

آموزش کار با نرم افزار Active-HDL

با سلام خدمت دوستان عزیز.

قبل اینکه بریم با محیط کار ساده نرم افزار آشنا بشیم بهتره یه بار دیگه یادآور بشم که چرا اصلا کد vhdl می نویسیم.

Vhdl مخفف عبارت زبان طراحی سخت افزار هست. همون طور که قبلا هم گفتیم در اصلا این کد ها بعد از کامپایل شدن اصلا روی یک پردازنده اجرا نمی شوند یعنی اصلا کد اجرایی نیستند. پس کجا قراره اجرا بشن؟

یه قطعه سخت افزاری داریم که به تعداد میلیونی انواع گیت ها رو داره. گیت های داخل اون کامل هستند یعنی میشه هر مداری رو با اون ساخت. این گیت ها قابل برنامه ریزی شدن هستند. یعنی میتونیم بهشون بگیم چجوری به هم وصل بشن تا مثلا یه نیم جمع کننده داشته باشیم. اما باید بازهم فقط با صفر و یک به سخت افزارمون که اسمش fpga هست بفهمونیم که چطوری گیت هاش بهم وصل بشن.

اما به لطف کامپیوتر دوستان اومدن یه زبان سطح بالا نزدیک به زبان انسان طراحی کردن که همین vhdl هست. این کد ها در نهایت توسط کمپایلر به صفر و یک تبدیل میشن و باید اینها رو روی fpga بریزیم تا مدارمون اونجا ساخته بشه. Fpga هم مانند مثل هر ریز پردازنده ای دارای یه سری پایه هستند که ورودی و خروجی ما میشن در اخر.

خب مسئله مهم همین بود که ما بفهمین تفاوت زبان vhdl با c و جاوا چیه. یعنی اصلا هدف این زبان طراحی سخت افزار با زبان هایی که تا الان خوندم فرق میکنه.

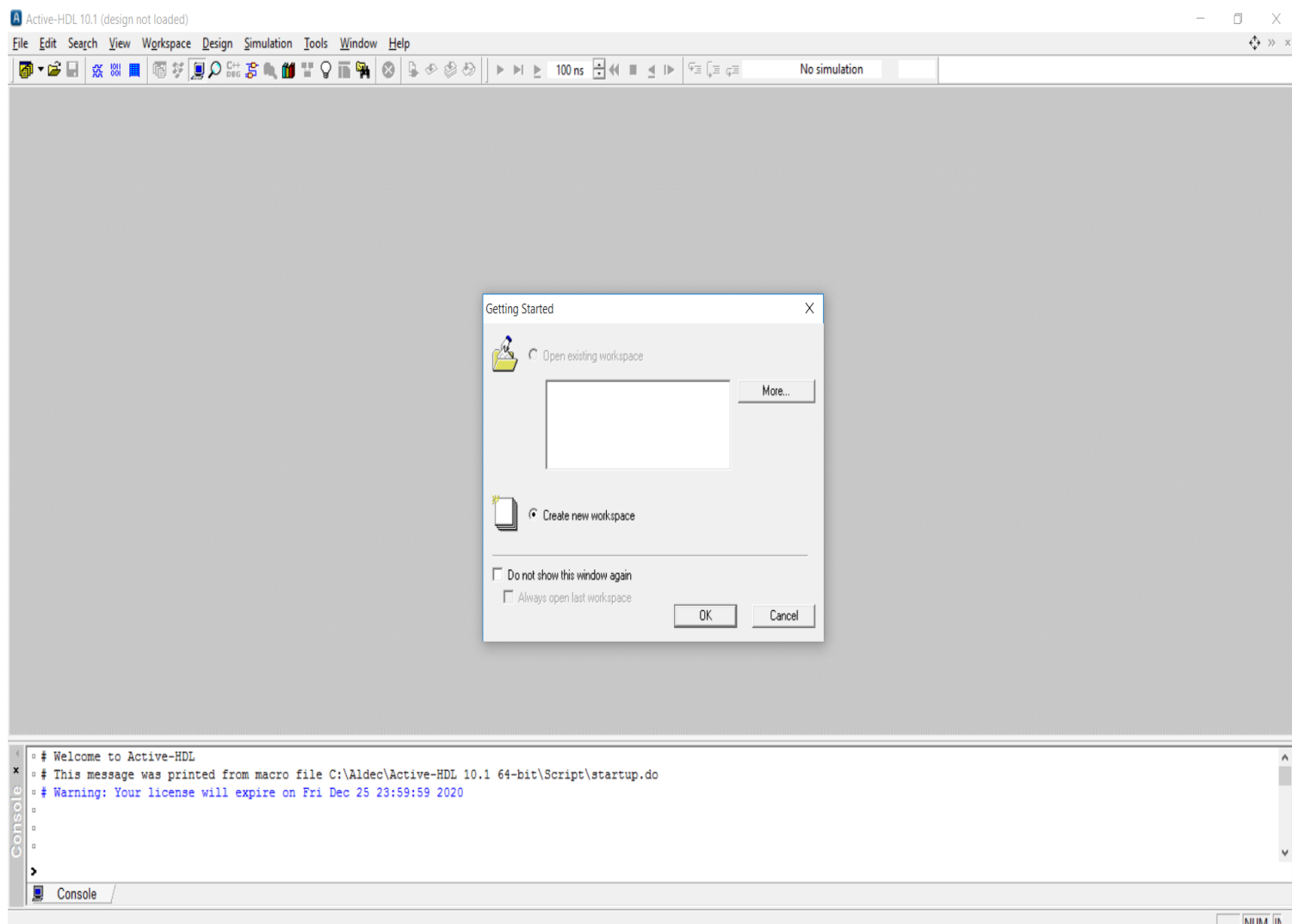
خب دوستان حالا هی سخته که ما کد بنویسیم و ببریم روی سخت افزار fpga تست کنیم خب پس برای راحتی کار میایم یه محیط شبیه سازی در کامپیوتر میسازیم یعنی همین نرم افزار که قراره آموزش بدیم و کد هامون رو به صورت شبیه سازی شده اونجا اجرا کنیم تا مراجعون به سخت افزار اصلی کم بشه.

خب پس ما کدهامون رو تویه این نرم افزار می نویسیم و به صورت فانتزی بهش ورودی میدیم (سنتر میکنیم) و مثل بقیه زبان های برنامه نویسی خروجی رو مشاهده می کنیم با این تفاوت که اینجا فقط ورودی و خروجی سینگنال منطقی ۰ و ۱ هستند! اما بعدها در صورت ادامه این کار خواهید دید که میشه مثلا یه سون سگمنت رو باهاش روشن کنید و اعداد رو باهاش نمایش بدید. یعنی یه شمارنده بسازید که اعداد رو به صورت دهدهی خودمون نمایش بده.

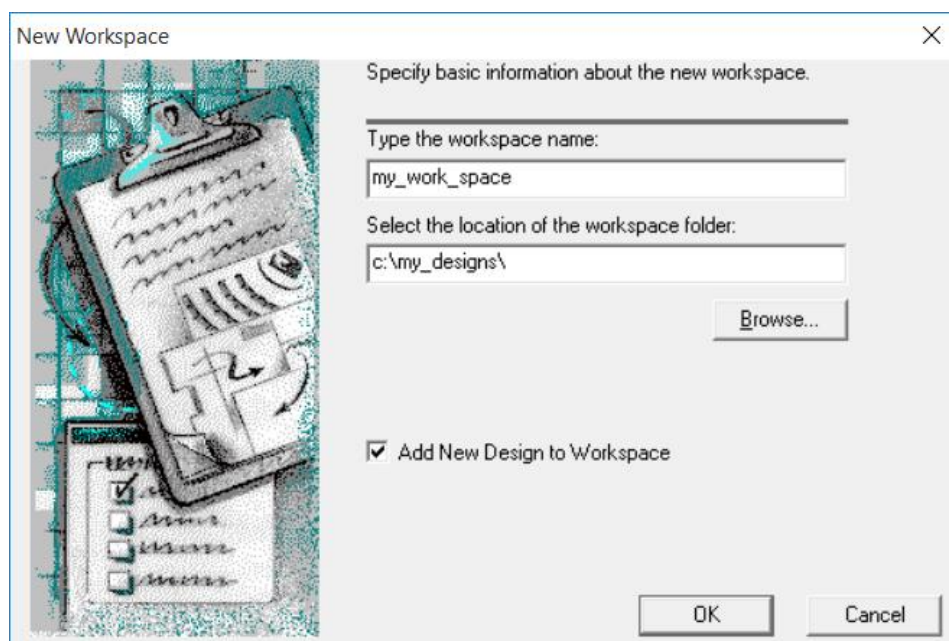
خب بریم سراغ آموزش نرم افزار:

صفحه بعد ----->

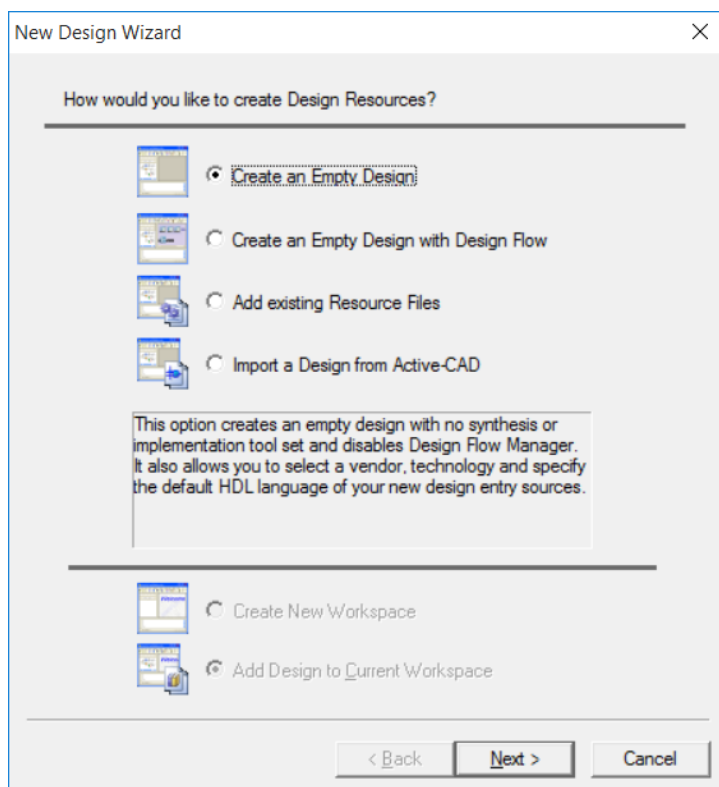
اگه موفق به کرک کردن نرم افزار شدید در صورتی که نرم افزار رو باز کنید صفحه زیر قابل مشاهده است:



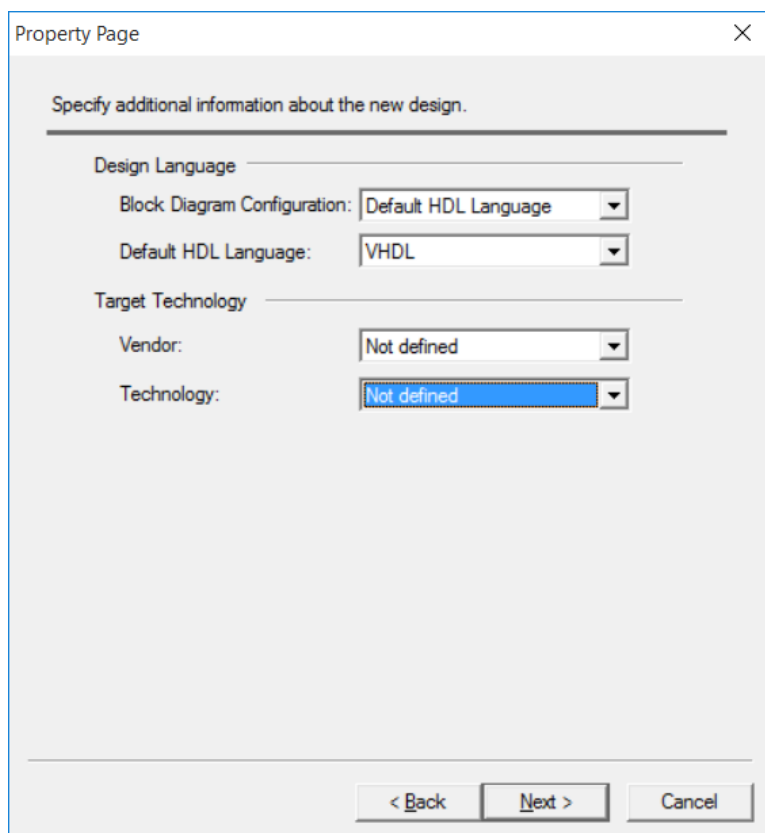
خوب چون فضای کاری (workspace) قبلا نداشتم باید به صورت پیش فرض یک فضای کاری جدید بسازیم. پس روی ok کلیک میکنیم. خوب من تا جایی که نیاز نباشه فضای کل نرم افزار رو عکس نمیگیرم و به پنجره فعال بسنده میکنم. در مرحله بعد پنجره زیر نمایش داده میشه.



خوب اسم فضای کاریتون رو به دلخواه وارد کنید و پیش برید تا پنجره زیر نمایش داده بشه.



Create an empty design رو انتخاب کنید و next بزنید. به پنجره زیر میرسد و تنظیمات این پنجره رو هم مته عکس زیر اعمال کنید:



و باز هم next و ... به پنجره زیر میرسید. در این پنجره اسم طرح خودتون رو وارد کنید.

New Design Wizard

Specify basic information about the new design.

Type the design name:

Select the location of the design folder:

☐ Do not create design directory

The name of the default working library of the design:

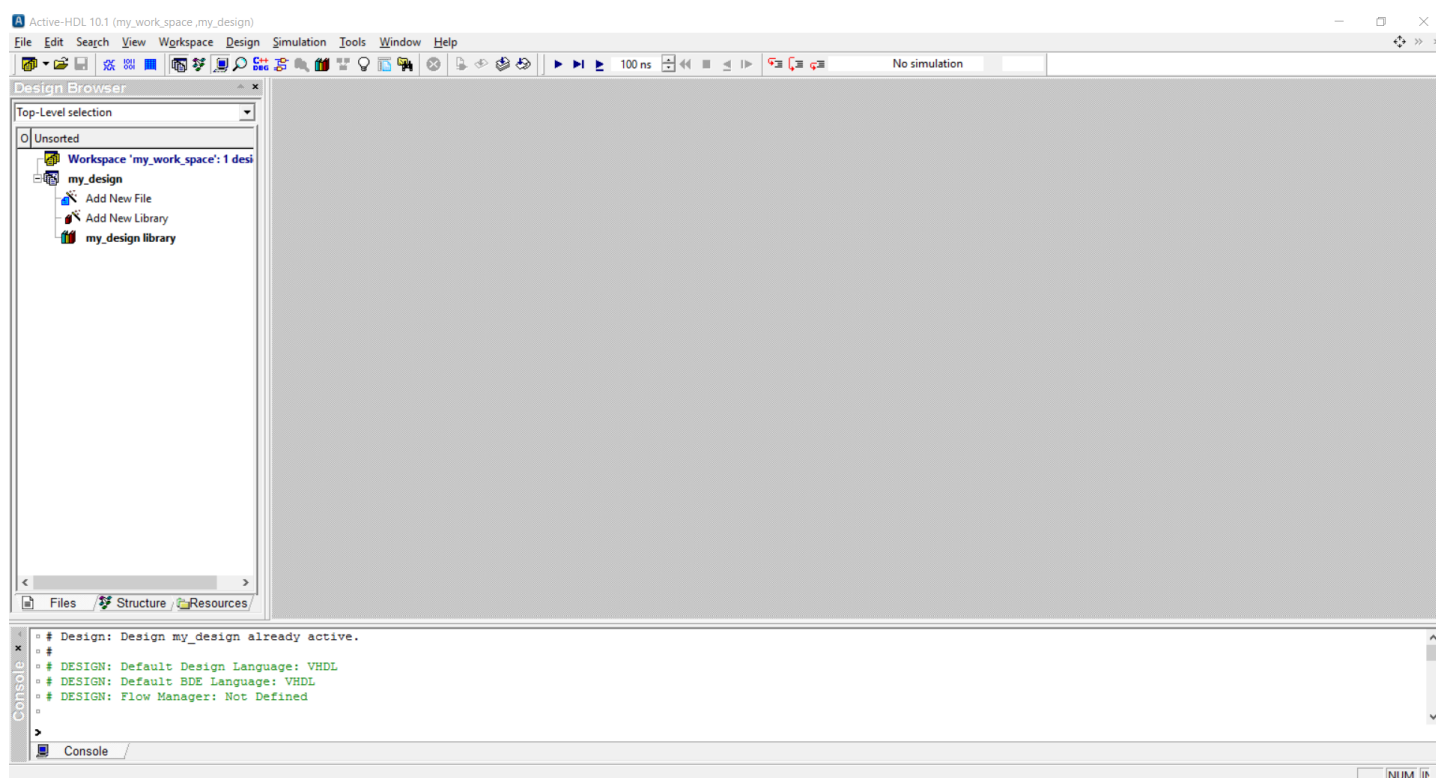
The name specified here will be used as the file name for the library files and as the logical name of the library. You can change the logical name later on.

Design file path:

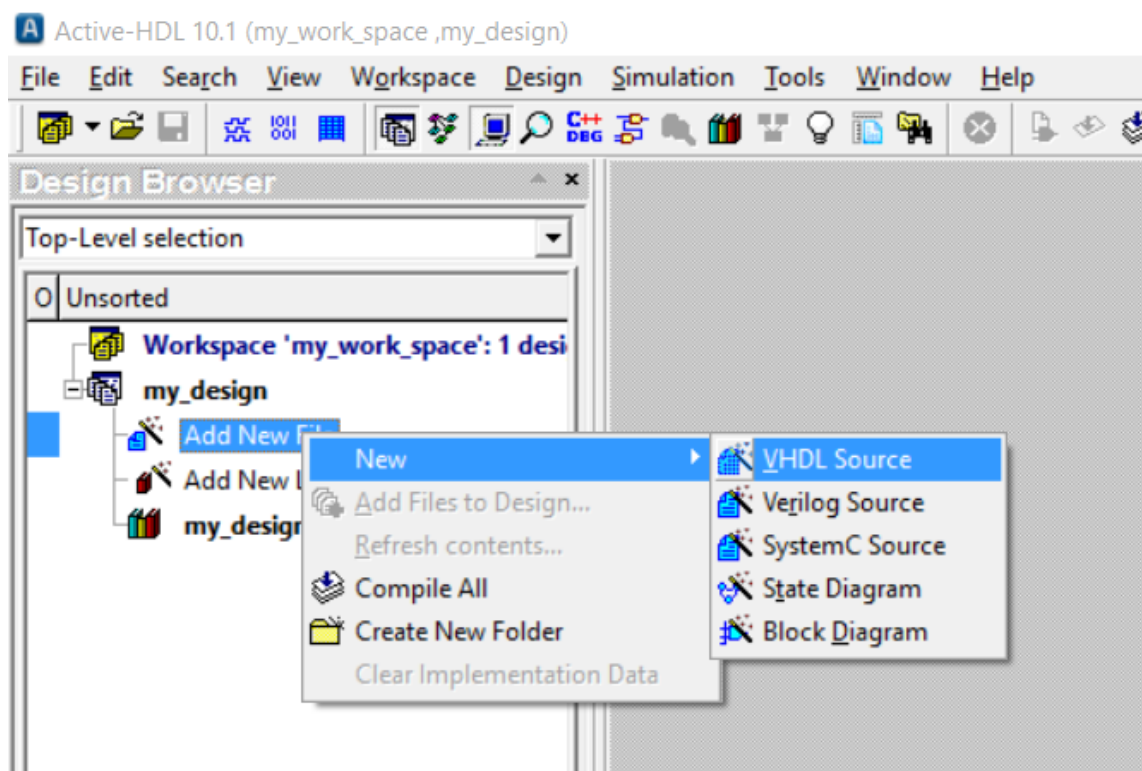
< Back Next > Cancel

بازم هم next رو بزنید و در ادامه finish. ... خب تا اینجا ما به فضای کاری و به محیط طراحی برای خودمون ساختیم.

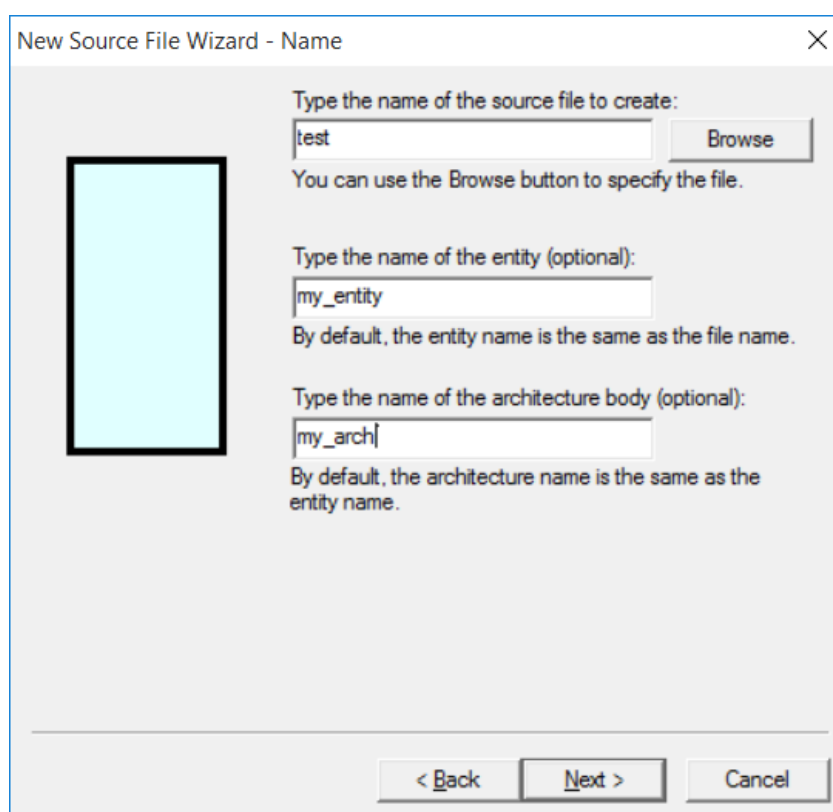
در نهایت پنجره کلی نرم افزار تون میشه مثل عکس زیر:



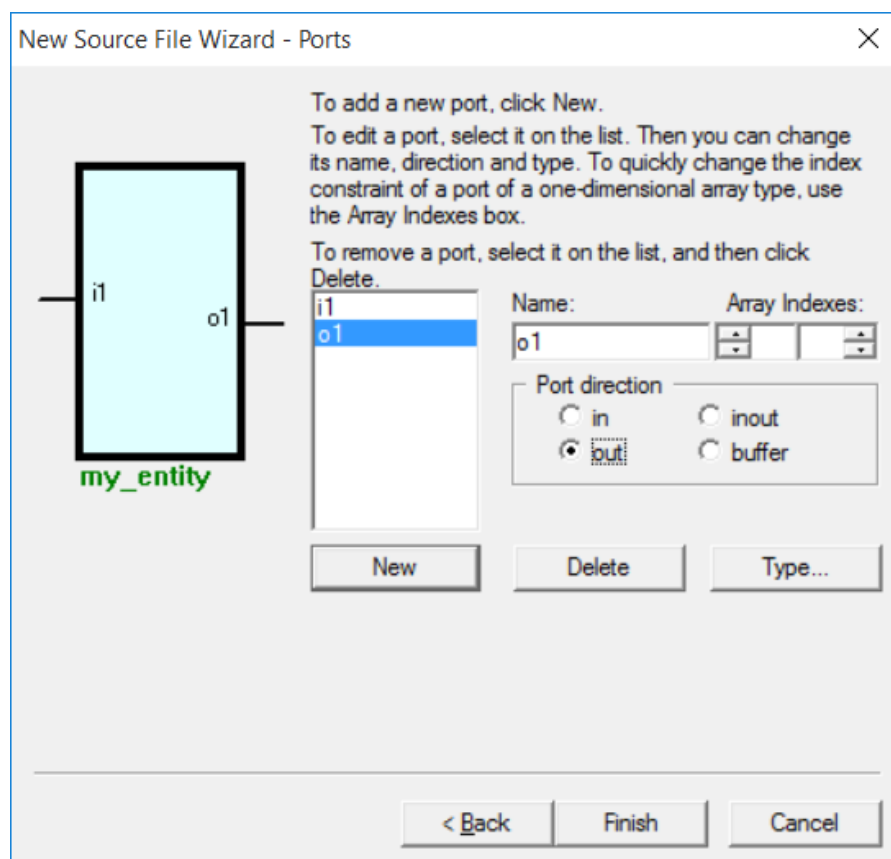
حالا باید یه فایل جدید vhdl بسازیم و کد های خودمون رو در داخل اون بنویسیم. ستون قسمت سمت چپ نرم افزار که طرح فعال شما رو نشون میده یه ایکون هست که نوشته add new file. روش راست کلیک کرده و طبق عکس زیر یه فایل vhdl بسازید:



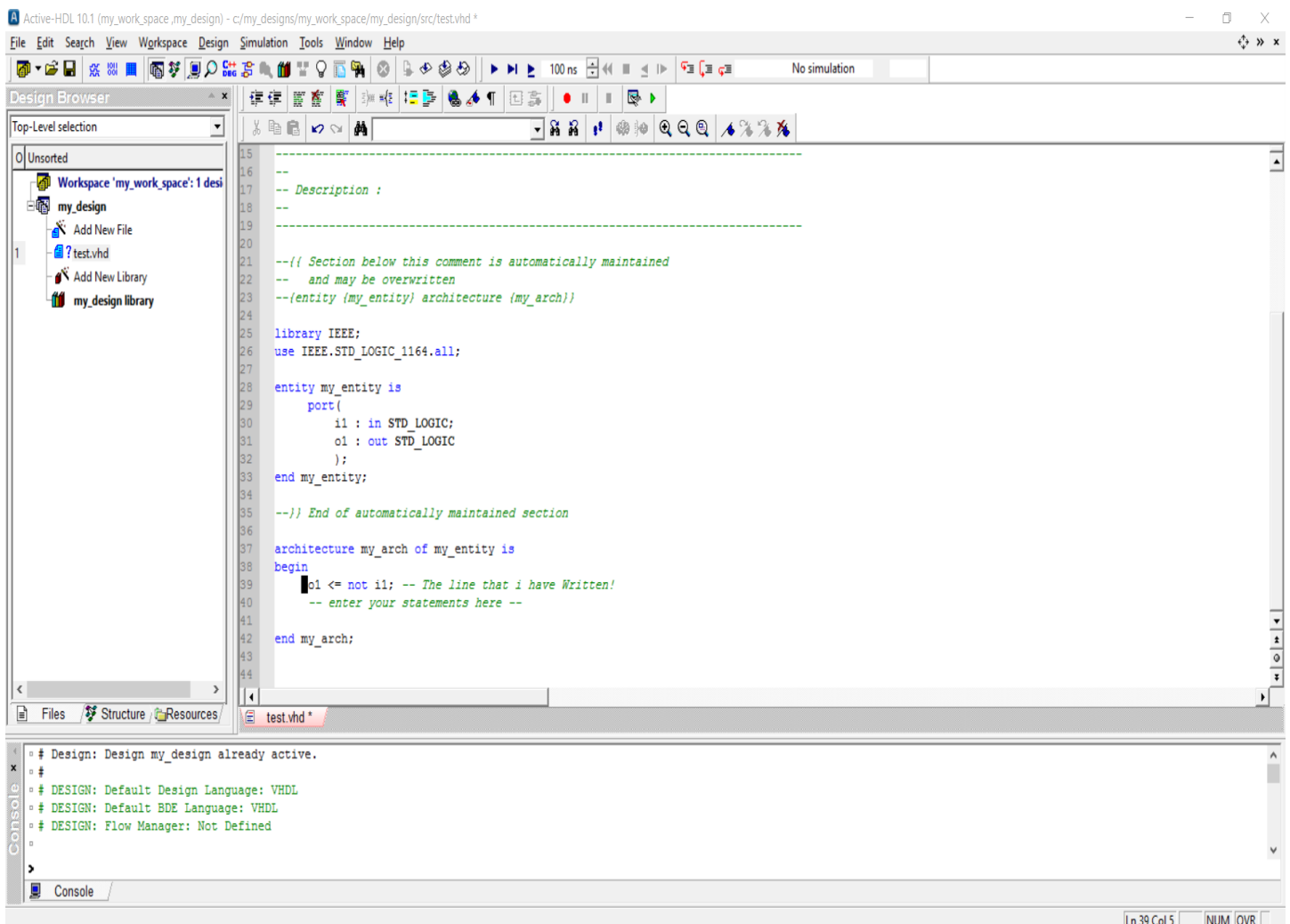
وقتی روی یه VHDL Source کلیک میکنید یه پنجره جدید اونجا هم next رو بزنید تا پنجره زیر ظاهر بشه.



تو این پنجره طبق اطلاعات درخواستی اسم فایل vhdl، اسم entity و اسم architecture رو وارد کنید و next.... تا به مرحله زیر برسید:



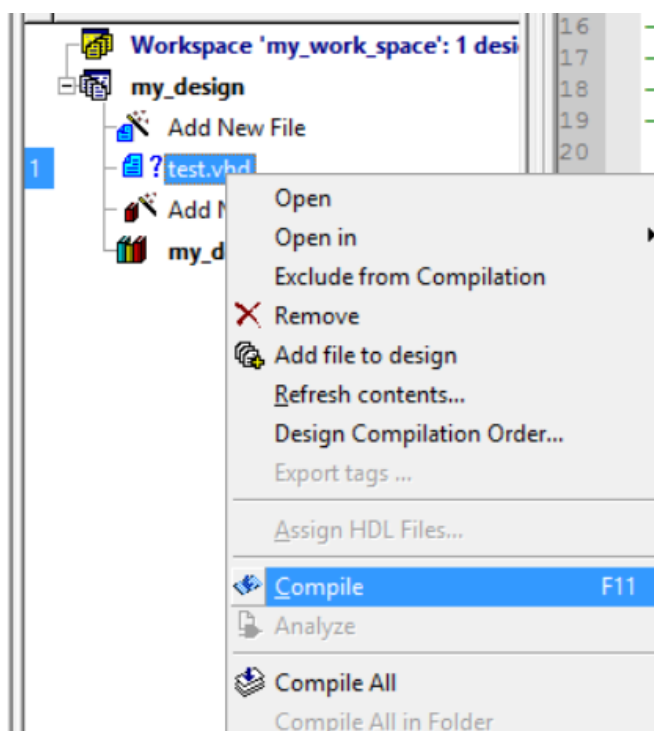
در اینجا هم ورودی و خروجی entity رو مشخص کنید. البته این برای ساختن یه کد پیش فرض هستید. اگر همه چیزی وارد نکنید بعد میتونید با کد نویسی در قسمت port ورودی و خروجی هاتون رو مشخص کنید. پس روی finish کلیک کنید تا فایل vhdل با کد های پیش فرض ساخته بشه:



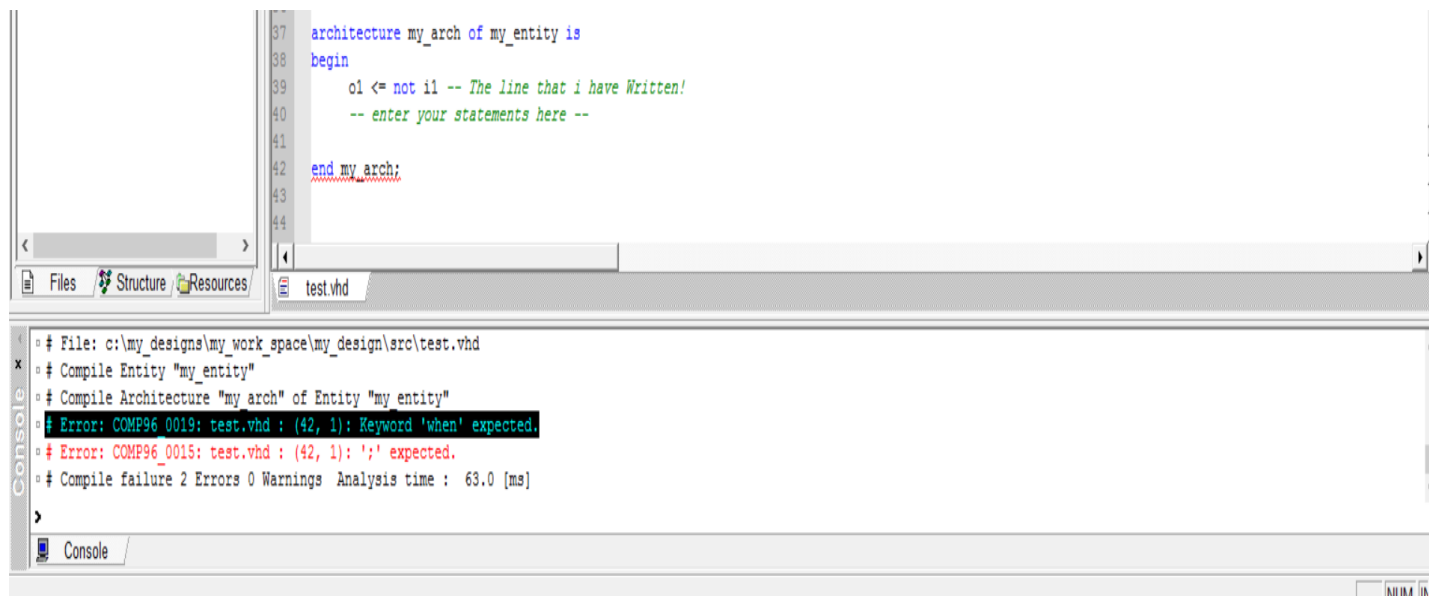
خوب تا الان یه فایل به نام test.vhdl ساختم. من خودم تو خط ۳۹ یه کد نوشتم تا مقدار not ورودی i1 رو به خروجی بفرسته.

```
37 architecture my_arch of my_entity is
38 begin
39     o1 <= not i1; -- The line that i have Written!
40     -- enter your statements here --
41
42 end my_arch;
```

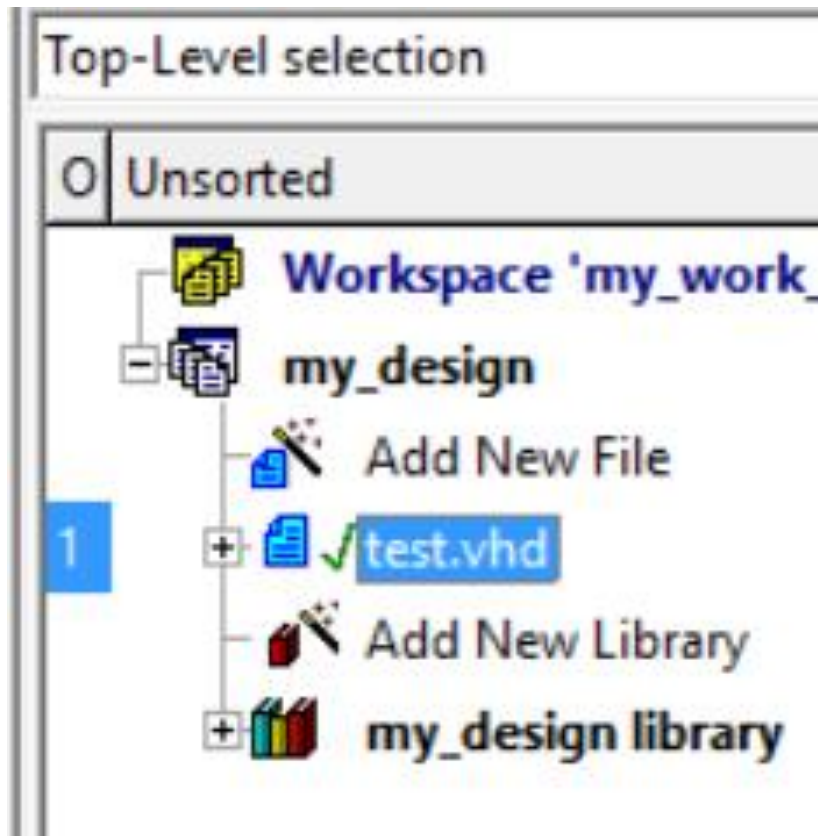
خوب حالا برای اینکه بتونیم کد رو اجرا کنیم و سنتر کنیم ابتدا باید کامپایل کنیم. لذا برای انجام این کار از روی ستون چپ روی فایل مذکور راست کلیک و کرده و کامپایل رو میزنیم:



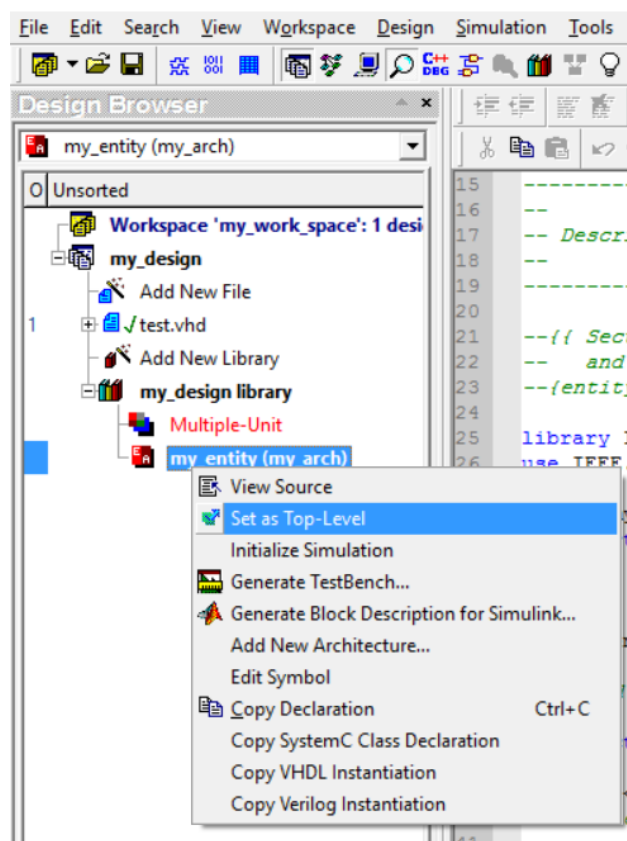
وقتی کد بدون خطا کامپایل بشه علامت ؟ کنار فایل تبدیل به یک تیک سبز رنگ تبدیل میشه. اگر خطایی رخ بده در قسمت پایین نرم افزار نمایش داده میشه: مثل عکس زیر که من ؛ اخر خط کدی نوشتم رو بر میدارم تا خطای کامپایل بده. به شکل زیر:



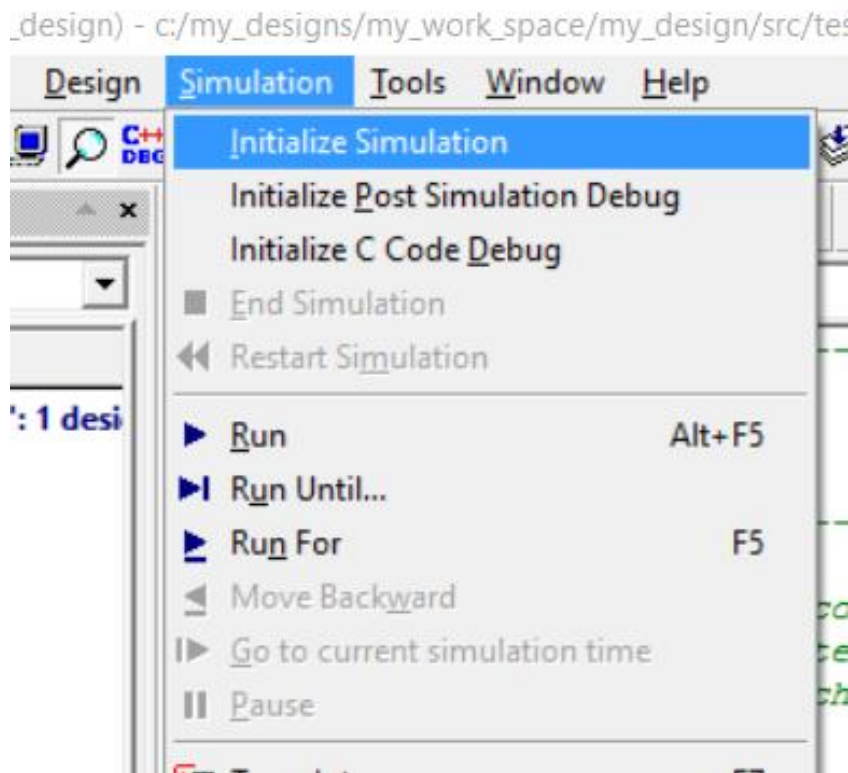
می بینید که خطا داده که ؛ انتظار می رود. البته ممکنه خطای دیگه هم بده که به گرامر های دیگه مربوط میشه و یک خطا مورد نظر ما باشه. ولی بالاخره خطا رو میده. در صورت کامپایل صحیح فایل به شکل زیر می شود:



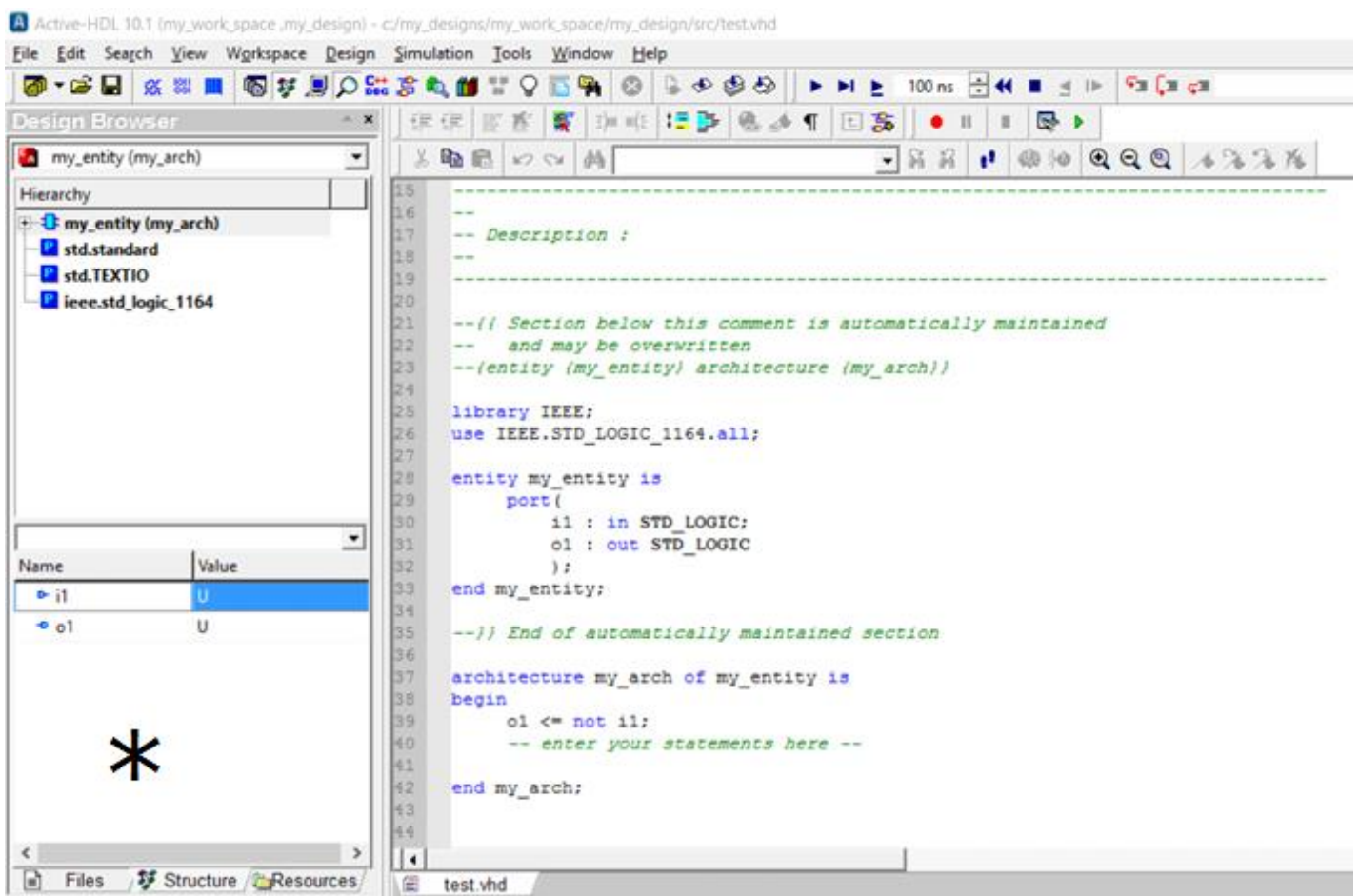
حالا می رسیم به قسمت اصلی تر. برای اینکه بتوانیم کدی که نوشتیم پس از کامپایل کردن سنتز کنیم باید محیط شبیه سازی رو آماده کنیم. ابتدا روی علامت + my_design library کلیک میکنیم تا به صورت زیر باز شود. در این جا روی My_entity(my_arch) راست کرده و set as top-level را انتخاب کرده.



حال در این مرحله از منوی simulation گزینه initialize simulation را انتخاب می کنیم.



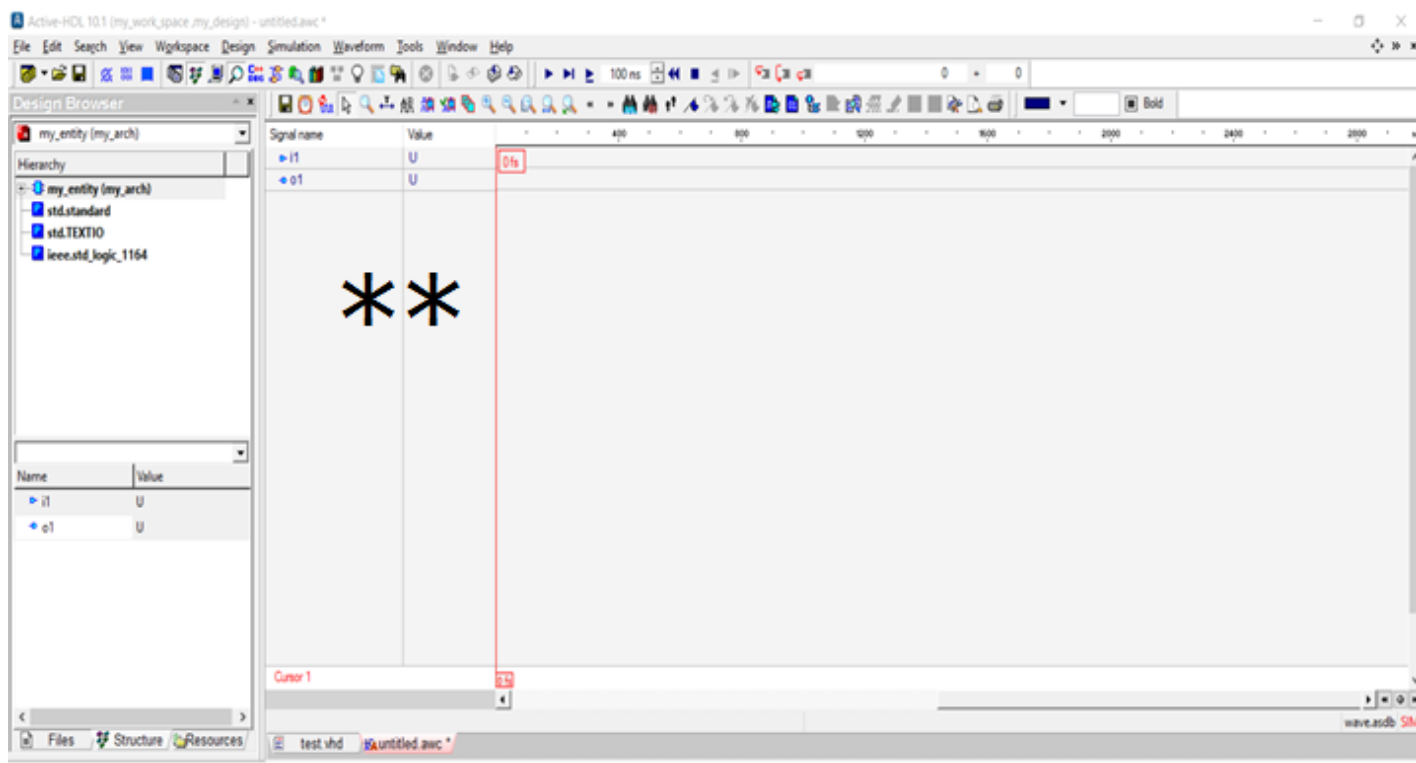
حال محیط شبیه سازی آماده به کار است. ستون سمت چپ نرم افزار به شکل زیر می شود:



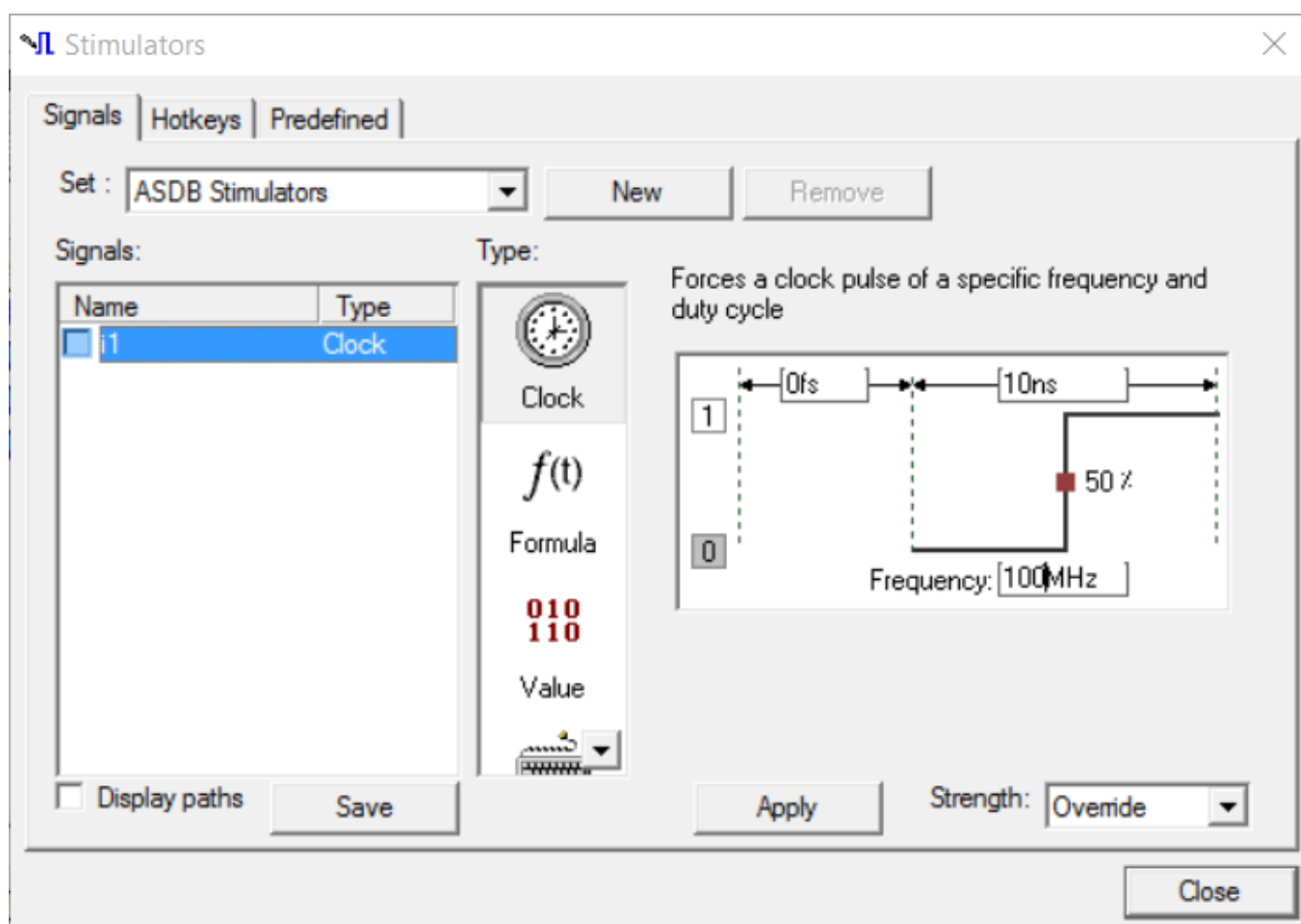
حال در قسمتی که با * مشخص شده است راست کلیک کرده و select all رو انتخاب میکنیم.

پس از این دوباره روی یکی از سیگنال های i1 یا o1 راست کلیک کرده و add to waveform را انتخاب می کنیم.

به شکل زیر :



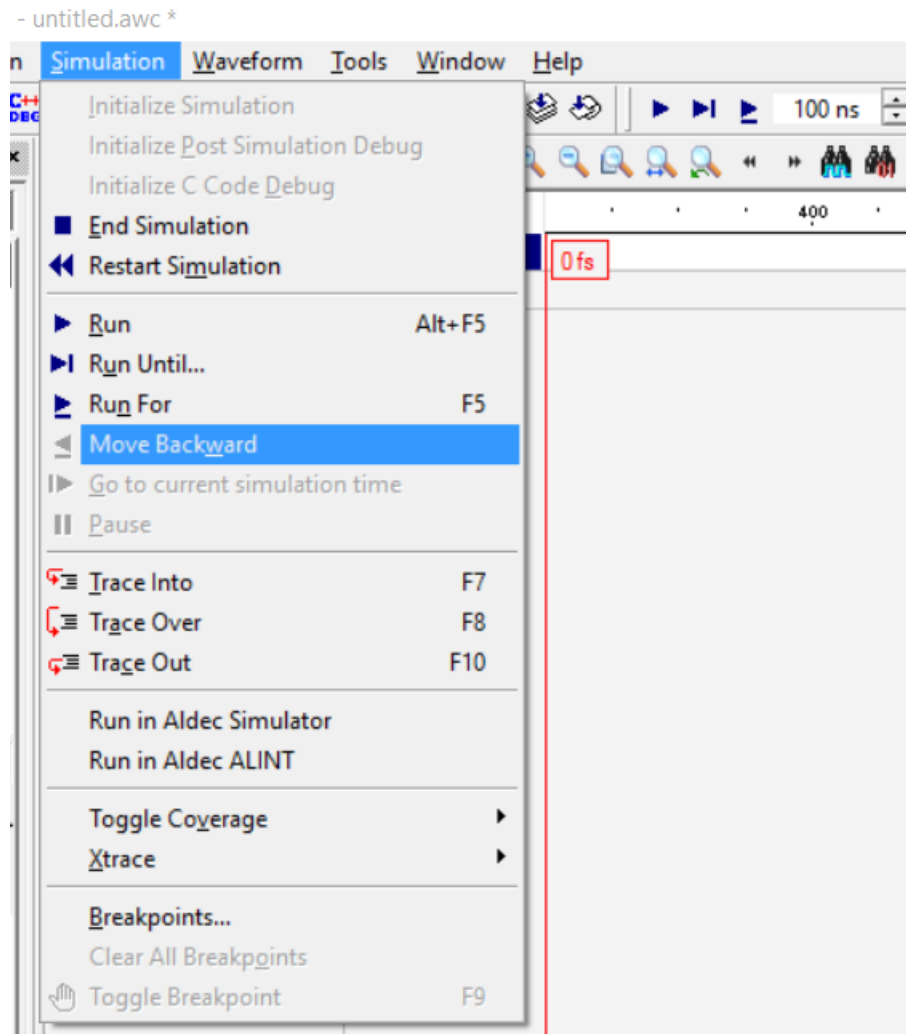
حال در قسمت ** روی سیگنال ورودی i1 راست کلیک کرده و از simulator را انتخاب کنید. پنجره زیر ظاهر می شود.



در این پنجره ابتدا رو علامت کلاک کلیک کنید و فرکانس کاری را روی 100mhz ببرید. این یعنی در هر ثانیه کلاک ۱۰۰ میلیون بار پالس میدهد. یعنی هر پالس کلاک ۱۰ نانو ثانیه طول می کشد. و در اخر روی apply کلیک میکنیم. البته شما می توانید مقادیر دلخواه برای فرکانس در نظر بگیرید.

در نهایت این پنجره رو می بندیم.

خب باید حالا کلاک رو روشن کنیم. برای این کار به منوی simulation برویم:



در این جا سه حال برای سنتر کردن هست:

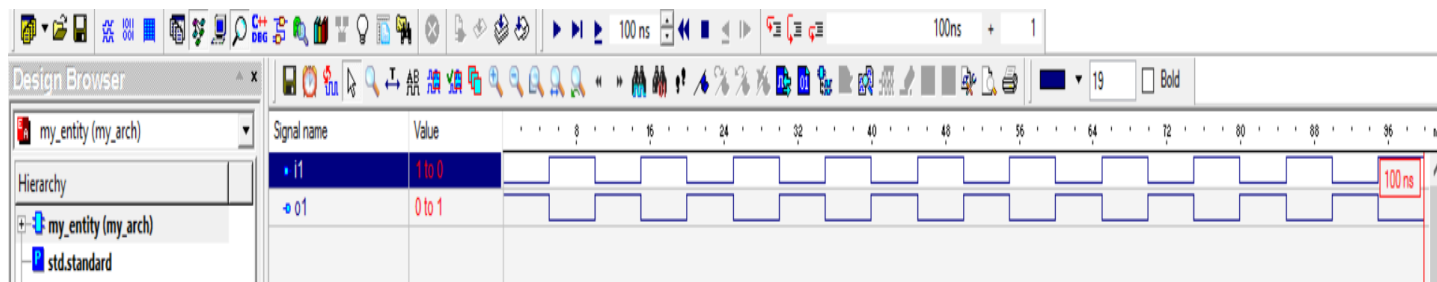
۱- run : این یعنی کلاک تا وقتی که آن را متوقف نکنیم در حال کار خواهد بود.

۲- run until : در صورتی کلیک روی این گزینه در پنجره باز شده مقداری رو که وارد کنید زمان کار کردن کلاک است. یا بهتره بگیم که مقدار زمانی که ما تست بگیریم.

۳- run for : این گزینه میگوید که برای یک زمان مشخص شده در صورت هر بار کلیک روی آن همان مقدار تست صورت میگیرد.

اگر دقت کنید در سمت راست عکس بالا عدد 100 ns نوشته شده است. در صورت کلیک روی run for برای اولین بار تست ۱۰۰ نانو ثانیه به جلو می رود یعنی ۱۰۰ نانو ثانیه سیستم کلاکش روشن میشود. اگر دوباره کلیک کنیم باز ۱۰۰ نانو ثانیه دیگر تست ادامه پیدا میکند و به ۲۰۰ نانو ثانیه میرسیم.

حال تست رو برای ۱۰۰ نانون ثانیه انجام می دهیم:



همانطور که می بینید خروجی o1 دقیقاً مخالف ورودی است. پس یعنی خروجی طبق نظر ماست و برنامه درست کار میکند.

خب همین قد از کار کردن با نرم افزار رو بدونیم فعلاً میتونیم تمام برنامه هایی که استاد تا اینجا گفته رو تست کنیم. اگر در کرک کردن یا کار کردن با برنامه به مشکل برخوردید حتماً کمکتون میکنیم.

تهیه کننده: امیرحسین قاسمی مقدم — ah.gh74@yahoo.com