UNIVERSITY OF TEXAS AT DALLAS

800 W Campbell Rd, Richardson, TX 75080, USA

VLSI DESIGN (EECT 6325)

Project Done On:

SIMULATION OF DIVIDE BY THREE CIRCUIT WITH 50% DUTY CYCLE

Team Members:

GOKUL SAI RAGHUNATH	GXR200021
ABHISHEK MAHESH KUMAR	AXM200255
SOMA RAJ KUMAR	RXS190135

Professor:

PROF. CARL SECHEN

ACKNOWLEDGEMENT

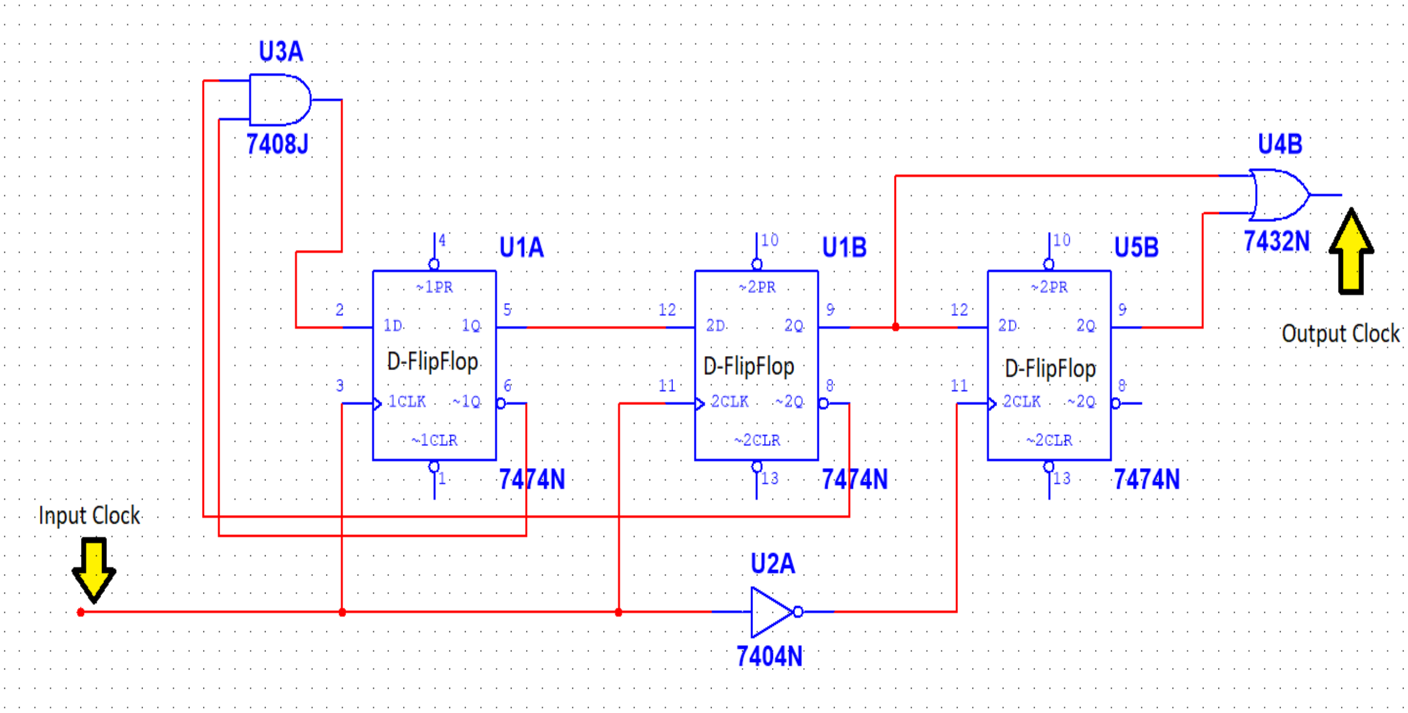
We are grateful to Professor CARL SECHEN for providing us an opportunity to explore and conduct projects based on VLSI Design.

We also take this opportunity to express our gratitude to MR VAIBHAV KUMARSWAMY SALIMATH for his guidance in conducting the project.

INTRODUCTION

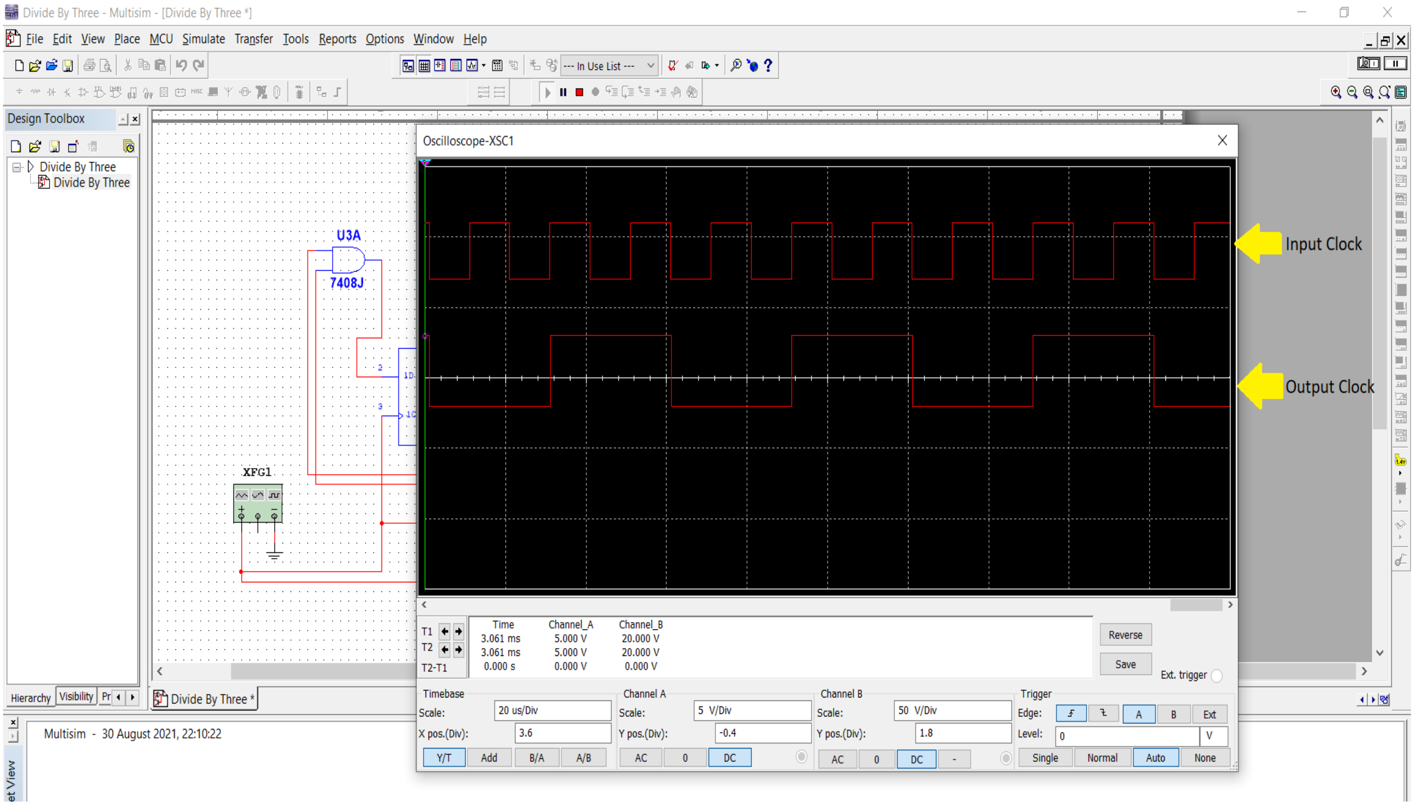
In this project we are demonstrating a clock divider of an odd integer 3, with a 50% duty cycle. In some designs it should be provided with a number of phase-related clocks to various components. In most cases, one can generate the needed clocks by dividing a master clock by a power of two (synchronous division). However, sometimes it is desirable to divide a frequency by an odd number. In these cases, such circuits help in overcoming the problem.

CIRCUIT DIAGRAM



The Circuit is constructed using Multisim Software by National Instruments. In this circuit we have used three D-Flipflops, a NOT gate, an AND gate and an OR gate.

SIMULATION IN MULTISIM



	Time	Channel_A	Channel_B
T1	3.061 ms	5.000 V	20.000 V
T2	3.061 ms	5.000 V	20.000 V
T2-T1	0.000 s	0.000 V	0.000 V

Reverse

Save

Ext. trigger ○

Timebase
Scale: 20 us/Div
X pos.(Div): 3.6
Y/T | Add | B/A | A/B

Channel A
Scale: 5 V/Div
Y pos.(Div): -0.4
AC | 0 | DC

Channel B
Scale: 50 V/Div
Y pos.(Div): 1.8
AC | 0 | DC | -

Trigger
Edge: | A | B | Ext
Level: 0 V
Single | Normal | Auto | None

Multisim - 30 August 2021, 22:10:22

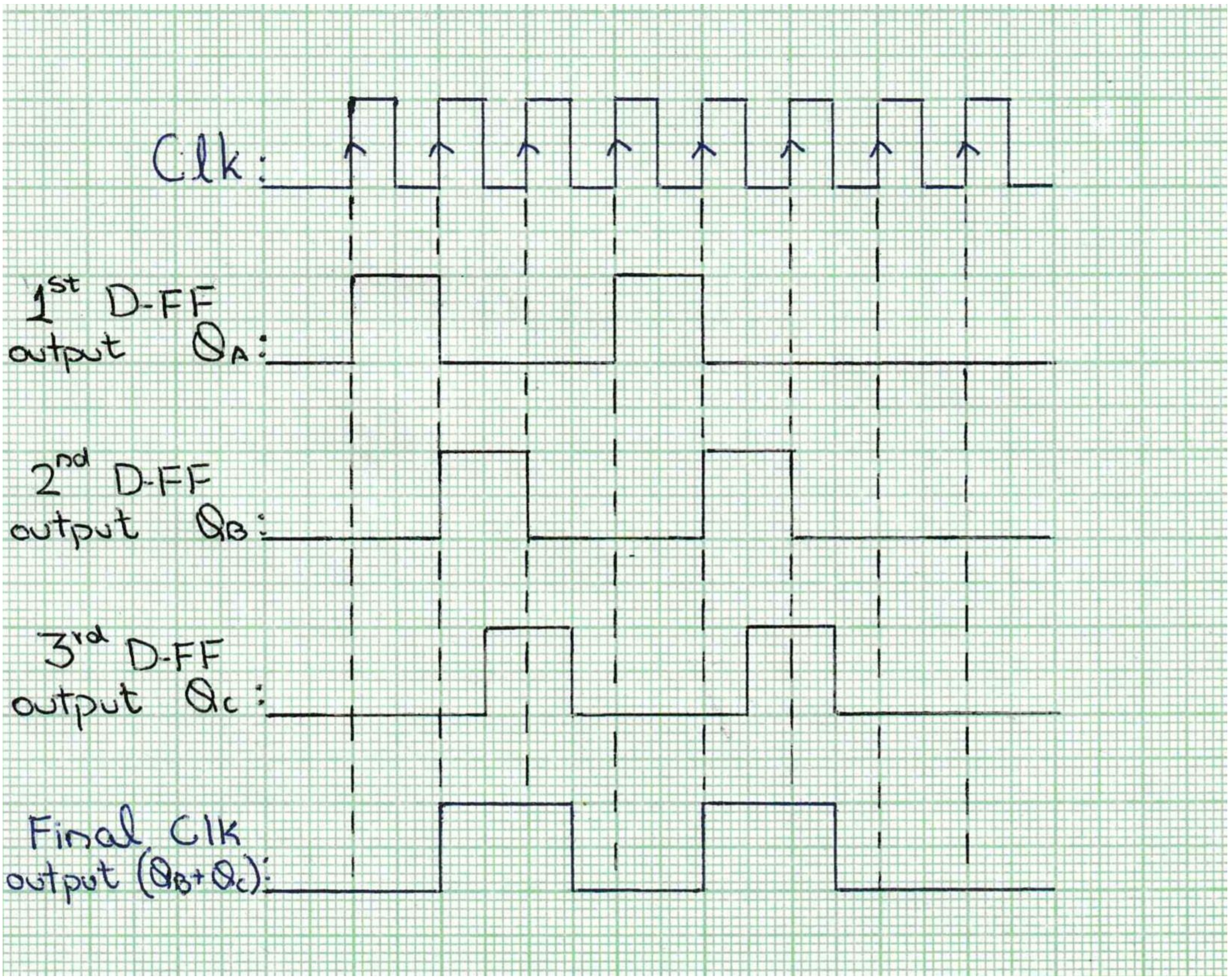
TRUTH TABLE

CLK	1st D-Flipflop O/P Q_A	2nd D-Flipflop O/P Q_B	3rd D-Flipflop O/P Q_C	Final Clock O/P $Q_B + Q_C$
--	0	0	0	0
↑	1	0	0	0
↑	0	1	0	1
↑	0	0	1	1
↑	1	0	0	0
↑	0	1	0	1
↑	0	0	1	1

WORKING PRINCIPLE

The three D-Flipflops work similar to a sequence generator for the sequence 001, the first two flipflops are triggered for the positive edge of the input clock pulse and the third flipflop is triggered to the negative edge of the clock. The output of the second flipflop is frequency divided by three but with 33% duty-cycle. Since the application it to implement a 50% duty-cycle, a third flipflop is used to shift the signal by half the clock cycle which is done by triggering this flipflop by negative edge of the clock. Now, the outputs of the second and third flipflop is applied to an OR gate in order to get a divide by three frequency with the required 50% duty-cycle.

WAVEFORM



VERILOG CODE IMPLEMENTATION IN BEHAVIORAL CODING STYLE

```
module clk_div3(clk,reset, clk_out);
input clk;
input reset;
output clk_out;
reg [1:0] pos_count, neg_count;

always @(posedge clk)
          if (reset) pos_count <=0;
         else if (pos_count ==2) pos_count <= 0;
         else pos_count<= pos_count +1;

always @(negedge clk)
          if (reset) neg_count <=0;
         else  if (neg_count ==2) neg_count <= 0;
         else neg_count<= neg_count +1;

assign clk_out = ((pos_count == 2) | (neg_count == 2));

endmodule
```

From the simulation of waveform from Multisim, it can be observed that the output remains high for one and a half clock cycles of the input and the next one and a half clock cycle will be low thereby constituting to a 50% duty-cycle clock signal as the output. In this Verilog code we have used two counters to keep track of the number of positive and negative edges of the input clock signal, the positive edge counter value is a divide by two frequency output and the negative edge counter value is same as positive edge counter value but is shifted by half clock signal with respect to the input and positive edge counter value as the always block for the same is negative edge triggered. The last code snippet which assigns the value to the clock out is the logic ‘OR’ operation done on the positive counter and negative counter value to obtain a divide by three frequency output with the required 50% duty-cycle.

VERILOG CODE FOR TESTBENCH

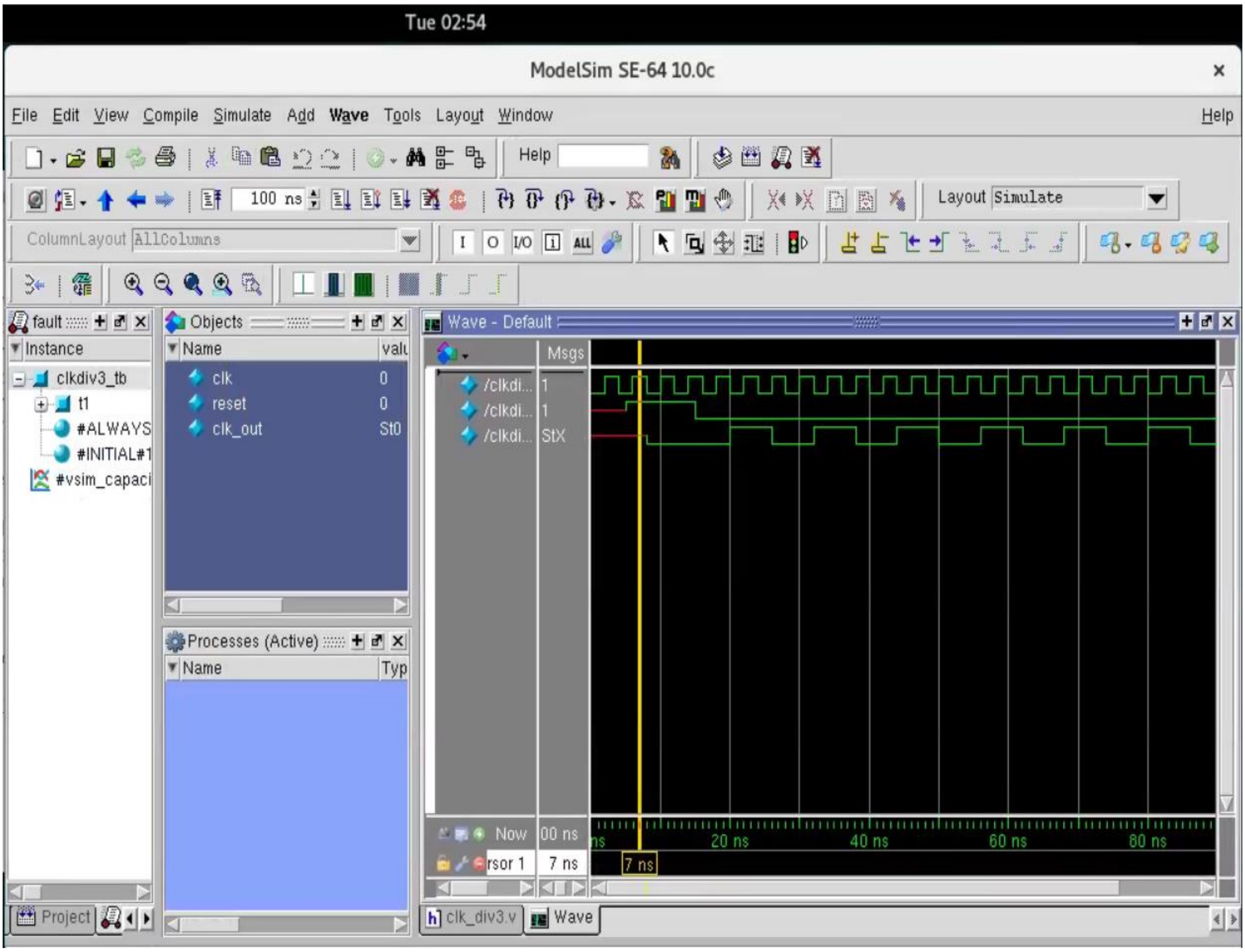
```
module clkdiv3_tb;
reg clk,reset;
wire clk_out;

clk_div3 t1(clk,reset,clk_out);
         initial
                clk= 1'b0;
        always
                #2  clk=~clk;
        initial
        begin
                #5 reset=1'b1;
                #10 reset=1'b0;
                #500 $finish;
        end
endmodule
```

The Verilog code is first simulated on ModalSim and then the result from the Verilog code simulation and MultiSim is realised and compared.

The clock pulse generated at the always block has a time period of 2 nanoseconds and a reset is applied with ON time of 5 nanoseconds. From 5 nanoseconds to 10 nanoseconds from start time, the reset is high during which the clock output remains low.

WAVEFORM RESULT FROM TESTBENCH



The input clock signal has a time period of 4 nanoseconds,

$F_{in} = 1 / (4*10^{-9}) = 250$ MHz

The output clock signal obtained has a time period of 12 nanoseconds,

$F_{out} = 1 / (12 * 10^{-9}) = 83.33$ MHz

Hence from the results of the input and output clocks, it can be determined that,

$F_{out} = F_{in} / 3$

CONCLUSION

The waveform output from the simulation through MultiSim and the simulation from the Verilog code through ModalSim is verified and realized. Hence, giving the desired output.