PES UNIVERSITY

Digital Design & Computer Organization Laboratory

UE20CS206

End Semester Assessment (B.Tech CSE)

August-December 2021

Project Report on

2-by-3-frequency-divider

Submitted by:

Name	SRN
Arunvenkat	PES2UG20CS068
Ashutosh	PES2UG20CS071
Ashutosh Routray	PES2UG20CS072
Claudius D'Souza	PES2UG20CS097

Date: 21st Nov 2021

Table of contents

Abstract of the project

Aim: Design an algorithm to simulate a 2-by-3-frequency divider in `iverilog`.

Working

- Has 2 inputs, the clock signal `clk` and the modulus control signal `mc`.
- The program divides the input frequency signal by reading the modulus control or `mc` value. (This is specified in the testbench)

Value of `mc`	Operation
`0010`	Divide by 2
`0011`	Divide by 3

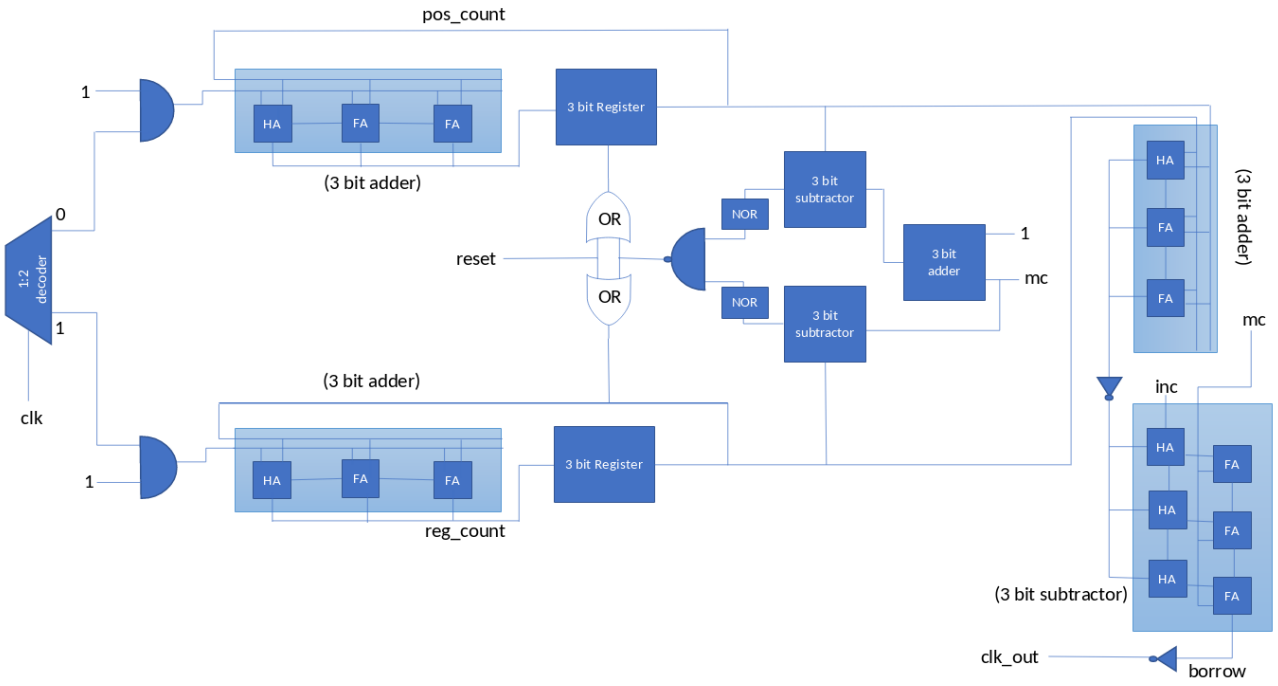
- The program runs of trackers for the main clock line `clk` based on positive and negative edges which triggers the output clock lines `clk_out` accordingly.
- The above is simulated using flip-flops, multiplexers, subtractors, decoders, registers and ripple carry adders.

Components:

Few major components used are Ripple carry adders, subtractors, decoders and registers.

1. **Decoders**: A combinational logic circuit that takes an n coded input and generates a maximum of 2^n unique outputs. In this project we make use of 1:2 decoder to divide the clock signal.
2. **Ripple carry adder**: A combinational logic circuit used to add 2 n-bit binary numbers (in our case 2 3-bit binary numbers). It consists of 1 half adder and n-1 full adders. Each adder generates a single bit output which is the sum of two corresponding single bit inputs. The carry generated from each adder is passed on to the next adder as input.
3. **Registers**: These are collection of flipflops, wherein each flipflop is used to store single bit data.Thus for an n bit input data to be stored, a n bit register is used which consists of n flipflops
4. **Subtractors**: It is a combinational logic circuit used to perform subtraction of 3 input bits, minuend, subtrahend and borrow in. It generates two outputs, difference and borrow out. Subtractors consist of n full adders and n half adders.

Circuit Diagram



The $\div 2/3$ circuit employs an OR gate to permit $\div 3$ operation if the modulus control (MC) is low or $\div 2$ operation if it is high.

Program Code

Verilog Code:

```
module frequency_divider(clk,reset,mc,pos_count,neg_count,clk_out);
    input [3:0] mc;
    input clk;
    input reset;
    output clk_out;

    output reg [3:0] pos_count, neg_count;
    reg c;
    wire [1:0] r_nxt;

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

    always @(negedge clk)
    if (reset)
    begin
        neg_count =0; pos_count=0;
    end
    else  if (neg_count ==(mc) && pos_count ==(mc+1))
    begin
        neg_count = 0;pos_count=0;
    end
    else
        neg_count= neg_count +1;

    assign clk_out = (pos_count +neg_count> (mc));
endmodule
```

Testbench Code for testing MC=2:

```
`timescale 1ns/100ps
module frequencydivider_tb;
reg clk,reset;
reg [3:0] mc;
wire [3:0] pos_count, neg_count;
```

```
wire clk_out;

frequency_divider t1(clk,reset,(mc-4'b0001),pos_count,neg_count,clk_out);
    initial
        begin
            clk= 1'b0;
            mc=4'b0010;
   end
   always
    #5  clk=~clk;
    initial
    begin
         reset=1'b1;
         #5 reset=1'b0;
         #200 $finish;
   end
   initial
$monitor("clk=%b,reset=%b,pos_count=%b,neg_count=%b,clk_out=%b",clk,reset,
         pos_count,neg_count,clk_out);
    initial
        begin
            $dumpfile("freqdiv.vcd");
            $dumpvars(0,frequencydivider_tb);
        end
endmodule
```

Testbench Code for testing MC=3:

```
`timescale 1ns/100ps
module frequencydivider_tb;
reg clk,reset;
reg [3:0] mc;
wire [3:0] pos_count, neg_count;
wire clk_out;

frequency_divider t1(clk,reset,(mc-4'b0001),pos_count,neg_count,clk_out);
    initial
        begin
            clk= 1'b0;
            mc=4'b0011;
        end
    always
        #5  clk=~clk;
    initial
```

```
        begin
            reset=1'b1;
            #5 reset=1'b0;
            #200 $finish;
        end
    initial
$monitor("clk=%b,reset=%b,pos_count=%b,neg_count=%b,clk_out=%b",clk,reset,
                    pos_count,neg_count,clk_out);
   initial
   begin
      $dumpfile("freqdiv.vcd");
      $dumpvars(0,frequencydivider_tb);
   end
endmodule
```

Output Screenshots

Execution command:

```
# for MC = 2
iverilog -o frequency_divider frequency_divider.v frequency_divider_tb_2.v

# for MC = 3
iverilog -o frequency_divider frequency_divider.v frequency_divider_tb_3.v
```

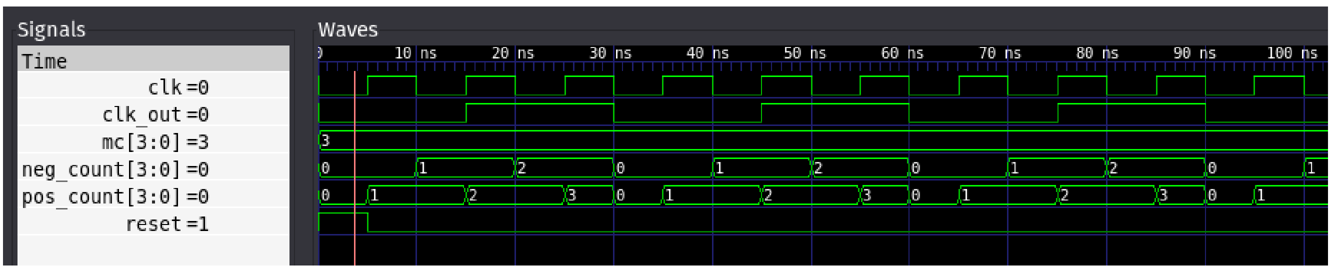
VVP Screenshot for MC=2:

```
VCD info: dumpfile freqdiv.vcd opened for output.
clk=0,reset=1,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=1
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
```

VVP Screenshot for MC=3:

```
VCD info: dumpfile freqdiv.vcd opened for output.
clk=0,reset=1,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
clk=0,reset=0,pos_count=0000,neg_count=0000,clk_out=0
clk=1,reset=0,pos_count=0001,neg_count=0000,clk_out=0
clk=0,reset=0,pos_count=0001,neg_count=0001,clk_out=0
clk=1,reset=0,pos_count=0010,neg_count=0001,clk_out=1
clk=0,reset=0,pos_count=0010,neg_count=0010,clk_out=1
clk=1,reset=0,pos_count=0011,neg_count=0010,clk_out=1
```

GTKWave Screenshot for MC=2:



GTKWave Screenshot for MC=3:

Applications

Frequency Divider Circuits or Frequency Dividers are an integral part of many communication and audio based systems like:

1. Frequency Synthesizers
2. Audio Equipment
3. Radar and Satellite Communication
4. Military Equipment
5. RF Devices