گزارش‌کارِ دومِ آزمایشگاهِ معماری کامپیوتر

تهیه و تنظیم: مبین خیری

شماره دانشجویی: 994421017

استاد راهنما: دکتر حاجی‌زاده

چکیده:

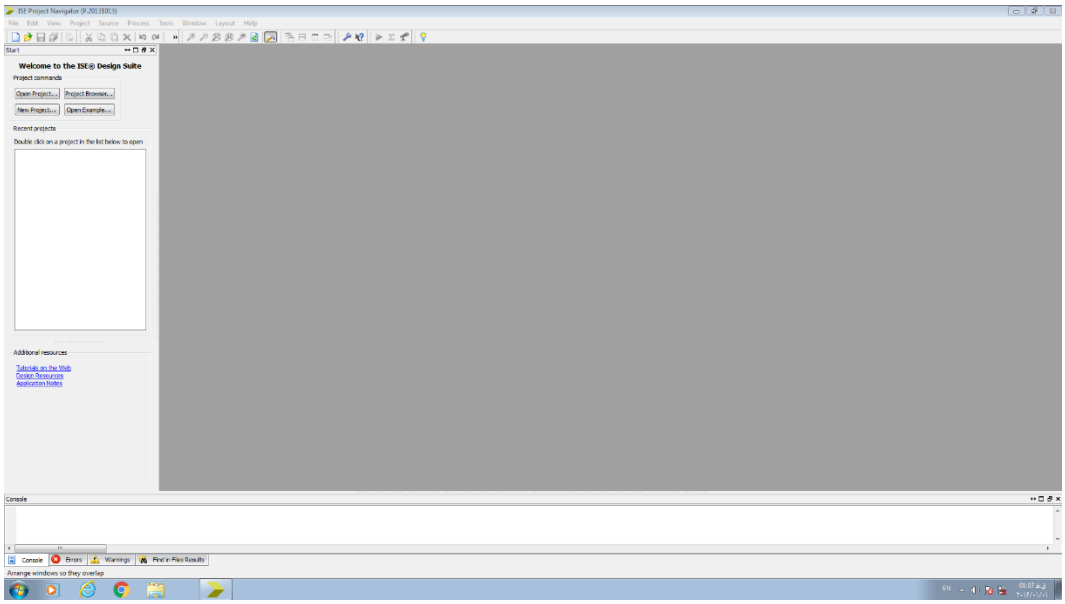
در این جلسه ابتدا اقداماتِ انجام‌شده در جلسه‌ی گذشته به‌طورِ کلی مرور و سپس به طراحی و پیاده‌سازیِ تعدادی نیم‌جمع‌کننده و تمام‌جمع‌کننده پرداخته شد.

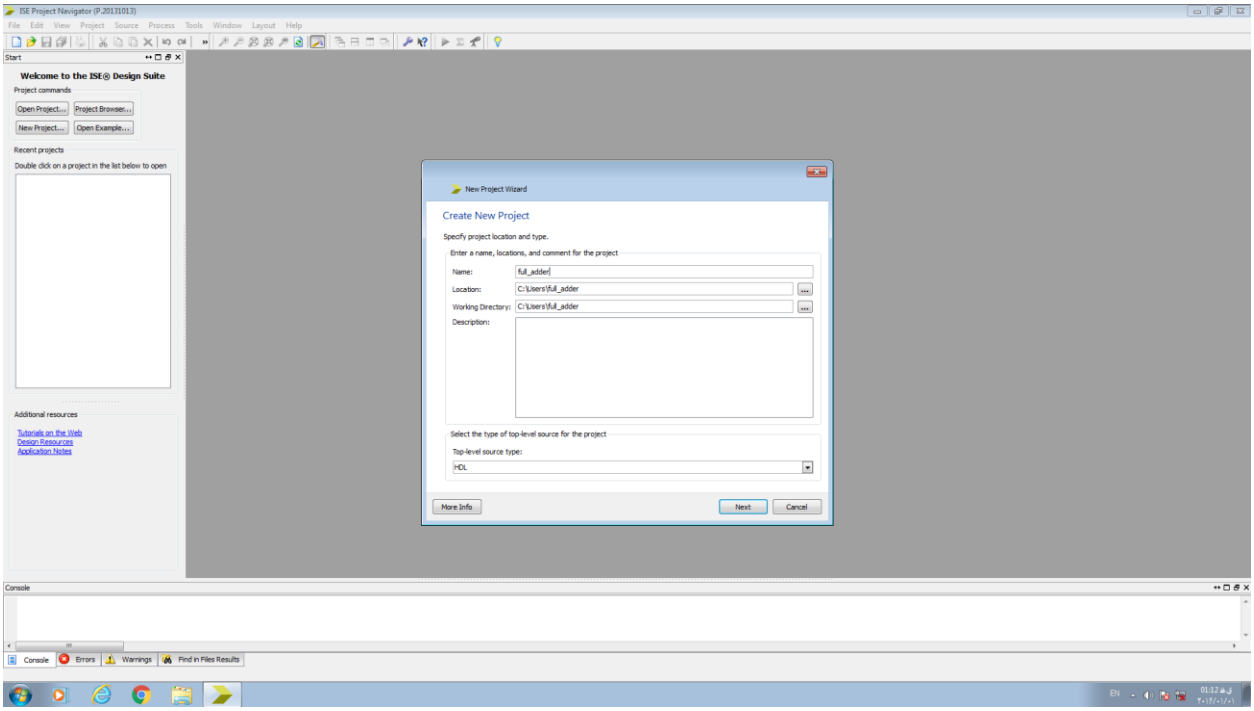
برای طراحِ این مدارها در محیطِ نرم‌افزارِ ISE از عملگر Bitwise و ساختارِ سلسله‌مراتبی نیز استفاده شد.

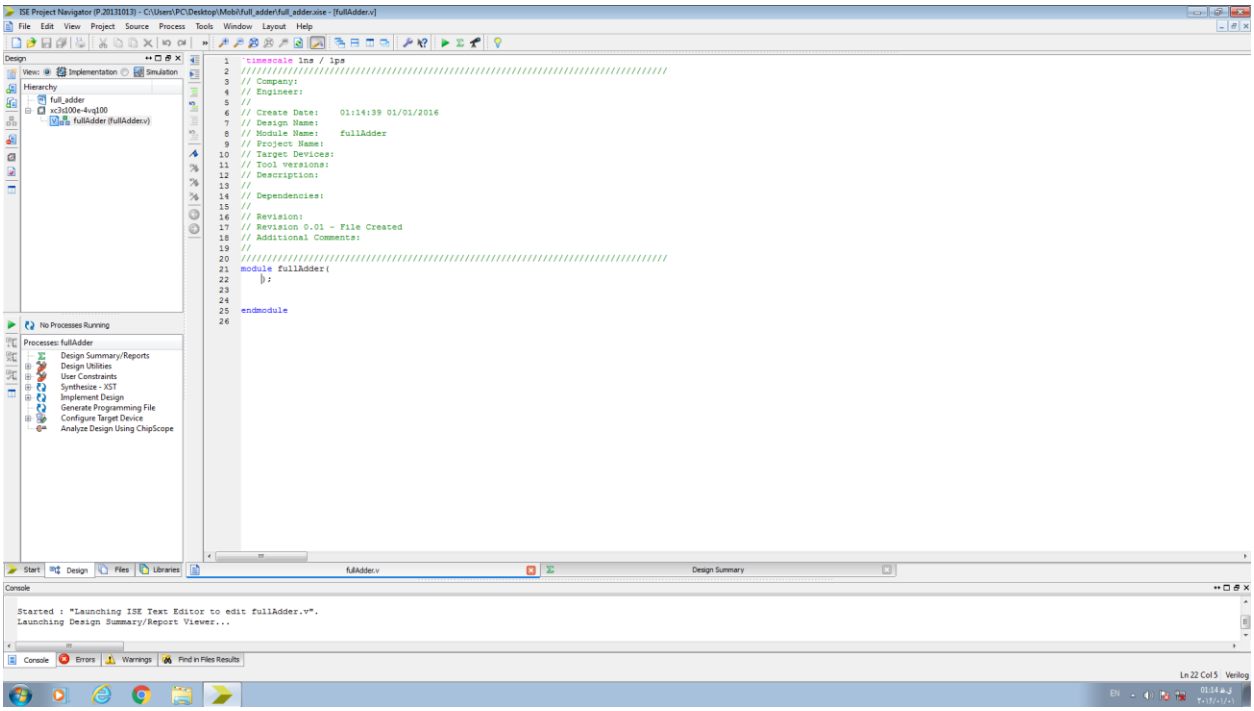
همچنین، مطابقِ طراحی‌های جلسه‌ی گذشته، برای بررسیِ عملکرد این برنامه‌ها هم تعدادی فایلِ تست نوشته و اجرا شدند. در طول این گزارش‌کار اقداماتِ انجام‌شده در طول جلسه را قدم به قدم و با ذکر جزئیات مرور خواهیم کرد.

همچون گذشته این بار هم پس از اجرای برنامه‌ی ISE، ابتدا پروژه‌ی جدیدی تعریف کرده و سپس یک فایلِ Source به آن اضافه می‌کنیم.

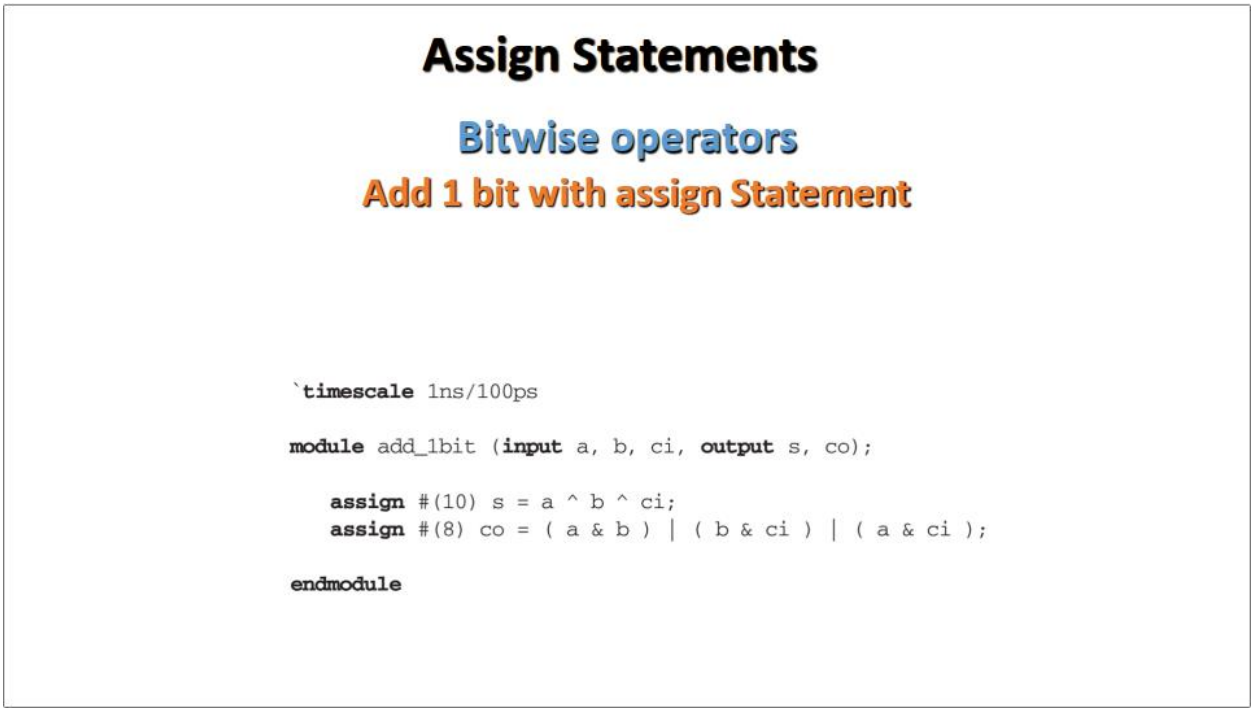
مراحلِ انجامِ این کار در تصاویرِ زیر آورده شده‌اند.



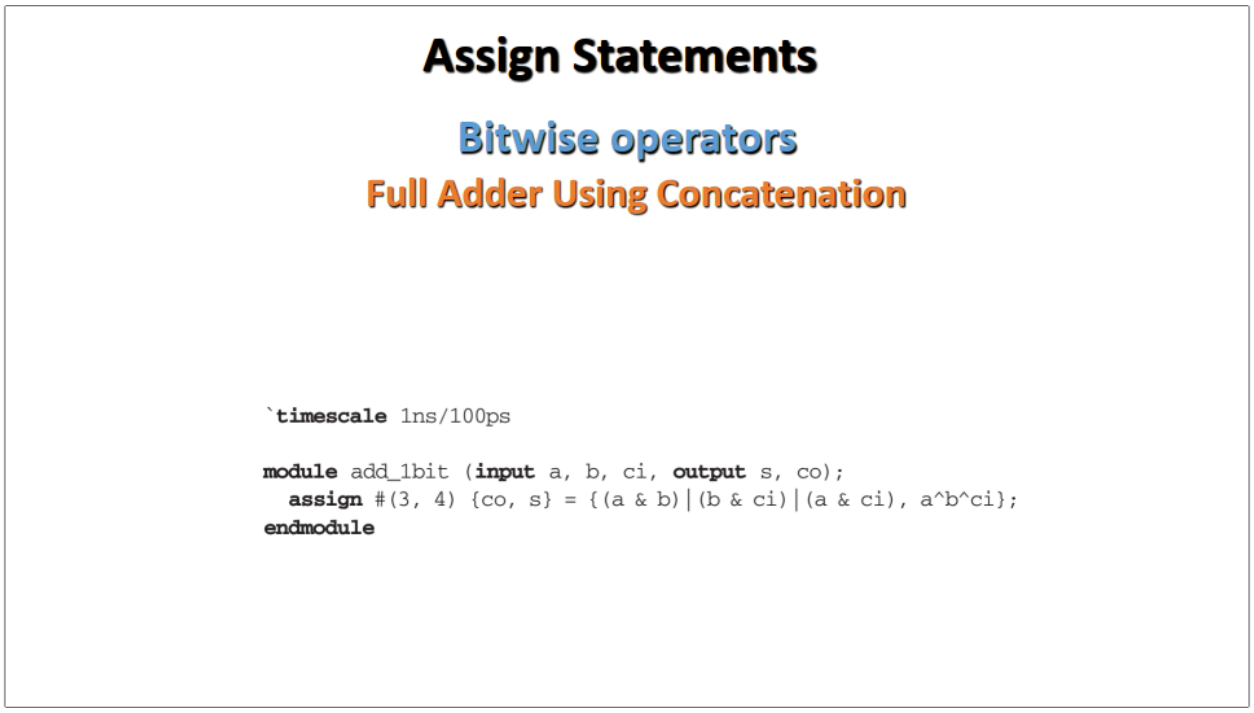

ISE Project Navigator (P.20131013)
Welcome to the ISE® Design Suite
Project commands
Open Project...
Project Browser...
New Project...
Open Example...
Recent projects
Double click on a project in the list below to open
Additional resources
Tutorials on the Web
Design Resources
Application Notes
New Project Wizard
Create New Project
Specify project location and type.
Enter a name, locations, and comment for the project
Name: full_adder
Location: C:\Users\full_adder
Working Directory: C:\Users\full_adder
Description:
Select the type of top-level source for the project
Top-level source type:
HDL
More Info
Next
Cancel
Console
Errors
Warnings
Find in Files Results


ISE Project Navigator (P.20131013) - C:\Users\PC\Desktop\Mobi\full_adder\full_adder.xise - [fullAdder.v]
Design
Hierarchy
full_adder
xc3s100e-4vq100
fullAdder (fullAdder.v)
No Processes Running
Processes: fullAdder
Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope
`timescale 1ns / 1ps
//
// Company:
// Engineer:
//
// Create Date: 01:14:39 01/01/2016
// Design Name:
// Module Name: fullAdder
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
module fullAdder(
);
endmodule
Start
Design
Files
Libraries
fullAdder.v
Design Summary
Console
Started : "Launching ISE Text Editor to edit fullAdder.v".
Launching Design Summary/Report Viewer...
Console
Errors
Warnings
Find in Files Results
Ln 22 Col 5
Verilog

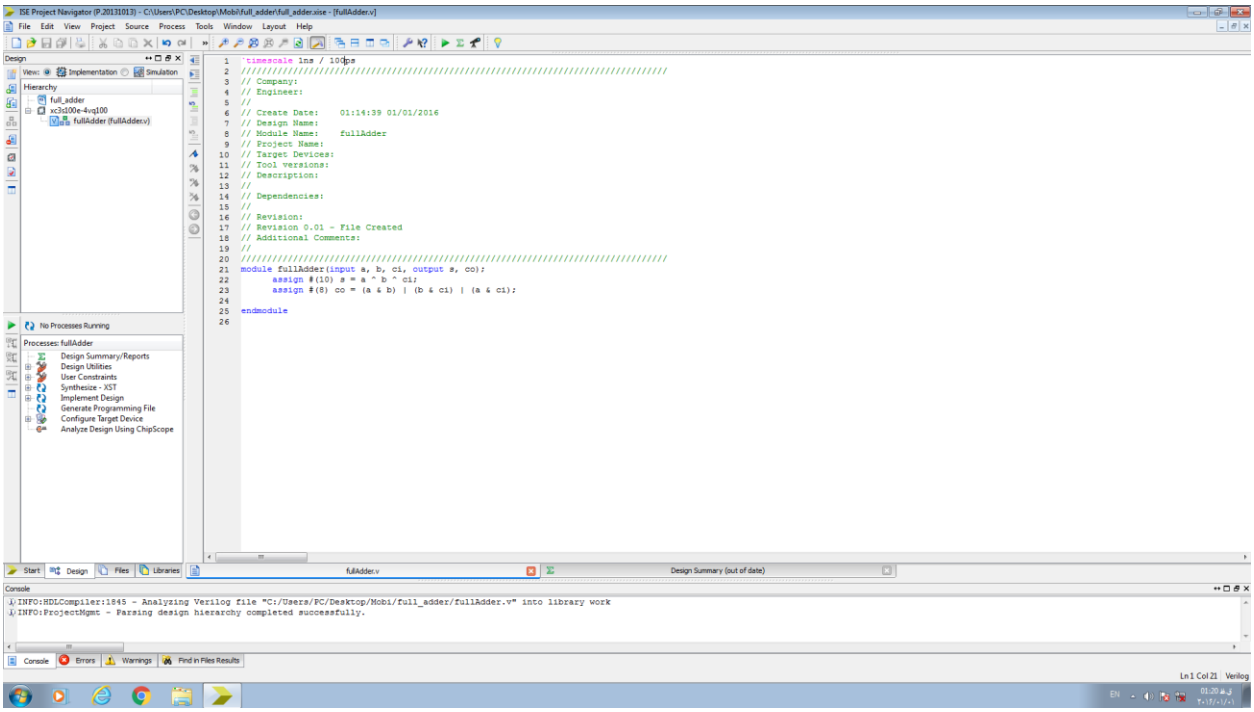
اولین قطعه کدی که برای طراحی یک adder نیاز به اجرای آن داریم در تصویر زیر آورده شده:



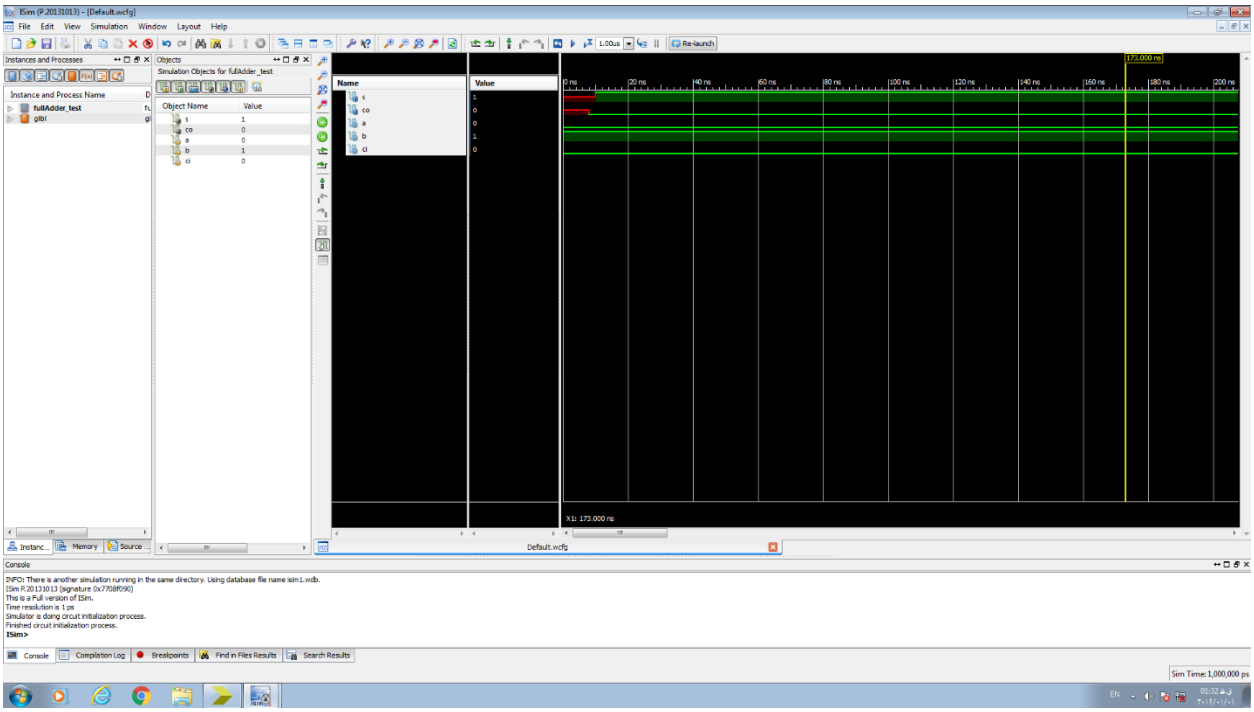
قطعه کد زیر نیز، شکل دیگری از پیاده‌سازیِ این مفهوم است:

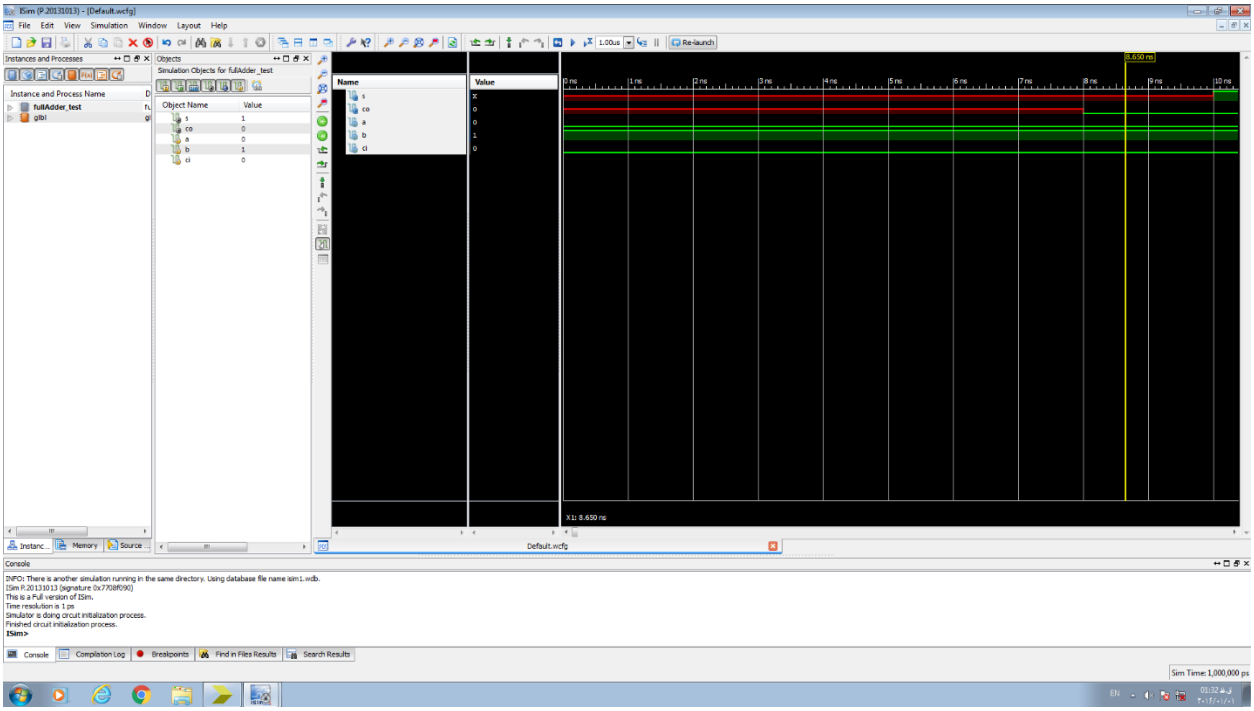


حال لازم است یکی از کدهای بالا را انتخاب کرده و آن را در Source ایجادشده برای پروژه‌ی جدید بنویسیم:



سپس با کامپایل کردن برنامه و اجرای قطعه‌ی تست به نتایجِ زیر دست می‌یابیم:





در مرحله‌ی بعد قصد داریم همان ساختارِ جمع‌کننده‌ی سابق را برای 4 بیت پیاده‌سازی کنیم.

کدها و مفاهیمِ مورد بررسی در شکلِ زیر آورده شده‌اند:

Hierarchical Structures

Vector declarations

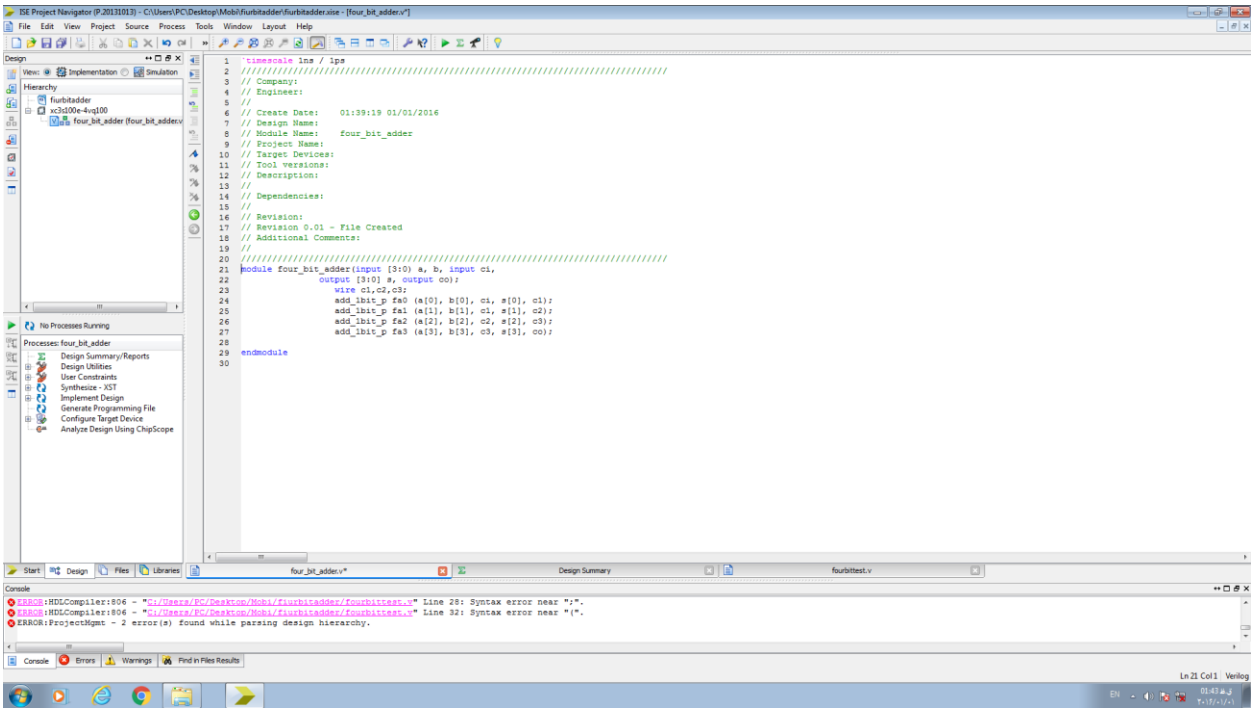
A 4-bit Adder by Wiring Four Full Adders

```
`timescale 1ns/100ps

module add_4bit (input [3:0] a, b, input ci,
                 output [3:0] s, output co);
   wire c1, c2, c3;

   add_1bit_p fa0 (a[0], b[0], ci, s[0], c1);
   add_1bit_p fa1 (a[1], b[1], c1, s[1], c2);
   add_1bit_p fa2 (a[2], b[2], c2, s[2], c3);
   add_1bit_p fa3 (a[3], b[3], c3, s[3], co);

endmodule
```

OR

```
add_1bit_p fa1 (.a(a[1]),.b(b[1],.ci(c1),.sum(s[1]),.co(c2));
```

شکلِ بعد، پیاده‌سازیِ این کد درونِ نرم‌افزارِ ISE را نشان می‌دهد:



پس از نوشتنِ این قطعه‌کد، لازم است بار دیگر برای ارزیابیِ عملکرد آن یک فایلِ تستِ مجزا بنویسیم.

فایلی که کدهای آن در شکلِ زیر آورده شده‌اند:

Hierarchical Structures

Vector declarations

A 4-bit Adder by Wiring Four Full Adders

```
module test_add4;
    // Inputs
    reg [3:0] a;
    reg [3:0] b;
    reg ci;
    // Outputs
    wire [3:0] s;
    wire co;
    // Instantiate the Unit Under Test (UUT)
    add_4bit uut (
        .a(a),
        .b(b),
        .ci(ci),
        .s(s),
        .co(co));
    initial begin
        // Initialize Inputs
        a = 0;
        b = 0;
        ci = 0;
        #50 a = 4'b1010; b = 4'b1001;
        #50 a = 4'b0010; b = 4'b1001;
        #50 a = 4'b1011; b = 4'b1101; ci = 1;
        // Wait 100 ns for global reset to finish
        #100;
        // Add stimulus here
    end
endmodule
```

در نهایت، فایلِ تستِ جدید را ساخته و عملکرد برنامه را در محیط Simulation یا شبیه‌سازی، ارزیابی می‌کنیم:

```
// Engineer:
//
// Create Date:   01:57:01 01/01/2016
// Design Name:   four_bit_adder
// Module Name:   C:/Users/PC/Desktop/Mobi/fiurbitadder/test44.v
// Project Name:  fiurbitadder
// Target Device:
// Tool versions:
// Description:
//
// Verilog Test Fixture created by ISE for module: four_bit_adder
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////

module test44;
      reg [3:0] a;
      reg [3:0] b;
      reg ci;
      wire [3:0] s;
      wire co;
      add_4bit(
      .a(a),
      .b(b),
      .ci(ci),
      .s(s),
      .co(co));
      initial begin
      a = 0;
      b = 0;
      ci = 0;
      #50 a = 4'b1010; b = 4'b1001;
      #50 a = 4'b0010; b = 4'b1001;
      #50 a = 4'b1011; b = 4'b1101; ci=1;
      #100


   end

endmodule
```

پایان.