DIGITAL DESIGN

Lab Project

Kshitiz Bansal 2019A7PS0012U
Dhruv Maheshwari 2019A7PS0020U

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Dr Nilesh Goel sir for providing us this opportunity wherein we could apply our knowledge in solving the problem. We came across many new things which would be helpful to us in the future.

We all are thankful to you for giving this assignment.

PROBLEM STATEMENT

Write a Verilog code and testbench which measure the intensity of ambient light (as a input BCD number between 0 to 50) and display it digitally using 16 bit number.

Link: https://www.edaplayground.com/x/CrAa

VERILOG CODE

```
module bcd_binary(input [7:0] bcd, output reg [15:0]out);
 reg [7:0] binary;
 real w, c;
 always @ (bcd)
        begin
         if (bcd[3:0]<4'b1010)
           begin
             binary=(bcd[7:4]*4'b1010) + {4'b0, bcd[3:0]};
             w=binary;
        c=w/8'b00110010;
        out=c*{16{1'b1}};
           end
     else
      binary={8{1'bx}};
   end
endmodule
```

LOGICAL EXPLAINATION

BCD number is valid from 0000 to 1001, states such as 1010, 1011, 1100, 1101, 1110 and 1111 are invalid states.

Hence, *bcd[3:0]<4'b1010* ensures that there are no invalid states taken into consideration.

*bcd[7:4]*4'b1010*, multiplies the first four digits of BCD input by 10.

{4'b0, bcd[3:0]}, the last 4 digits are added to the number obtained above. These two steps are done to get the binary equivalent of the BCD number.

c=w/8'b00110010; out=c{16{1'b1}};* this is used to get the output according to the formula $\frac{x*(2^{16}-1)}{50}$ where x and 50 are in binary.

In case the last 4 digits are >=1010, the else statement takes care of it and assigns null value.

TESTBENCH

```
module testbench;
  reg [7:0] bcd;
  wire [15:0] binary;
  initial
    begin
      $dumpfile("dump.vcd");
      $dumpvars (1, testbench);
      bcd=8'b00000000;
    end
  always
        #1 bcd=bcd+1;
  initial
        #81 $stop;


  initial
    begin
    $monitor("Input in BCD is: %8b, output in Binary: %16b, output in
Decimal is %5d", bcd, binary, binary);
        end
  bcd_binary b1 (bcd, binary);
endmodule
```

LOGICAL EXPLAINATION

We start by declaring reg [7:0] bcd; wire [15:0] binary;

We initialize the value of BCD number as 0 and use *always* statement to increase the value of BCD number by 1 after a delay of 1 ns. This cycle goes on for 81 times as there are few inputs which are illegal in BCD system.

WAVEFORM

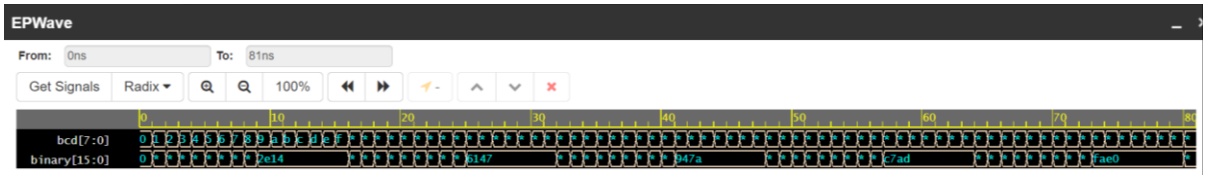

Fig 1: Showing complete waveform

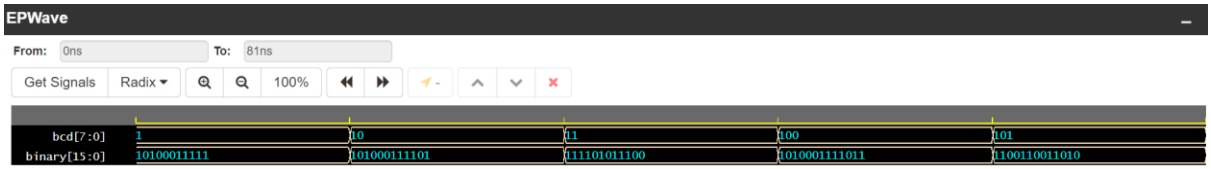

Fig 2: Showing input and output for BCD number from 0 to 5

For eg, BCD input is 00000011 (i.e. 3 in decimal), the output is 111101011100, which is 3932 (in decimal) according to formula $\frac{3*(2^{16}-1)}{50}$.

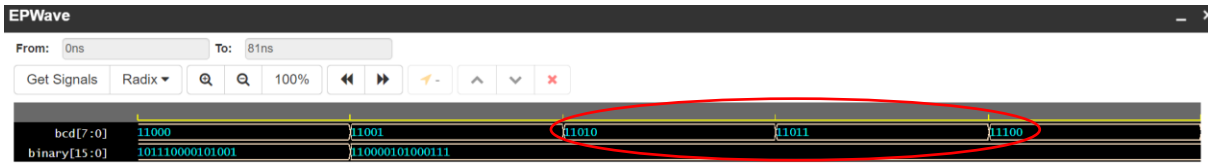

Fig 3: Showing invalid BCD number

For eg, BCD input is 00011010, the last 4 digits are >=1010, hence the else statement in the code executes and we don't get any output for this number.