# PSG COLLEGE OF TECNOLOGY

(AUTONOMOUS INSTITUTION)

COIMBATORE-641 004


TOPIC: TUT-2 IMPLEMENTATION OF 4-bit COUNTER USING VIVDO 2018.2 AND ZYBO z7-10

BRANCH: ELECTRONICS AND COMMUNICATION ENGINEERING.

SUBJECT CODE: 19L019-FPGA

SUBMITTED BY: SHEENA S-20L141

DHEVAA DHARSHINIE -20L108

JEYASHRI-20L116

**AIM:**

To implement a counter on zybo z710 board using vivado 2018.2 by generating bitstream file

**SOFTWARE USED**:

xilinx vivado 2018.2

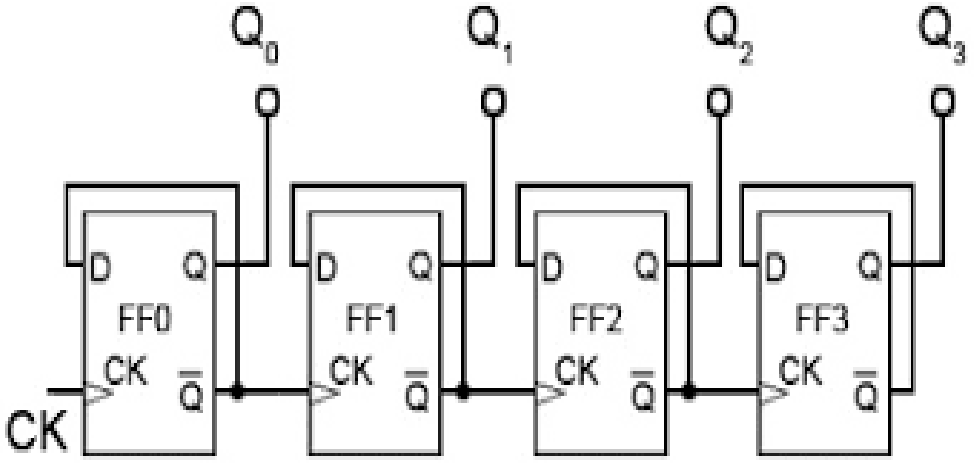
**THEORY:**

## 4-bit binary counter

In **Synchronous Counter**, the external clock signal is connected to the clock input of EVERY individual flip-flop within the counter so that all of the flip-flops are clocked together simultaneously (in parallel) at the same time giving a fixed time relationship. In other words, changes in the output occur in "synchronisation" with the clock signal.

The result of this synchronisation is that all the individual output bits changing state at exactly the same time in response to the common clock signal with no ripple effect and therefore, no propagation delay.

**COUNTER CIRCUIT:**



## SIMULATION CODE:

## DESIGN MODULE:

PROJECT MANAGER - counterzybo2

Project Summary | counterzybo2.v | clockdivider.v

D:/vivado_projects/counterzybo2/counterzybo2.srcs/sources_1/new/counterzybo2.v

```
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module counterzybo2(input clk, reset, output[3:0] counter
24      );
25  reg [3:0] counter_up;
26  wire clkout1; //internal signal for the divided clock output
27  wire clkout2; //internal signal for the divided clock output
28  wire clkOut; //
29  clockdivider UclkDiv (
30      .clk(clk), // master clock
31       .clkout1(clkout1), // 381.4697265625Hz Q17
32       .clkout2(clkout2), //190.73486328125Hz Q18
33      .clkOut(clkOut) //clock out in the BUFG CLOCK WHICH IS INPUT TO THE COUNTER
34  );
35  // up counter
36  always @(posedge clkOut or posedge reset)
37  begin
38  if(reset)
39   counter_up <= 4'd0;
40  else
41   counter_up <= counter_up + 4'd1;
42  end
43  assign counter = counter_up;
44  endmodule
45
```

## TIME DIVISION MODULE:

Project Summary | counterzybo2.v | clockdivider.v

D:/vivado_projects/counterzybo2/counterzybo2.srcs/sources_1/new/clockdivider.v

```
22
23  module clockdivider(
24  input clk,
25  output clkout1,
26  output clkout2,
27  output clkOut
28  );
29
30  wire Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20, Q21,Q22,Q23,Q24,Q25;
31  assign clkout1 = Q17;
32  assign clkout2 = Q18;
33
34
35  // input clock is 100Mhz, Q = 50MHz
36
37  FDCE #(
38  .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
39  ) FDCE_0 (
40        .Q(Q0),      // Data output
41        .C(clk),     // Clock input
42        .CE(1'b1),   // Clock enable input
43        .CLR(1'b0),  // Asynchronous clear input
44        .D(!Q0)      // Data input
45  );
```

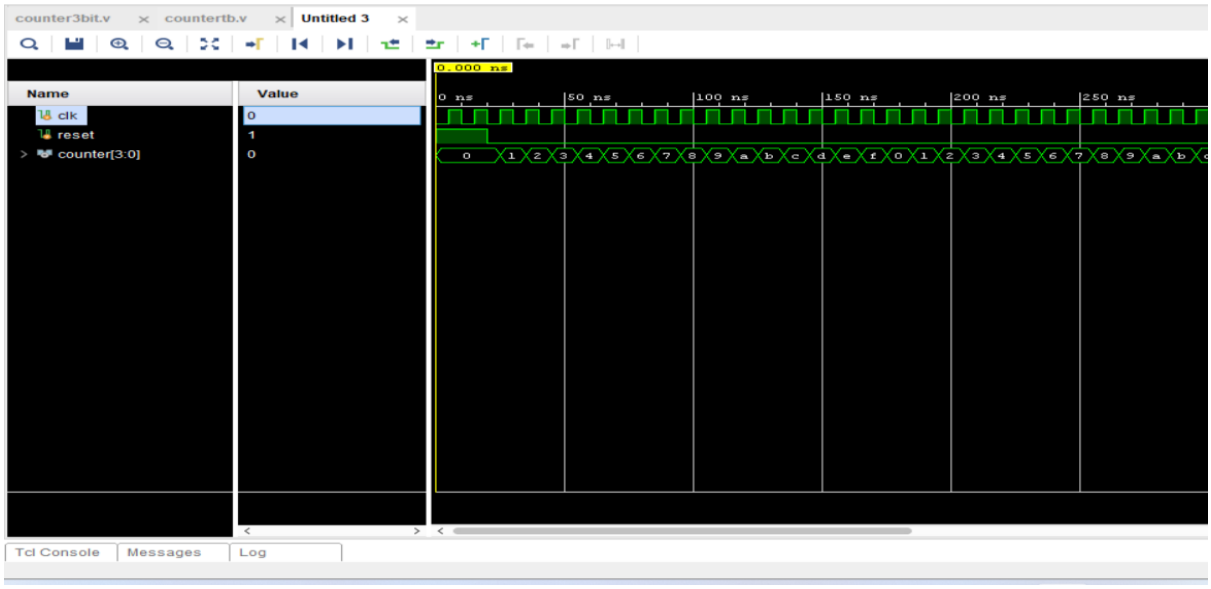
```
// input clock is 2.9802322Hz, Q = 1.49011611Hz
FDCE #(
.INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCE_25 (
      .Q(Q25),       // Data output
      .C(Q24),       // Clock input
      .CE(1'b1),     // Clock enable input
      .CLR(1'b0),  // Asynchronous clear input
      .D(!Q25)       // Data input
);


BUFG U_BUFG (
      .O(clkOut),      // Clock buffer output
      .I(Q25)       // Clock buffer input
);
```

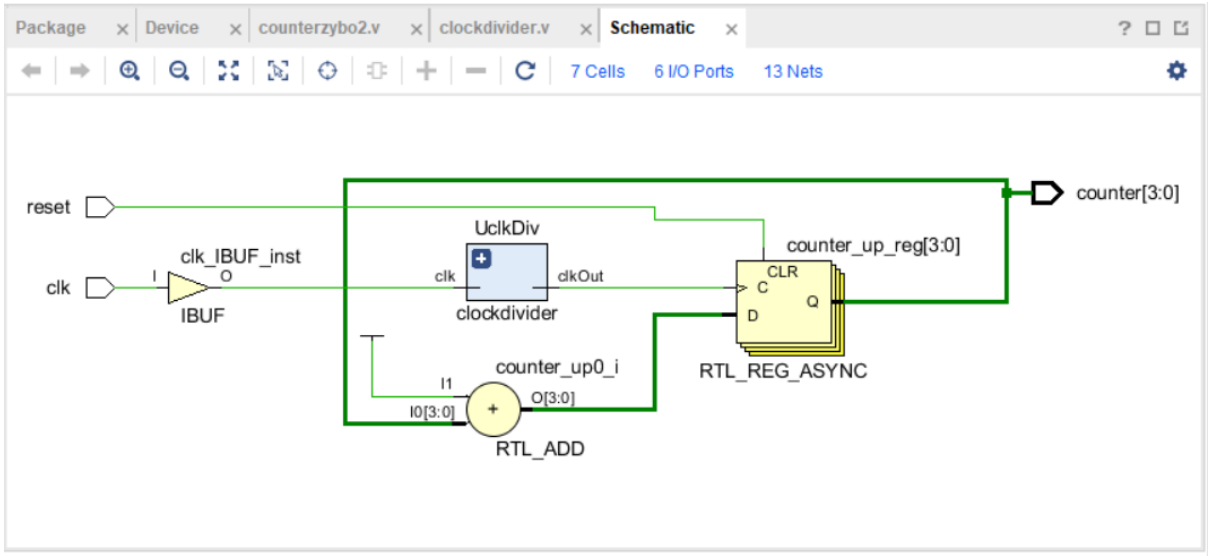
```
endmodule
```

# Simulation outputs:

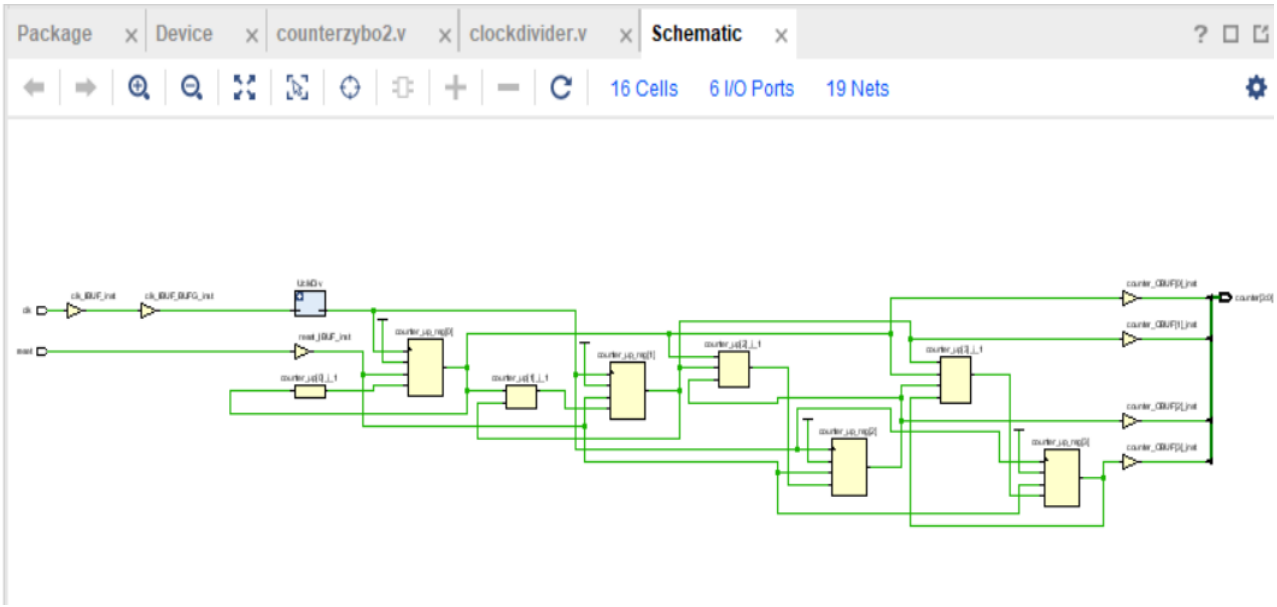
# Synthesis results:



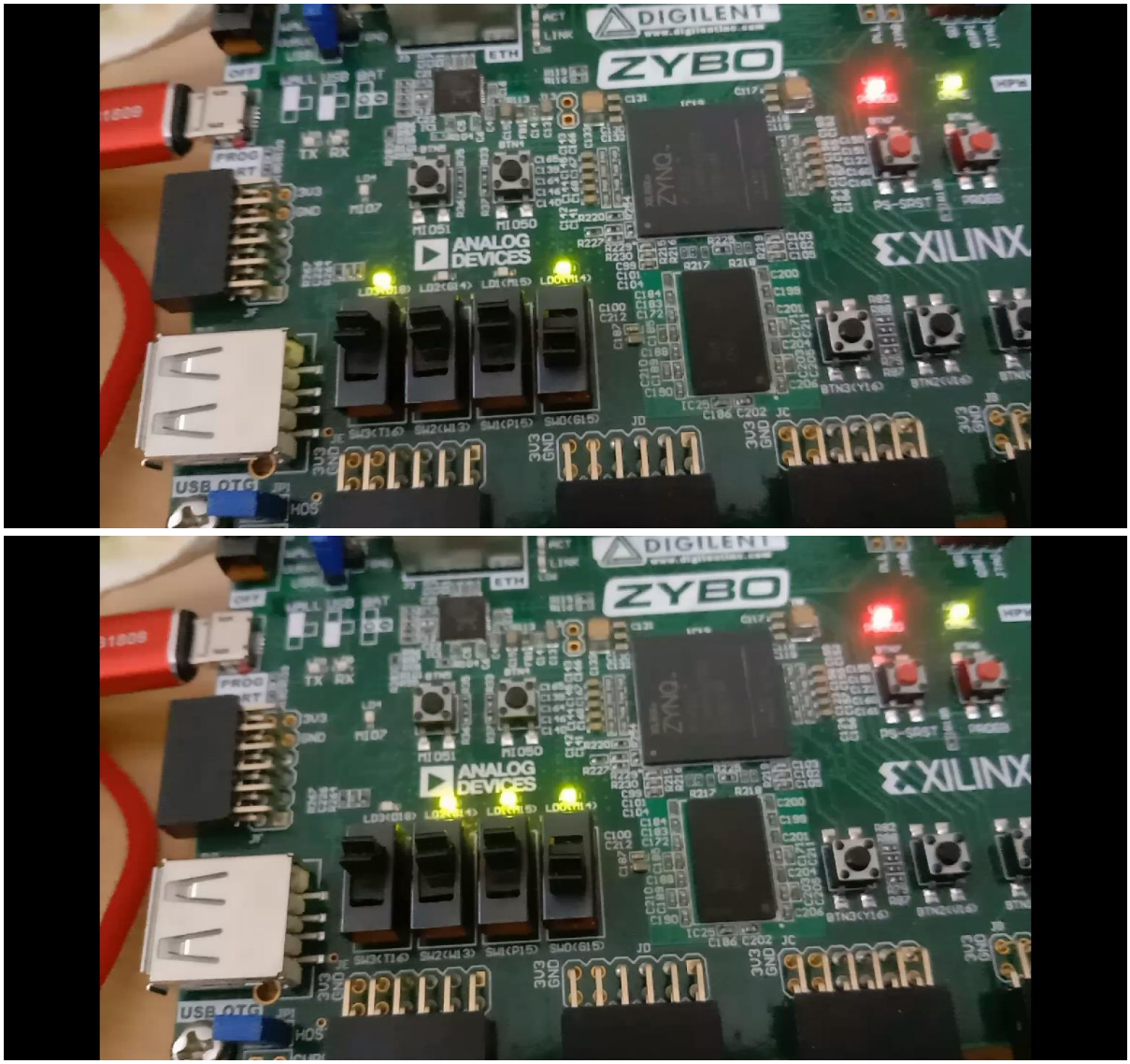
# RTL Schematic:




# SYNTHESIS SCHEMATIC:




# BOARD IMPLEMENTATION:




**RESULT:**

Thus 4-bit counter is been implemented using vivado by generating bitfile and downloading the same in zybo board.