EX. NO: 08	COUNTERS
DATE :	

AIM: To verify the functionalities of Counters using Verilog HDL program.

SOFTWARE USED: Xilinx Vivado

HARDWARE USED: Basys3 FPGA Board

COUNTERS:

Verilog HDL Code in Structural:

```
module updown_counter(up, down, clk, A);
   input up, down, clk;
   output [3:0] A;
   wire [11:0] X;
   wire [3:0] B;
   not(X[0], up);
   and(X[1], X[0], down);
   or(X[2], up, X[1]);
   t_flipflop T1(X[2], clk, A[0], B[0]);
   int_con I1(A[0], up, B[0], X[1], X[3], X[4], X[5]);
   t_flipflop T2(X[5], clk, A[1], B[1]);
   int_con I2(A[1], X[3], B[1], X[4], X[6], X[7], X[8]);
   t_flipflop T3(X[8], clk, A[2], B[2]);
   int_con I3(A[2], X[6], B[2], X[7], X[9], X[10], X[11]);
   t_flipflop T4(X[11], clk, A[3], B[3]);
endmodule
module int_con(a, b, c, d, h, i, j);
   input a, b, c, d;
   output wire h, i, j;
   and(h, a, b);
   and(i, c, d);
   or(j, i, h);
endmodule
module t_flipflop(T, clk, Q, Qbar);
   input T, clk;
   output reg Q, Qbar;
   initial begin
      Q = 0;
      Qbar = 1;
   end
   always @(posedge clk) begin
      if (T) begin
         Q = ~Q;
         Qbar = ~Q;
```

```
        end
        else begin
            Q = Q;
            Qbar = ~Q;
        end
    end
endmodule
```

Test bench for Counter Structural:

```
module tb_updown_counter;
    reg up, down, clk;
    wire [3:0] A;
    updown_counter C(
        .up(up),
        .down(down),
        .clk(clk),
        .A(A) );
    always begin
    #5 clk = ~clk;
    end
 initial begin
        clk = 0;
        up = 0;
        down = 0;
        #10 up = 1; down = 0;
        #10 up = 1; down = 0;
        #10 up = 0; down = 1;
        #10 up = 0; down = 0;
        #10 up = 0; down = 0;
        #10 up = 1; down = 1;
        #10; $finish;
    end
initial begin
$monitor("Time: %0t | up: %b | down: %b | A: %b", $time, up, down, A);
end
endmodule
```

Verilog HDL Code in Behavioral:

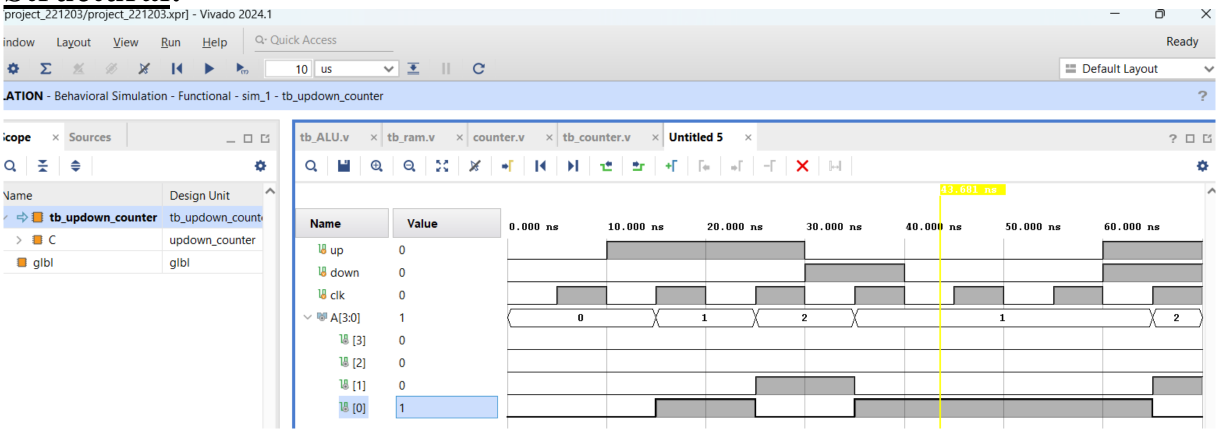
```
module up_down_counter (clk, rst, up, down, count);
    input wire clk, rst, up, down;
    output reg [3:0] count;
    always @(posedge clk or posedge rst) begin
        if (rst) begin
            count <= 4'b0000;
        end else if (up && !down && count < 4'b1111) begin
            count <= count + 1;
        end else if (down && !up && count > 4'b0000) begin
            count <= count - 1;
        end
    end
endmodule
```

<u>Test bench for Counter Behavioral</u>:

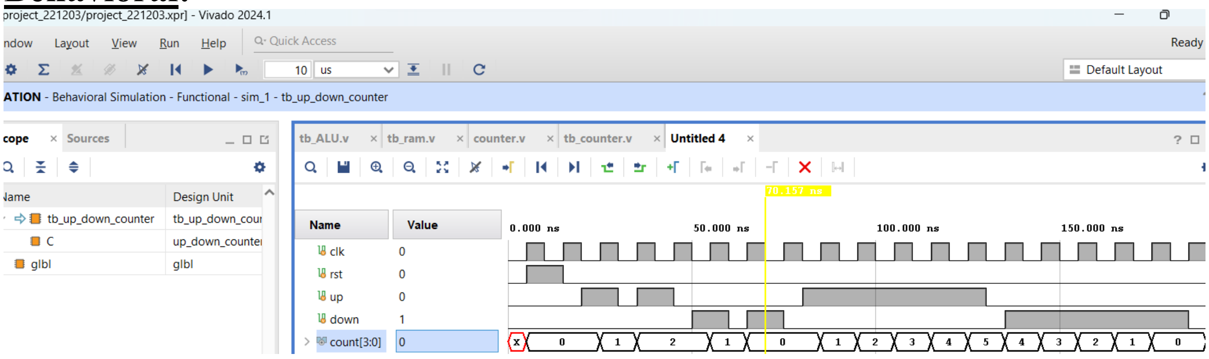
```
module tb_up_down_counter;
    reg clk, rst, up, down;
    wire [3:0] count;
    up_down_counter C(
        .clk(clk),
        .rst(rst),
        .up(up),
        .down(down),
        .count(count)
    );
    initial begin
        clk = 0;
        rst = 0;
        up = 0;
        down = 0;
        #5 rst = 1;
        #10 rst = 0;
        #5 up = 1;
        #10 up = 0;
        #5 up = 1;
        #10 up = 0;
        #5 down = 1;
        #10 down = 0;
        #5 down = 1;
        #10 down = 0;
        #5 up = 1;
        #50 up = 0;
        #5 down = 1;
        #50 down = 0;
        #5 $finish;
    end
    always #5 clk = ~clk;
    initial begin
        $monitor("Time: %0t, Reset: %b, Up: %b, Down: %b, Count: %d",
                $time, rst, up, down, count);
    end
endmodule
```

SIMULATION WAVEFORM:

Structural:



Behavioral:



HARDWARE OUTPUT FOR STRUCTRAL MODEL:

I/O ports:

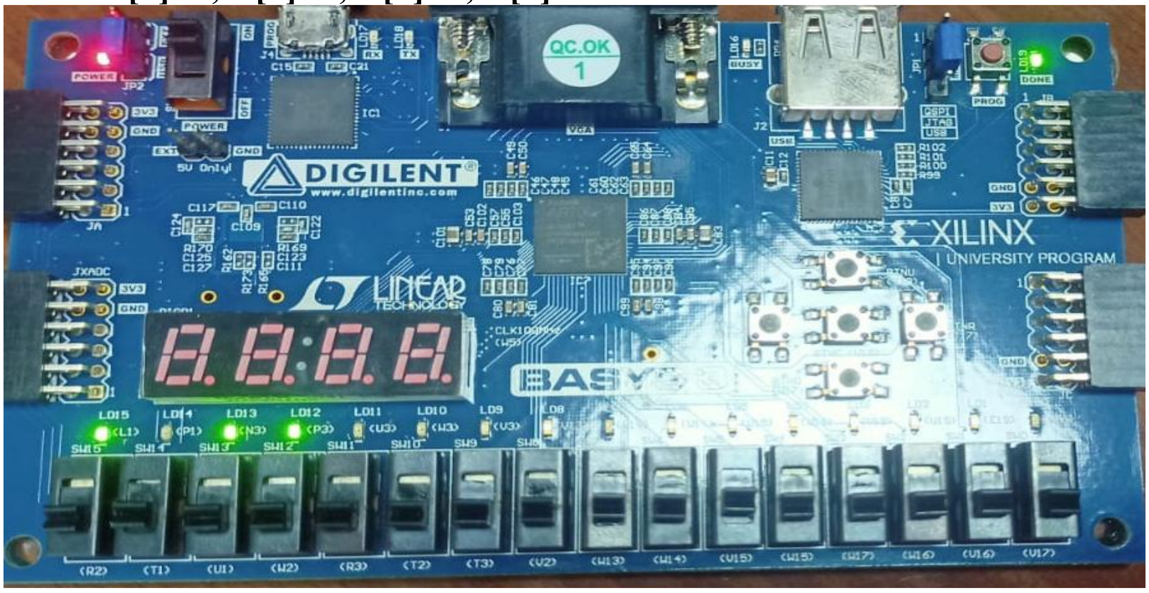
Inputs:	Outputs:
clk: W5	A[3]: L1
down:T1	A[2]: P1
up: R2	A[1]: N3
	A[0]: P3

For

clk=1,up=0,down=0

Output:

A[3]=1, A[2]=0, A[1]=1, A[0]=1



CIRCUIT DIAGRAM:

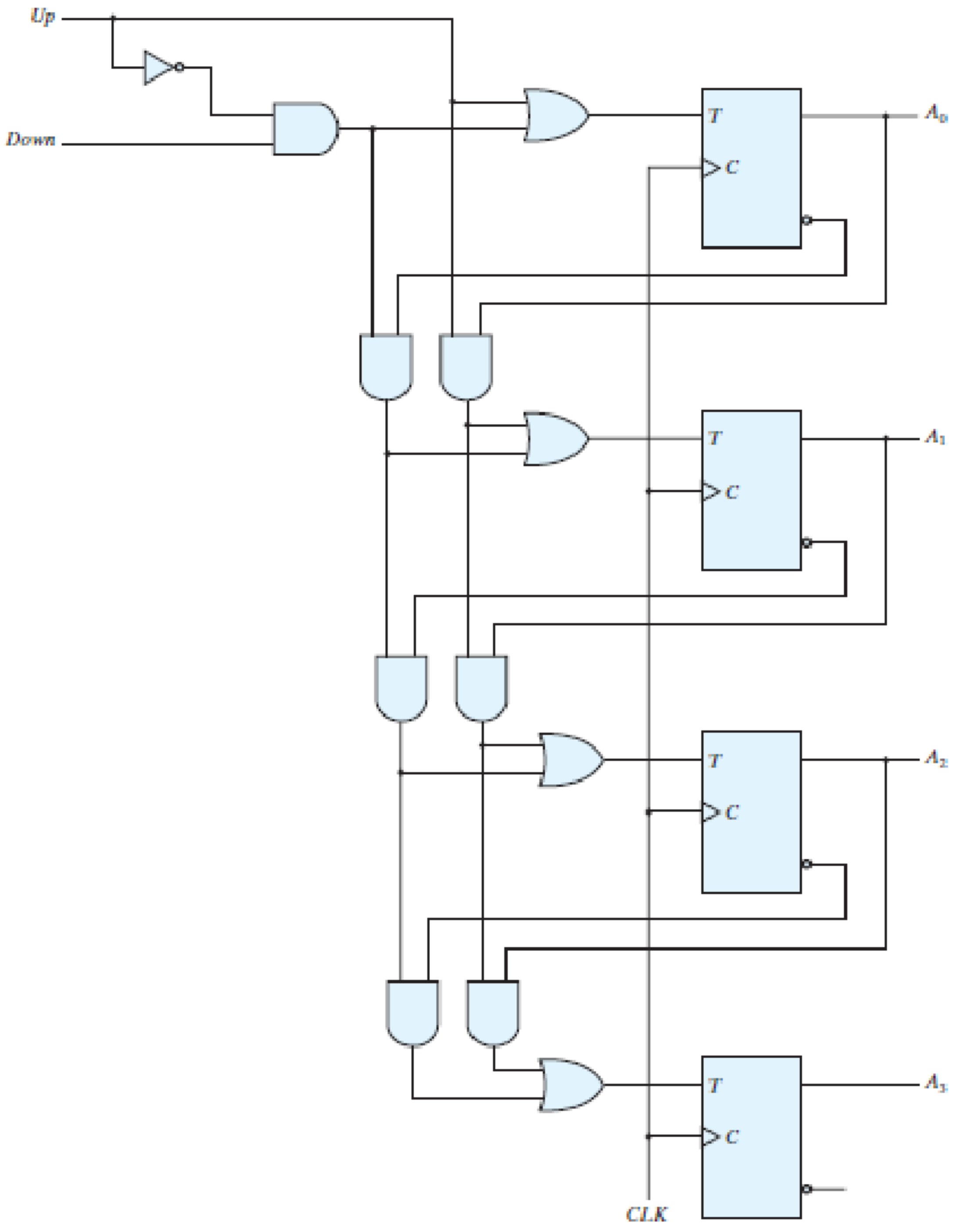


FIGURE 6.13
Four-bit up–down binary counter

RESULT:

The Counters has been successfully designed and implemented using Verilog HDL, and its functionalities have been thoroughly verified through simulation with Xilinx Vivado software on the Basys3 FPGA board.