# LAB 07
# OPEN ENDED LAB

**OBJECTIVE:**

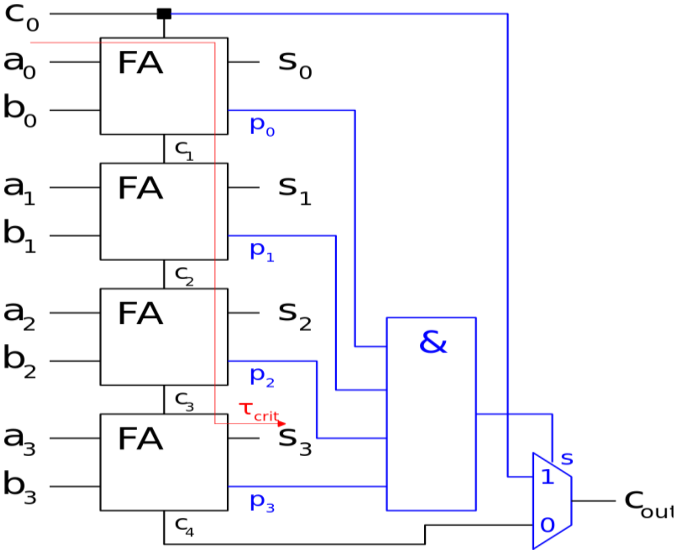
To Design, implement and simulate 16 – bits Carry Skip adder.

**HARDWARE / SOFTWARE REQUIRED:**

Hardware which is used in 16-bits carry skip adder is 4 full adders,1-AND gate and 1 multiplexer. We have used a software Xilinx 14.7 for performing 16 bits carry skip adder.

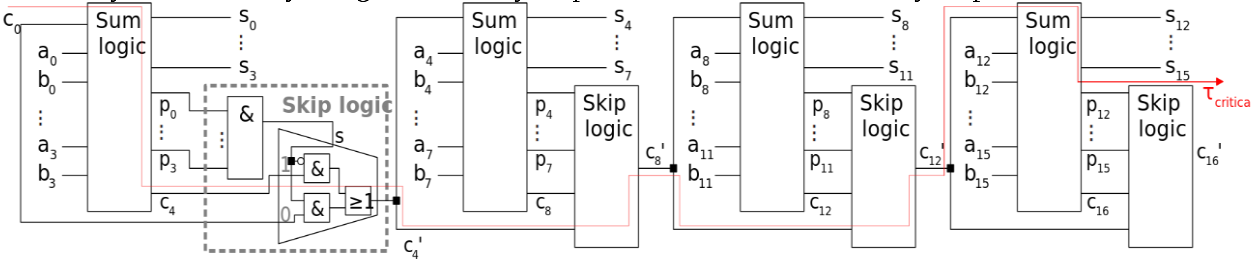
**DIAGRAM:**

**METHODOLOGY:**



**CARRY SKIP ADDER**

A carry-skip adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder.



16-bit Carry Skip Adder

**Why We Prefer CSA over RCA // Merits of CSA:**

A carry-skip adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder.

Unlike other fast adders, carry-skip adder performance is increased with only some of the combinations of input bits. This means, speed improvement is only probabilistic.

CSA compared to RCA consume more power, more area and less delay. To overcome the problem of power and area,the CSA design makes use of efficient full adders. Among the tested three efficient full-adders N-10T is found to be

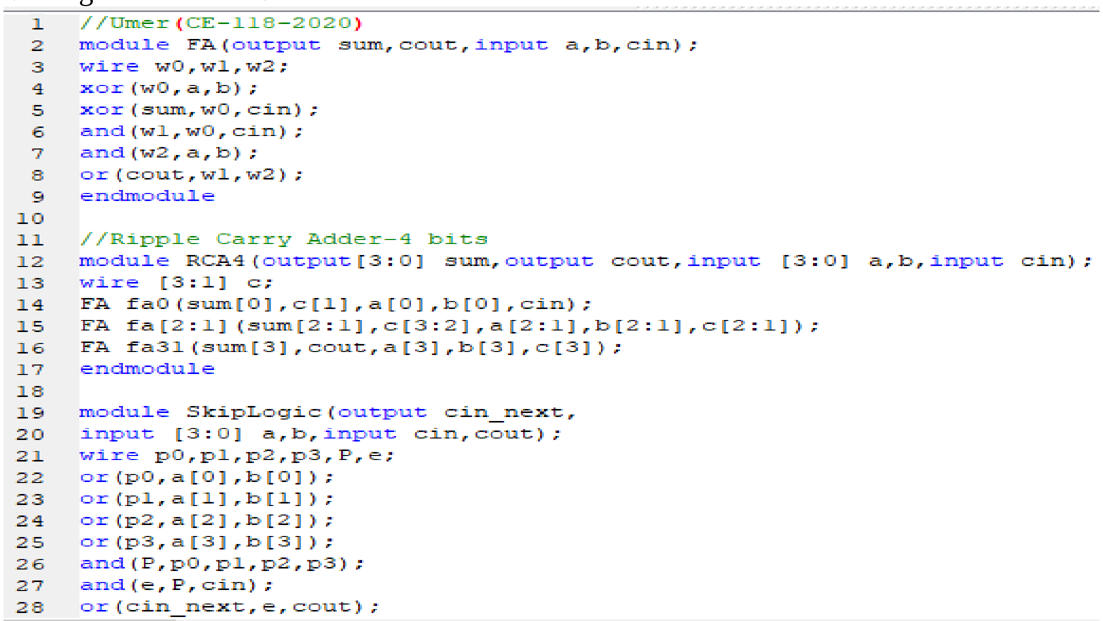
the most efficient full-adder. Simulation results show that CSA is faster than RCA. Therefore in applications which require fast and accurate addition operations, the CSA is preferred over RCA.

Furthermore, 8-block CSA is faster than 4-block CSA and 4-block CSA is faster than 2-block CSA too. Also the tree stages of CSA consume more power and area when compared to RCA.

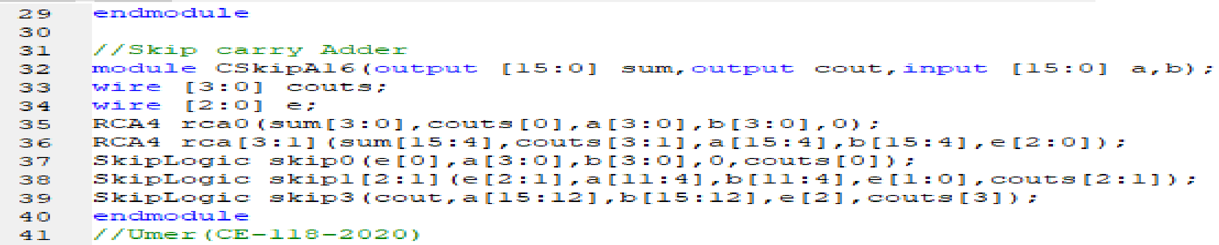
## SOURCE CODE:

### 16 Bits Carry Skip Adder

Verilog Module Code:

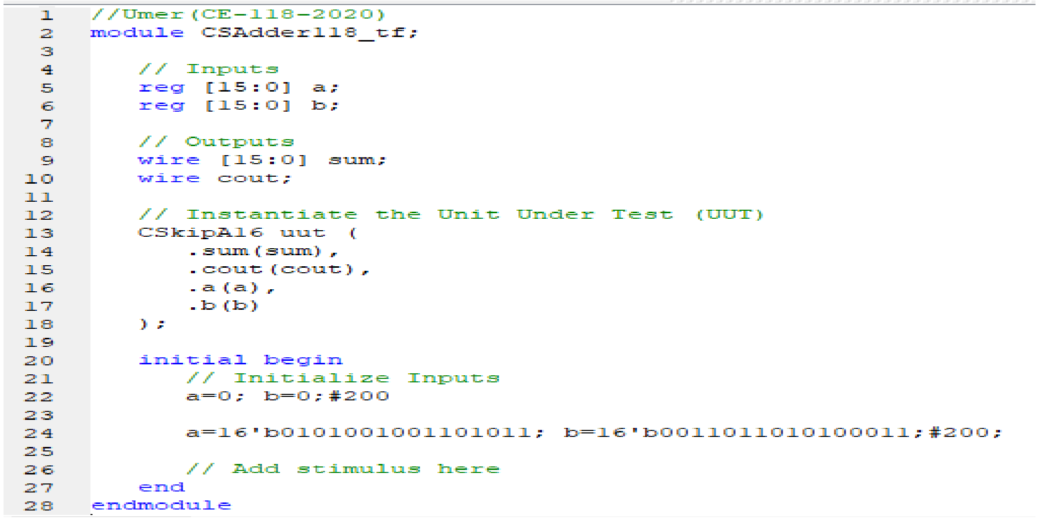


```
//Umer(CE-118-2020)
module FA(output sum,cout,input a,b,cin);
wire w0,w1,w2;
xor(w0,a,b);
xor(sum,w0,cin);
and(w1,w0,cin);
and(w2,a,b);
or(cout,w1,w2);
endmodule

//Ripple Carry Adder-4 bits
module RCA4(output[3:0] sum,output cout,input [3:0] a,b,input cin);
wire [3:1] c;
FA fa0(sum[0],c[1],a[0],b[0],cin);
FA fa[2:1](sum[2:1],c[3:2],a[2:1],b[2:1],c[2:1]);
FA fa31(sum[3],cout,a[3],b[3],c[3]);
endmodule

module SkipLogic(output cin_next,
input [3:0] a,b,input cin,cout);
wire p0,p1,p2,p3,P,e;
or(p0,a[0],b[0]);
or(p1,a[1],b[1]);
or(p2,a[2],b[2]);
or(p3,a[3],b[3]);
and(P,p0,p1,p2,p3);
and(e,P,cin);
or(cin_next,e,cout);
```



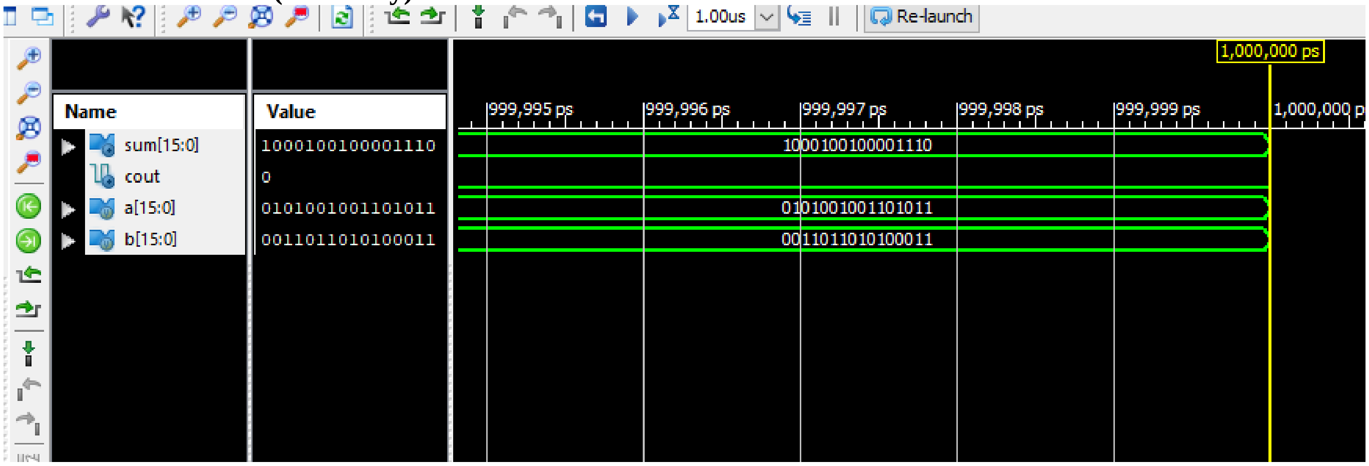
```
endmodule

//Skip carry Adder
module CSkipA16(output [15:0] sum,output cout,input [15:0] a,b);
wire [3:0] couts;
wire [2:0] e;
RCA4 rca0(sum[3:0],couts[0],a[3:0],b[3:0],0);
RCA4 rca[3:1](sum[15:4],couts[3:1],a[15:4],b[15:4],e[2:0]);
SkipLogic skip0(e[0],a[3:0],b[3:0],0,couts[0]);
SkipLogic skip1[2:1](e[2:1],a[11:4],b[11:4],e[1:0],couts[2:1]);
SkipLogic skip3(cout,a[15:12],b[15:12],e[2],couts[3]);
endmodule
//Umer(CE-118-2020)
```

Verilog Test Fixture Code:



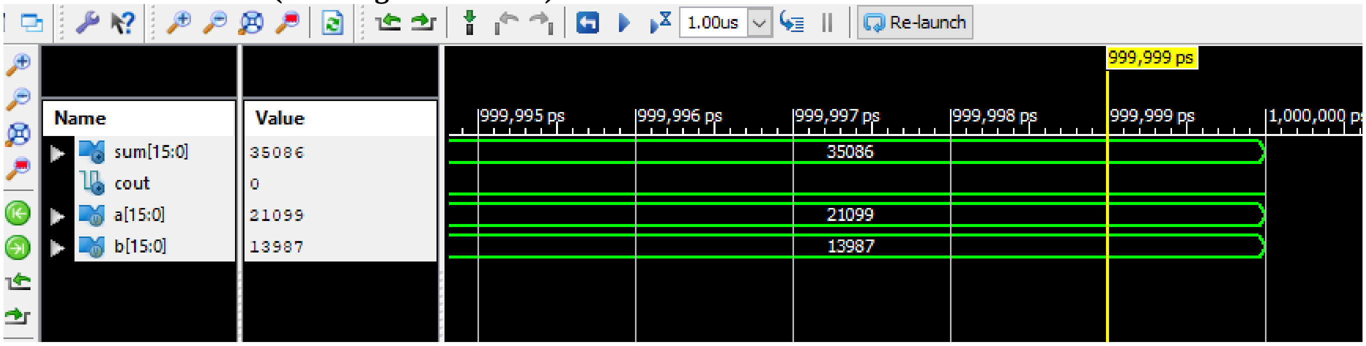
```
//Umer(CE-118-2020)
module CSAdder118_tf;

    // Inputs
    reg [15:0] a;
    reg [15:0] b;

    // Outputs
    wire [15:0] sum;
    wire cout;

    // Instantiate the Unit Under Test (UUT)
    CSkipA16 uut (
        .sum(sum),
        .cout(cout),
        .a(a),
        .b(b)
    );

    initial begin
        // Initialize Inputs
        a=0; b=0;#200

        a=16'b0101001001101011; b=16'b0011011010100011;#200;

        // Add stimulus here
    end
endmodule
```

## Result / Wave Form: (In Binary)



| Name | Value |
|---|---|
| sum[15:0] | 1000100100001110 |
| cout | 0 |
| a[15:0] | 0101001001101011 |
| b[15:0] | 0011011010100011 |

## Result / Wave Form: (In Unsigned Decimal)



| Name | Value |
|---|---|
| sum[15:0] | 35086 |
| cout | 0 |
| a[15:0] | 21099 |
| b[15:0] | 13987 |

## CONCLUSION:

From this open ended lab we learned that Carry skip ahead adder addition of 16 bits needs 33 inputs, like Input A with 16 bits, Input B with 16 bits followed by a Carry input of 1 bit. The resulting output is a 17 bits output with 16 bits Sum output and a 1 bit Carry output.

We also learned that the carry skip adder provides a compromise between a ripple carry adder and a CLA adder. It consists of a speed up carry chain called skip chain. This chain defines the distribution of ripple carry blocks, which compose the skip carry blocks, constituting skip adder. It consists of a special circuit which quickly detects if all bits to be added are different. CSA divides the words to be added into blocks.