## EXPERIMENT-8
## REALIZATION OF COUNTERS

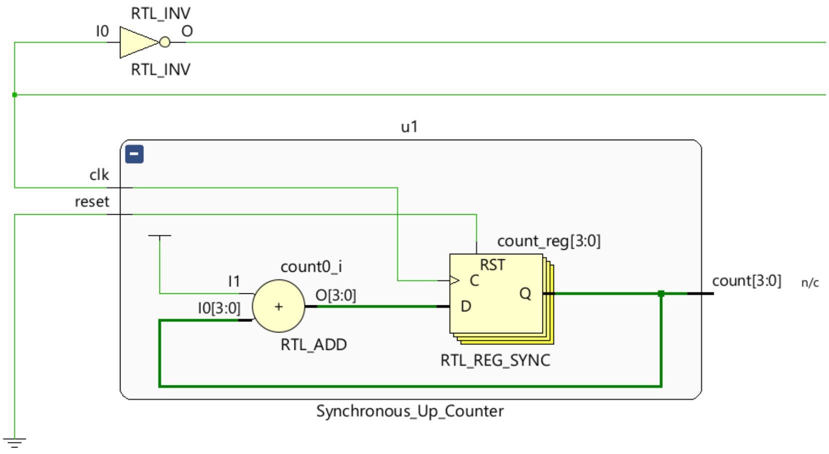
1. AIM**:** To Design, simulate and implement Counters in Behavioral Flow Modeling
2. **SOFTWARE USED:** Xilinx Vivado 2022.2
3. **PROCEDURE:**
    - Open the Xilinx Vivado project navigator.
    - Open the Design source and go to add / create sources
    - Create new file, give appropriate name save it.
    - Open the file in the editor and write the Verilog code.
    - Open the Design source and go to add / create sources to create the test bench
    - Open the editor and write the Verilog code for test bench.
    - After completion of Verilog code set your test bench as top module in simulation settings and Run Simulation.
    - To view RTL schematic, select RTL Analysis in which select open elaborate design, select Schematic.

## Counters

**1.1 AIM:** Synchronous UP Counter

**1.2 SOFTWARE USED:** Xilinx Vivado 2022.2

### 1.3 RTL Schematic:



### 1.4 VERILOG CODE :

```
    module Synchronous_Up_Counter(count,clk,reset);
output reg [3:0]count;
input clk,reset;

initial
begin
count<=0;
end
always@(posedge clk)
begin
if(reset)
count<=0;
else
count<=count+1;
end
endmodule
```

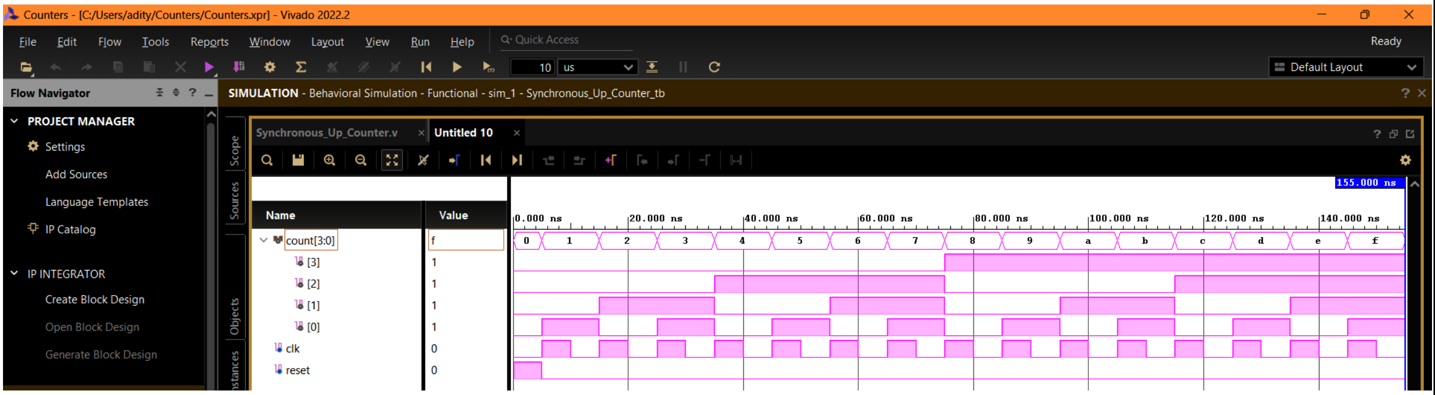
### 1.5 TEST BENCH :

```
module Synchronous_Up_Counter_tb();
wire [3:0]count;
reg clk,reset;

Synchronous_Up_Counter u1(count,clk,reset);

always #5 clk=~clk;
initial
begin
reset=1;
clk=0;
#5 reset=0;
#150 $finish;
end
endmodule
```

## 1.6 WAVEFORM:



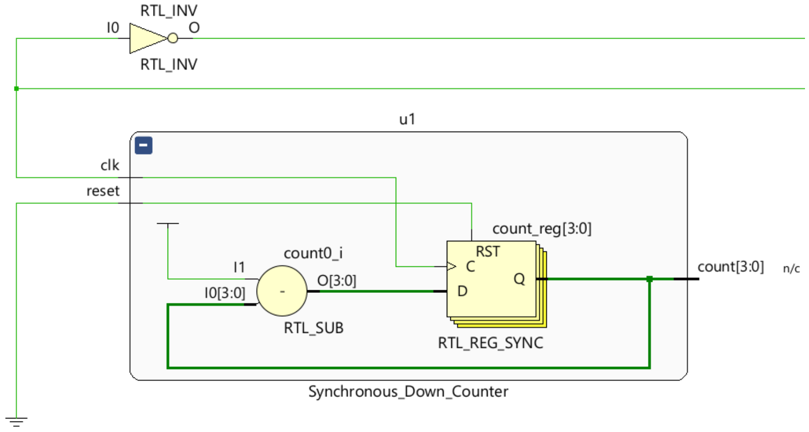
## 1.7 RESULT:

Synchronous UP Counter is implemented in Behavioral Flow Modeling.

**1.1 AIM:** Synchronous Down Counter

**1.2 SOFTWARE USED:** Xilinx Vivado 2022.2

### 1.3 RTL Schematic:



### 1.4 VERILOG CODE :

```
    module Synchronous_Down_Counter(count,clk,reset);
output reg [3:0]count;
input clk,reset;

initial
begin
count<=15;
end
always@(posedge clk)
begin
if(reset)
count<=0;
else
count<=count-1;
end
endmodule
```

### 1.5 TEST BENCH :

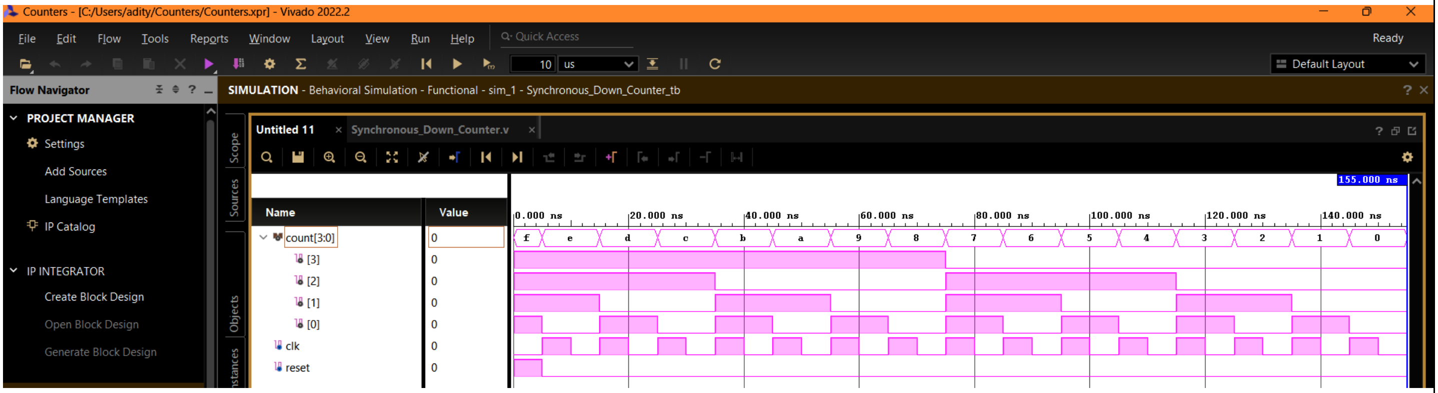
```
module Synchronous_Down_Counter_tb();
wire [3:0]count;
reg clk,reset;

Synchronous_Down_Counter u1(count,clk,reset);

always #5 clk=~clk;
initial
begin
reset=1;
clk=0;
#5 reset=0;

#150 $finish;
end
endmodule
```

## 1.6 WAVEFORM:



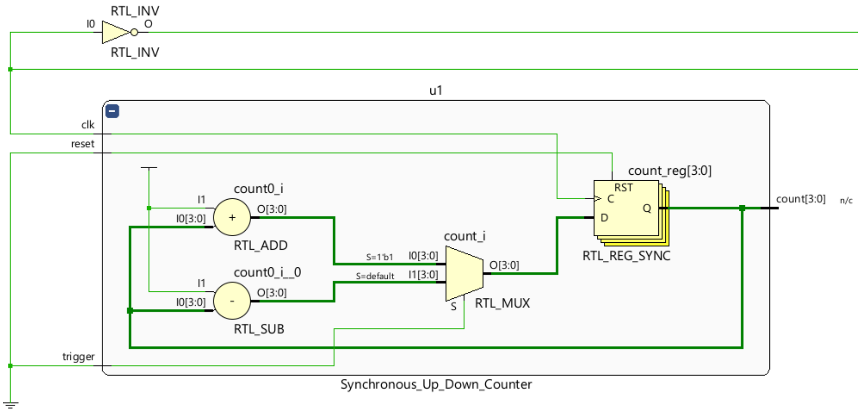
## 1.7 RESULT:

Synchronous Down Counter is implemented in Behavioral Flow Modeling.

**1.1 AIM:** Synchronous UP Down Counter

**1.2 SOFTWARE USED:** Xilinx Vivado 2022.2

## 1.3 RTL Schematic:



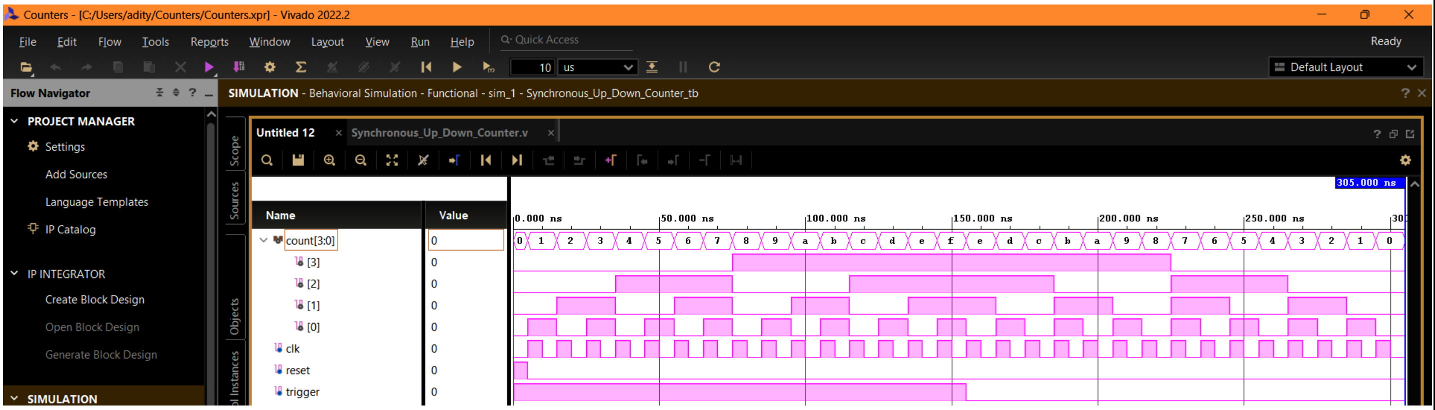
## 1.4 VERILOG CODE :

```
    module Synchronous_Up_Down_Counter(count,clk,reset,trigger);
output reg [3:0]count;
input clk,reset,trigger;
initial
begin
count<=0;
end
always@(posedge clk)
begin
if(reset)
count<=0;
else if(trigger)
count<=count+1;
else
count<=count-1;
end
endmodule
```

## 1.5 TEST BENCH :

```
module Synchronous_Up_Down_Counter_tb();
wire [3:0]count;
reg clk,reset,trigger;
Synchronous_Up_Down_Counter u1(count,clk,reset,trigger);
always #5 clk=~clk;
initial
begin
reset=1;
clk=0;
trigger=1;
#5 reset=0;
#150 trigger=0;
#150 $finish;
end
endmodule
```

## 1.6 WAVEFORM:



| Name | Value |
|---|---|
| count[3:0] | 0 |
| [3] | 0 |
| [2] | 0 |
| [1] | 0 |
| [0] | 0 |
| clk | 0 |
| reset | 0 |
| trigger | 0 |

## 1.7 RESULT:

Synchronous UP Down Counter is implemented in Behavioral Flow Modeling.