گزارش‌کارِ اولِ آزمایشگاهِ معماری کامپیوتر

تهیه و تنظیم: مبین خیری

شماره دانشجویی: 994421017

استاد راهنما: دکتر حاجی‌زاده

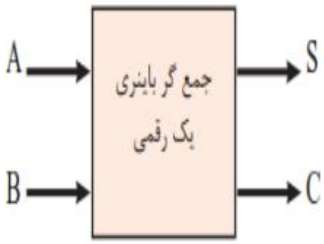
چکیده:

در جلسه‌ی نخستِ درسِ آزمایشگاهِ معماریِ کامپیوتر، درباره‌ی ساختارِ درونی و بیرونیِ نیم‌جمع‌کننده‌ها و تمام‌جمع‌کننده‌ها بحث شد و همچنین بخشِ دیگری از زمانِ کلاس جهتِ آشناییِ کلی با محیط نرم‌افزارِ ISE و دستوراتِ ابتداییِ زبان سطح‌بالای Verilog صرف شد.

در گزارشِ پیش‌رو بنا داریم ابتدا نگاهی گذرا به ساختارِ کلی نیم‌جمع‌کننده‌ها و تمام‌جمع‌کننده‌ها انداخته و سپس به شرح اقداماتِ انجام‌شده در محیطِ نرم‌افزارِ ISE برای ذخیره و تست کردن چند خط کد بپردازیم.

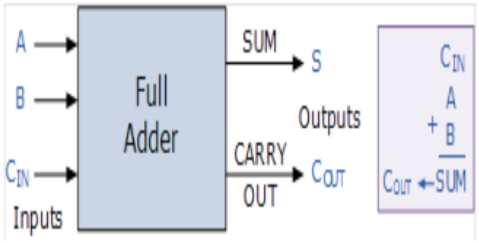
تصویرها و جداول زیر ساختارِ کلیِ انواعِ جمع‌کننده‌ها را نشان می‌دهند که پیش از این در درس مدارهای منطقی با آن‌ها آشنا شده‌ایم:

نیمه جمع کننده(Half adder)

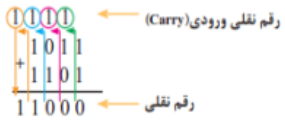


A	B	C	S	
0	0	0	0	0 + 0 = 0
0	1	0	1	0 + 1 = 1
1	0	0	1	1 + 0 = 1
1	1	1	0	1 + 1 =10

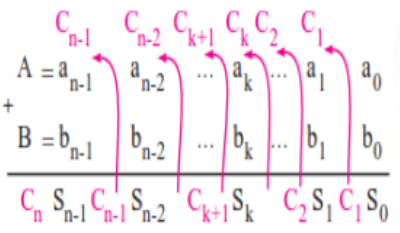
تمام جمع کننده(Full adder)

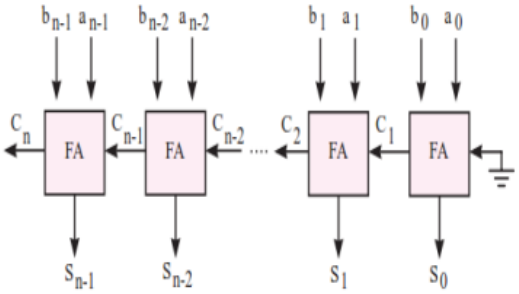


C-in	B	A	Sum	C-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

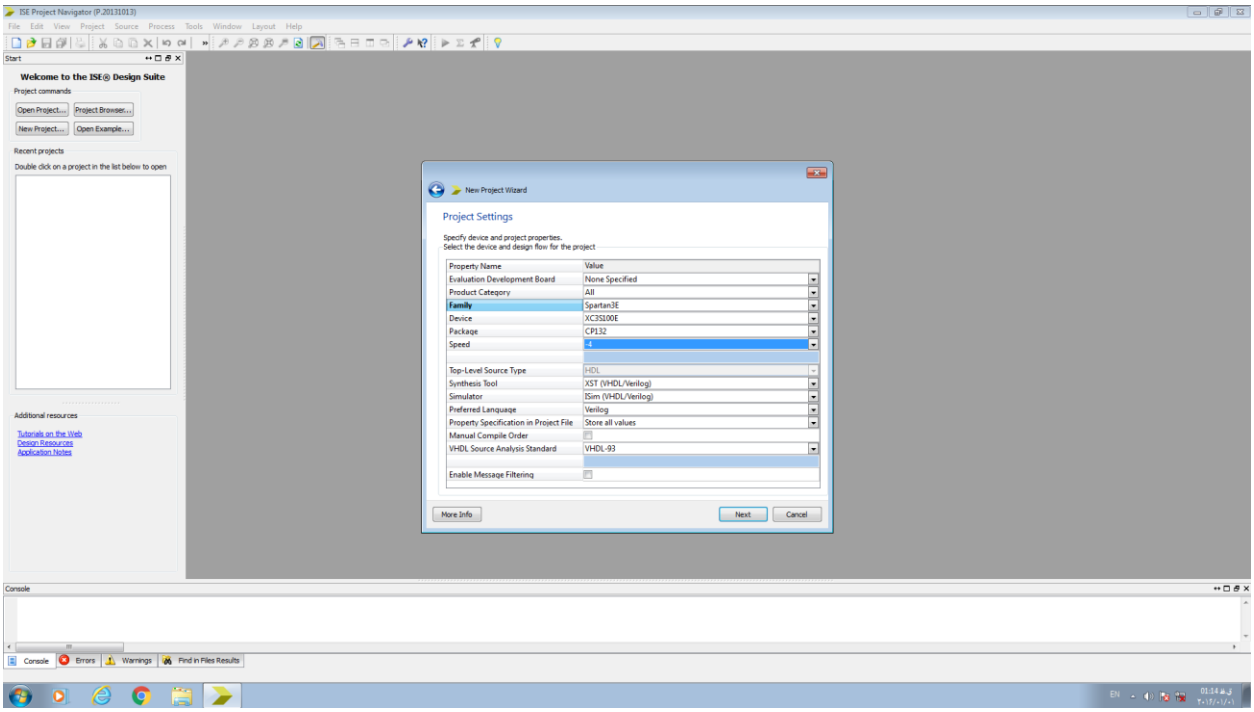


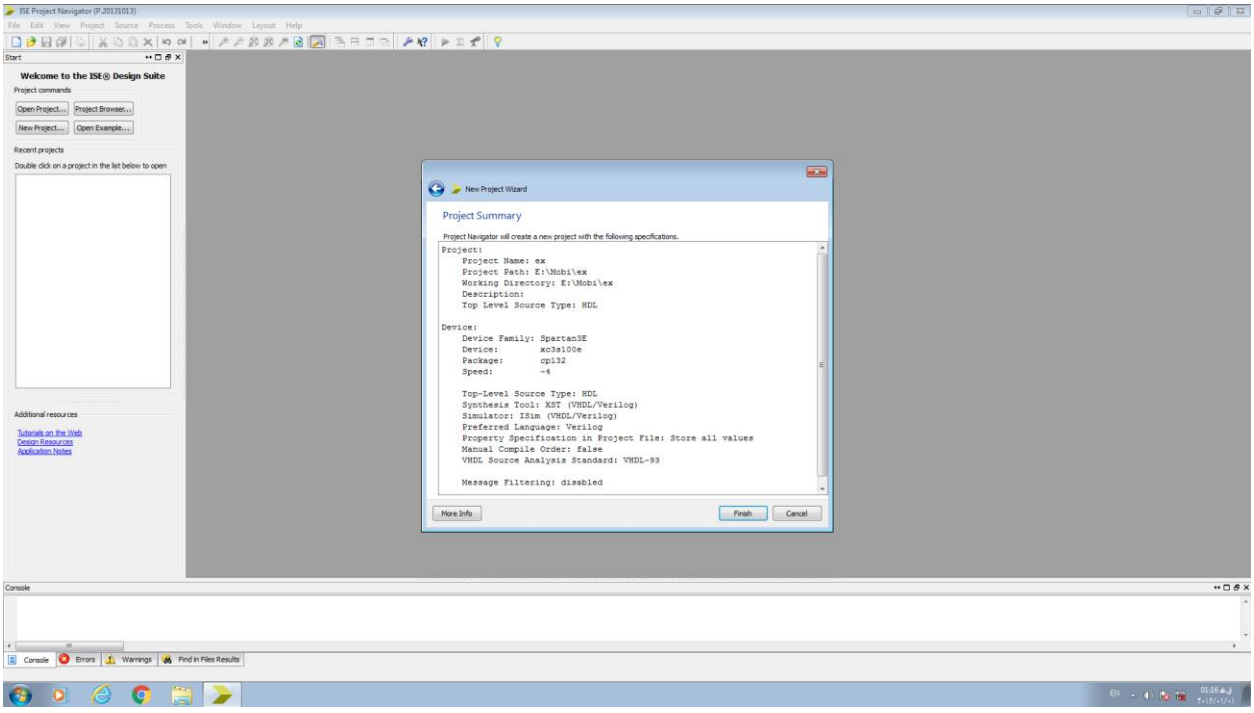
ادامه

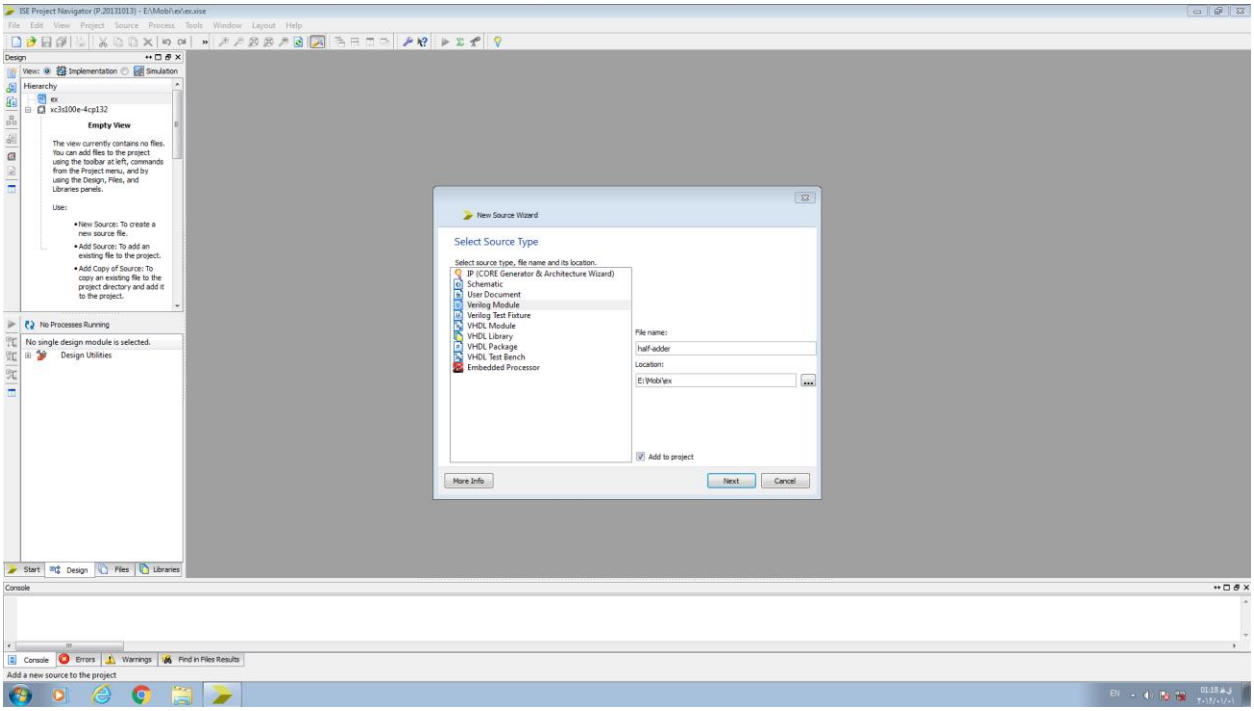


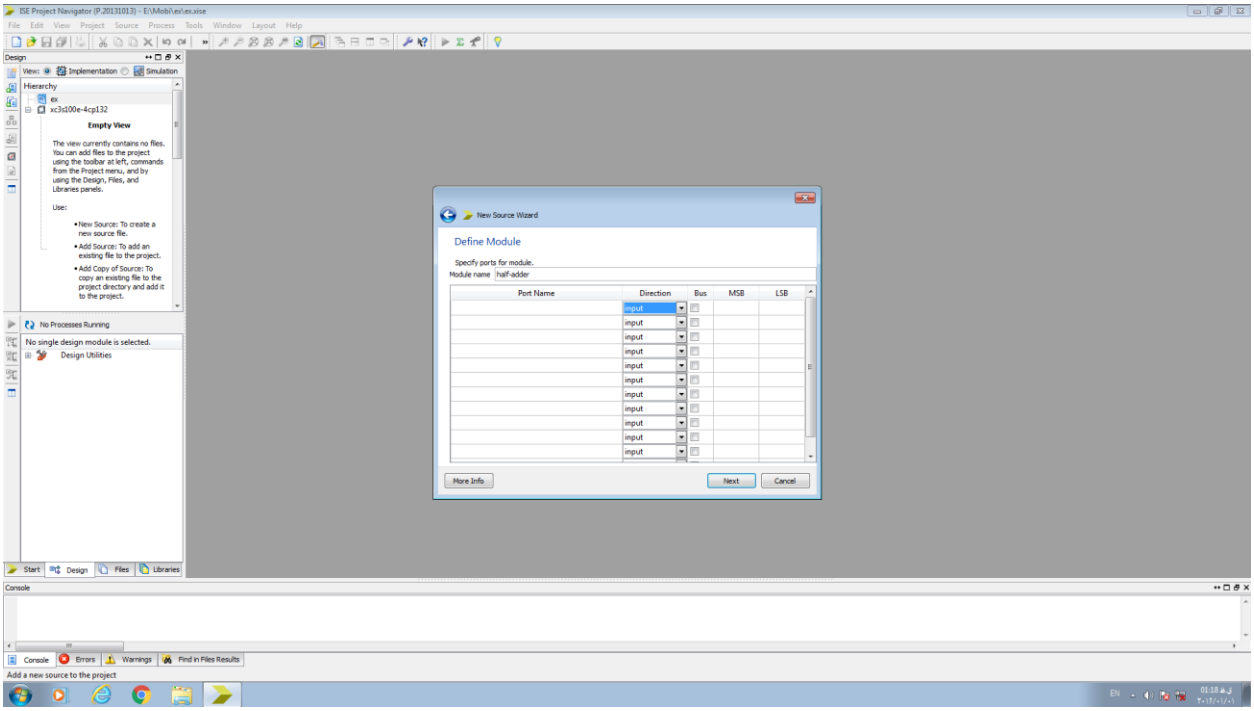


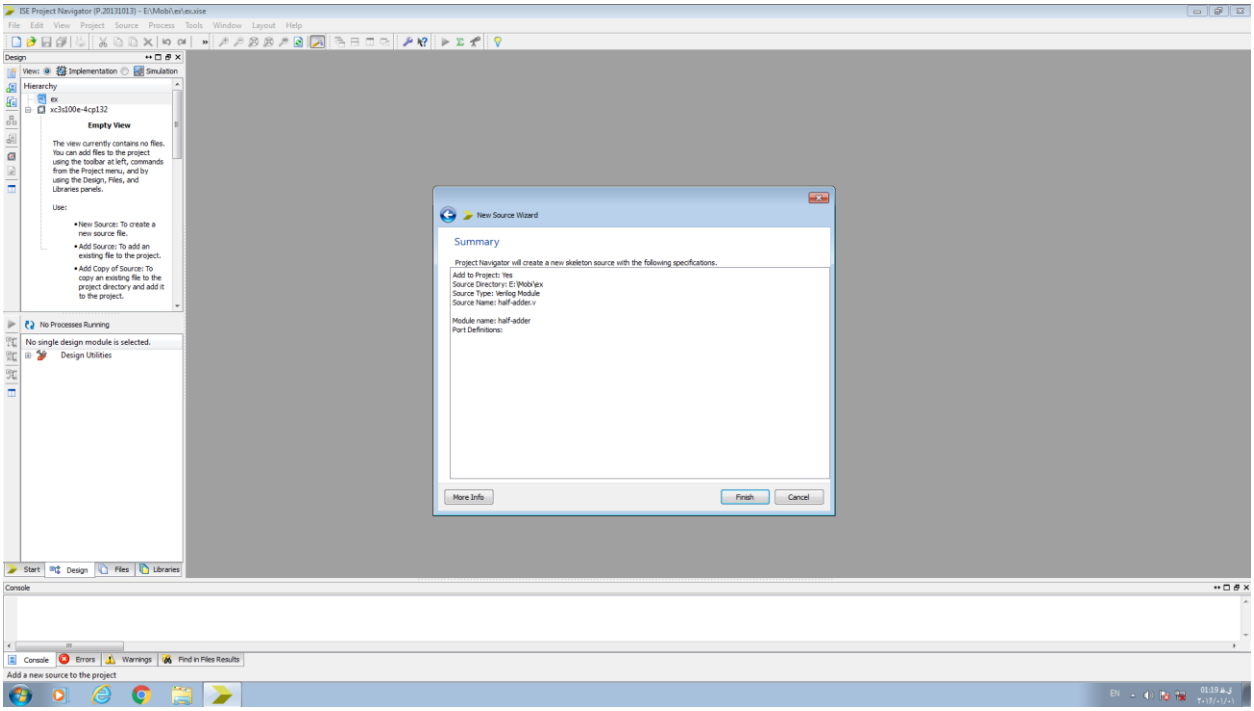
هنگام اضافه کردن یک پروژه‌ی جدید به نرم‌افزار، ابتدا پارامترها بر اساسِ سخت‌افزارِ هدف تعیین کرده و به این شکل پروژه‌ی جدیدی را تعریف می‌کنیم:

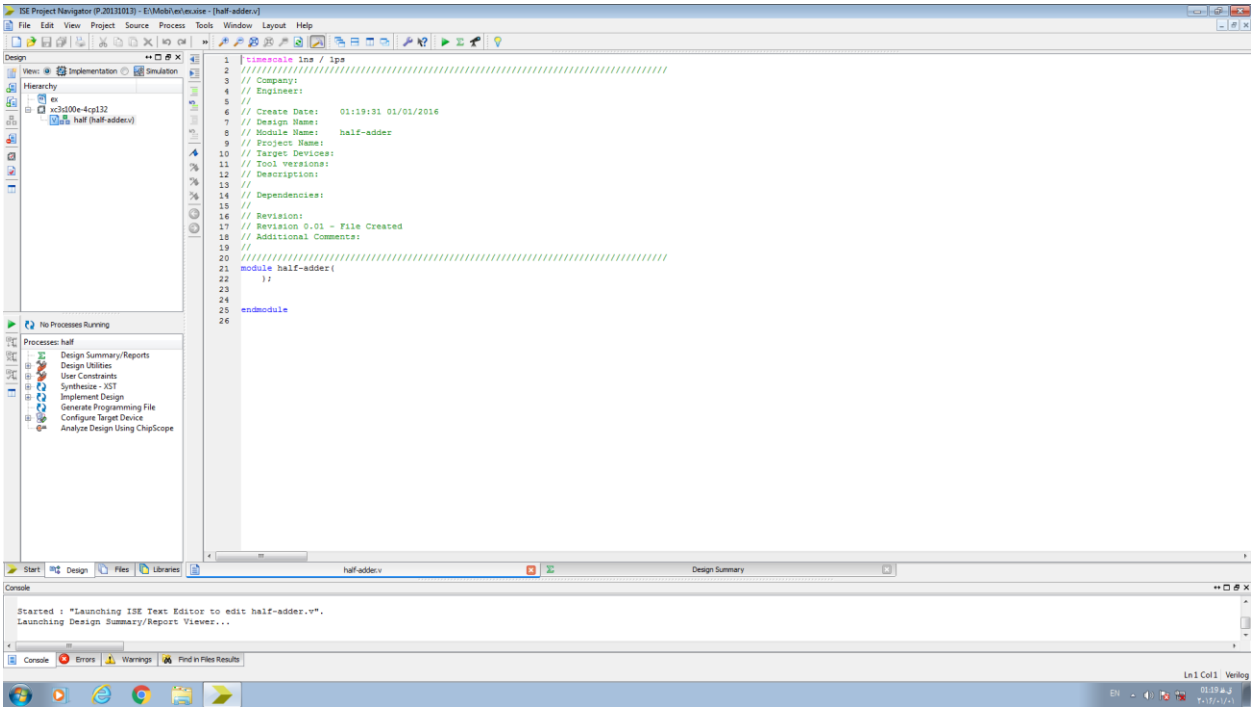





New Source Wizard
Select Source Type
Select source type, file name and its location.
IP (CORE Generator & Architecture Wizard)
Schematic
User Document
Verilog Module
Verilog Test Fixture
VHDL Module
VHDL Library
VHDL Package
VHDL Test Bench
Embedded Processor
File name:
half-adder
Location:
E:\Mobi\ex
Add to project
More Info
Next
Cancel


New Source Wizard
Define Module
Specify ports for module.
Module name half-adder
Port Name
Direction
Bus
MSB
LSB
input
More Info
Next
Cancel


ISE Project Navigator (P.20131013) - E:\Mobi\ex\ex.xise
New Source Wizard
Summary
Project Navigator will create a new skeleton source with the following specifications.
Add to Project: Yes
Source Directory: E:\Mobi\ex
Source Type: Verilog Module
Source Name: half-adder.v
Module name: half-adder
Port Definitions:
More Info
Finish
Cancel
Add a new source to the project

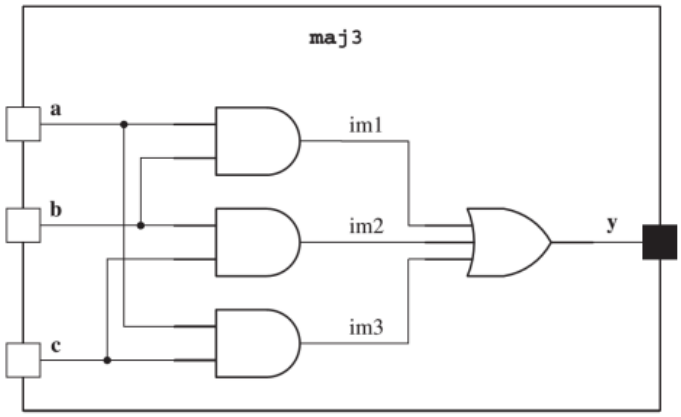

ISE Project Navigator (P.20131013) - E:\Mobi\ex\ex.xise - [half-adder.v]
timescale 1ns / 1ps
// Company:
// Engineer:
// Create Date: 01:19:31 01/01/2016
// Design Name:
// Module Name: half-adder
// Project Name:
// Target Devices:
// Tool versions:
// Description:
// Dependencies:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
module half-adder(
);
endmodule
Started : "Launching ISE Text Editor to edit half-adder.v".
Launching Design Summary/Report Viewer...

در مرحله‌ی بعد ابتدا ساختاری را که قصد داریم به کمکِ نرم‌افزارِ ISE طراحی کنیم، بررسی کرده و برای پیاده‌سازیِ کدهای مربوط به آن وارد عمل می‌شویم:

Gate Level Logic

Gate Primitives

Majority Circuit



Gate Level Logic

Gate Primitives

Majority Circuit

```
module maj3 (a, b, c, y);
    input a, b, c;
    output y;
    wire im1, im2, im3;

    and #(2, 4)
        ( im1, a, b ),
        ( im2, b, c ),
        ( im3, c, a );
    or #(3, 5) (y, im1, im2, im3);

endmodule
```

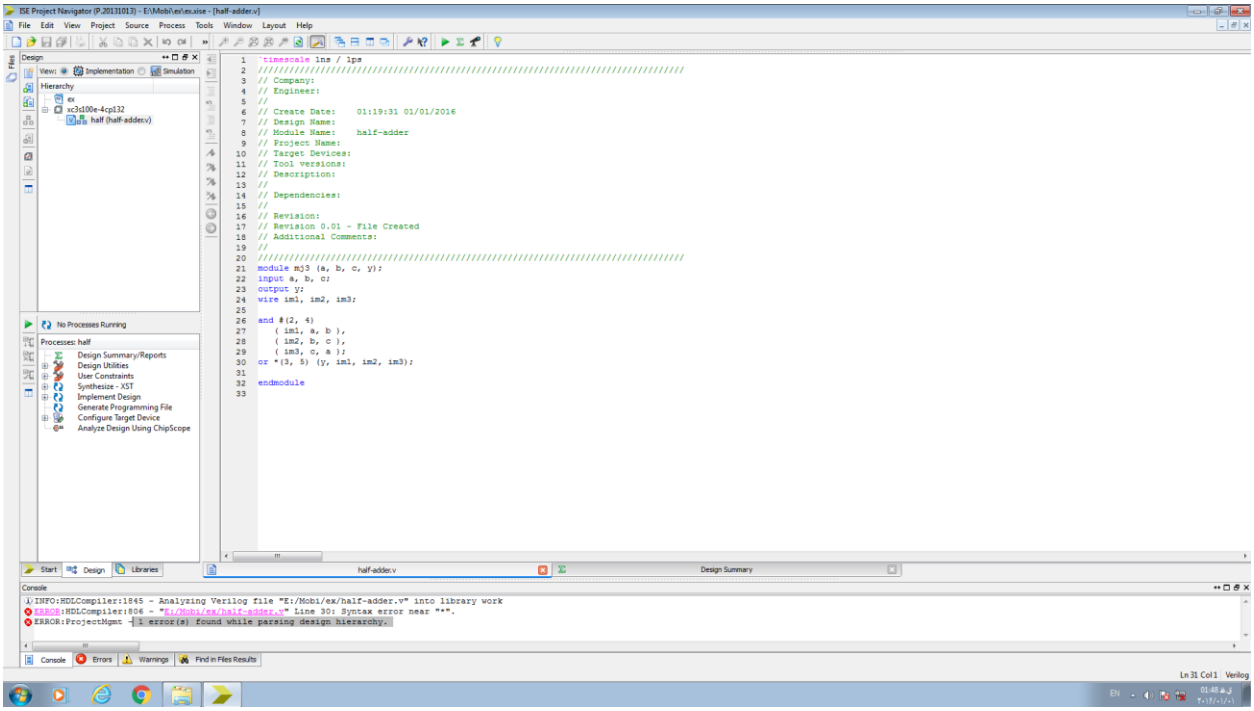
Assign Statements

Bitwise operators

maj3 with assign Statement

```
`timescale 1ns/100ps

module maj3 (input a, b, c, output y);
   assign #(4) y = ( a & b ) | ( b & c ) | ( a & c );
endmodule
```



قطعه‌کدهای زیر برای تستِ ماژولِ طراحی شده به کار رفته و رفتار و عملکرد آن را در یک محیطِ جدید شبیه‌سازی می‌کنند:

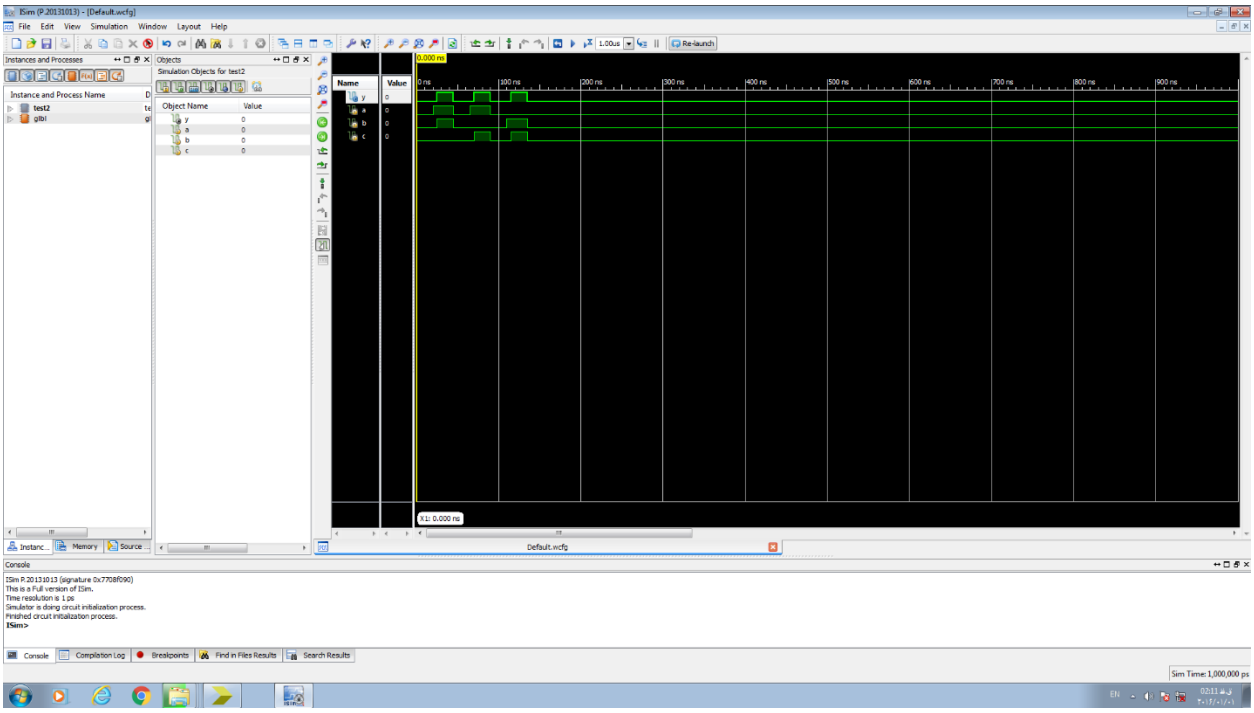
Gate Level Logic

Gate Primitives

Test for Majority Circuit

```
module test_maj;
    // Inputs
    reg a;
    reg b;
    reg c;
    // Outputs
    wire y;
    // Instantiate the Unit Under Test (UUT)
    maj3 uut (
        .a(a),
        .b(b),
        .c(c),
        .y(y));
    initial begin
        // Initialize Inputs
        a = 0; b = 0; c = 0;
        #20 a = 1;
        #5  b = 1;
        #20 a = 0; b = 0; c = 0;
        #20 a = 1;
        #5  c = 1;
        #20 a = 0; b = 0; c = 0;
        #20 b = 1;
        #5  c = 1;
        #20 a = 0; b = 0; c = 0;
    end

endmodule
```

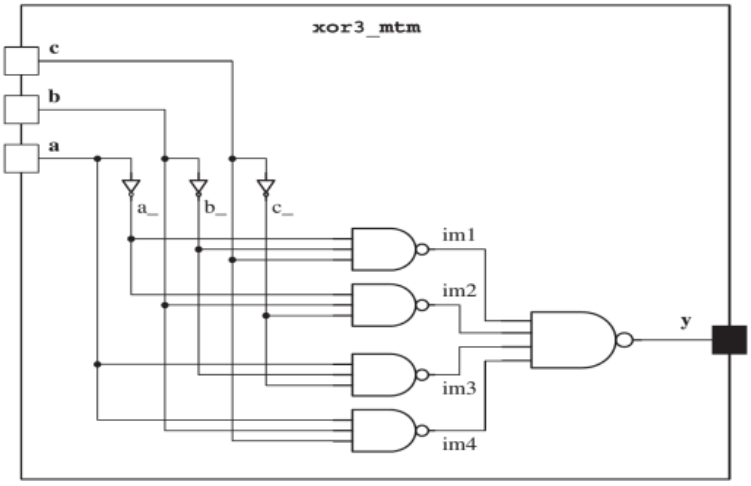


در مرحله‌ی آخر قصد داریم به کمک این نرم‌افزار، یک گیتِ XOR را مشابهِ ساختارِ زیر طراحی کنیم:

Gate Level Logic

Three input XOR

XOR Circuit



Gate Level Logic

Three input XOR

XOR Circuit

```
module xor3_mtm (input a, b, c, output y);
   wire a_, b_, c_;
   wire im1, im2, im3, im4;

   not #(1:3:5, 2:4:6)
      ( a_, a ),
      ( b_, b ),
      ( c_, c );
   nand #(2:4:6, 3:5:7)
      ( im1, a_, b_, c ),
      ( im2, a_, b, c_ ),
      ( im3, a, b_, c_ ),
      ( im4, a, b, c );
   nand #(2:4:6, 3:5:7) (y, im1, im2, im3, im4);

endmodule
```

Min : type : Max Delay

Assign Statements

Bitwise operators

xor3 with assign Statement

```
`timescale 1ns/100ps

module xor3 (input a, b, c, output y);
    assign y = a ^ b ^ c;
endmodule
```