additionally, when we reach this value, we reset the counter.

The same is true for clk_div_5 except we use a 3 bit counter instead and compare the signal to 3’b100 and reset the counter and flip clk_div_5 whenever this is hit.

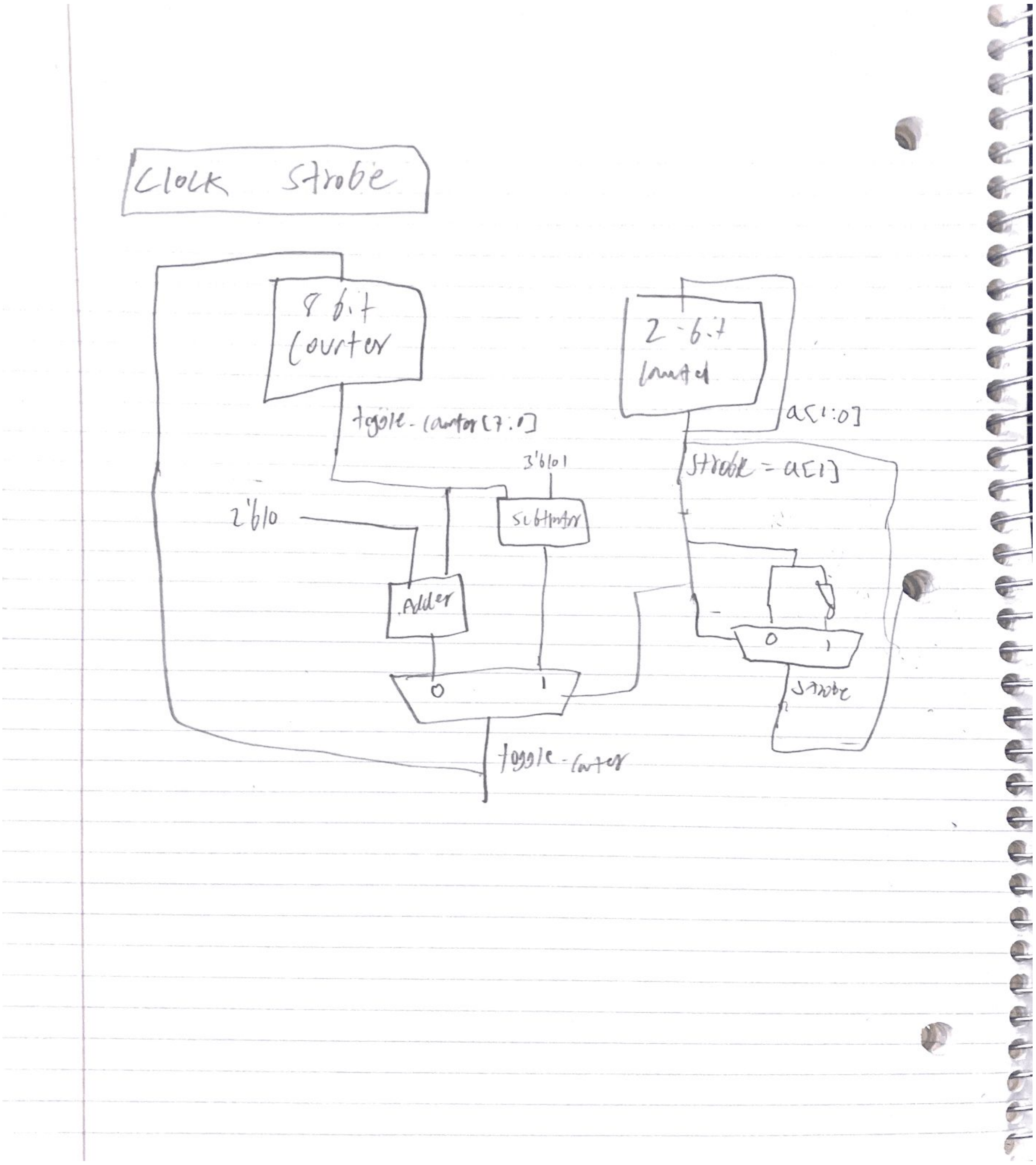
Lastly, the same is true for clk_div_100 and clk_div_200, except we use a 7 bit counter and when that counter is 100, we set clk_div_100 on and then on the next posedge we set it back to 0. Lastly, we flip clk_div_200 every time clk_div_100 is on.

Next is the schematic for the 33% duty cycle clocks and the or of them:



We essentially create two of these that trigger on different edges of the clock and then or the final outputs together.

Last but not least is the schematic for the strobe counter:

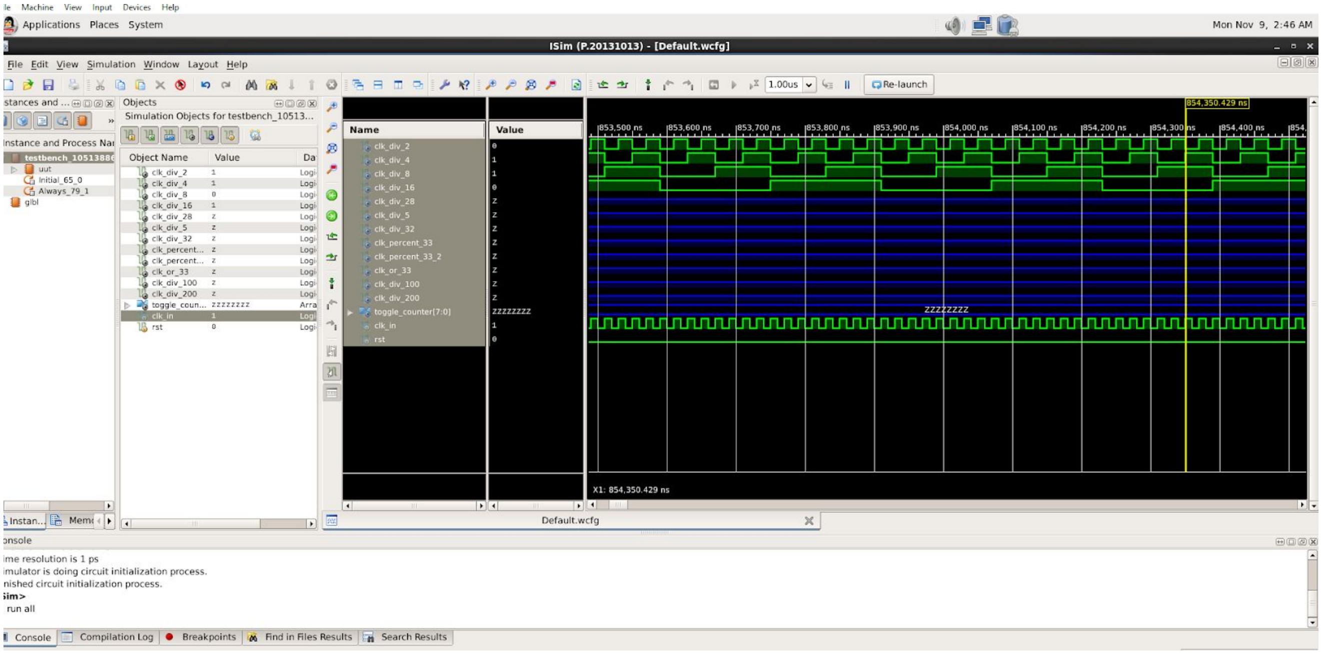


This is an 8bit counter that counts up by two except when strobe is active, in which case it will subtract by 4 instead.

Simulation Documentation

In order to test my design, I ran a simulation using VISM. The simulation was run on a test bench that alternates the clk_in value to simulate as the system clock and pass in as inputs to the various modules.

First we look at the simulation of the clk_div_2 module, which contains clk_div_2, clk_div_4, clk_div_8, and clk_div_16:



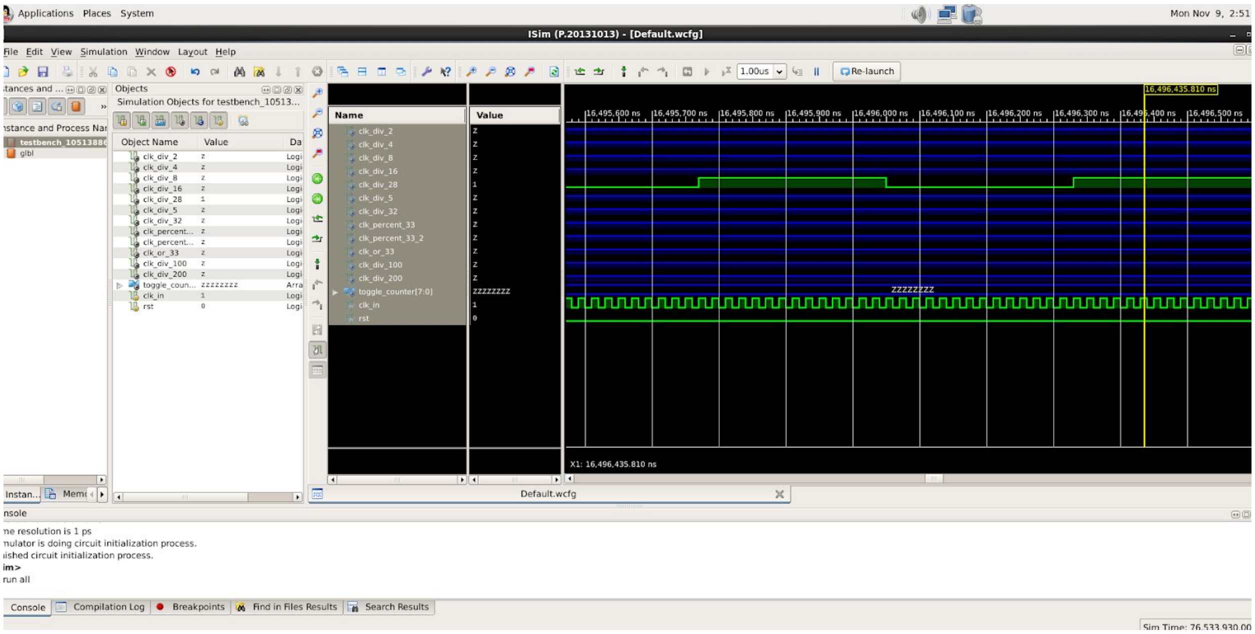
Object Name	Value	Da
clk_div_2	1	Logi
clk_div_4	1	Logi
clk_div_8	0	Logi
clk_div_16	1	Logi
clk_div_28	z	Logi
clk_div_5	z	Logi
clk_div_32	z	Logi
clk_percent...	z	Logi
clk_percent...	z	Logi
clk_or_33	z	Logi
clk_div_100	z	Logi
clk_div_200	z	Logi
toggle_coun...	zzzzzzzz	Arra
clk_in	1	Logi
rst	0	Logi

Name	Value
clk_div_2	0
clk_div_4	1
clk_div_8	1
clk_div_16	0
clk_div_28	Z
clk_div_5	Z
clk_div_32	Z
clk_percent_33	Z
clk_percent_33_2	Z
clk_or_33	Z
clk_div_100	Z
clk_div_200	Z
toggle_counter[7:0]	ZZZZZZZZ
clk_in	1
rst	0

```
onsole
ime resolution is 1 ps
imulator is doing circuit initialization process.
nished circuit initialization process.
im>
 run all
```

As can be seen, the waveforms for each clk is accurate. clk _div_16 is twice as large as clk_div_8 which is twice as large as clk_div_4 ...

Next we look at the simulation for clk_div_28:



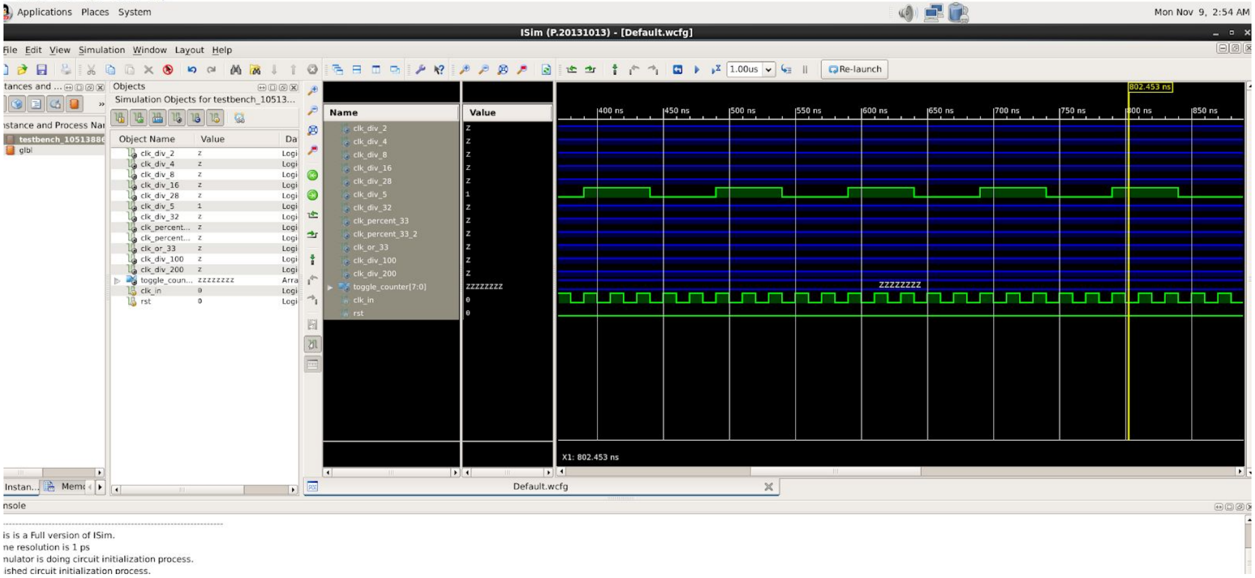
Object Name	Value	Da
clk_div_2	z	Logi
clk_div_4	z	Logi
clk_div_8	z	Logi
clk_div_16	z	Logi
clk_div_28	1	Logi
clk_div_5	z	Logi
clk_div_32	z	Logi
clk_percent...	z	Logi
clk_percent...	z	Logi
clk_or_33	z	Logi
clk_div_100	z	Logi
clk_div_200	z	Logi
toggle_coun...	zzzzzzzz	Arra
clk_in	1	Logi
rst	0	Logi

Name	Value
clk_div_2	Z
clk_div_4	Z
clk_div_8	Z
clk_div_16	Z
clk_div_28	1
clk_div_5	Z
clk_div_32	Z
clk_percent_33	Z
clk_percent_33_2	Z
clk_or_33	Z
clk_div_100	Z
clk_div_200	Z
toggle_counter[7:0]	ZZZZZZZZ
clk_in	1
rst	0

```
nsole
ne resolution is 1 ps
nulator is doing circuit initialization process.
ished circuit initialization process.
im>
run all
```

As can be seen, the waveforms for clk_div_28 is clearly a divide-by-28 clk cycle

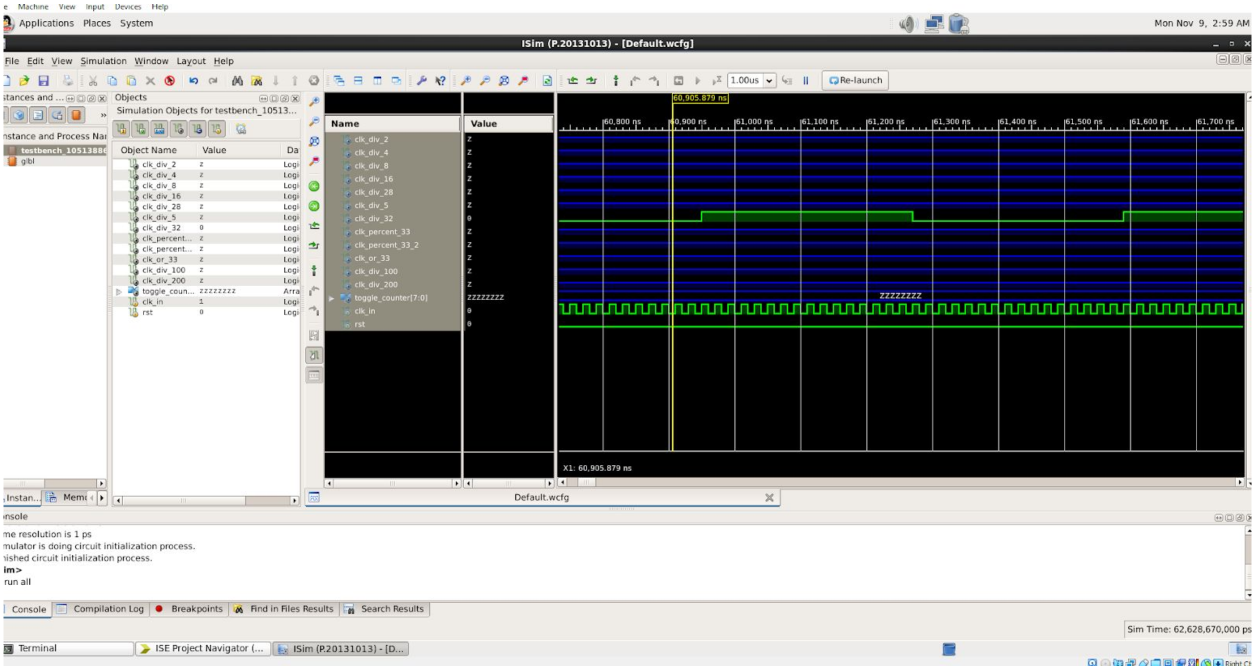
Next we look at the simulation for clk_div_5:



```
nsole
---------------------------------------------------------------
is is a Full version of ISim.
ne resolution is 1 ps
nulator is doing circuit initialization process.
ished circuit initialization process.
```

As can be seen, clk_siv_5 is a divide by 5 clock with a 50% duty cycle

Next we look at the simulation for clk_div_32:

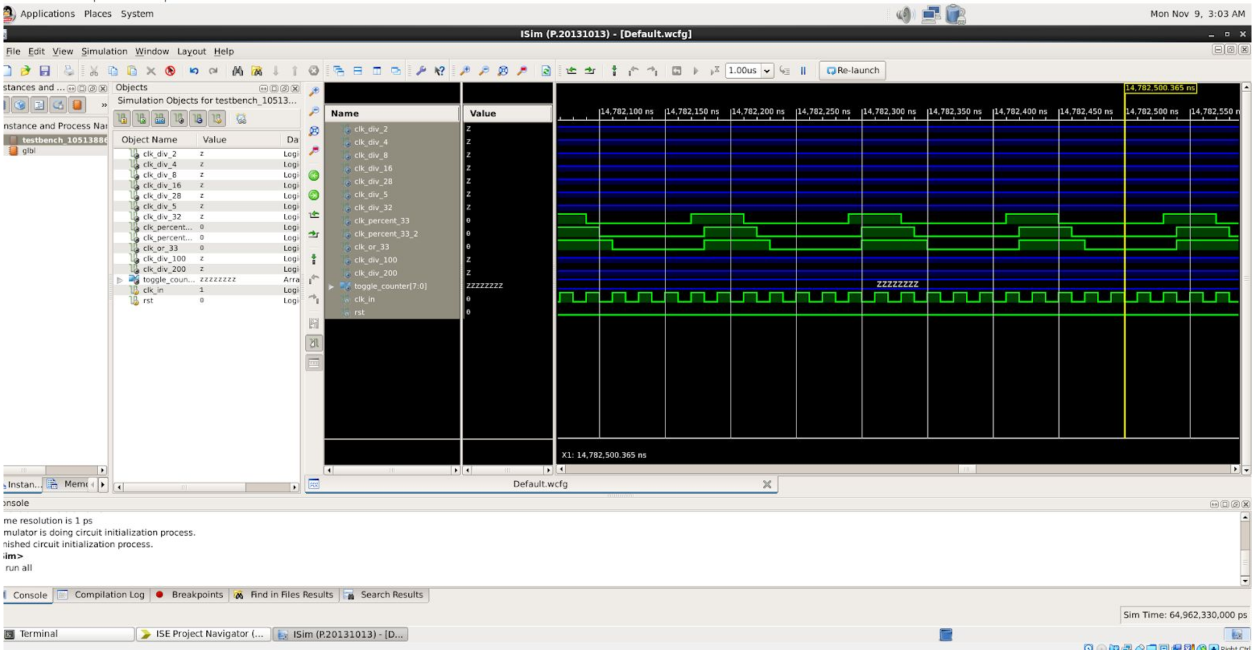


```
nsole
me resolution is 1 ps
mulator is doing circuit initialization process.
nished circuit initialization process.
im>
run all
```

Sim Time: 62,628,670,000 ps

As can be seen, clk_siv_32 is a divide by 32 clock

Next we look at the simulation of the two 33% duty cycle clocks and the or of them:



Object Name	Value	Da
clk_div_2	z	Logi
clk_div_4	z	Logi
clk_div_8	z	Logi
clk_div_16	z	Logi
clk_div_28	z	Logi
clk_div_5	z	Logi
clk_div_32	z	Logi
clk_percent...	0	Logi
clk_percent...	0	Logi
clk_or_33	0	Logi
clk_div_100	z	Logi
clk_div_200	z	Logi
toggle_coun...	zzzzzzzz	Arra
clk_in	1	Logi
rst	0	Logi

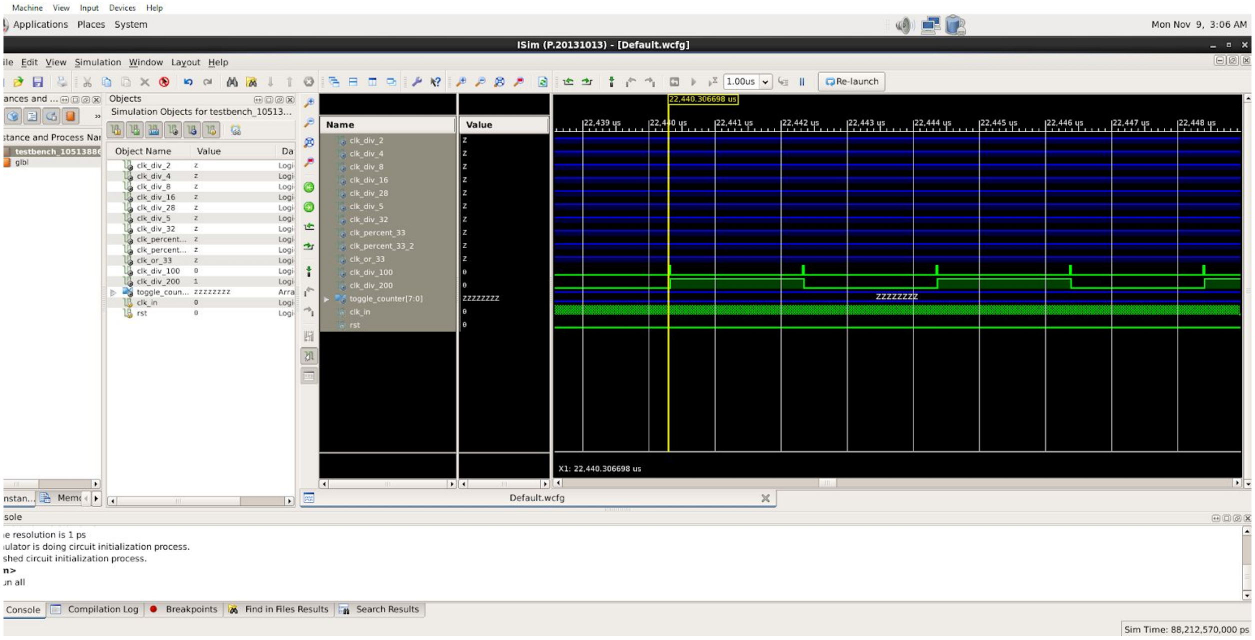
Name	Value
clk_div_2	Z
clk_div_4	Z
clk_div_8	Z
clk_div_16	Z
clk_div_28	Z
clk_div_5	Z
clk_div_32	Z
clk_percent_33	0
clk_percent_33_2	0
clk_or_33	0
clk_div_100	Z
clk_div_200	Z
toggle_counter[7:0]	ZZZZZZZZ
clk_in	0
rst	0

```
onsole
me resolution is 1 ps
mulator is doing circuit initialization process.
nished circuit initialization process.
im>
 run all
```

Sim Time: 64,962,330,000 ps

As can be seen, the two 33% duty cycle clocks are active 33% of the time and are offset by one edge. The value below them, the clk_or_33 is a 41% duty cycle clock that is active when either 33% duty cycle clock is active.

Next we look at the clk_div_200 simulation:



Object Name	Value	Da
clk_div_2	z	Logi
clk_div_4	z	Logi
clk_div_8	z	Logi
clk_div_16	z	Logi
clk_div_28	z	Logi
clk_div_5	z	Logi
clk_div_32	z	Logi
clk_percent...	z	Logi
clk_percent...	z	Logi
clk_or_33	z	Logi
clk_div_100	0	Logi
clk_div_200	1	Logi
toggle_coun...	zzzzzzzz	Arra
clk_in	0	Logi
rst	0	Logi

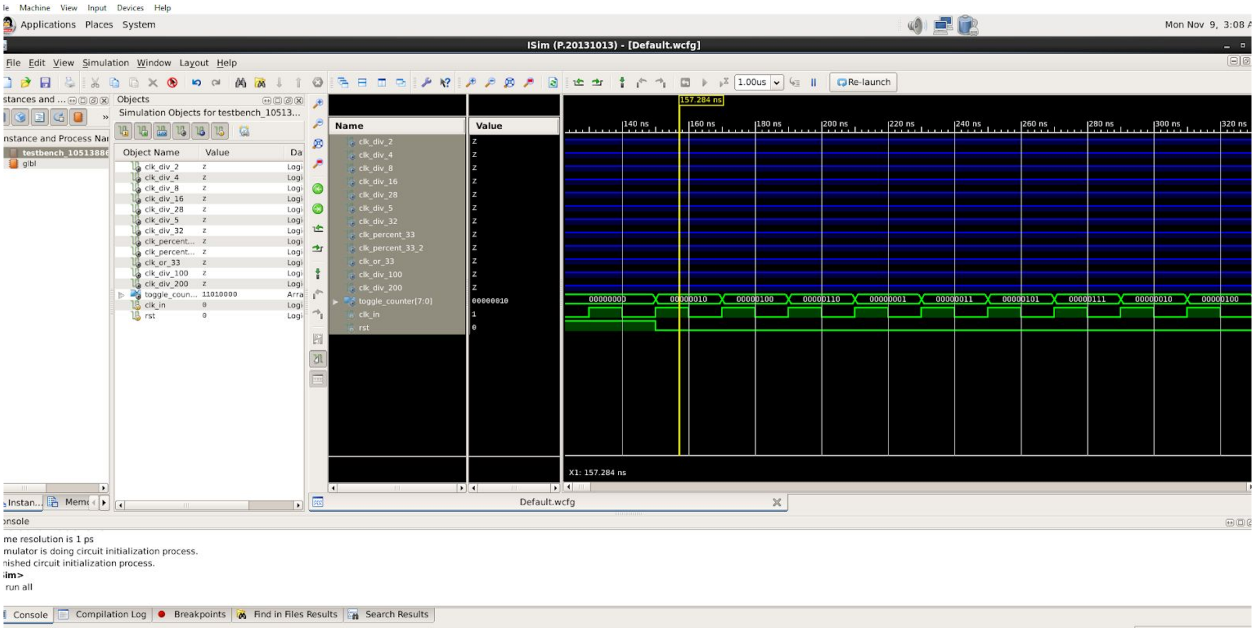
Name	Value
clk_div_2	Z
clk_div_4	Z
clk_div_8	Z
clk_div_16	Z
clk_div_28	Z
clk_div_5	Z
clk_div_32	Z
clk_percent_33	Z
clk_percent_33_2	Z
clk_or_33	Z
clk_div_100	0
clk_div_200	0
toggle_counter[7:0]	ZZZZZZZZ
clk_in	0
rst	0

```
sole
e resolution is 1 ps
ulator is doing circuit initialization process.
shed circuit initialization process.
n>
un all
```

Sim Time: 88,212,570,000 ps

As can be seen, the clk_div_100 is only active 1% of the time, making it a 1% duty cycle clock. Then, the clk_div_200 is switches every time clk_div_100 is active, making it a 50% duty cycle clock.

Next we look at the strobe clock simulation:



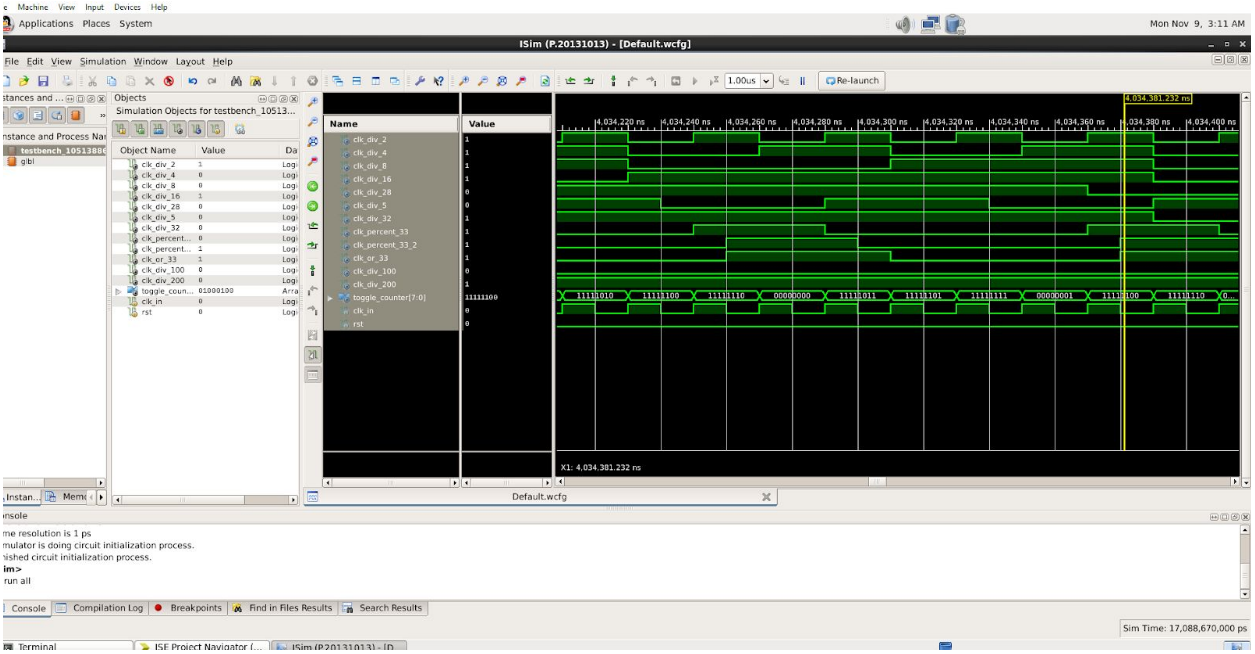
Object Name	Value	Da
clk_div_2	z	Logi
clk_div_4	z	Logi
clk_div_8	z	Logi
clk_div_16	z	Logi
clk_div_28	z	Logi
clk_div_5	z	Logi
clk_div_32	z	Logi
clk_percent...	z	Logi
clk_percent...	z	Logi
clk_or_33	z	Logi
clk_div_100	z	Logi
clk_div_200	z	Logi
toggle_coun...	11010000	Arra
clk_in	0	Logi
rst	0	Logi

Name	Value
clk_div_2	Z
clk_div_4	Z
clk_div_8	Z
clk_div_16	Z
clk_div_28	Z
clk_div_5	Z
clk_div_32	Z
clk_percent_33	Z
clk_percent_33_2	Z
clk_or_33	Z
clk_div_100	Z
clk_div_200	Z
toggle_counter[7:0]	00000010
clk_in	1
rst	0

```
me resolution is 1 ps
mulator is doing circuit initialization process.
nished circuit initialization process.
im>
run all
```

As can be see, the strobe clock counts in the following manner: 0, 2, 4, 6, 1, 3, 5, 7 …

Lastly, we will look at all of them simulated together:



Object Name	Value	Da
clk_div_2	1	Logi
clk_div_4	0	Logi
clk_div_8	0	Logi
clk_div_16	1	Logi
clk_div_28	0	Logi
clk_div_5	0	Logi
clk_div_32	0	Logi
clk_percent...	0	Logi
clk_percent...	1	Logi
clk_or_33	1	Logi
clk_div_100	0	Logi
clk_div_200	0	Logi
toggle_coun...	01000100	Arra
clk_in	0	Logi
rst	0	Logi

Name	Value
clk_div_2	1
clk_div_4	1
clk_div_8	1
clk_div_16	1
clk_div_28	0
clk_div_5	0
clk_div_32	1
clk_percent_33	1
clk_percent_33_2	1
clk_or_33	1
clk_div_100	0
clk_div_200	1
toggle_counter[7:0]	11111100
clk_in	0
rst	0

```
me resolution is 1 ps
mulator is doing circuit initialization process.
nished circuit initialization process.
im>
run all
```

Conclusion

In conclusion, this lab took me some time to understand, especially in terms of the mechanics for creating the clocks. However, once that was understood, the actual programming of the clocks was not all too difficult. The main problem for me was the conceptual understanding of everything that was going on. With lots of time however, I began to understand and now I am happy to say my understanding of clocking systems has greatly increased!