DIGITAL DESIGN AND COMPUTER ORGANISATION – DDCO

MINI-PROJECT

THEME: → DESIGNING AND IMPLEMNENTING A CIRCUIT THAT COUNTS NUMBER OF 1s IN A GIVEN BINARY INPUT USING ICARUS VERILOG SOFTWARE

DONE BY:

RAHUL RAMACHANDRA KINI

SYNOPSIS

Icarus Verilog is a hardware description language used to design and document electronic systems. Verilog HDL allows designers to design at various levels of abstraction. It basically including designing of hardware circuits with include involvement of Logic Gates, Combination and Sequential Logic circuits, Finite state machines, counters , Abstract Microprocessor Design etc. It involves involves writing a circuit code and testbench code .

CIRCUIT CODE

Circuit Code involves importing , creation , or the usage of modules. These modules could be either of basic gates or flipflops or ALU design based on the requirements circuit design. Module definition is very important aspect. We can define the kinds and number of inputs and output ports , later assign output port with Logical operators and Operands.

TESTBENCH CODE:

TestBench code involves instantiates and imports the modules created and marks the beginning of simulation using ‘begin’ keyword. It initiates inputs and stimulates the output at different time delay and finishes simulation with keyword ‘end’ keyword. Time point delay can be give using ‘#’ symbol followed by delay. It also include displaying of output in the iverilog environment based on user feed and display statements. $dumpvars is used to specify which variables or signals should be included in VCD file during output simulation. VCD file is standard format used to record waveform simulation in GTKWave window. Using $monitor, we can display print statements our input or outputs in iverilog terminal or environment

OVERVIEW OF PROJECT:

The main theme of our Mini Project is to design and implement circuit for counting number of One's in given binary input. We make of simple for looping construct, wherein we iterate and go through each and every bit of 8 bit binary input given and examine the presence of One's. We also initialise a counter element (initially to Zero), which keep the count of One's in the given when iterated. As on the Control finds the presence of One, counter element is incremented. 8 Sample testcases as been taken and number of One's are counted. Henceforth the binary input and Number of One's are represented in GTKWave window in the form of Waveforms. Upon compiling the circuit logic and testbench logic, output waveform is recorded in a VCD file and Output is displayed in GTKWave window.

Let say binary input is 7

In binary it is → **00000111**

Number of Ones→ 3

CIRCUIT CODE:

```
module count_ones (
   input [7:0] binary_number,   // 8-bit input representing a binary number
   output reg [3:0] num_of_ones  // 4-bit output representing the number of '1's in the binary number
);
   always @* begin
      num_of_ones = 0;  // Initialize the count of '1's to zero

      // Loop through each bit of the 8-bit binary_number input
      for (integer i = 0; i < 8; i = i + 1) begin
         // Check if the current bit is '1'
         if (binary_number[i])
            num_of_ones = num_of_ones + 1;  // Increment the count if the current bit is '1'
      end
   end
endmodule
```

TESTBENCH CODE:

```
`timescale 1ns / 1ps  // Specify the simulation timescale (1 nanosecond resolution, 1 picosecond precision)


module tb_count_ones;
   reg [7:0] binary_number;  // 8-bit register for the binary number input
   wire [3:0] num_of_ones;   // 4-bit wire for the output number of '1's


   count_ones uut (
      .binary_number(binary_number),
      .num_of_ones(num_of_ones)
   );


   initial begin
      // $dumpfile("sim.vcd");
      // $dumpvars(0, testbench);


      binary_number = 8'b00000000;  // Initialize binary_number to 8'b00000000
      #10;  // Wait for 10 time units
```

```
    binary_number = 8'b00000001;  // Change binary_number to 8'b00000001
    #10;
    binary_number = 8'b00000111;
    #10;
    binary_number = 8'b01010101;
    #10;
    binary_number = 8'b11010101;
    #10;
    binary_number = 8'b01110011;
    #10;
    binary_number = 8'b11111111;
    #10;
    $finish;  // Finish simulation

  end

  initial begin
    $monitor("binary_number = %b, num_of_ones = %d", binary_number, num_of_ones);
  end
endmodule
```

TERMINAL OUTPUT:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\iverilog\bin> iverilog -o m code.v testbench.v
PS C:\iverilog\bin> vvp m
VCD info: dumpfile sim.vcd opened for output.
binary_number = 00000000, num_of_ones =  0
binary_number = 00000001, num_of_ones =  1
binary_number = 00000111, num_of_ones =  3
binary_number = 01010101, num_of_ones =  4
binary_number = 11010101, num_of_ones =  5
binary_number = 01110011, num_of_ones =  5
binary_number = 11111111, num_of_ones =  8
testbench.v:30: $finish called at 70000 (1ps)
PS C:\iverilog\bin> gtkwave sim.vcd

GTKWave Analyzer v3.3.100 (w)1999-2019 BSI

[0] start time.
[70000] end time.
```

GTKWAVE OUTPUT: