Dayananda Sagar College of Engineering

Department of Electronics and Communication Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru – 560 078.

(An Autonomous Institute affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment and Accreditation Council (NAAC) with 'A' grade

OPEN ENDED EXPERIMENT

Course: ABNC-IV Laboratory (DIGITAL CIRCUIT DESIGN WITH FPGA- HDL LAB) Semester : 4
Course Code: 22ECL471 Date:
Lab Batch: A3

A Report on

DECADE COUNTERS

Submitted by

1DS22EC052	**BHUVAN KUMAR S**
1DS22EC054	**CHAITRA KRISHNA GOUDA**

Faculty In-charge
Prof. Manasa R. K.
Prof. Ninu Rachel Philip

Signature of Faculty In-charge

DEVELOP A VERILOG CODE FOR THE DECADE COUNTER

INTRODUCTION :

In the field of digital electronics, counters are integral components widely used in various applications such as digital clocks, frequency counters, and event counters. Among these, the decade counter holds a significant place due to its ability to count from 0 to 9 cyclically. This report delves into the design and implementation of a decade counter using Verilog, a hardware description language (HDL) known for its effectiveness in digital system design.

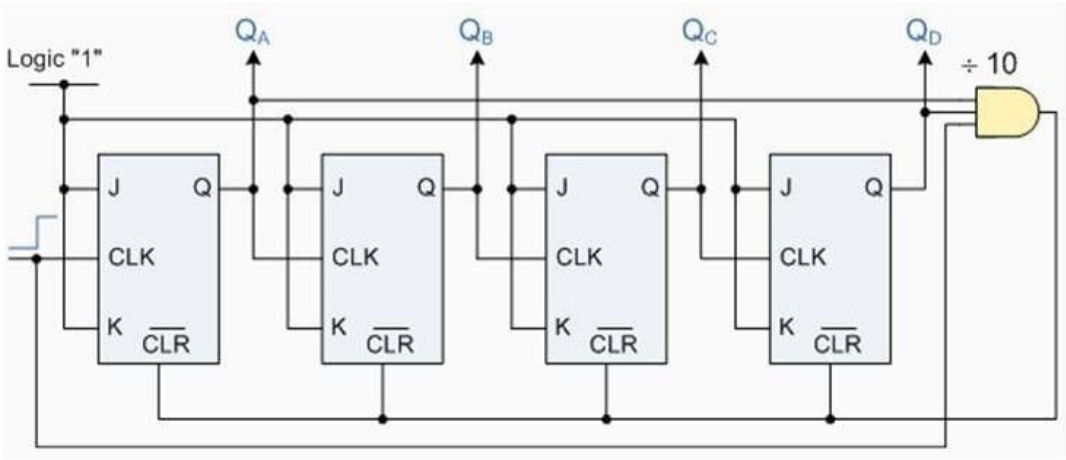
A decade counter, also known as a mod-10 counter, it is a digital circuit that counts the number of pulses or clock signals it receives, incrementing its output by one for every ten input pulses. The design of a decade counter typically involves flip-flops, which are basic memory elements in digital electronics. By connecting flip-flops in a specific configuration, we can create a counter that progresses through a defined sequence of states. It is a type of asynchronous counter, which means that the output of each stage is used as the input for the next stage, allowing the counter to count in a sequential manner. It is a type of counter that counts ten distinct states, typically from 0 to 9. Once it reaches its maximum count, it resets to zero and starts the counting a new cycle. This functionality makes it ideal for applications requiring precise timing and sequence control.

Decade counters are commonly used in digital systems and electronic devices, such as digital clocks, timers, and frequency dividers, to count events or measure time intervals. They are typically designed to count up to a maximum value of nine, after which they reset to zero, ready to start counting again.

Verilog, as an HDL, provides a robust platform for describing digital systems and is widely used for designing and verifying digital circuits. The language allows designers to describe the behavior and structure of electronic systems at multiple levels of abstraction, from the algorithmic level down to the gate level. Verilog's capability to simulate the behavior of a digital circuit before actual hardware implementation ensures that designs are error-free and optimized for performance.

By the end of this report, readers will have a comprehensive understanding of how to design and implement a decade counter using Verilog.

DESIGN :



TRUTH TABLE:

Clock Count	Output bit Pattern				Decimal Value
	Q_D	Q_C	Q_B	Q_A	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Output back to zero				

PROGRAM :

VERILOG CODE	TESTBENCH

module decade_counter(en, clock, count);
input en, clock;
output reg [3:0] count;
always @(posedge clock)
begin
if(en)
begin
if (count>=4'd0 && count<4'd10)
count<=count+4'd1;
else
count<=4'd0;
end
else
count<=4'd0;
end
endmodule
	`timescale 1ns/1ps
	module decadecounter_tb;
	wire [3:0] count;
	reg en,clock;
	decade_counter dut(.en(en), .clock(clock),
	.count(count));
	initial begin
	$display($time, " << Starting the Simulation >>");
	en=0;
	clock=0;
	#20 en=1'd1;
	end
	always
	#5 clock=~clock;
	initial
	$monitor ($time , "clock= %b, count= %d, en=
	%b", clock,count, en);
	endmodule

RTL CODE:

```
module decoder_counter(clk, rst, q);
input clk, rst;
output [3:0]q;
reg [3:0]q;
reg clkdiv;
reg[22:0] div;
initial q = 4'd0;
always @ (posedge clk )
begin
div=div+1'd1;
clkdiv=div[22];
end
always@(posedge clkdiv or posedge rst)
begin
if (rst == 1'd1)
q = 4'd0;
else if (q== 4'd9)
q = 4'd0;
else
q=q+1;
end
endmodule
```

(Advance) Verilog Code of Decade Counter :

```
module decade_counter ( output reg [3:0] q,
input clk );
always @(posedge clk)
q <= q = = 9 ? 0 : q + 1;
endmodule
```

PIN ASSIGNMENT:

clock: set_location_assignment PIN_P11 -to CLOCK_50

rst: set_location_assignment PIN_C10 -to SW[0]

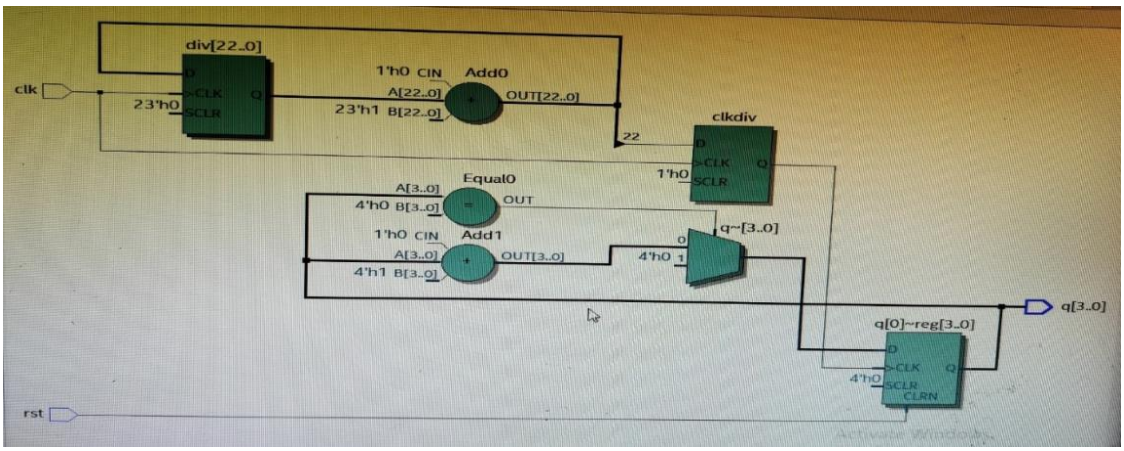
q[3]: set_location_assignment PIN_A8 -to LEDR[0]

q[2]: set_location_assignment PIN_A9 -to LEDR[1]

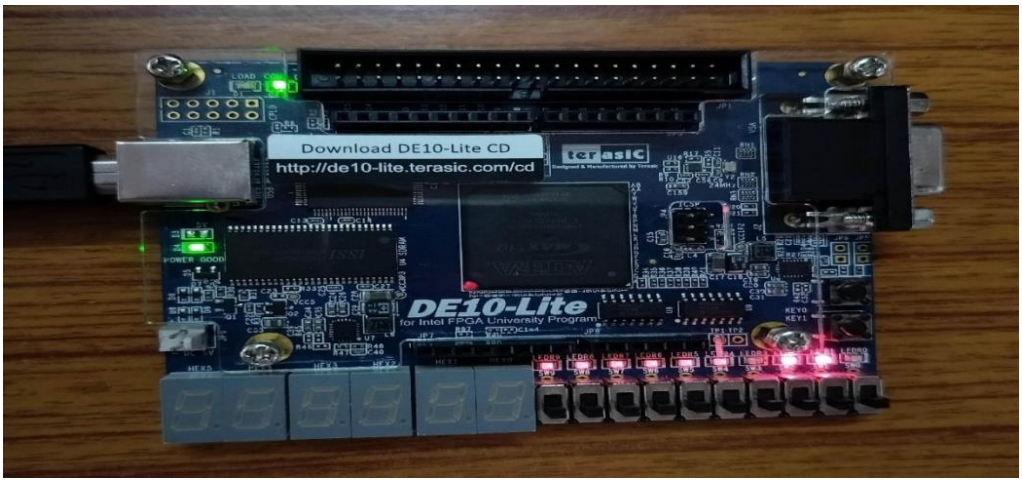
q[1]: set_location_assignment PIN_A10 -to LEDR[2]

q[0]: set_location_assignment PIN_B10 -to LEDR[3]

RTL Viewer:



SIMULATION RESULTS:



APPLICATIONS:

1. Digital Clocks: Decade counters are used to display hours, minutes, and seconds in digital clocks.

2. Timers: Decade counters are used to implement timers that count down from a set value to zero.

3. Frequency Division: Decade counters can be used to divide a high-frequency signal by 10, generating a lower-frequency signal.

4. Medical Devices: Decade counters are used in medical devices, such as heart rate monitors and blood pressure monitors.

5. Aerospace: Decade counters are used in aerospace applications, such as counting navigation data and sensor readings.

RESULT DISCUSSION:

Design and Simulated Decade Counter with Verilog Code using FPGA implementation.

CONCLUSION:

In conclusion, the decade counter project exemplifies the effective use of Verilog and FPGA technology in digital design. The design and implementation of a decade counter using Verilog and FPGA highlight the synergy between hardware description languages and reprogrammable hardware platforms.