Experiment No. 07

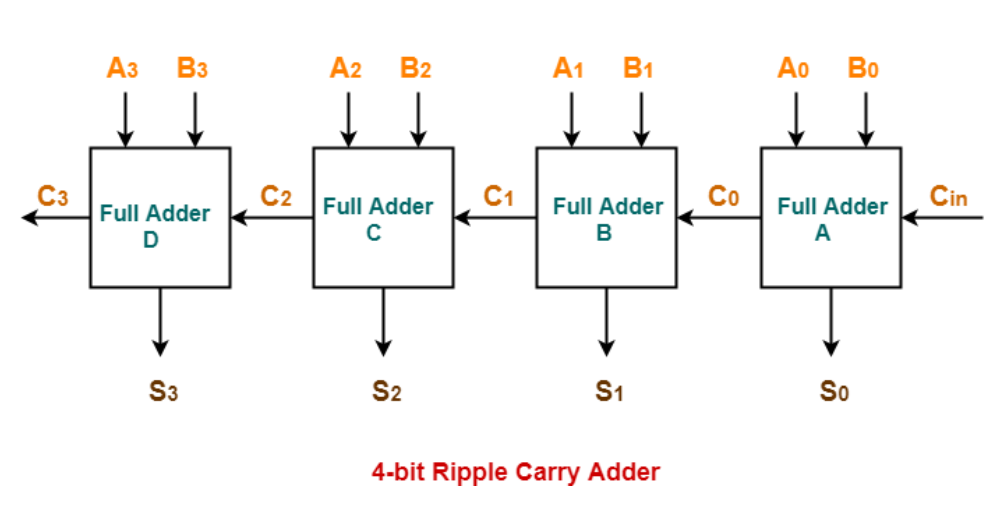
**4 bit ripple carry adder using Verilog HDL**

**OBJECTIVE:** To implement 4 bit ripple carry adder using Verilog HDL .

**SOFTWARE**: EDA playground

**THEORY:**

Ripple carry adder is a combinational logic circuit used for the purpose of adding two n-bit binary numbers. 4-bit ripple carry adder is used for adding two 4-bit binary numbers. N-bit ripple carry adder is used for adding two N-bit binary numbers.

****

* Ripple Carry Adder works in different stages.
* Each full adder takes the carry-in as input and produces carry-out and sum bit as output.
* The carry-out produced by a full adder serves as carry-in for its adjacent most significant full adder.
* When carry-in becomes available to the full adder, it activates the full adder.
* After full adder becomes activated, it comes into operation.

**PROCEDURE:**

1. Open EDA playground
2. Write Verilog code and test bench
3. Select tool and click on open EV waveform
4. Generate Waveform.

**OUTPUT:**

**Code :**

// Code your testbench here

// or browse Examples

module test;

  reg[3:0] a,b;

  reg c\_in;

  wire [3:0] s;

  wire c\_out;

  full\_adder4 dut (

    .a(a),

    .b(b),

    .c\_in(c\_in),

    .s (s),

    .c\_out (c\_out));

initial begin

    $dumpvars(1,test);

    #5;

    a=0;

    b=0;

    c\_in=0;

    #5;

    a=2;

    b=3;

    #5

    c\_in=1;

    #5

    c\_in=0;

    a=4'b1111;

    b=4'b1111;

    #5

    c\_in =1;

    #5

    $finish;

  end

endmodule

**Testbench :**

// Code your design here

module full\_adder(

  input a,b,c\_in,

  output s,c\_out);

  wire net1,net2,net3;

  xor (net1,a,b);

  xor(s,net1,c\_in);

  and(net3,a,b);

  and(net2,net1,c\_in);

  or(c\_out,net2,net3);

  endmodule

module full\_adder4(

      input[3:0] a,b,

      input c\_in,

      output[3:0] s,

      output c\_out );

      wire[3:0] carry;

      full\_adder fa0(

        .a (a[0]),

        .b (b[0]),

        .c\_in (c\_in),

        .s(s[0]),

        .c\_out (carry[0]));

full\_adder fa1(

  .a (a[1]),

  .b (b[1]),

  .c\_in (carry[0]),

  .s(s[1]),

  .c\_out (carry[1]));

  full\_adder fa2(

    .a (a[2]),

    .b (b[2]),

    .c\_in (carry[1]),

    .s(s[2]),

    .c\_out (carry[2]));

      full\_adder fa3(

        .a (a[3]),

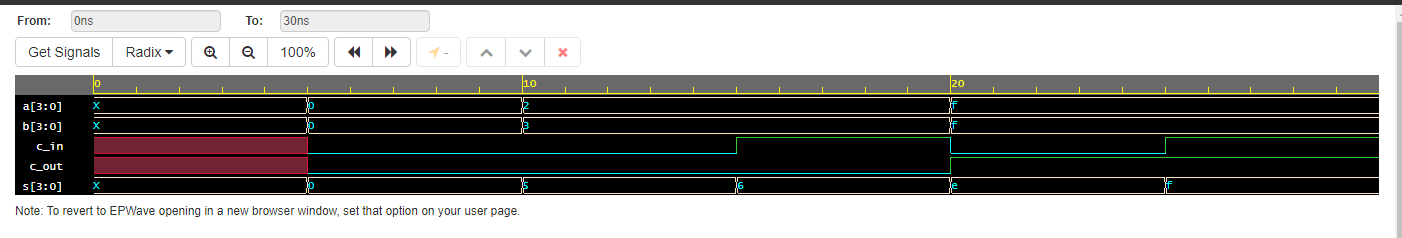
        .b (b[3]),

        .c\_in (carry[2]),

        .s(s[3]),

        .c\_out (c\_out));

      Endmodule

** Waveform:**

**CONCLUSION:**

**Implementation of 4 bit ripple carry adder using Verilog HDL was successfully performed. The output of the 4 bit ripple carry adder was generated and verified.**