به نام خدا

VLSI : فاز B پروژه

نام و نام خانوادگی : علی یداللهی شماره دانشجویی : 400102233

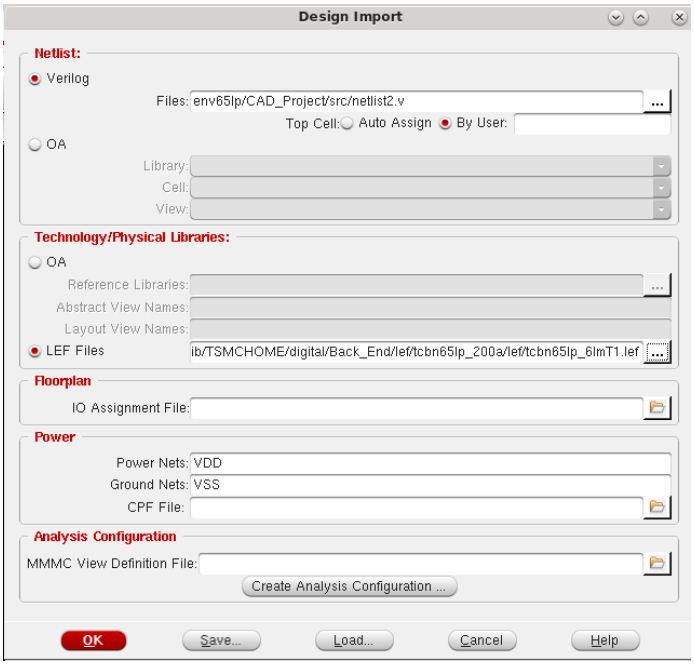
قسمت Design compiler

ماژول اصلی ما CORDIC_ip است و یک ساب ماژول به نام CORDIC_element دارد. ابتدا ساب ماژول را در design_compiler کامپایل کرده و سپس ماژول اصلی را کامپایل می کنیم. چندین بار با چند کلاک مختلف report گرفته شده که نتایج به این صورت است: (دستورات اجرا در design compiler در یک فایل آورده شده اند.) (در ماژول اولیه ورودی ها و خروجی ها به صورت رجیستر تعریف شده بودند بنابراین نیازی به تغییر در این وضعیت نبود)س

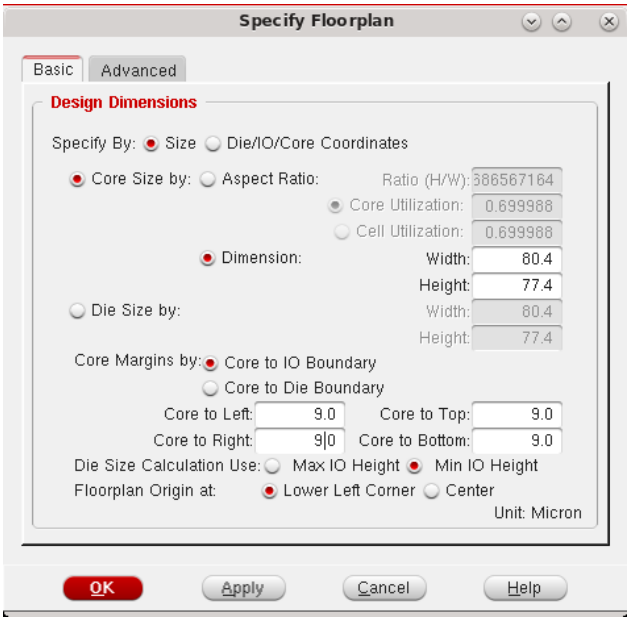
clock	power	Area	Energy
5ns	0.7832	2899	3.916
2ns	1.9354	2809	3.8708
1ns	3.8710	2809	3.871
0.8ns	4.8363	2809	3.869
0.6ns	6.3295	3010	3.7977
0.5ns	7.7080	3043	3.854
0.4ns	9.8118	3267	3.9247
0.37ns	10.8311	3335.76	4
0.35ns	12.2978	3590	4.3

قسمت innovus

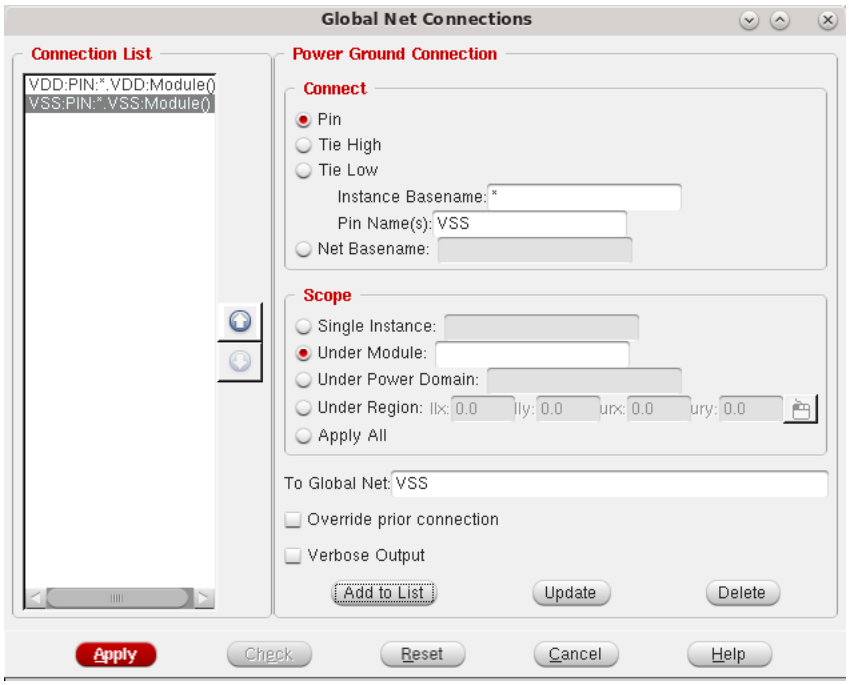
- Import کردن netlist



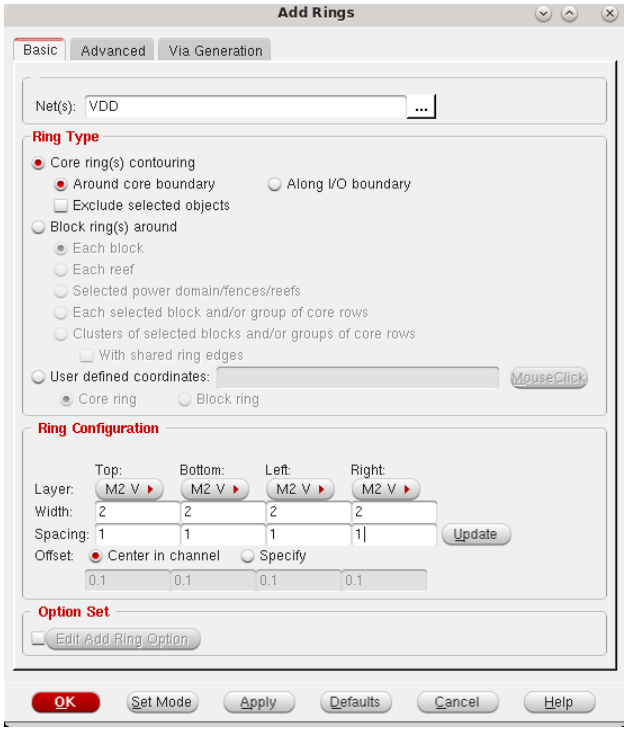
- Specify floorplan

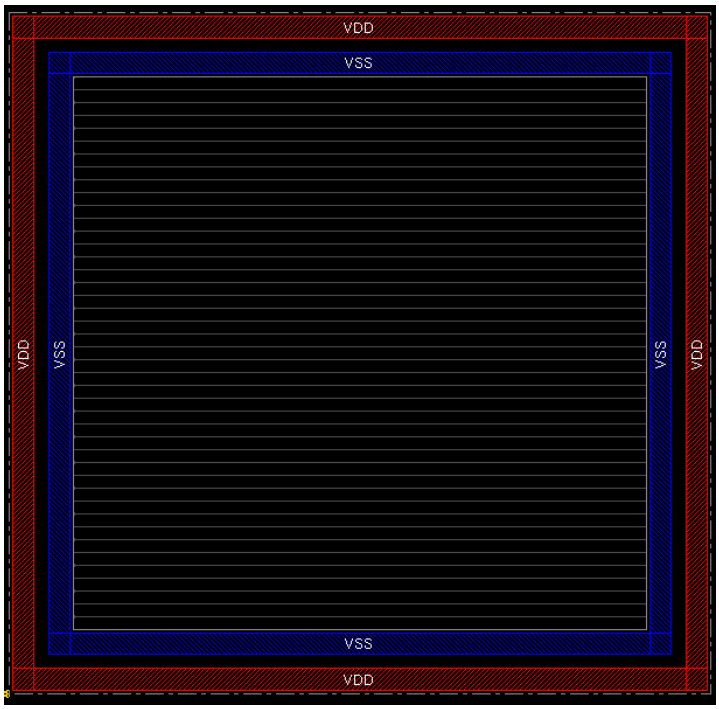


- حال باید VDD و VSS را به عنوان global net connections مشخص کنیم:

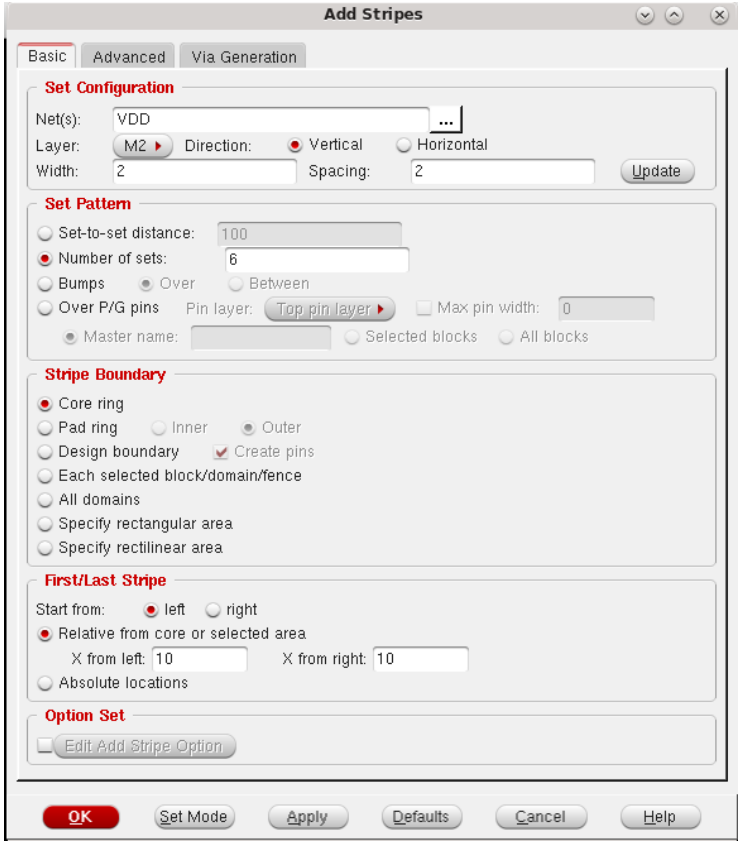


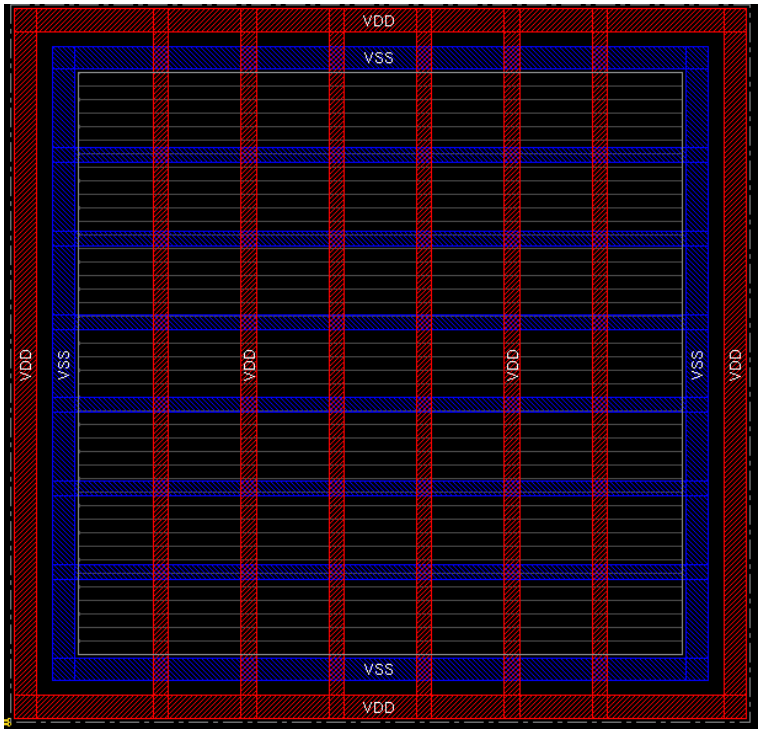
- در این مرحله ring ها را اضافه می کنیم:



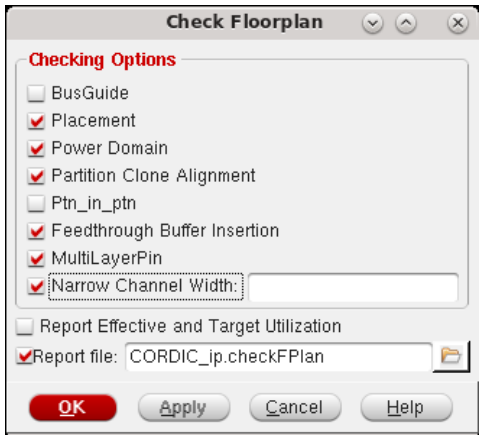


- وصل کردن strip ها


Add Stripes
Basic
Advanced
Via Generation
Set Configuration
Net(s):
VDD
Layer:
M2
Direction:
Vertical
Horizontal
Width:
2
Spacing:
2
Update
Set Pattern
Set-to-set distance:
100
Number of sets:
6
Bumps
Over
Between
Over P/G pins
Pin layer:
Top pin layer
Max pin width:
0
Master name:
Selected blocks
All blocks
Stripe Boundary
Core ring
Pad ring
Inner
Outer
Design boundary
Create pins
Each selected block/domain/fence
All domains
Specify rectangular area
Specify rectilinear area
First/Last Stripe
Start from:
left
right
Relative from core or selected area
X from left:
10
X from right:
10
Absolute locations
Option Set
Edit Add Stripe Option
OK
Set Mode
Apply
Defaults
Cancel
Help


VDD
VSS
VDD
VSS
VDD
VDD
VSS
VDD
VSS
VDD

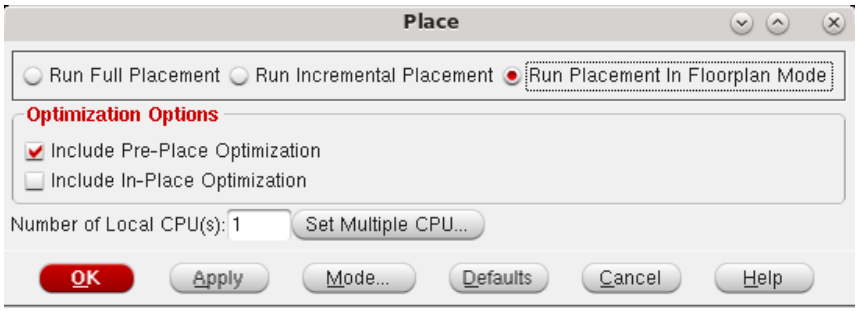
- Check floorplan

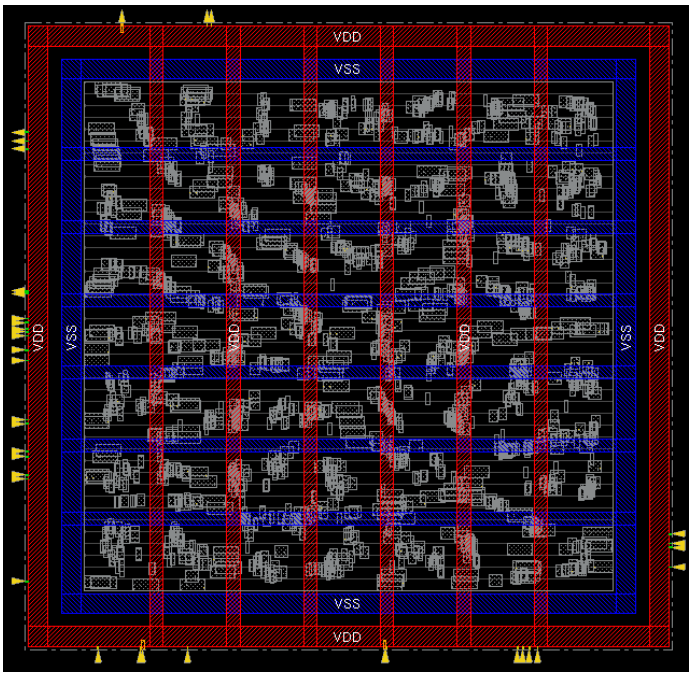



```
Checking Preroutes.....
No. of regular pre-routes not on tracks : 0
 Calling CheckPlace .....
Begin checking placement ... (start mem=628.8M, init mem=630.8M)
*info: Placed = 0
*info: Unplaced = 1802
Placement Density:70.00%(4356/6223)
Finished checkPlace (cpu: total=0:00:00.2, vio checks=0:00:00.0; mem=630.8M)
Calling VerifyPowerDomain .....
```

تقریبا 70 درصد از سطح چیپ اشغال شده است.

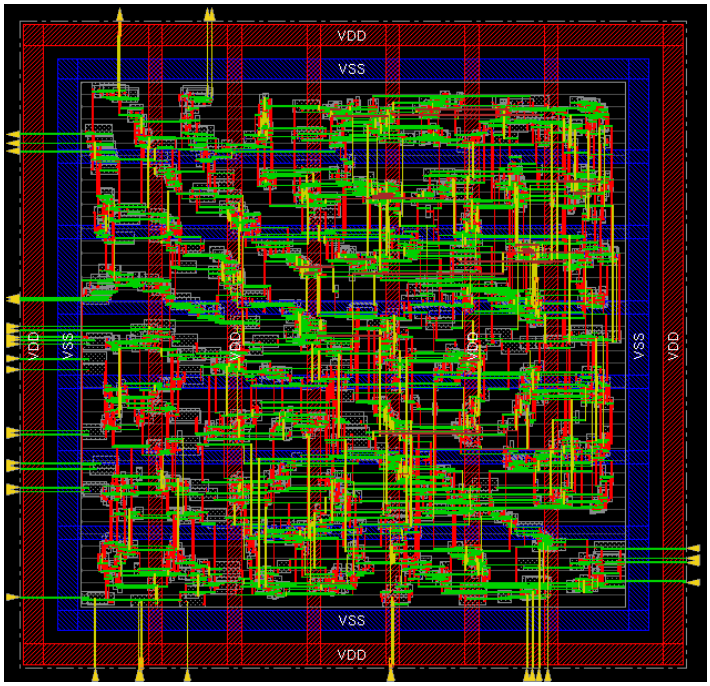
- Placement




VDD
VSS
VDD
VSS
VDD
VDD
VDD
VSS
VDD
VSS
VDD

- Routing

Trial Route
Trial Route Effort Level
prototyping
low effort
medium effort
high effort
use routing guide
CORDIC_ip.guide
save routing to
CORDIC_ip.route
ignore routing obstruct
handle partition
max. route layer:
7
OK
Mode...
Cancel
Help




```
Congestion distribution:

Remain  cntH            cntV
----------------------------------------
 -5:    0        0.00%  24       0.61%
 -4:    0        0.00%  8        0.20%
 -3:    0        0.00%  11       0.28%
 -2:    0        0.00%  14       0.35%
 -1:    0        0.00%  24       0.61%
----------------------------------------
  0:    0        0.00%  37       0.93%
  1:    0        0.00%  45       1.14%
  2:    0        0.00%  69       1.74%
  3:    2        0.05%  76       1.92%
  4:    2        0.05%  149      3.76%
  5:    3956    99.90%  3503    88.46%


Total length: 1.216e+04um, number of vias: 10087
M1(H) length: 5.363e+02um, number of vias: 5422
M2(V) length: 4.817e+03um, number of vias: 3849
M3(H) length: 5.244e+03um, number of vias: 727
M4(V) length: 1.320e+03um, number of vias: 84
M5(H) length: 2.360e+02um, number of vias: 5
M6(V) length: 5.800e+00um, number of vias: 0
M7(H) length: 0.000e+00um
```

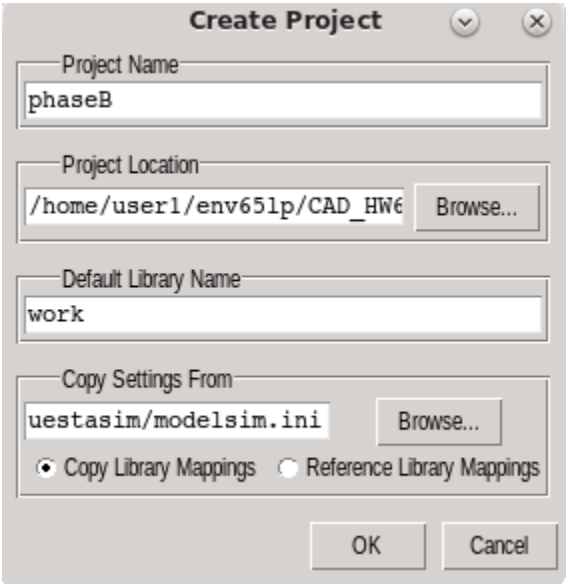
توان مصرفی:

```
innovus 1>
innovus 1> report area
Depth  Name          #Inst  Area (um^2)
-------------------------------------
0      CORDIC ip     1217   3590.28
1
innovus 2>
```

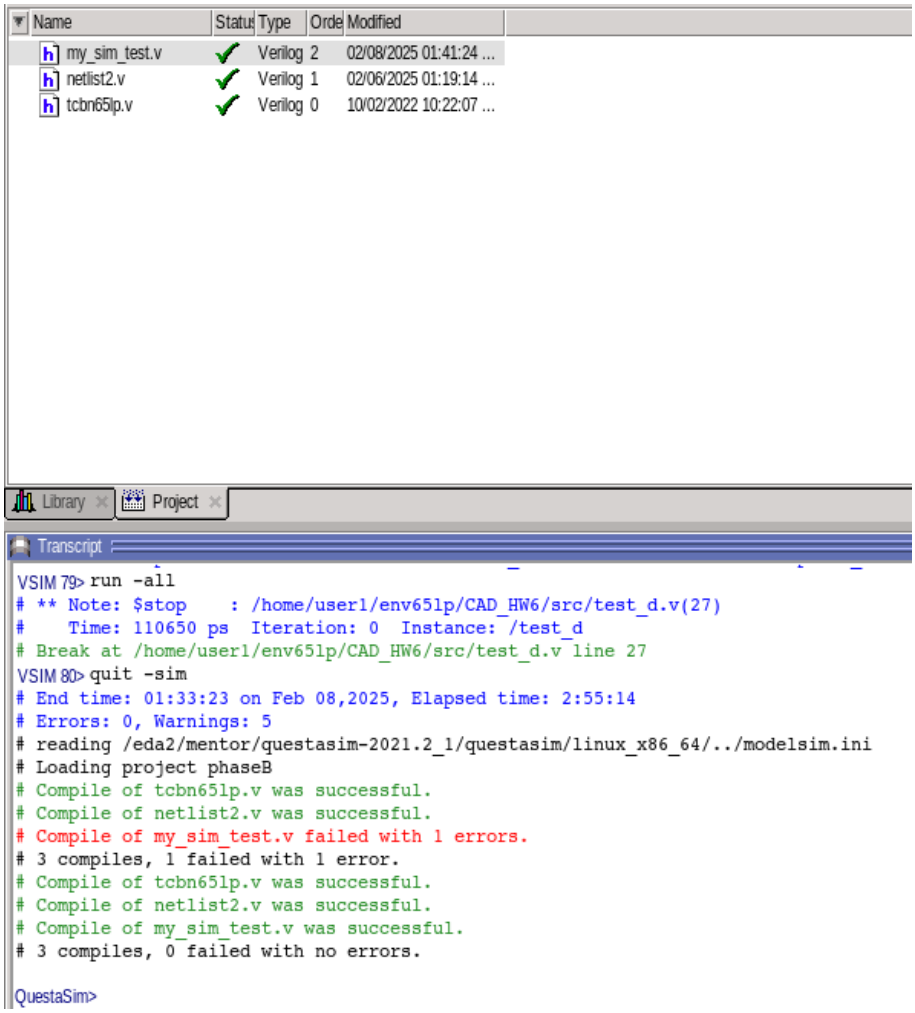
تقریبا برابر با توانی است که design compiler به ما اعلام کرد.

قسمت Questasim

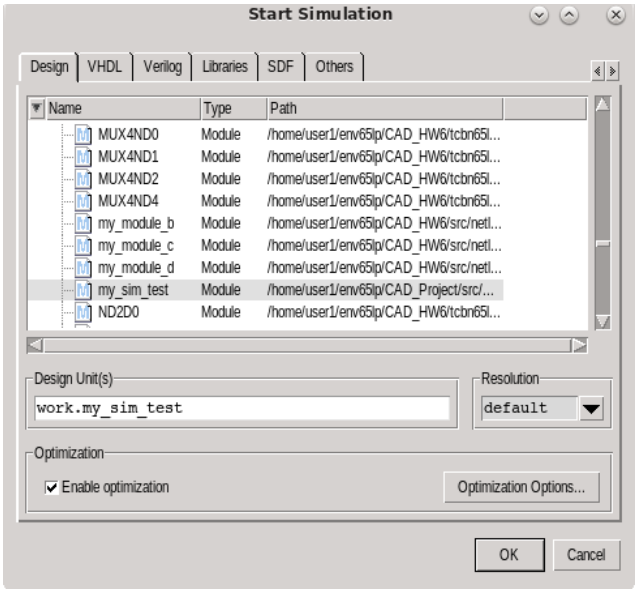
پس از باز نرم افزار یک پروژه جدید ایجاد می کنیم:

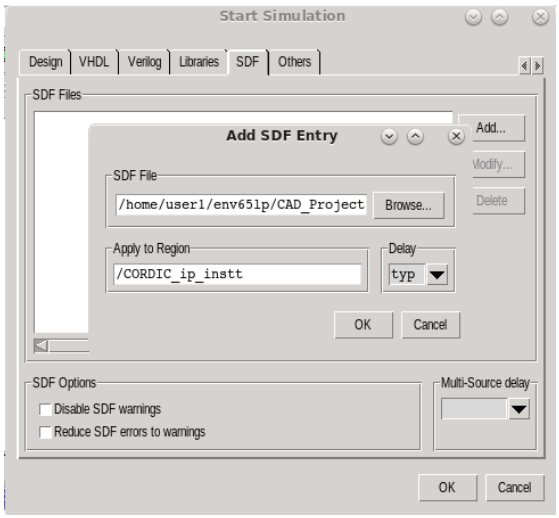


فایل netlist ، فایل مربوط به تکنولوژی و تست بنچ را به پروژه اضافه می کنیم و این 3 فایل را کامپایل می کنیم:



برای انجام دادن شبیه سازی فایل تست بنچ و sdf. را اضافه می کنیم:





از آنجایی که در ماژول CORDIC_ip از ماژول دیگری instance گرفته شده بود در اجرای شبیه سازی مشکل به وجود آمد بنابراین ماژول ها را طوری تغییر دادم که به صورت یک ماژول در آیند سپس یک بار دیگر شبیه سازی کردم:

در تست بنچ (my_sim.v) داده های تولید شده توسط متلب را از دو فایل می خوانیم. خروجی های ماژول در فایل دیگری ذخیره می شوند. سپس این ماژول را بار دیگر به اسکریپت متلب دیگری می دهیم تا درصد صحت عملکرد ماژول بعد از سنتز شدن در design_compiler به دست آید. (کلاک در تست بنچ برابر با 1 نانوثانیه تنظیم شده است). نتیجه خروجی صحت سنجی در متلب صحت عملکرد ماژول سنتزشده را تایید می کند:

(ماژول با 1000 داده مختلف ورودی تست شده است):

```
Correct Results:
        1000
```