بسم الله

علی ایمانقلی 810197692

پروژه 1 طراحی سیستم های نهفته میتنی بر FPGA

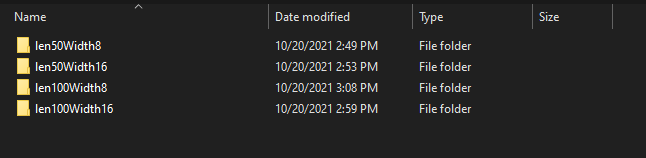
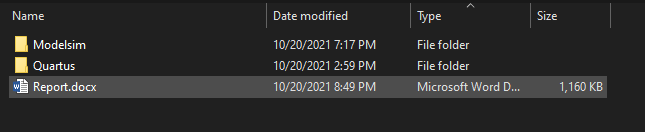
بخش اول: مقدمه و خلاصه

بخش دوم: تصویر مسیر داده و واحد کنترل و توضیحات پیرامون آن

بخش سوم: بررسی روند کلی کد های Verilog و تصاویر نتایج خروجی

بخش چهرم: بررسی روند کلی سنتز در ابزار Quartus و تصاویر خروجی ها و مقایسه بین خروجی های بدست امد.

ساختارفایل ها:



قسمت Quartus به تفکیک

**بخش اول:**

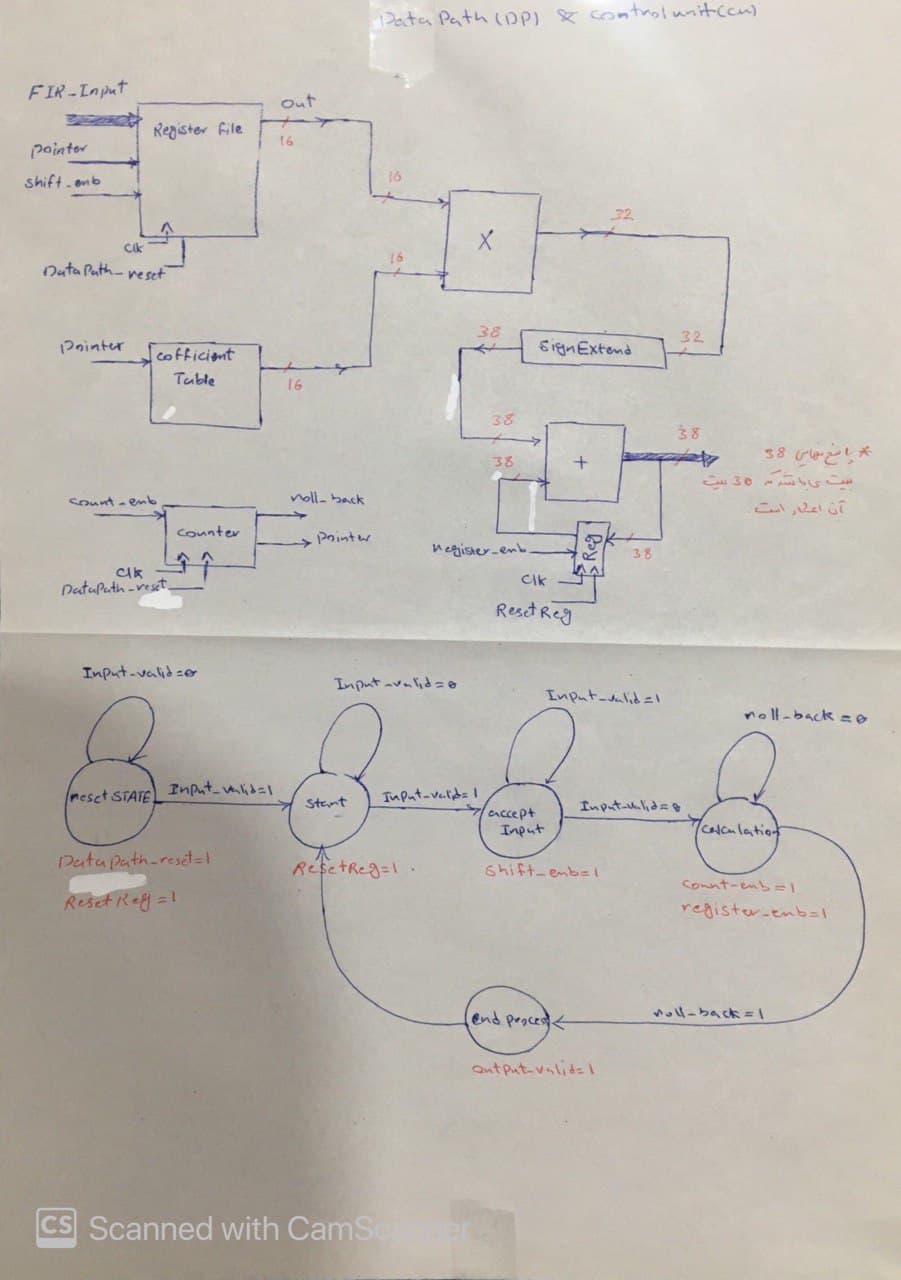
در این پروژه هدف انجام یک فیلتر خطی FIR بوده تا صوت بدست امده از نرم افزار متلب را فیلتر کند.

این پروژه با زبان توصیف سخت افزار Verilog نوشته شده است و به صورت پارامتری فیلتر خطی FIR را پیاده سازی میکند.

نکته ی حائز اهمیت ساختار serial فیلتر طراحی شده می باشد که در بخش دوم، مسیر داده و واحد کنترل آن را می توان مشاهده کرد.

\*\*\* کد به نحوی نوشته شده است که نام متغیر ها گویا ی کاربرد ان ها باشد و در برخی موارد کامنت گذاری نیز شده است. \*\*\*

**بخش دوم:**

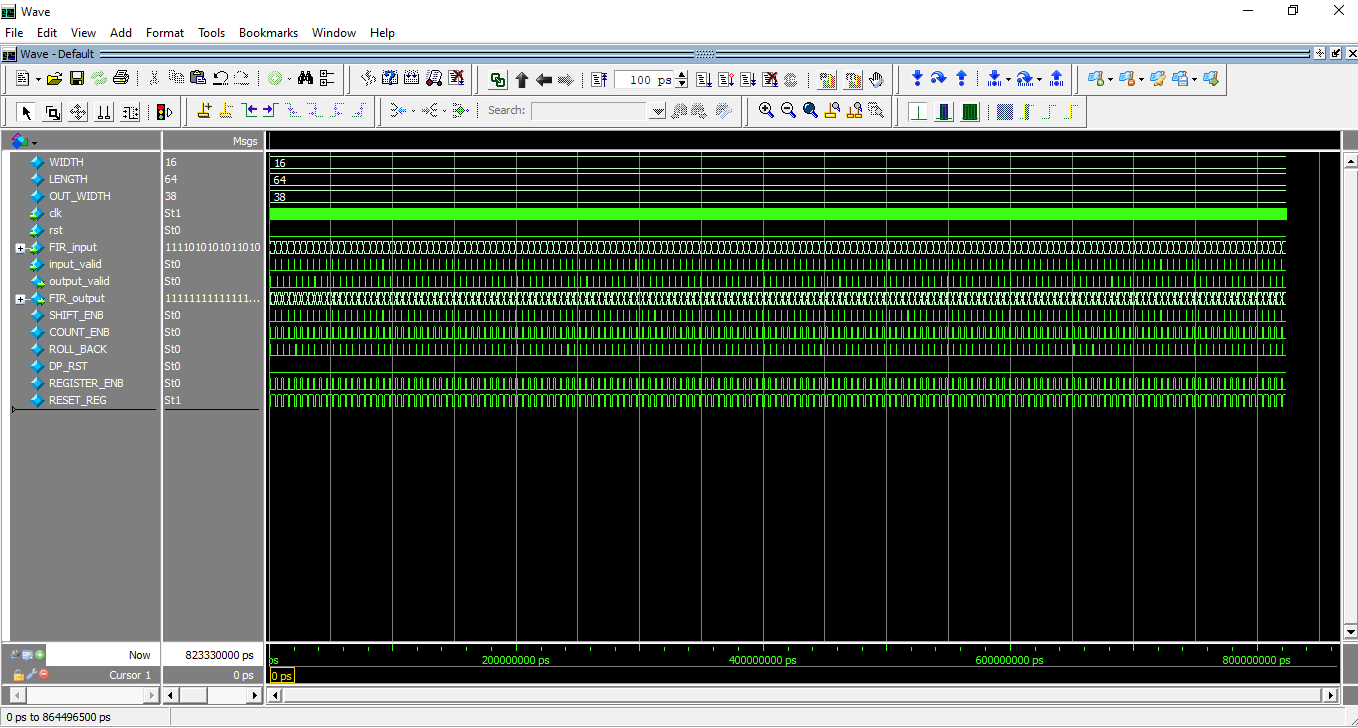


**بخش سوم:**

این بخش از دو قسمت مسیر داده و واحد کنترل تشکیل شده است.

در ابتدا سیستم با دریافت Input\_Valid ورودی FIR\_Inputرا دریافت کرده و پس از محاسبات خروجی را در FIR\_Output قرار می دهد و سیگنال Output\_Valid را فعال میکند.

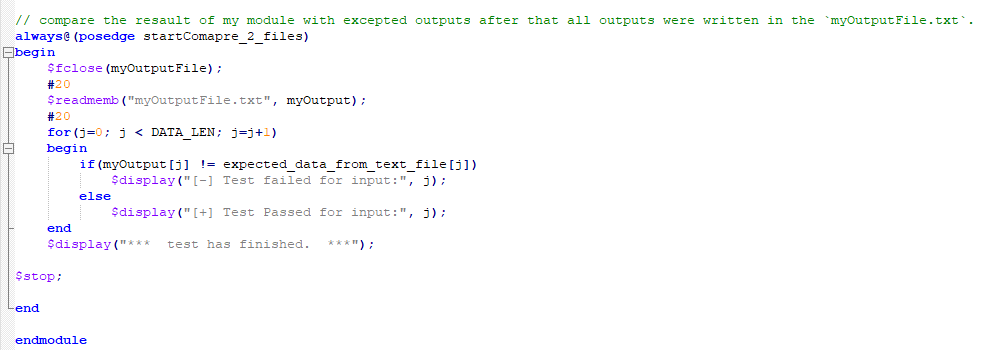
نتایج : Simulation



در قسمت test bench کد نوشته شده است که خروجی ما را با خروجی معیار مقایسه می کند.

روند این کد به این صورت است که منتظر می ماند تا ما خروجی را در یک فایل به نام myOutputFile.txt بنویسیم ( این مکانیزم توسط سیگنال startComapre\_2\_files مدریت می شود.) و سپس شروع به مقایسه یا فایل مرجع می کند.

\*\*\* فایل myOutputFile.txt خروجی برنامه می باشد که در پوشه یModelsim قرار دارد.

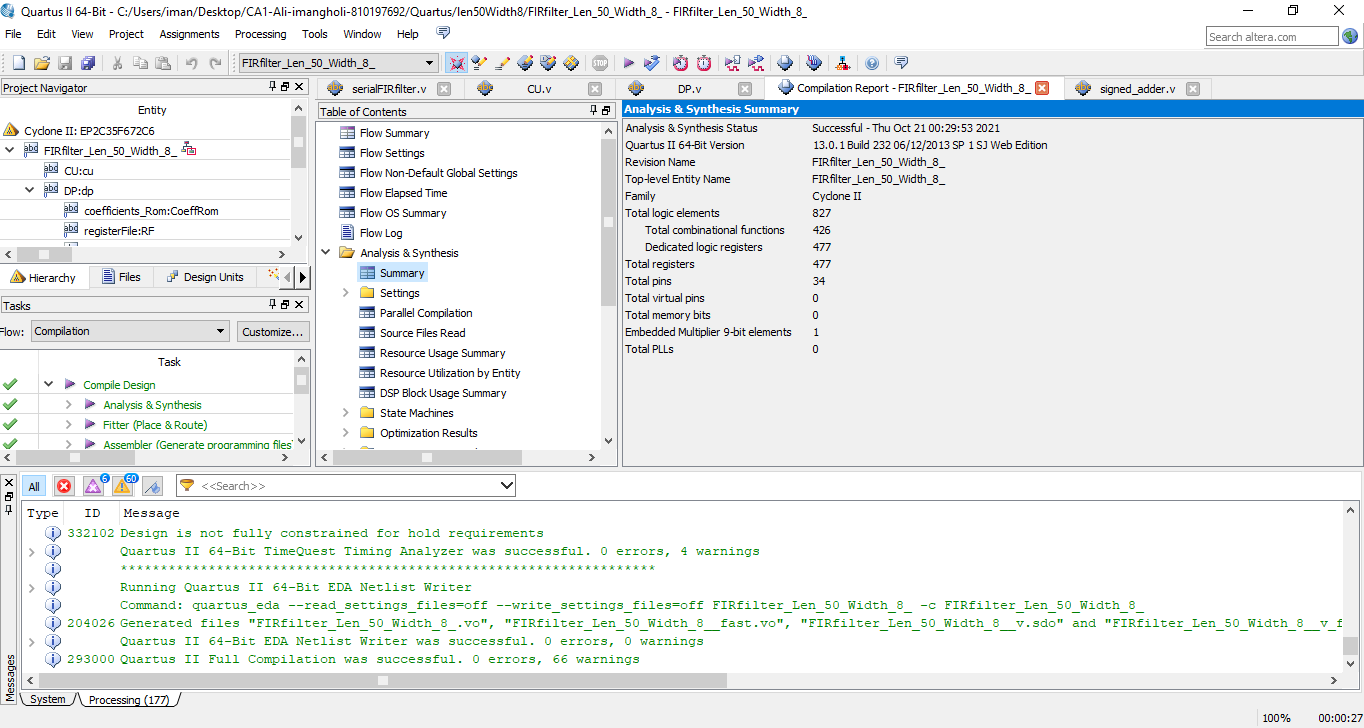
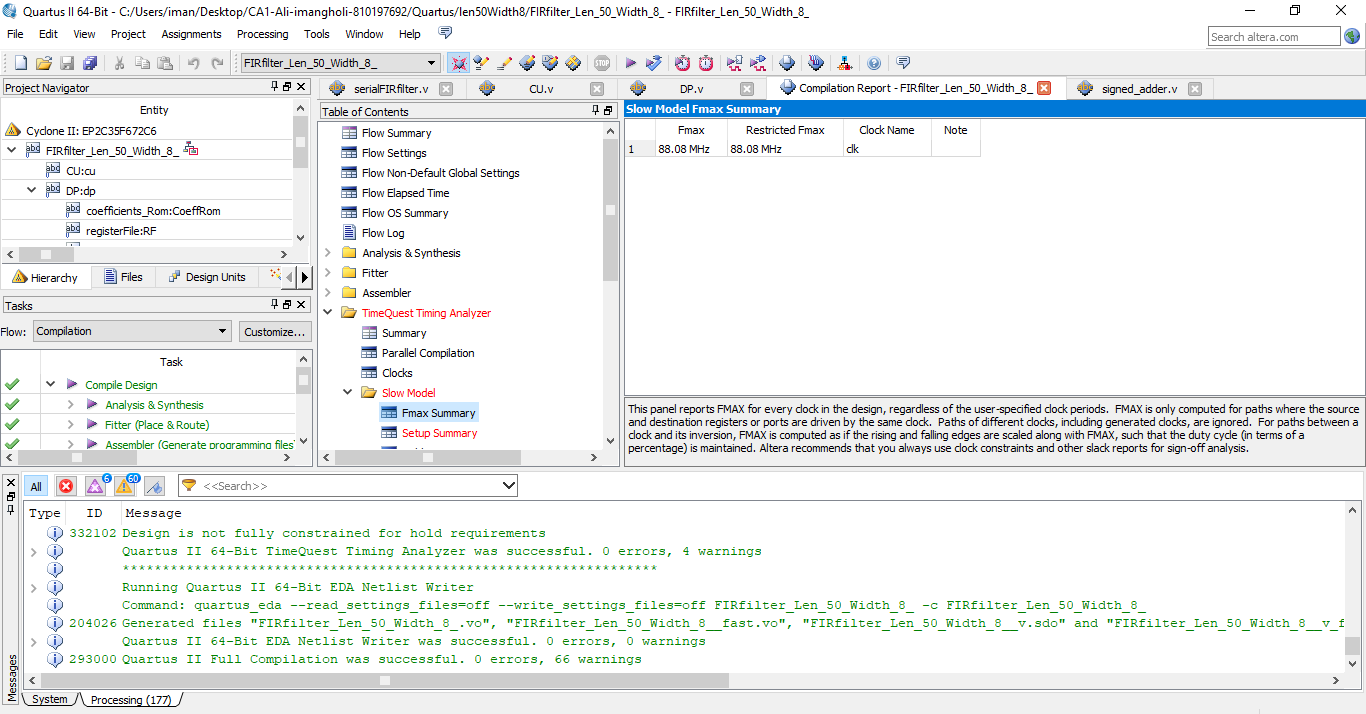
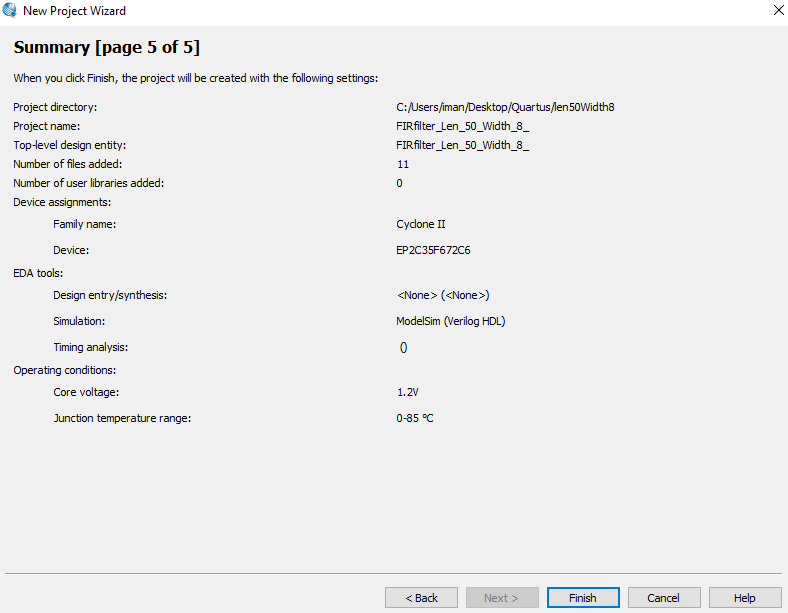
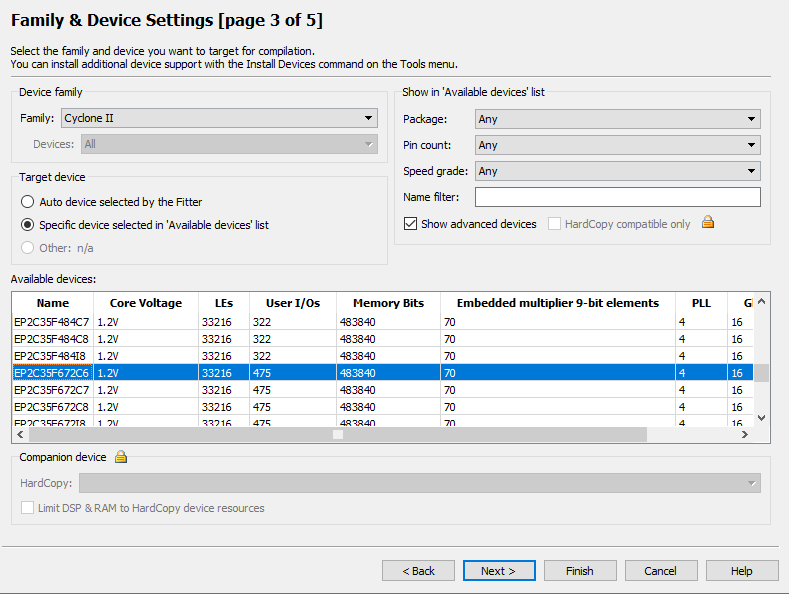


**بخش چهارم:**

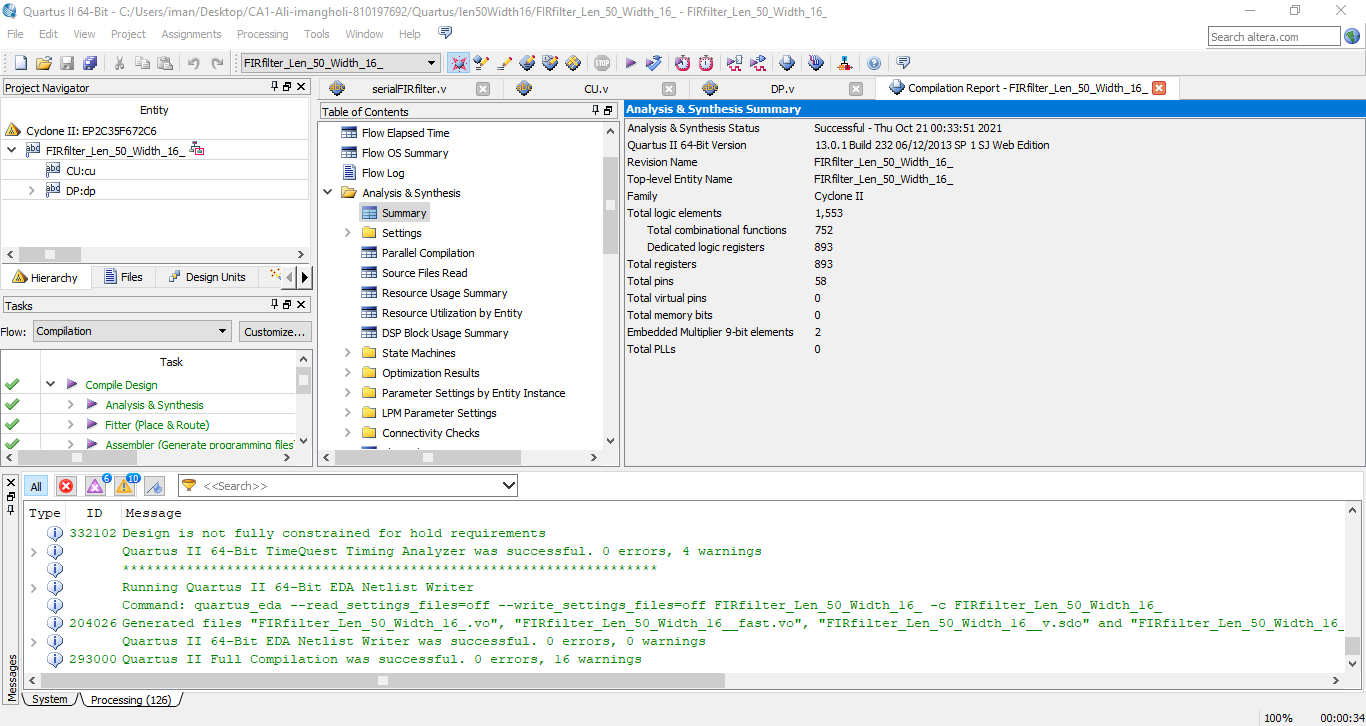
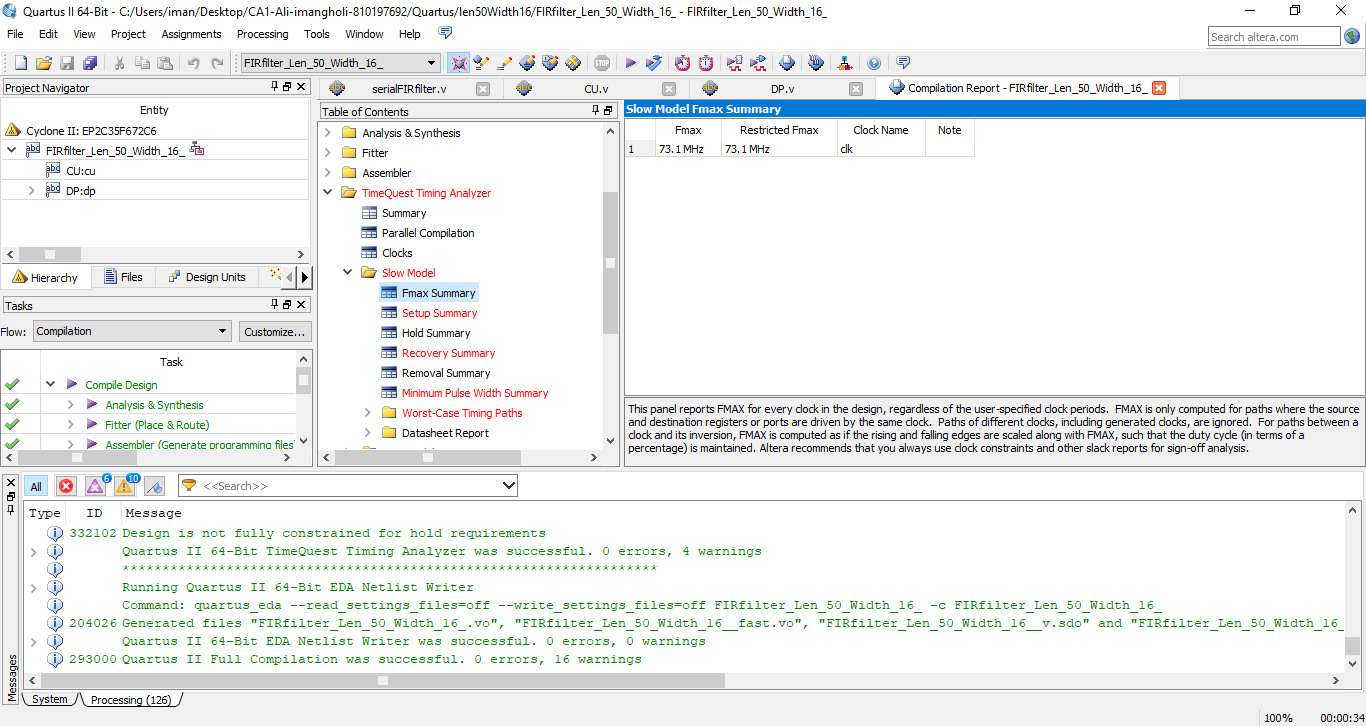
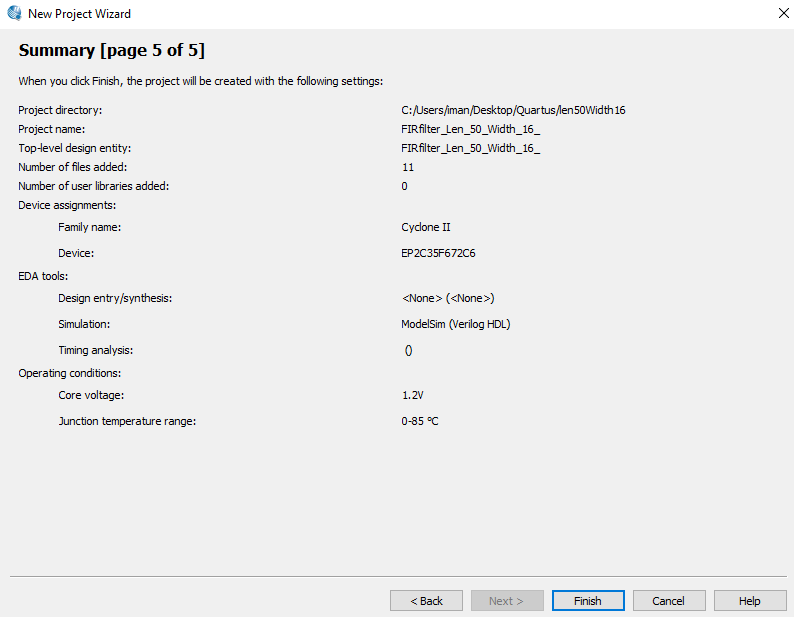
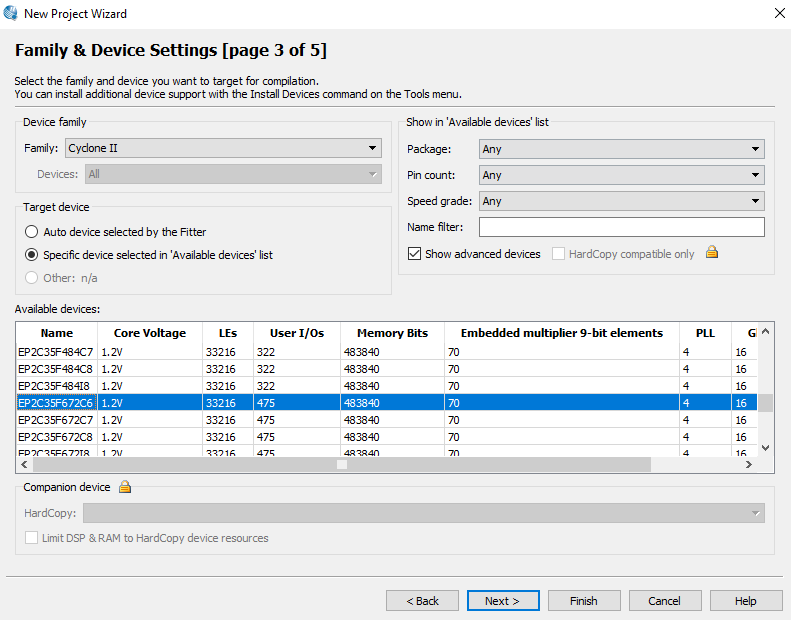
به تفکیک نتیاج برای 4 حالت خواسته شده در زیر آورده شده است:

\*\*\* برای تمامی حالت ها از خانواده ی cycloneII و مدلی که در ویدیو ی راهنمایی نشان داده شد استفاده شده است . \*\*\*

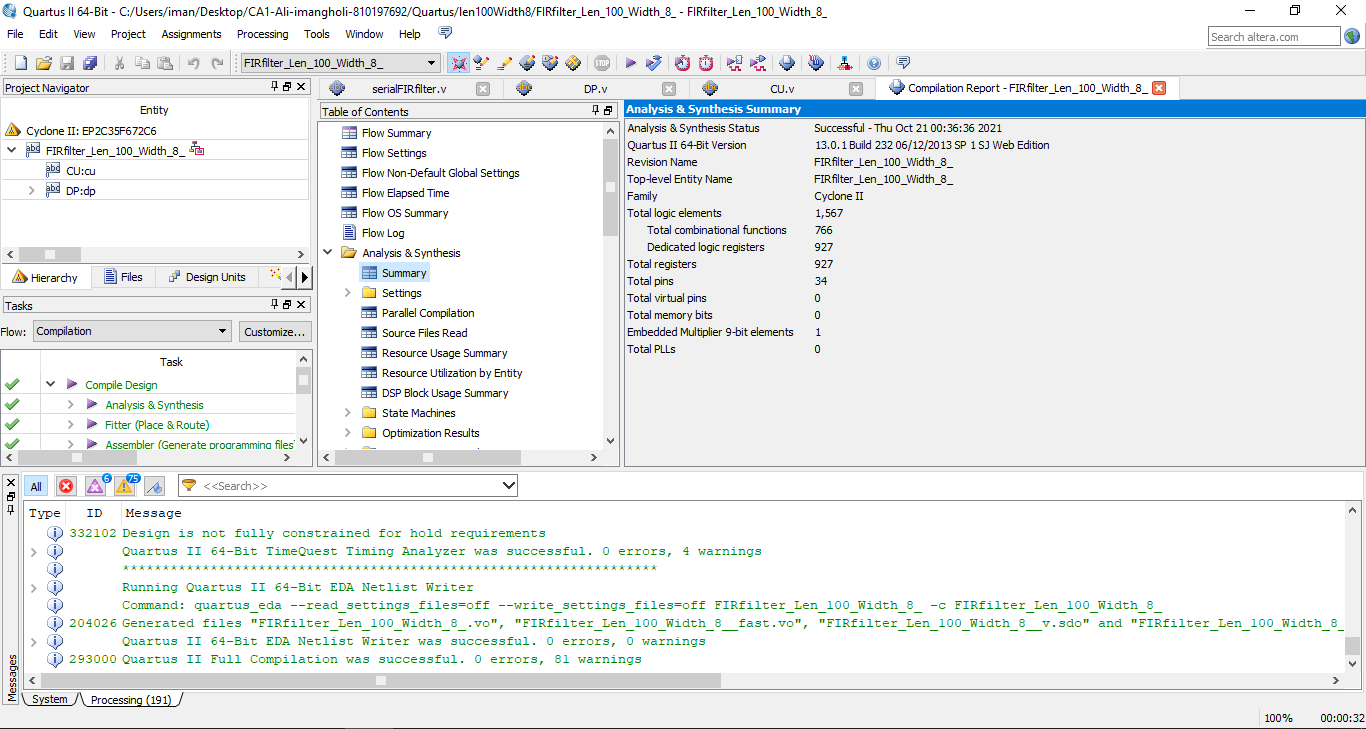
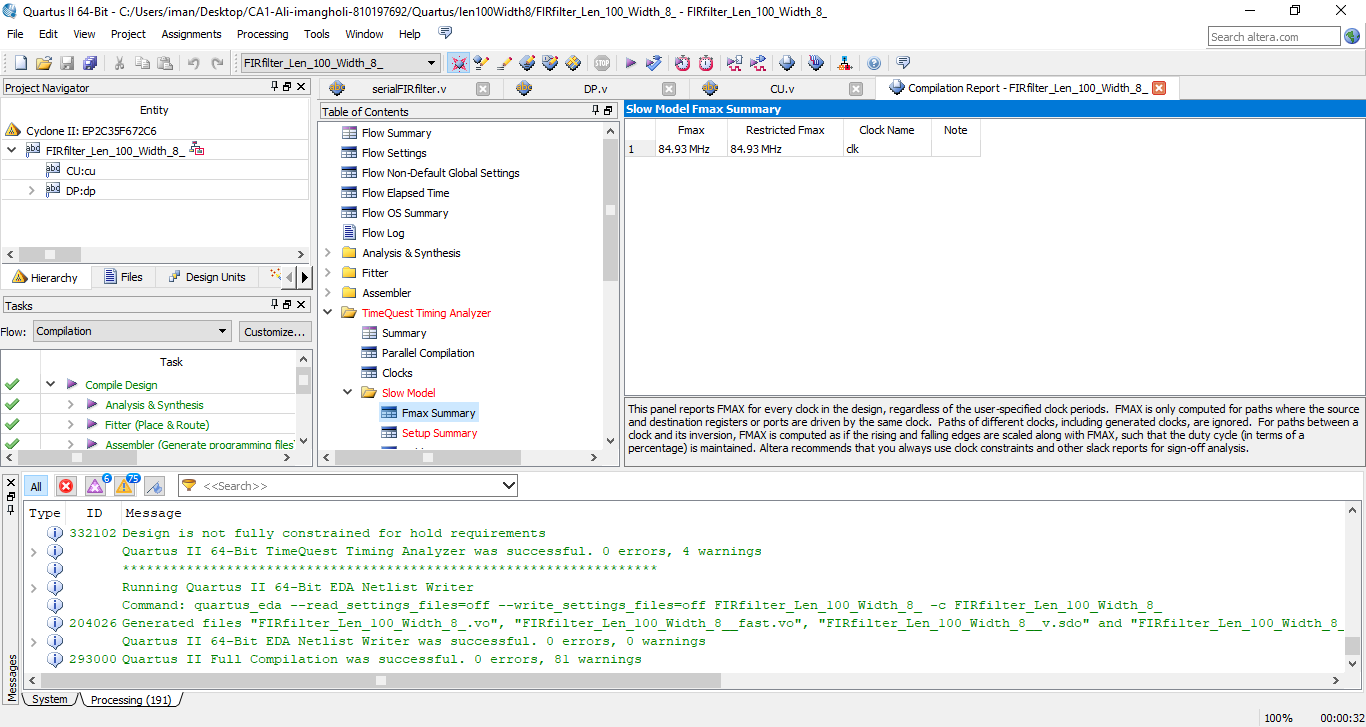
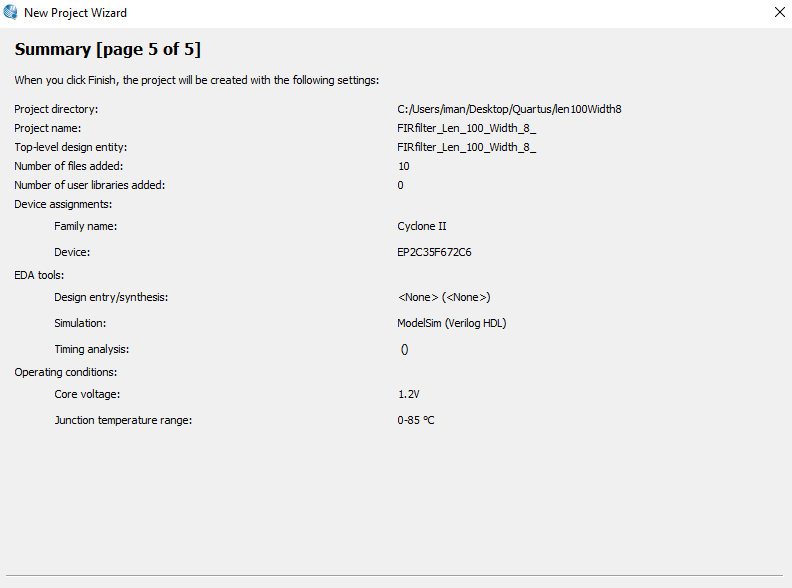
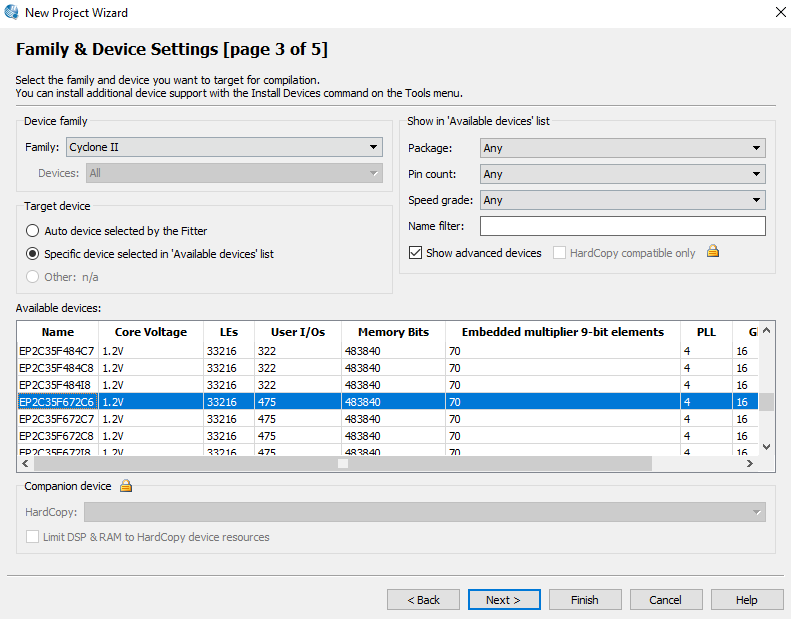
len50Width8:



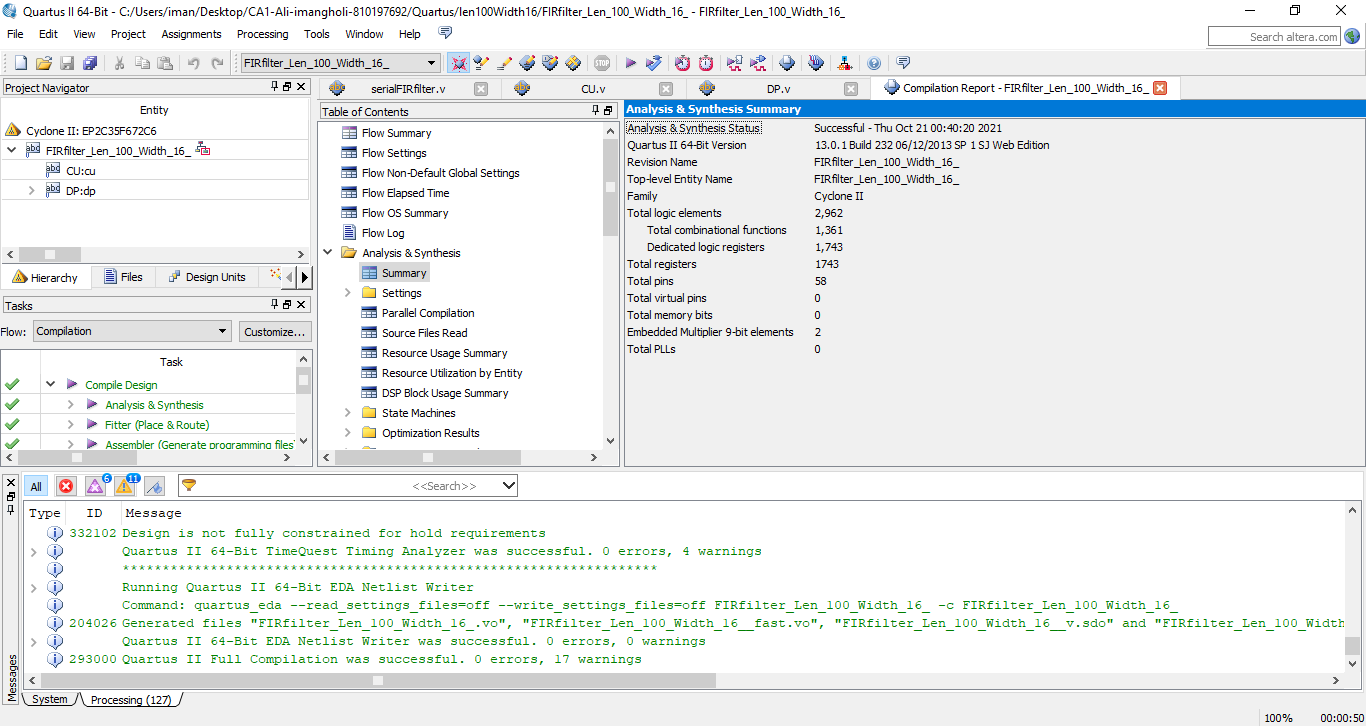
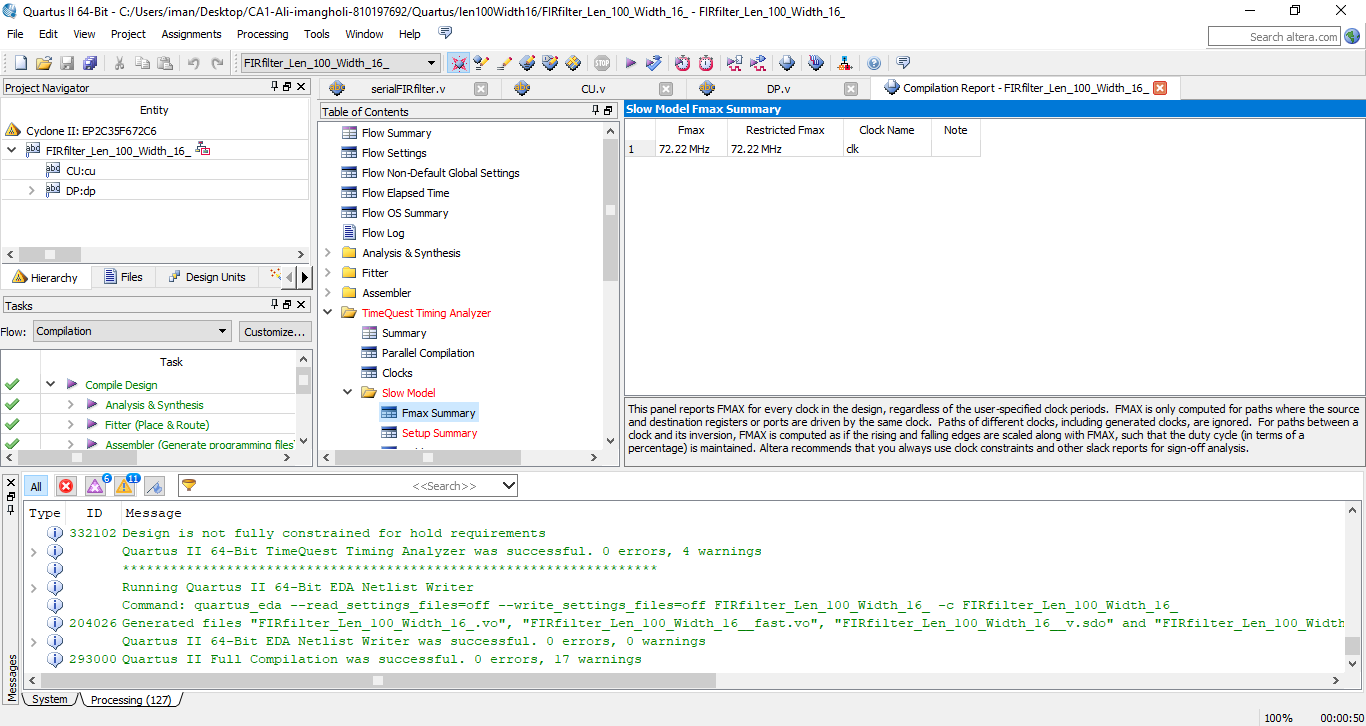
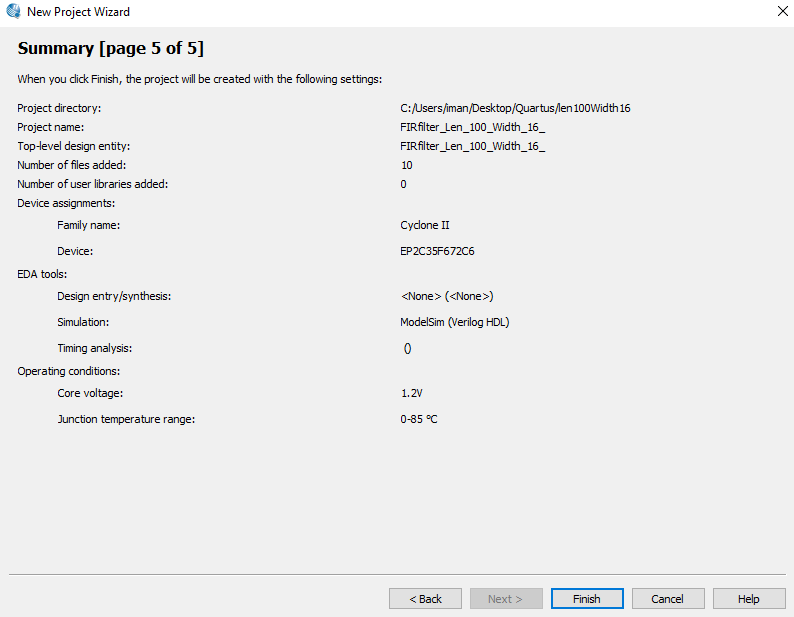
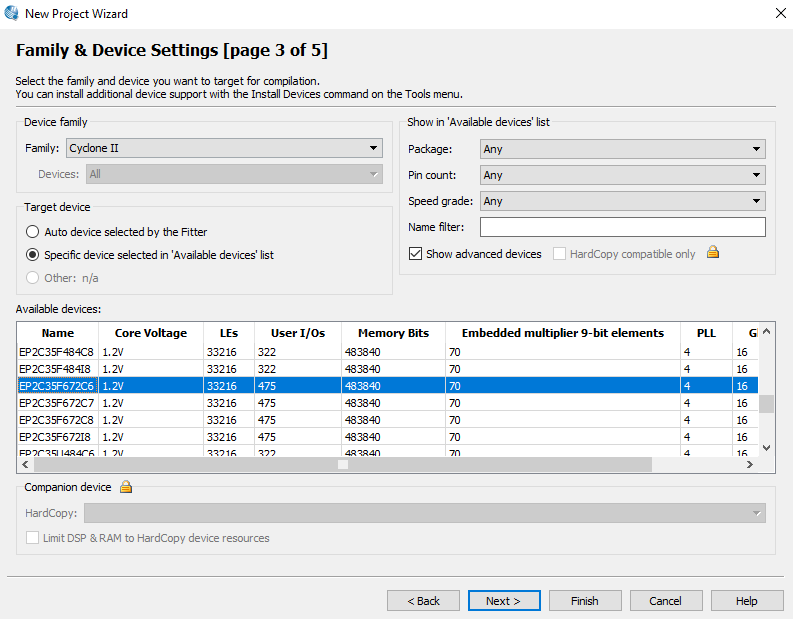
len50Width16:



len100Width8:



len100Width16:



مقایسه ی بین 4 حالت بالا:

|  |  |  |
| --- | --- | --- |
| Total registers | Total logic elements |  |
| 477 | 827 | len50Width8 |
| 893 | 1553 | len50Width16 |
| 927 | 1567 | len100Width8 |
| 1743 | 2962 | len100Width16 |

مقایسه بین حالت های بالا:

Total logic elements:

len50Width8 < len50Width16 < len100Width8 < len100Width16

همانطور که از جدول بالا مشخص است، با افزایش width و length تعداد قابل توجه ای المان منطقی به مدار اضافه می شود.

Total registers:

len50Width8 < len50Width16 < len100Width8 < len100Width16

همانطور که از جدول بالا مشخص است، با افزایش width و length تعداد قابل توجه ای رجیستر(فیلپ فلاپ) به مدار اضافه می شود.

\*\*\* با توجه به جدول بالا به صورت تقریبی، رجیستر ها (فیلپ فلاپ ها ) نصف المان های منطقی هر سطر را تشکیل می دهند.