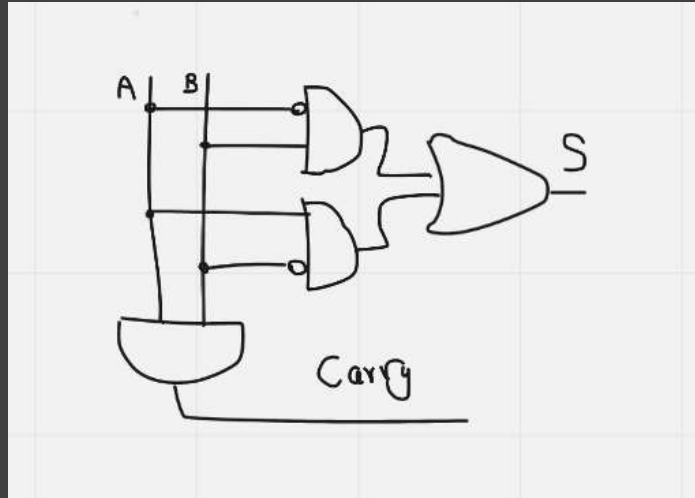


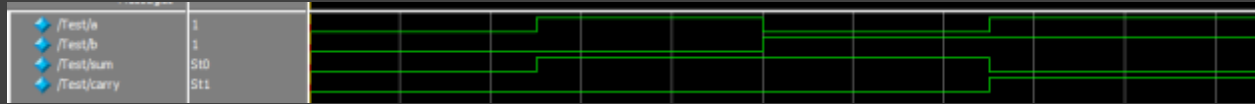
۱. طراحی نیم جمع کننده



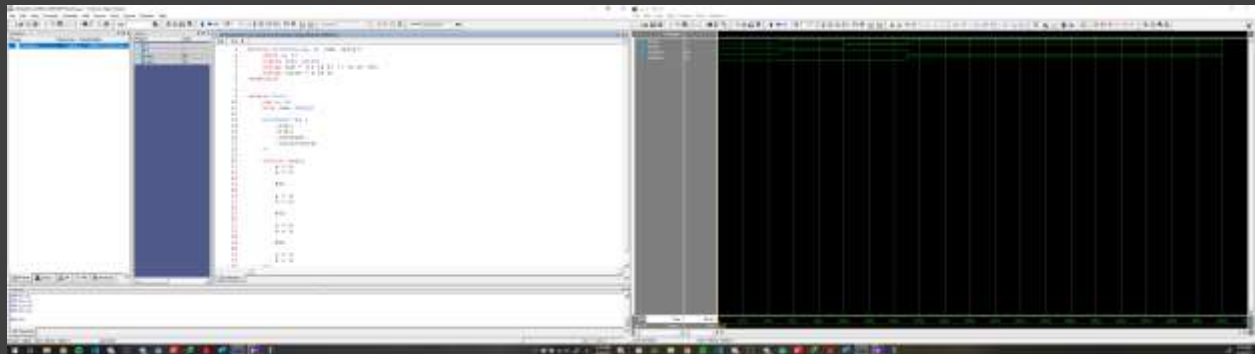
```
module HalfAdder(a, b, sum, carry);  
    input a, b;  
    output sum, carry;  
    assign sum = (!a && b) || (a && !b);  
    assign carry = a && b;  
endmodule
```

```
module Test;  
    reg a, b;  
    wire sum, carry;  
  
    HalfAdder ha (  
        .a(a),  
        .b(b),  
        .sum(sum),  
        .carry(carry)  
    );  
  
    initial begin  
        a = 0;  
        b = 0;  
  
        #50  
  
        a = 1;  
        b = 0;  
  
        #50  
  
        a = 0;  
        b = 1;  
  
        #50  
  
        a = 1;  
        b = 1;  
  
    end  
endmodule
```

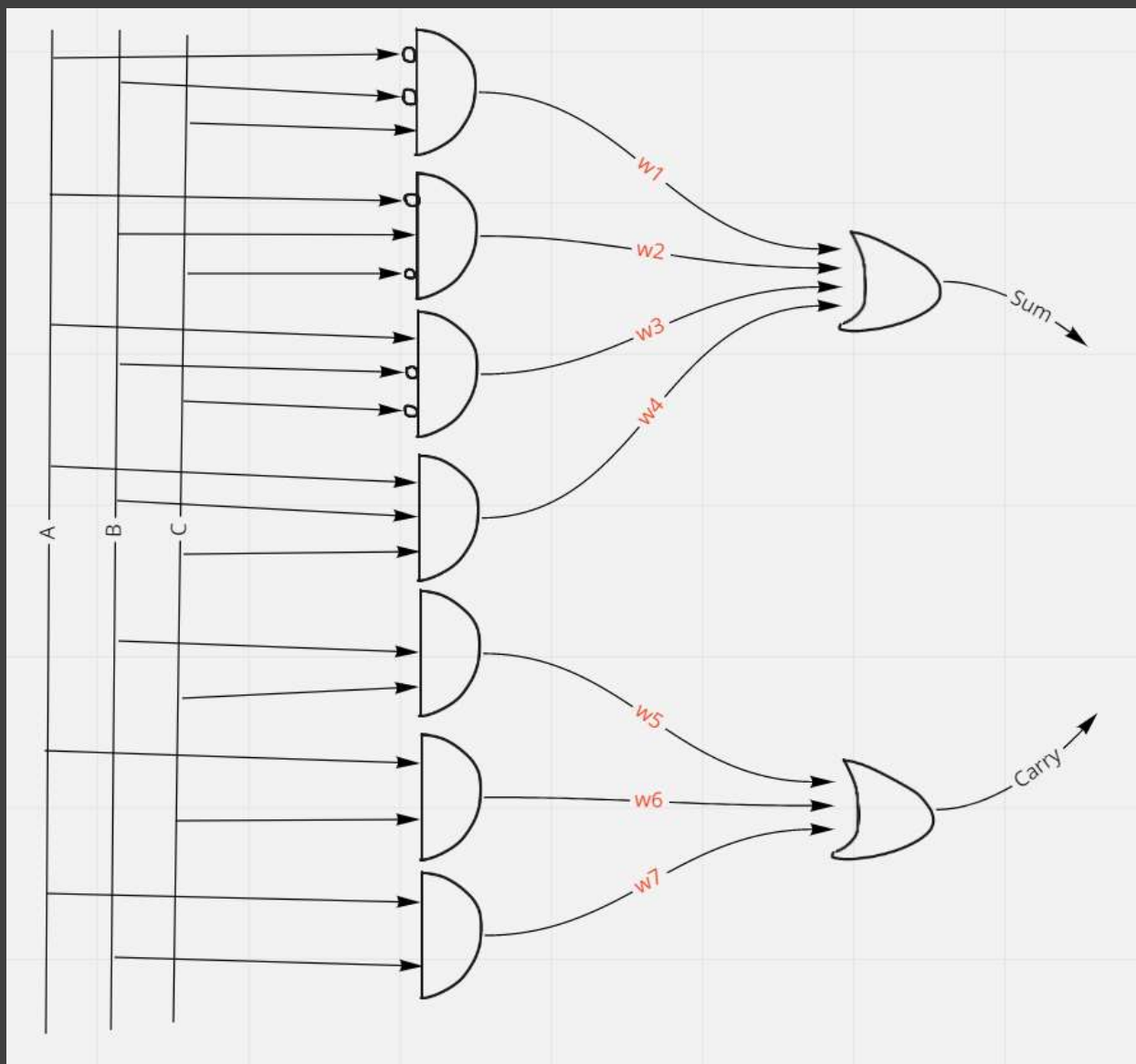
شبیه سازی شکل موج با ModelSim



اسکرین شات نهایی



۲. طراحی تمام جمع کننده



```
module FullAdder(a, b, c, sum, carry);
    input a, b, c;
    output sum, carry;
    wire w1 = !a && !b && c;
    wire w2 = !a && b && !c;
    wire w3 = a && !b && !c;
    wire w4 = a && b && c;
    wire w5 = b && c;
    wire w6 = a && c;
    wire w7 = a && b;
    assign sum = w1 && w2 && w3 && w4;
    assign carry = w5 && w6 && w7;
endmodule
```

```
module TestFullAdder;
    reg a, b, c;
    wire sum, carry;

    FullAdder ha (
        .a(a),
        .b(b),
        .c(c),
        .sum(sum),
        .carry(carry)
    );

    initial begin
        a = 0;
        b = 0;
        c = 0;

        #50
        a = 1;
        b = 0;
        c = 0;

        #50
        a = 0;
        b = 1;
        c = 0;
```

```
#50
a = 1;
b = 1;
c = 0;
```

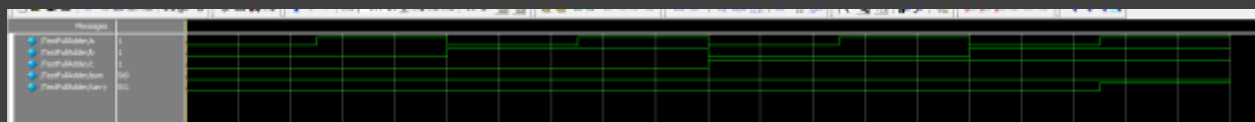
```
#50
a = 0;
b = 0;
c = 1;
```

```
#50
a = 1;
b = 0;
c = 1;
```

```
#50
a = 0;
b = 1;
c = 1;
```

```
#50
a = 1;
b = 1;
c = 1;
```

```
end
endmodule
```





۳. جمع کننده ۴ بیتی

طراحی جمع کننده ۴ بیتی اینطوریه که از یک نیم جمع کننده و سه تمام جمع کننده استفاده می کنیم و carry ها رو پاس میدم. البته این مدل برای cascade کردن مناسب نیست ولی اگه اولیشو تمام جمع کننده کنیم اوکی میشه. در پروژه شماره ۴ من از جمع کننده ۴ بیتی ای استفاده می کنم که ۴ تا تمام جمع کننده یک بیتی دارد. یعنی فقط از خط ۱۹ تا ۲۴ یکم تغییر می کنه و نحوه کردن اینطوریه که FullAdder رو به کار می بریم و مقدار C ورودی را صفر می دهیم.

کد وریلاگ

```
module FourBitAdder(a, b, c, sum, carry);
    input [3:0] a;
    input [3:0] b;
    input c;
    output [3:0] sum;
    output carry;
    wire carry_01;
    wire carry_12;
    wire carry_23;
endmodule

module FourBitAdderTest;
    reg [3:0] a;
    reg [3:0] b;
    reg c;
    wire [3:0] sum;
    wire carry;

    HalfAdder bit_1 (
        .a(a[0]),
        .b(b[0]),
        .sum(sum[0]),
        .carry(carry_01)
    );

    FullAdder bit_2 (
        .a(a[1]),
        .b(b[1]),
        .c(carry_01),
```



```
        .sum(sum[1]),
        .carry(carry_12)
    );

    FullAdder bit_3 (
        .a(a[2]),
        .b(b[2]),
        .c(carry_12),
        .sum(sum[2]),
        .carry(carry_23)
    );

    FullAdder bit_4 (
        .a(a[3]),
        .b(b[3]),
        .c(carry_23),
        .sum(sum[3]),
        .carry(carry)
    );

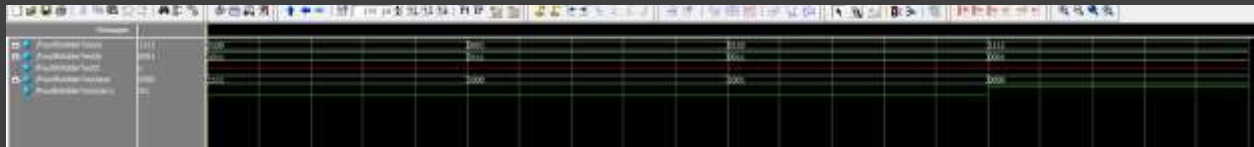
    initial begin
        a = 4'b0100;
        b = 4'b1011;

        #50

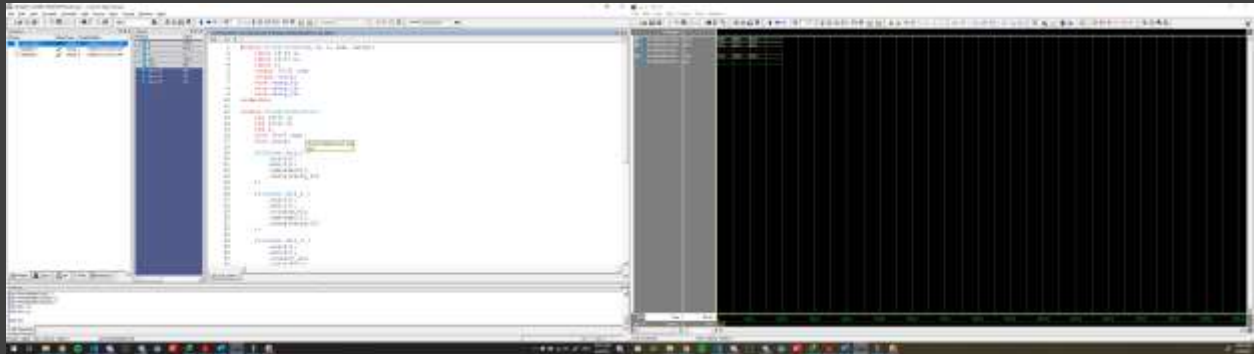
        a = 4'b0001;
        b = 4'b0111;

        #50

        a = 4'b0110;
        b = 4'b0011;
    end
endmodule
```



اسکرین شات از شکل موج یه تست بیشتر داره چون می خواستم ببینم اگه overflow بده چی میشه. که در اسکرین شات بالا آخرین تست جمع ۱۱۱۱ و ۰۰۰۱ هست که در اسکرین شات پایین نیست.



۴. Cascade to 16bit adder

Changes in 4\_bit\_adder.v

```
FullAdder bit_1 (  
    .a(a[0]),  
    .b(b[0]),  
    .c(0),  
    .sum(sum[0]),  
    .carry(carry_01)  
);
```

به مشکل برخورددم

کد وریلاگ

```
module SixteenBitAdder(a, b, c, sum, carry);  
    input [15:0] a;  
    input [15:0] b;  
    input c;  
    output [15:0] sum;  
    output carry;  
    wire carry_block_01;  
    wire carry_block_12;  
    wire carry_block_23;  
endmodule  
  
module SixteenBitAdderTest;  
    reg [15:0] a;  
    reg [15:0] b;  
    reg c;  
    wire [15:0] sum;  
    wire carry;  
  
    FourBitAdder bit_1 (  
        .a(a[3:0]),  
        .b(b[3:0]),  
        .c(0),  
        .sum(sum[3:0]),  
        .carry(carry_block_01)  
    );
```

```
FourBitAdder bit_2 (  
    .a(a[7:4]),  
    .b(b[7:4]),  
    .c(carry_block_01),  
    .sum(sum[7:4]),  
    .carry(carry_block_12)  
);  
  
FourBitAdder bit_3 (  
    .a(a[11:8]),  
    .b(b[11:8]),  
    .c(carry_block_12),  
    .sum(sum[11:8]),  
    .carry(carry_block_23)  
);  
  
FourBitAdder bit_4 (  
    .a(a[15:12]),  
    .b(b[15:12]),  
    .c(carry_block_23),  
    .sum(sum[15:12]),  
    .carry(carry)  
);  
  
initial begin  
    a = 16'd30;  
    b = 16'd12;  
    c = 1'b0;  
  
    #50  
  
    a = 16'd30;  
    b = 16'd12;  
    c = 1'b0;  
  
    #50  
  
    a = 16'd41;  
    b = 16'd2;  
    c = 1'b0;  
  
    #50
```

```
a = 16'd10;  
b = 16'd16;  
c = 1'b1;  
end  
endmodule
```

در کد FourBitAdderTest چون خود FourBitAdder ها Instanciate می شوند و در FourBitAdder ها ما assign داریم ولی اینجا نداریم برای همین عملی انجام نمی دهد. تمامیت کد وریلاگی که زدم از نظر منطقی درسته فقط باید اون قسمت assign ها انجام بشه. اگه دستی بنویسم خیلی فرم ساختار یافته نمیگیره و قشنگ نمیشه. با جستجو کردن به function رسیدم...

علیرضا جعفرتاش

۹۸۳۱۱۲۰۲۹