

دانشگاه شمال



موضوع:

**آموزش استفاده از نرم افزار**

# **Xilinx ISE Project Navigator**

(طراحی و ساخت مدارات FPGA)



تهیه کننده: **آرمین کام فیروزی**

رشته تحصیلی: **کارشناسی ناپيوسته سخت افزار کامپيوتر**

استاد: **مهندس رضا فانی**

اسفند ۸۵

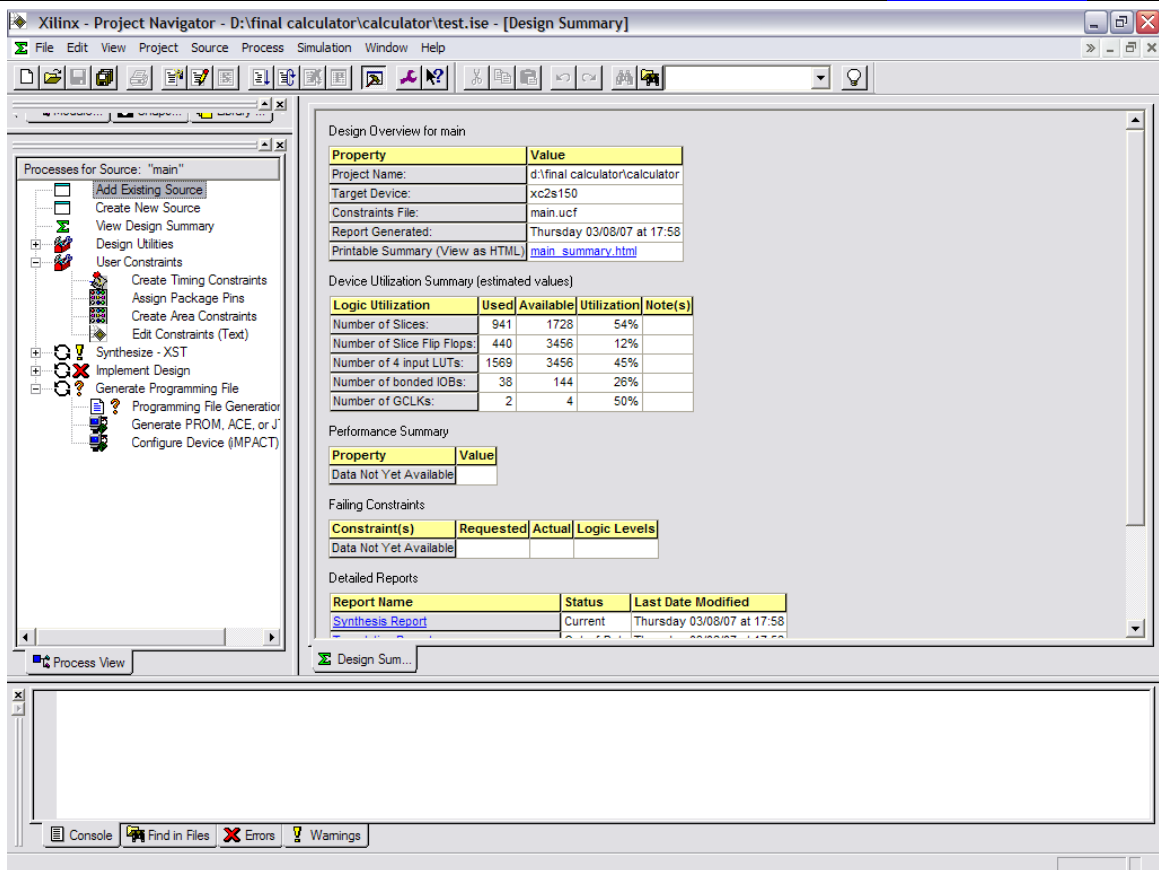
برای طراحی مدارات دیجیتال توسط FPGA یا CPLD از نرم افزار فوق العاده قدرتمند Xilinx ISE استفاده میکنیم که در حال حاضر تا این تاریخ نسخه ۸ آن به بازار آمده که ما از نسخه 7.1 محصول سال ۲۰۰۵ این شرکت استفاده میکنیم.

نام دیگر این نرم افزار Project Navigator است.



نصب این نرم افزار به راحتی امکان پذیر است و کافی است که یکی یکی CD های آن را داخل دستگاه گذاشته و نصب کنید. کلیه مراحل نصب به راحتی قابل انجام است و نکته خاصی در نصب آن وجود ندارد. فقط پس از نصب بهتر است به منظور اجرای بهتر نرم افزار Service Pack ویژه ای که برای این نسخه آماده شده نیز نصب گردد که در واقع نسخه جاری را ارتقاء می دهد.

پس از نصب از قسمت Start در قسمت Xilinx ISE نرم افزار را اجرا می نماییم. البته زیر برنامه های دیگری نیز در این قسمت می باشد که ما برنامه اصلی یعنی Project navigator را استفاده می کنیم. محیط نرم افزار چیزی شبیه به شکل زیر است.

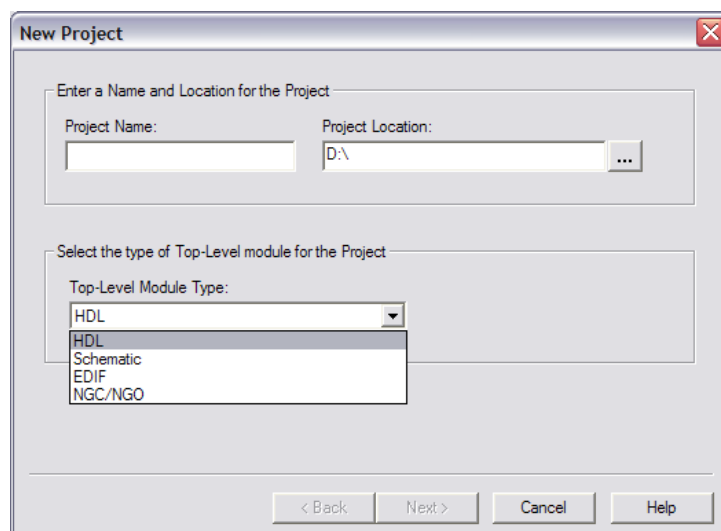


شکل ۱ (محیط نرم افزار)

همیشه در اول کار از منوی File گزینه Close Project را میزنیم تا پروژه ای که از قبل باز بوده بسته شود.

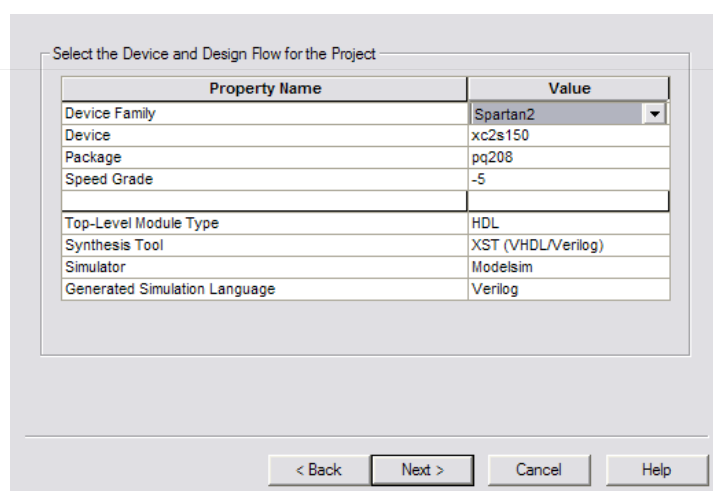
## شروع یک پروژه:

برای شروع از گزینه فایل گزینه New project را کلیک میکنیم. در صفحه باز شده نام پروژه، مسیر کاری ذخیره پروژه و نوع پروژه را انتخاب می‌نماییم.

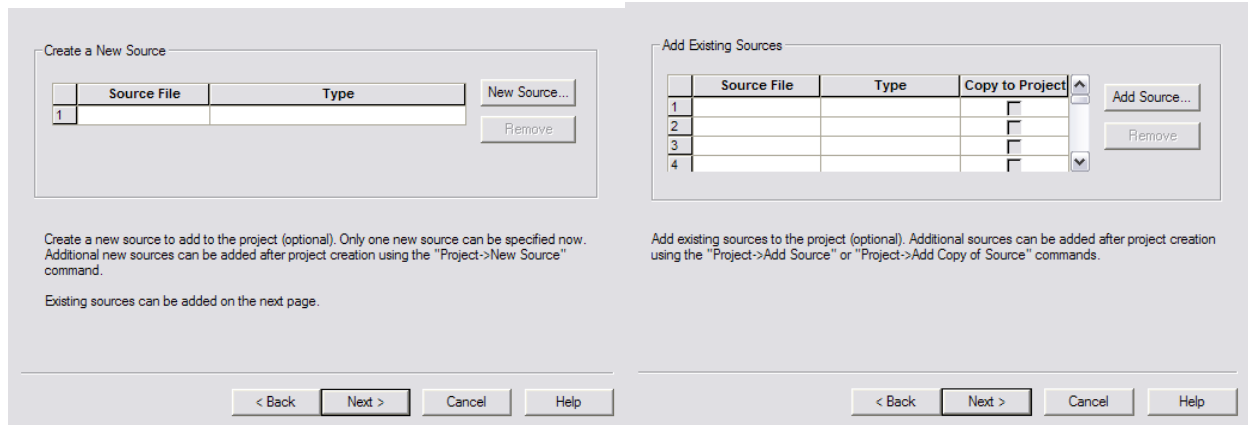


در صفحه بعد نوع IC که قرار است با آن کار کنیم و مشخصات فنی آن را تعیین می‌کنیم. مشخصات مذکور بر روی بردی که از آن استفاده می‌کنیم بر روی تراشه FPGA یا CPLD حک شده است. این مشخصات شامل موارد زیر می‌باشد:

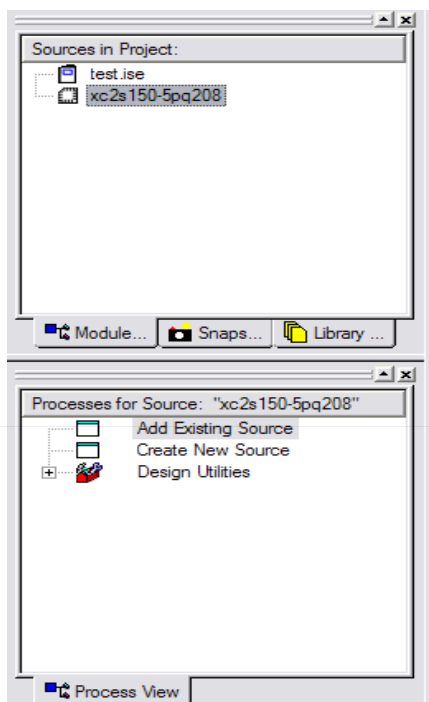
- خانواده IC مثل سری های Spartan یا Virtex و ...
- نوع وسیله و مدل آن که بر اساس تعداد گیت روی تراشه تعیین شده. مثلا XC2S150
- نوع بسته بندی و Package تراشه که باز در روی تراشه نوشته شده.
- سرعت و تأخیر آن
- Module Type: که نوع برنامه‌ریزی و کار ما با IC را مشخص میکند
- ابزار سنتز که در اینجا Xilinx Synthesis Tool یا همان XST است.
- نرم افزار شبیه ساز و Simulation که هم از خود ISE می‌توان استفاده کرد و هم از Modelsim و ...
- زبان برنامه نویسی شامل VHDL, Verilog و ...



در دو قسمت بعد می‌توانیم یک سری فایل‌های از پیش نوشته شده را اضافه کنیم. مانند Includeها در زبان C که برای شروع نیازی به این کار نیست.

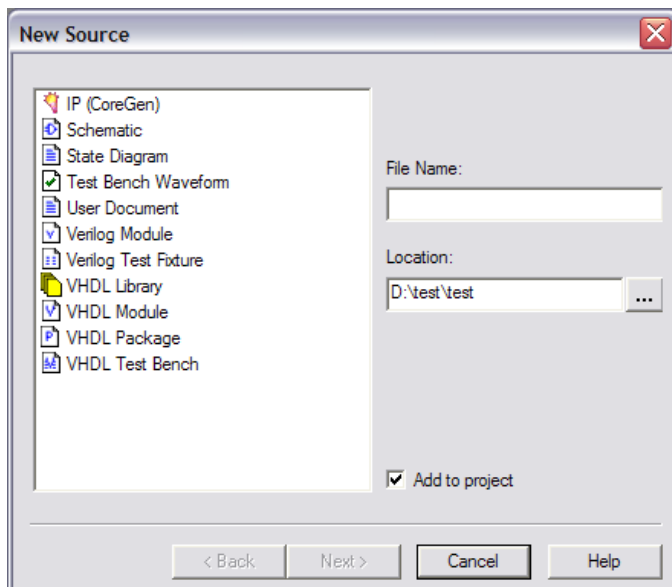


و بدین ترتیب اطلاعات اولیه و مشخصات کلی تراشه و نوع پروژه را تعریف می‌کنیم و وارد محیط نرم افزار می‌شویم.



در سمت چپ صفحه نام پروژه و محیط کاری ظاهر شده که فعلاً خالی است. بر روی مشخصات پروژه کلیک راست کرده و گزینه **New Source** را انتخاب می‌کنیم. می‌خواهیم شروع به طراحی و نوشتن برنامه کنیم. یک **Source** می‌تواند به صورتهای زیر باشد.

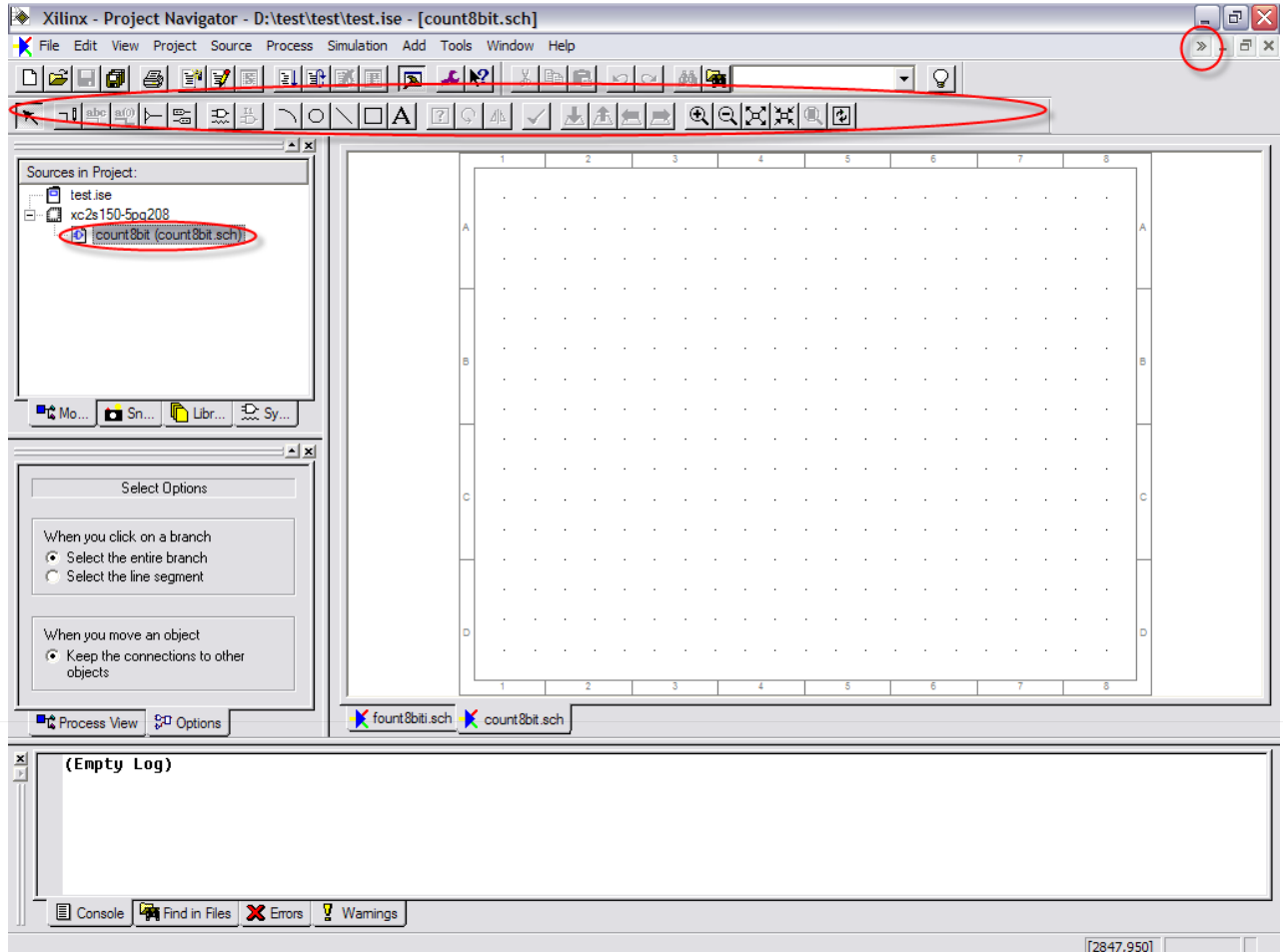
- شماتیک: که مدار را میکشیم و نرم افزار آن را کد میکند و به گیت‌های FPGA تبدیل میکند.
- **Test bench**: که برای تهیه یک فایل تست پروژه و شبیه سازی استفاده می‌شود.



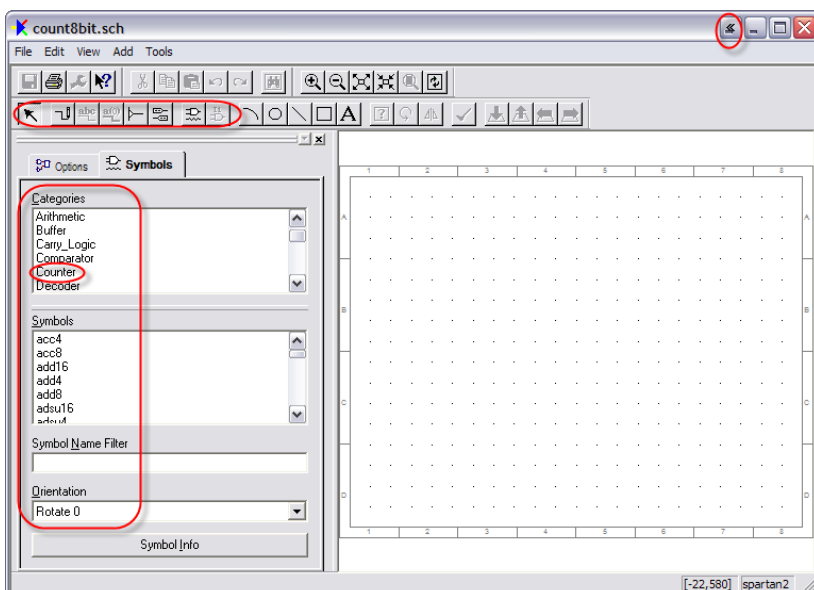
- **Verilog Module**: برای نوشتن برنامه به زبان Verilog
- **VHDL Module**: برای نوشتن برنامه به زبان VHDL
- **VHDL Library**: جهت تعریف کتابخانه VHDL برای

در ابتدا به معرفی **Source** و طراحی شماتیک می‌پردازیم.

اسم فایل را نیز در محل مربوطه مشخص می کنیم. مثلا شمارنده ۸ بیتی (Counter8biti) یک محیط طراحی شماتیک باز میشود که برای بهتر دیدن این محیط میتوانیم روی علامت >> در بالا سمت راست این صفحه را جدا کنیم و راحت تر کار کنیم. نام فایل نیز زیر قسمت مشخصات پروژه ظاهر میشود:



در این محیط ابزارهای طراحی شماتیک موجود است و همچنین انواع نمادها و ابزارهای مورد نیاز برای طراحی



شماتیک هر نوع مدار.

(در این شکل محیط طراحی را با کلیک

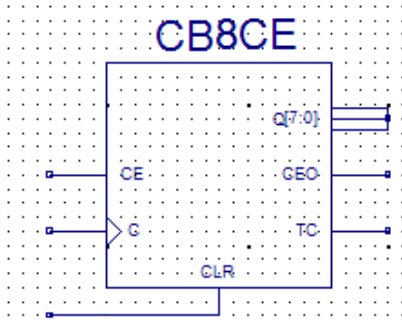
روی >> جدا کرده ایم).

در بخش Symbols بسیاری از نمادها و

گیت‌های از پیش تعریف شده وجود دارد

که به راحتی می توان آنها را انتخاب و بر

روی نقشه قرار داد و طراحی را انجام



داد. مثل بافر، انواع گیتها و ...

مثلا یک شمارنده را از بخش شمارنده ها انتخاب و روی مدار قرار دهیم.

برای مشاهده Datasheet مربوط به هر کدام از این اشیا می توانیم روی آن

کلیک راست کرده و گزینه Object Properties را انتخاب کنیم و در

صفحه جدید Symbol Info را کلیک کنیم. سپس نرم افزار یک فایل PDF که شامل کلیه اطلاعات فنی می-

باشد تولید میکند که میتوانیم آن را ذخیره و یا چاپ کنیم. ابزارهای متداول ترسیم در نوار ابزار قابل دسترس

است.



۱- ابزار انتخاب

۲- رسم Wire (سیم)

۳- برچسب گذاری (Label)

۴- تغییر نام گذرگاه (Bus)

۵- Bus Tap که میتوان از یک گذرگاه چند بیتی، یک یا چند بیت آن را جدا کرد

۶- قرار دادن I/O روی مدار

۷- اضافه کردن یک سمبول به مدار

۸- نمایش یک نام دلخواه (با لیبل فرق دارد)

۹- تست مدار شماتیک و نمایش ایرادات آن

۱۰- Push برای وارد شدن داخل یک object و مشاهده جزئیات آن و یا برنامه داخلی آن

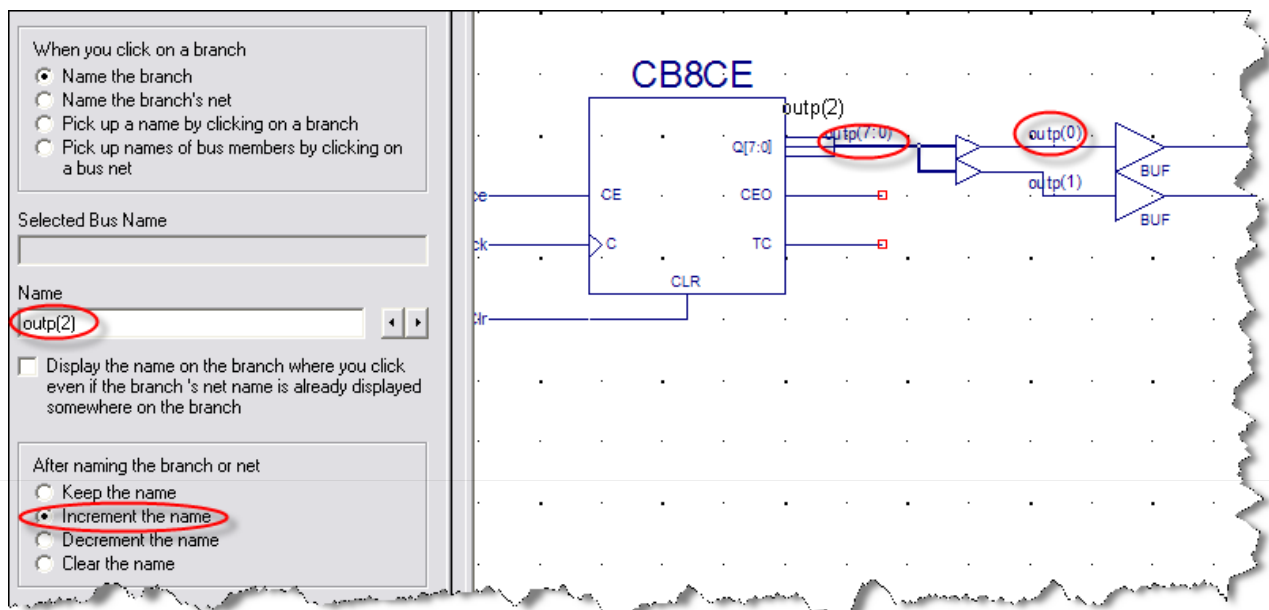
۱۱- برگشت به حالت قبل و بیرونی تر (pop)

۱۲- Forward و Back

توسط ابزار Wire به تمام ورودیها و خروجی ها سیم وصل میکنیم.

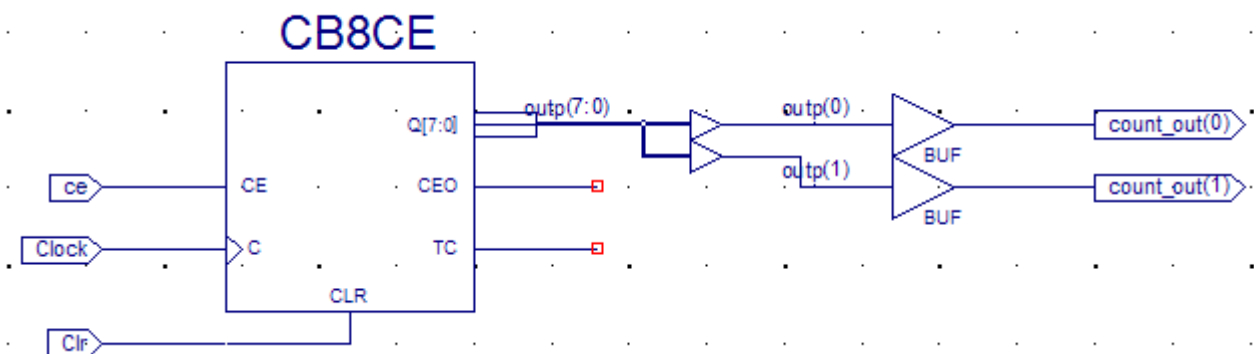
با ابزار Bus tap مثلا از ۸ بیت خروجی شمارنده، دو بیت آن را جدا میکنیم.

با ابزار لیبل برای ورودی و خروجیها لیبل میگذاریم و برای آن دو بیت نیز لیبل خروجی را به همراه شماره بیت در داخل پرانتز مینویسیم. مطابق شکل. مهم این است که اسم لیبلها یکی باشد. حتی اگر سیمها به هم وصل نباشد اما دو لیبل هم نام وجود داشته باشد، یعنی این دو لیبل به هم متصلند. از این کار برای زمانی که مدار پیچیده باشد و نخواهیم همه اشیا در یک منطقه رسم شود و شکل شلوغ شود استفاده میکنیم. بهتر است برای خروجیها بافر بگذاریم. در قسمت سمبول بافر را میتوانید پیدا کنید. اسامی بعد از بافر دلخواه است.



با انتخاب Increment the name در صفحه لیبل، شماره ها به صورت خودکار زیاد میشود و ۰،۱،۲ و ... را خود نرم افزار اضافه میکند.

برای ارتباط با محیط خارج، نیاز است که یک سری ورودی و خروجی به مدار اضافه کنیم. با استفاده از ابزار I/O این کار را انجام میدهیم.





برای بررسی صحت طراحی شماتیک روی علامت ✓ در نوار ابزار کلیک میکنیم.

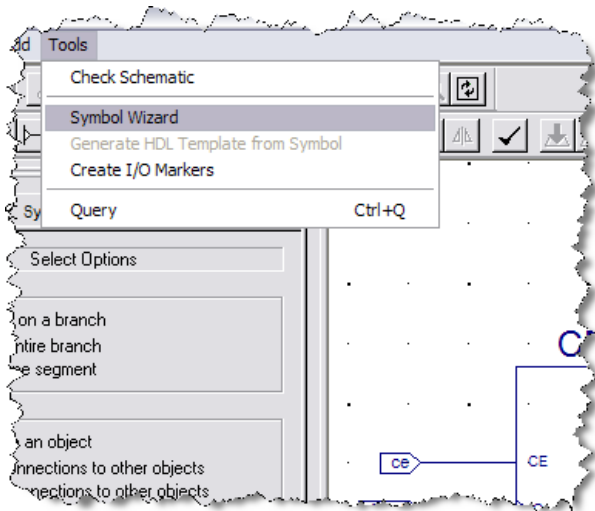
برای ذخیره مدار Save را میزنیم.

برای راحتی کار در مدارات بزرگ، میتوانیم یک شماتیک را به صورت یک سیمبول در آوریم. مثلا کل مدار

فوق را سمبل کنیم.

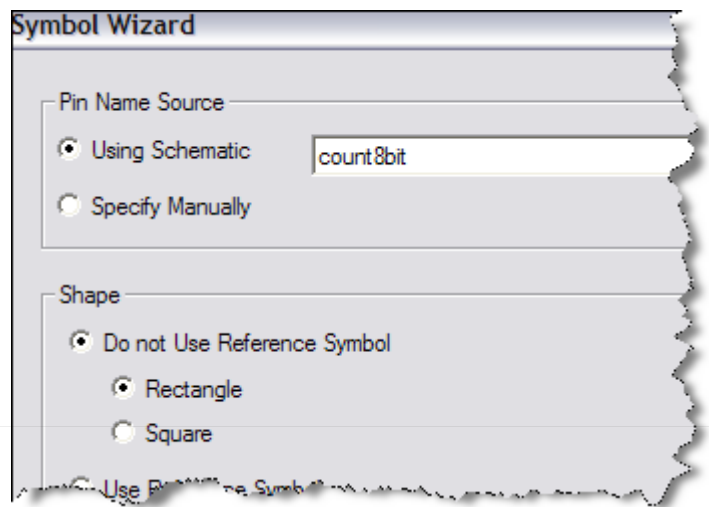
برای این کار از Tool گزینه Symbol Wizard را

میزنیم و اطلاعات خواسته شده را وارد می کنیم.



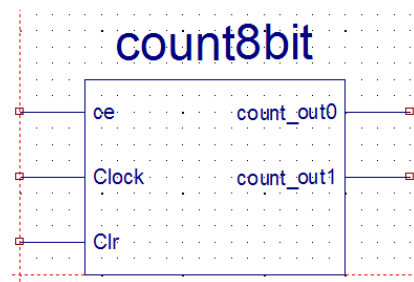
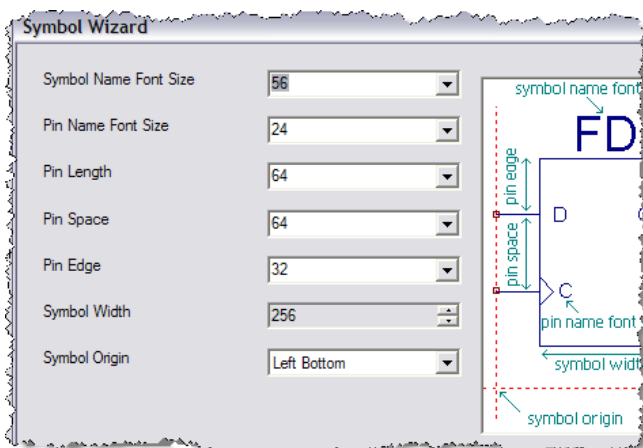
اسم سمبل، شکل آن و ترتیب پایه های ورودی و خروجی آن.

سایر مشخصات مانند اندازه شکل و غیره را نیز وارد میکنیم.



و بدین ترتیب از مدار یک سیمبول میسازیم که این سیمبل

نیز در بخش سیمبولها اضافه شده است.



تا اینجا برنامه ریزی به روش شماتیک را یاد گرفتیم.

برای برنامه نویسی به روش کد نویسی دوباره روی نام پروژه کلیک راست کرده و از گزینه New Source

گزینه Verilog Module و یا VHDL module را انتخاب میکنیم. در صفحه ظاهر شده نرم افزار به

صورت خودکار ورودی و خروجی ها و نوع آنها را می پرسد و کد مربوط به تعریف ورودی و خروجی ها را خودش تولید می کند.

در قطعه کد تولید شده بهتر است خط اول را که مربوط به تعریف زمانبندی است را پاک کنیم. چون برای شبیه سازی کاربرد دارد و در مدار عملی دچار مشکل می شود.

Module Name test2

Port Name	Direction
a	input
b	input
c	input
sum	output
out	output
	input
	input

```

1  timescale 1ns / 1ps
2  //////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:      11:16:03 03/10/07
7  // Design Name:
8  // Module Name:     test2
9  // Project Name:
10 // Target Device:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////
21 module test2(a, b, c, sum, out);
22     input a;
23     input b;
24     input c;
25     output sum;

```

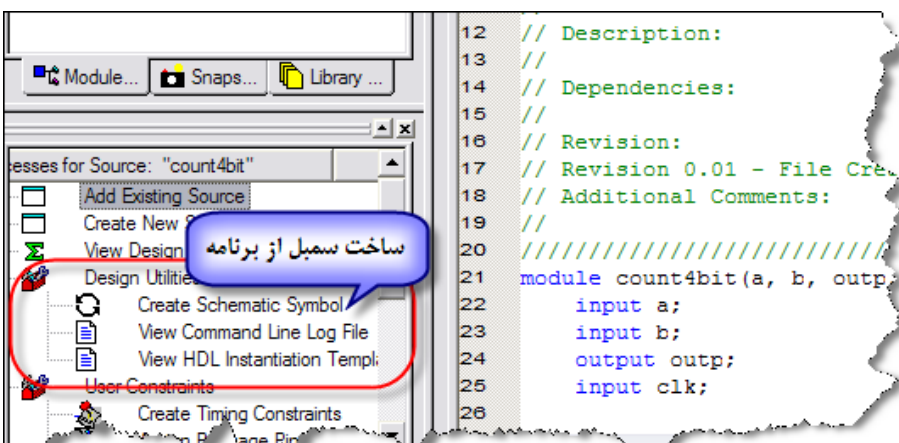
سپس با توجه به نوع زبان برنامه نویسی، کدهای مربوطه را برای عملیات خاص می نویسیم که در اینجا به آموزش زبان Verilog نمی پردازیم.

از ابزار Language Template در نوار ابزار می توانیم قطعه کدهای از پیش نوشته شده ای را بیابیم و یا با نوع برنامه نویسی و

دستورات آن آشنا شویم.



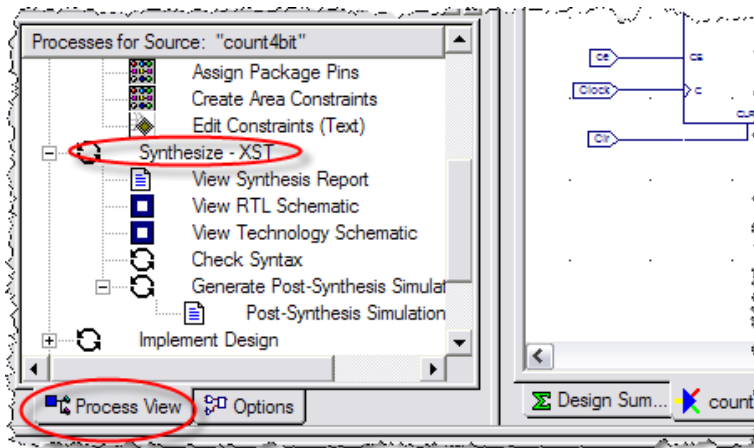
برنامه نوشته شده را می توان به صورت یک بلوک در آورد. برای این کار در قسمت سمت چپ گزینه Design Utilities و سپس Create schematic Symbol را میزنیم و بلوک مورد نظر را می سازیم.



این بلوک در بخش Symbols اضافه می شود.

## سنتز کردن:

برای این کار روی نام برنامه کلیک میکنیم و از قسمت سمت چپ گزینه Process View روی Synthesize XST دوبار کلیک میکنیم.



سایر گزینه هایی که در این بخش موجود است به این شرح می باشد.

View RTL Schematic: نمای RTL

مداری که طراحی و برنامه نویسی شده و عملا ساخته شده نمایش داده می شود و

تکنولوژی پیاده سازی روی مدار را نشان

میدهد..

View Technology Schematic: نقشه مدار را به صورت ریزترین جزئیات یعنی گیت های AND و OR

که در مدار ساخته شده نمایش می دهد و جالب است وقتی مداری را طراحی کردید این بخش را حتما ببینید تا محتویات FPGA و اتفاقاتی که واقعا در مدار می افتد را ببینید در این قسمت یک سری بافر ها هم ساخته اضافه میشود و اتفاقات سخت افزاری واقعی است که در مدار می افتد.

## شبیه سازی کردن:

شبیه سازی به منظور تست مدار قبل از ساخت و مشاهده خروجی ها و عکس العمل مدار به ازای ورودی های

مختلف صورت می گیرد.

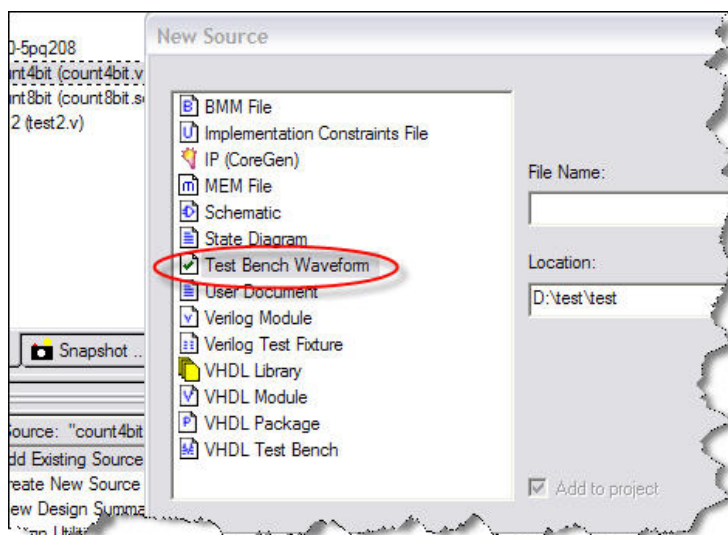
برای این کار روی نام مدار یا برنامه نوشته

شده از سمت چپ کلیک راست کرده و

New Source را انتخاب می کنیم و در

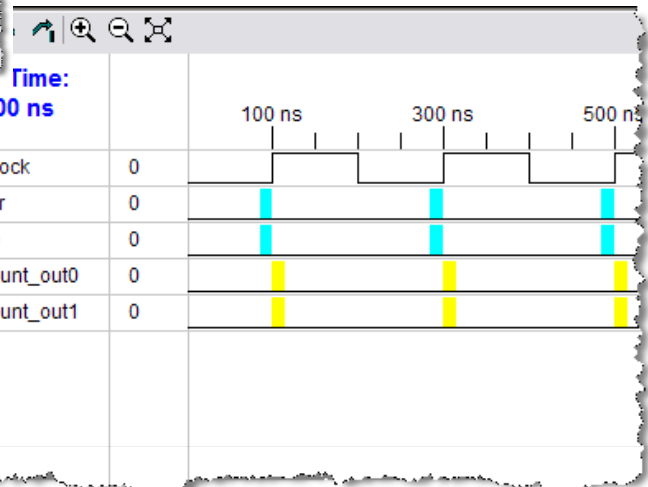
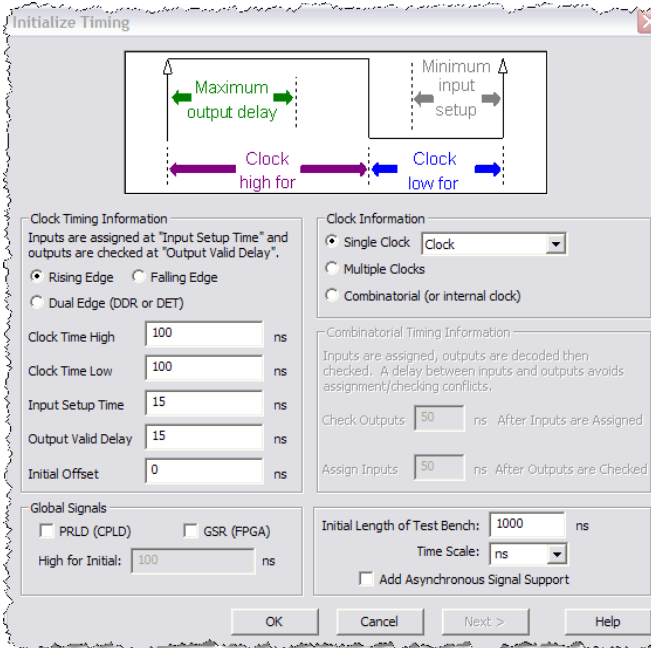
صفحه جدید Test bench waveform را

می زنیم و یک نام به آن اختصاص می دهیم.



در صفحه جدید اطلاعات مربوط به زمانبندی ها و کلاک سیستم را وارد می کنیم.

در صفحه جدید تمام سیگنالهای ورودی و خروجی نمایش داده شده است که می توان مقادیر مختلفی را به ورودی داد و خروجی ها و جزئیات مربوط به آن را مشاهده کرد.



با کلیک ماوس روی هر کدام از سیگنالها مقدار آن را می توان به صفر و یا یک تغییر داد. برای انجام عمل سیموله کردن

خود نرم افزار ISE امکاناتی را در اختیار ما می گذارد اما بهتر است که از نرم افزار قدرتمند Modelsim که پس از نصب با این نرم افزار لینک می شود استفاده کنیم. در قسمت Process view بخش مربوط به برنامه Modelsim قابل مشاهده است که انواع شبیه سازی از جمله Behavioral و یا شبیه سازی واقعی به صورت پیاده سازی شده روی FPGA را انجام می دهد. با دبل کلیک روی هر کدام از این گزینه ها برنامه Modelsim باز شده و عمل شبیه سازی صورت می گیرد.

صفحه Wave از برنامه Modelsim شکل موجهای شبیه سازی شده را نشان می دهد.

از ابزارهای Run.Zoom و غیره میتوان عملیات سیموله را به بهترین شکل ممکن بررسی کرد.

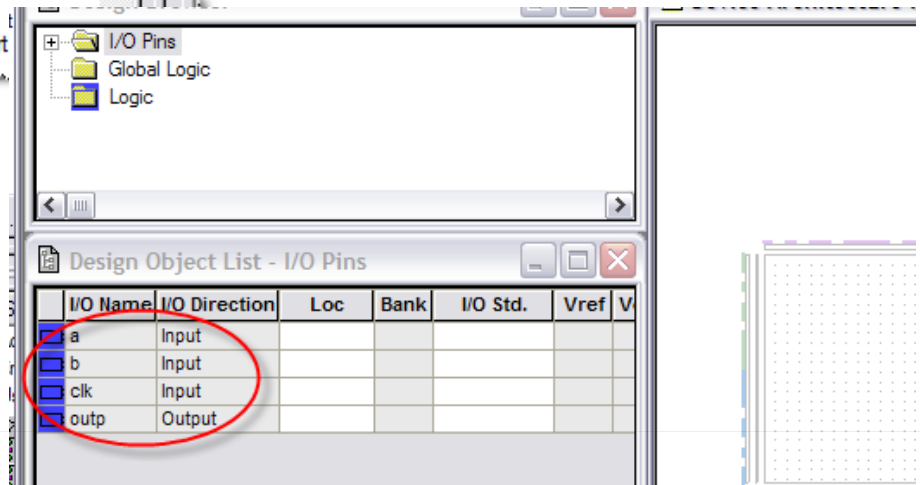
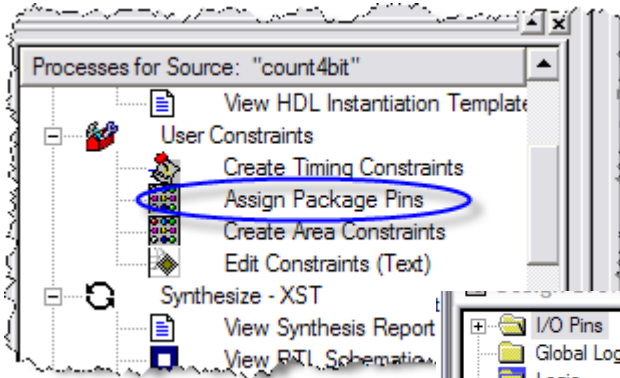
برنامه شماتیک و یا VHDL یا Verilog را هم میتوان به همین شکل سیموله کرد.

شبیه سازی Post translate رفتار واقعی روی مدار را نشان میدهد.

## Implement کردن:

در این قسمت میخواهیم برنامه را طوری آماده کنیم که بتوان روی IC آن را منتقل کرد. پینهای ورودی و خروجی را هم در این قسمت تعریف میکنیم.

در قسمت User Constraints با دوبار کلیک کردن روی Assign Package Pins تمامی پینهای ورودی و خروجی را میتوان تعریف کرد که بستگی به برد دارد.



با نوشتن حرف p جلوی هر شماره این کار را انجام میدهیم. مثلا جلوی clk مقدار p77 را وارد میکنیم. یعنی پین ۷۷ به کلاک اختصاص دارد.

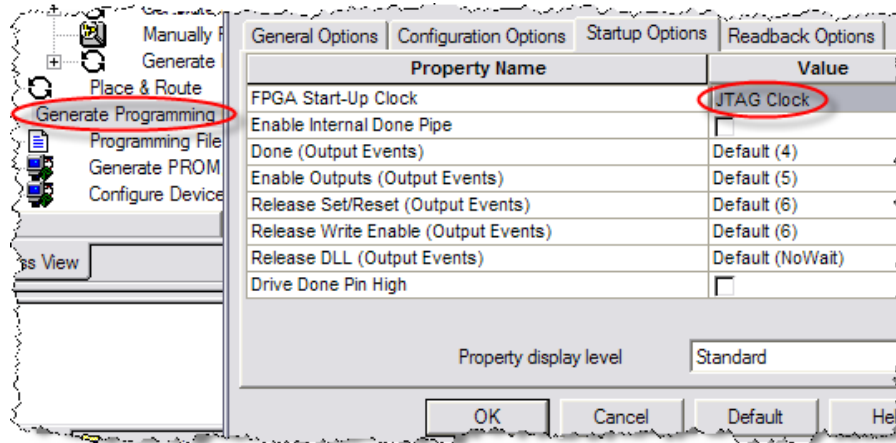
در این قسمت همچنین میتوان یک خروجی یا ورودی را بافر نیز کرد. برای این کار در قسمت Delay جلوی هر ورودی یا خروجی مقدار IBUF را انتخاب میکنیم.

سپس یک فایل با پسوند ucf ایجاد میشود. البته این فایل را به صورت دستی نیز میتوان نوشت اما این روش بهتر است. برای اینکه مطمئن شویم این IC با این پینها مشکلی ندارد گزینه Implement را میزنیم. اگر یک علامت ✓ سبز رنگ در کنار آن به وجود آمد یعنی مشکلی وجود ندارد.

## Program کردن IC و ریختن برنامه روی آن:

برای برنامه ریزی IC اگر از CPLD استفاده کرده باشیم چون برنامه ما درون آن ذخیره می شود و با قطع جریان برق از بین نمی رود، کفایت یکبار IC را پروگرام کنیم. اما IC های FPGA با قطع جریان برق برنامه درون آنها از بین می رود لذا از یک قطعه جانبی به نام PROM استفاده می کنیم که باید برنامه نوشته شده را روی

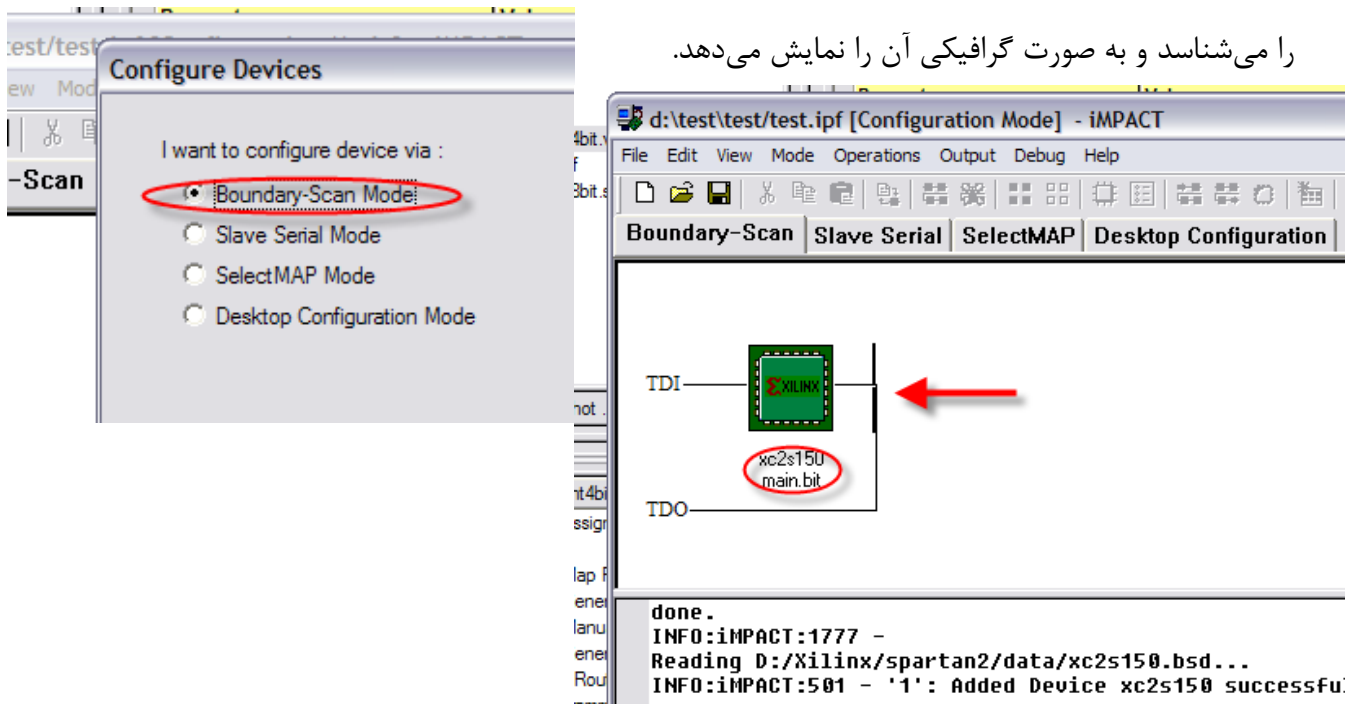
PROM بریزیم تا با قطع جریان برق برنامه از بین نرود. خود FPGA را هم می توان به تنهایی پروگرام کرد اما با قطع برق برنامه آن پاک شده و مجدداً باید پروگرام شود. برای برنامه ریزی خود FPGA روی گزینه Generate Programming file کلیک راست کرده و در قسمت Properties وارد Startup Options و در قسمت Startup Options می شویم و گزینه FPGA Startup Clock را روی گزینه JTAG Clock قرار می دهیم.



و سپس گزینه Configure Device IMPACT را دوبار کلیک می کنیم. قبل از کلیک باید دستگاه را به کامپیوتر متصل کرده باشیم. روش JTAG مربوط به اتصال کابل به دستگاه را می گویند. البته FPGAها بعضی پروگرامهای مخصوص هم دارند که ما در اینجا استفاده نمی کنیم.

در صفحه ای که باز می شود مدهای مختلف Programming را می توان انتخاب کرد که این مدها بستگی به تنظیمات IC دارد که از روی Datasheet آن می توان آنها را ست کرد. در اینجا از Boundary scan mode استفاده کرده ایم. در صفحه بعد Automatically connect و بعد Finish را میزنیم. بعد از مدتی نرم افزار IC

را می شناسد و به صورت گرافیکی آن را نمایش می دهد.



مثلا در اینجا IC xc2s150 شناخته شده است. فایلی که برای برنامه ریزی FPGA استفاده می شود پسوند .bit دارد که باید آن را به Assign IC کنیم. برای این کار روی شکل کلیک راست کرده و Assign new configuration file را انتخاب میکنیم و سپس فایل مورد نظر که در اینجا به نام main.bit است را Assign میکنیم.

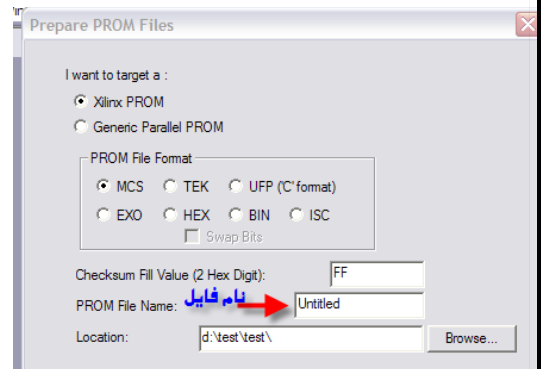
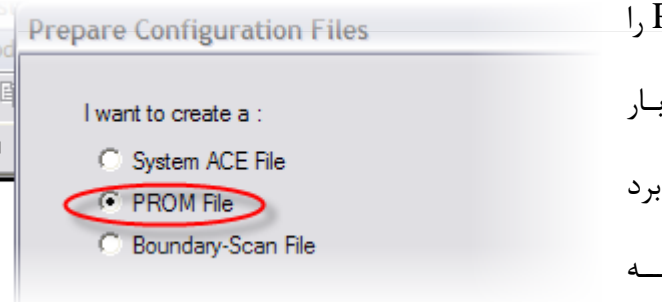
در اینجا نوبت به پروگرام کردن میرسد. روی IC کلیک راست کرده و Program را میزنیم. در FPGAها گزینه Verify نباید انتخاب شود. و سپس پروگرام کردن شروع می شود. اگر این عمل با موفقیت انجام شود پیغام موفقیت ظاهر شده و روی دستگاه نیز پین done روشن می شود.

### پروگرام کردن با استفاده از PROM:

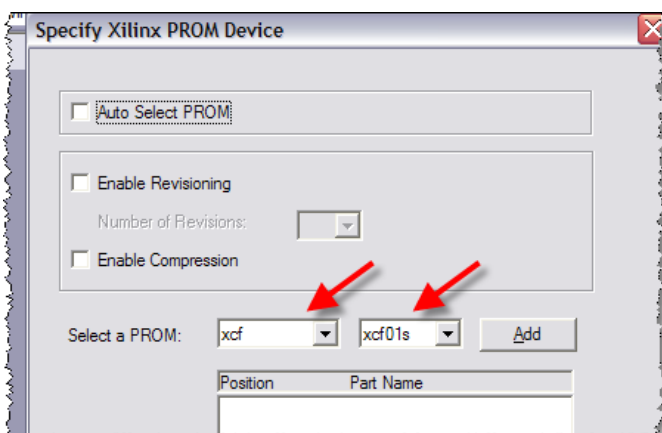
در صورتی که از PROM برای ذخیره برنامه و پروگرام IC استفاده کنیم باید روی گزینه Generate Programming File کلیک راست کرده و در قسمت Properties در بخش Startup options گزینه

CCLK را انتخاب می کنیم. حال باید فایل PROM را Generate کنیم. برای اینکار روی Generate Prom دوبار کلیک کرده و فایل PROM را بسازیم. قبل از آن باید روی برد PROM را تعبیه

کرده باشیم که با استفاده از Datasheet دستگاه این امر را انجام می -



دهیم. در صفحه جدید گزینه

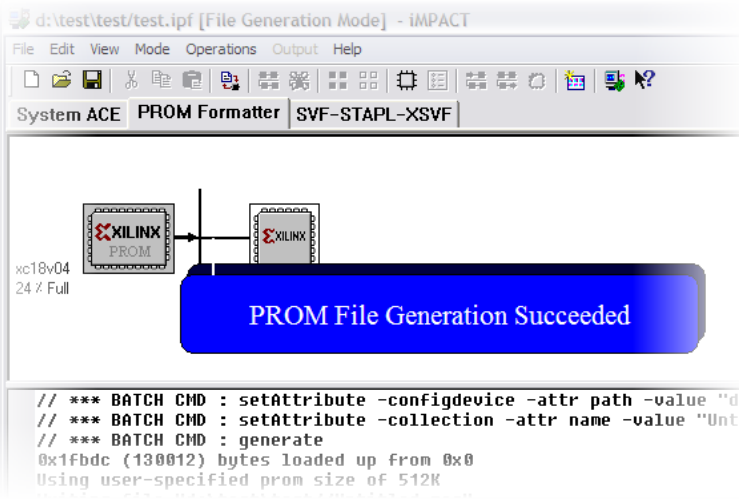


PROM را انتخاب کرده و به صفحه بعد می رویم. در صفحه جدید مشخصات PROM را مانند نام فایل آن را وارد می کنیم. معمولاً سایر قسمتها به صورت پیش فرض

می ماند. در صفحه بعد مشخصات PROM استفاده شده در مدار مانند مدل و سری آن را انتخاب می کنیم.

همچنین اگر از چند PROM در مدار استفاده شده باشد می توانیم چند تا را Add کنیم. سپس به صفحات بعد می رویم. با کلیک روی دکمه Add file فایلی را که می خواهیم در PROM ذخیره شود را انتخاب می کنیم که پسوند Bit دارد.

می توانیم چند فایل را هم بریزیم که خود نرم افزار از ما می پرسد. سپس با صفحه ای گرافیکی شبیه حالت قبل



مواجه می شوید که در آن تصویر PROM نیز اضافه شده است. مطابق شکل زیر:

و وقتی که برای Generate کردن فایل سوال کرد گزینه Yes را انتخاب می کنیم و PROM برنامه ریزی می شود. حالا باید FPGA هم برنامه ریزی و تنظیم شود. دوباره مثل حالت قبل

روش JTAG را از قسمت مربوطه انتخاب کرده و روی Configure Device دوبار کلیک می کنیم و مثل حالت قبل گزینه ها را انتخاب می کنیم. مثل boundary Scan mode و ...

سپس فایل PROM را می خواهد که آن را انتخاب و روی PROM میریزیم و سپس فایل bit. مربوط به FPGA را می خواهد که آن را هم انتخاب می کنیم.

PROMها هم به صورت سریال و هم به صورت موازی می توانند به FPGA متصل شوند که ما از روش سریال استفاده کردیم.

سپس روی تصویر PROM کلیک راست کرده و گزینه Program را می زنیم و Verify را هم انتخاب شده نگه می داریم.

در این حالت دیگر نیازی به پروگرام کردن FPGA نیست چون PROM این کار را انجام می دهد اگر مشکلی پیش آمد یکبار منبع تغذیه را قطع کرده و دوباره پروگرام کردن PROM را تکرار می کنیم.

با آرزوی موفقیت برای شما