Experiment-5

Design Verilog HDL to implement Decimal adder.

Aim: To design Verilog HDL code for Decimal adder

Numbers considered: 9 and 8

9 + 8 = 17

1001 + 1000 = 10001

The sum 17 is more than 9. So, add 6 to it.

17+6 = 23 (in binary 23 is 10111)

The least significant 4 bits (which is "0111") represent the units digit and the MSB (4th bit which is '1') bit represents the tens digit.

Verilog code:

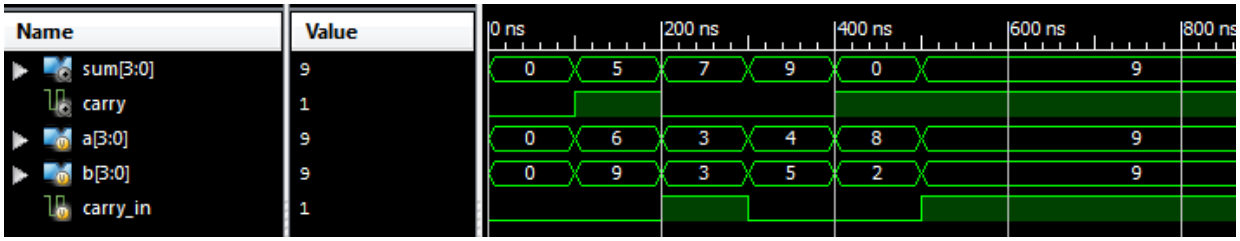
```
module bcd_adder(a,b,carry_in,sum,carry);
   input [3:0] a,b;
   input carry_in;
   output reg [3:0] sum;
   output reg carry;
   reg [4:0] sum_temp;
   always @(a,b,carry_in)
   begin
      sum_temp = a+b+carry_in;
      if(sum_temp > 9)
         begin
         sum_temp = sum_temp+6;
         carry = 1;
         sum = sum_temp[3:0];
         end
      else
         begin
         carry = 0;
         sum = sum_temp[3:0];
         end
   end
endmodule
```

Testbench for BCD adder:

```
module tb_bcdadder;
   reg [3:0] a;
   reg [3:0] b;
   reg carry_in;
   wire [3:0] sum;
```

```
    wire carry;
    bcd_adder uut ( .a(a), .b(b), .carry_in(carry_in), .sum(sum), .carry(carry));
    initial begin
        a = 0;  b = 0;  carry_in = 0;   #100;
        a = 6;  b = 9;  carry_in = 0;   #100;
        a = 3;  b = 3;  carry_in = 1;   #100;
        a = 4;  b = 5;  carry_in = 0;   #100;
        a = 8;  b = 2;  carry_in = 0;   #100;
        a = 9;  b = 9;  carry_in = 1;   #100;
    end
endmodule
```

Timing Diagram



Result

Verilog HDL code for Decimal adder is designed and simulated.