DIGITAL SYSTEM DESIGN LAB CSE-308L

SEMESTER:6TH

LAB REPORT # 5

RING COUNTERS

Submitted By: *Zainab Khalid*
Registration No:*19PWCSE1743*
Section: A
Submitted to: *Mam Madeeha Sher*

DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING
UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR

Lab 5

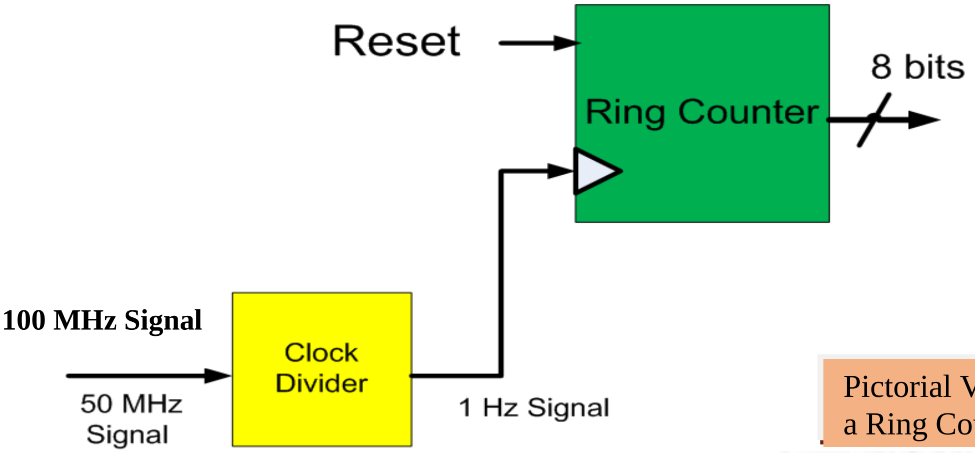
Implementation of a 8 bit Ring Counter

Objective:

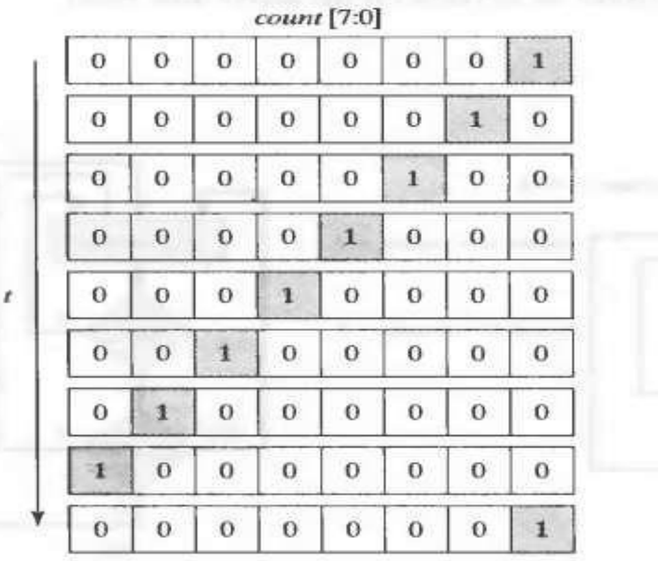
- To become familiarized with behavior level modeling
- To be able to implement sequential circuits using Verilog
- To Implement an 8 Bit Ring Counter on Spartan 6 FPGA starter kit.

Block Diagram:

The Spartan 6 kit has a clock source of 100 MHz. If we use it in applications like counters, the counter will count at an incredibly fast speed and we will not be able to see the output. Your task is to divide the 100 MHz frequency into a 1 Hz frequency. The Module Clock divider is responsible will be responsible for it.
The functional detail of the 8-bit Ring Counter is shown in the following figures.



Pictorial View of the Operation of a Ring Counter

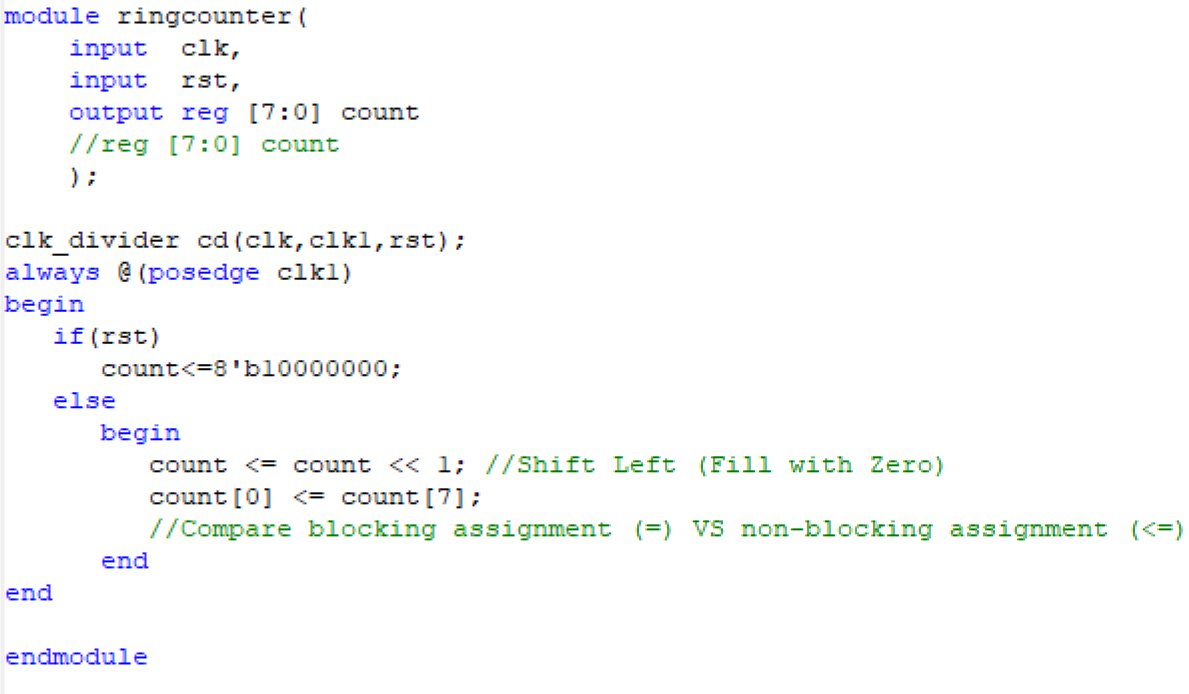


0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1

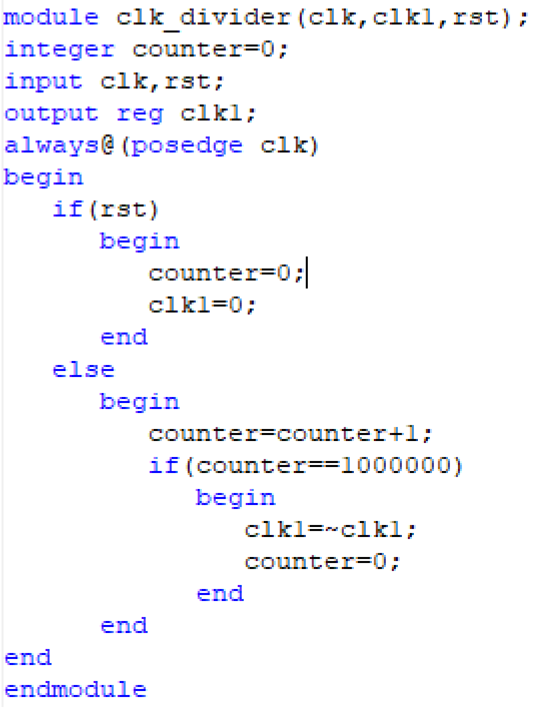
I/O Connection:

The output of the Ring Counter should be connected to 8 LEDs on Spartan 3 Starter Kit. The clock input is connected to pin “T9” on the board. The LEDs should turn on and off one by one from left to Right with an interval of 1 second.

CODE:

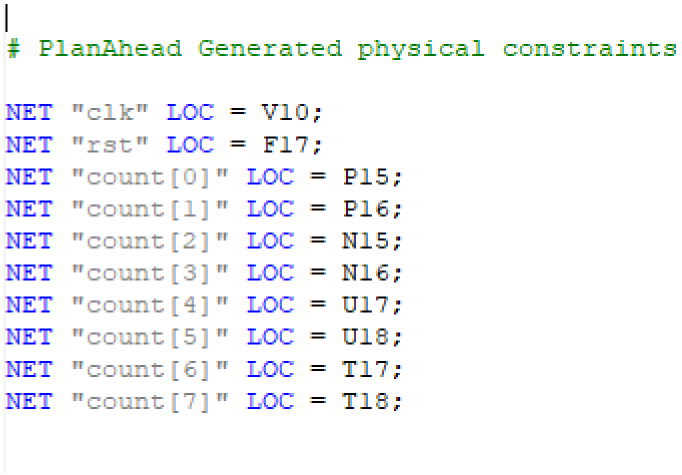


```
module ringcounter(
    input  clk,
    input  rst,
    output reg [7:0] count
    //reg [7:0] count
    );

clk_divider cd(clk,clk1,rst);
always @(posedge clk1)
begin
   if(rst)
      count<=8'b10000000;
   else
      begin
         count <= count << 1; //Shift Left (Fill with Zero)
         count[0] <= count[7];
         //Compare blocking assignment (=) VS non-blocking assignment (<=)
      end
end

endmodule
```



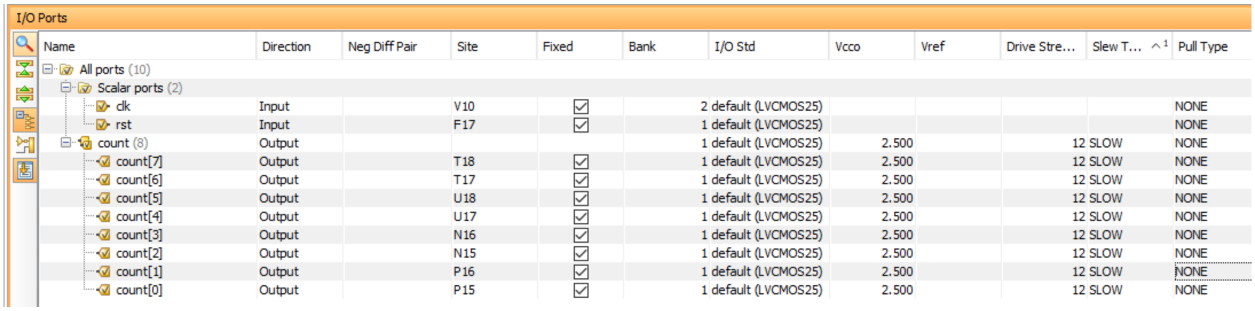
```
module clk_divider(clk,clk1,rst);
integer counter=0;
input clk,rst;
output reg clk1;
always@(posedge clk)
begin
   if(rst)
      begin
         counter=0;
         clk1=0;
      end
   else
      begin
         counter=counter+1;
         if(counter==1000000)
            begin
               clk1=~clk1;
               counter=0;
            end
      end
end
endmodule
```

UCF FILE:



```
# PlanAhead Generated physical constraints

NET "clk" LOC = V10;
NET "rst" LOC = F17;
NET "count[0]" LOC = P15;
NET "count[1]" LOC = P16;
NET "count[2]" LOC = N15;
NET "count[3]" LOC = N16;
NET "count[4]" LOC = U17;
NET "count[5]" LOC = U18;
NET "count[6]" LOC = T17;
NET "count[7]" LOC = T18;
```

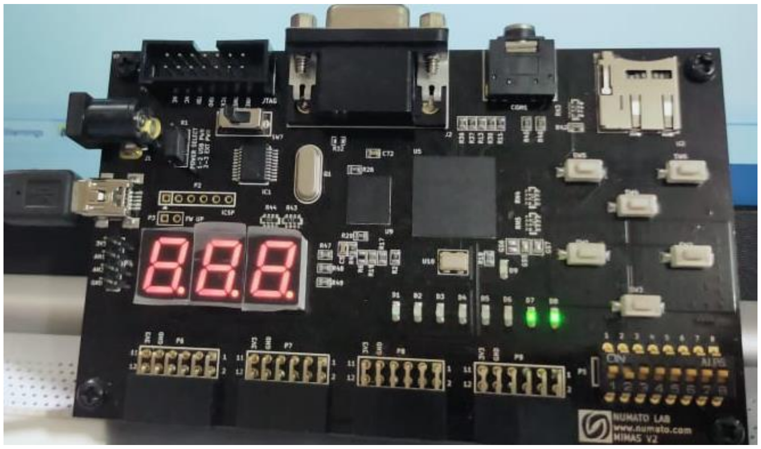
I/O PIN PLANNING:

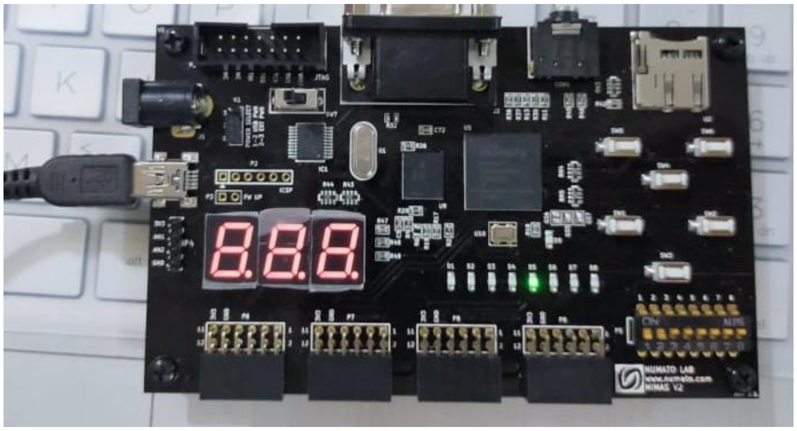


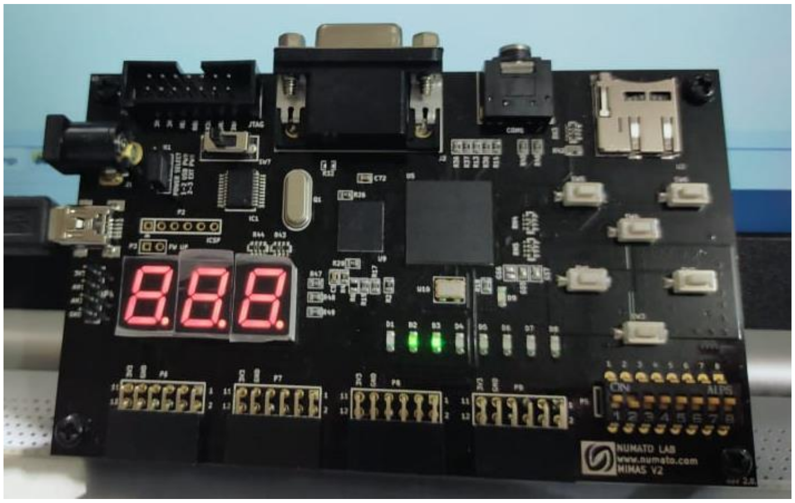
I/O Ports

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stre...	Slew T...	Pull Type
All ports (10)											
Scalar ports (2)											
clk	Input		V10	☑	2	default (LVCMOS25)					NONE
rst	Input		F17	☑	1	default (LVCMOS25)					NONE
count (8)	Output				1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[7]	Output		T18	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[6]	Output		T17	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[5]	Output		U18	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[4]	Output		U17	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[3]	Output		N16	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[2]	Output		N15	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[1]	Output		P16	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE
count[0]	Output		P15	☑	1	default (LVCMOS25)	2.500		12	SLOW	NONE

OUTPUT:




JTAG
NUMATO LAB
www.numato.com
MIMAS V2


JTAG
NUMATO LAB
www.numato.com
MIMAS V2