به نام خدا



گزارشکار دستورکار شماره 1

ارائه دهندگان:

زهره ابوعلی شمشیری

امین چیت سازان

نام استاد: مهندس سید مجتبی موسوی

**مدار تمام جمع کننده تک بیتی:** مداری است که شامل سه ورودی (دو عدد تک بیتی و یک Carry input که برای دریافت و اعمال بیت Carry از مراحل قبل) و دو خروجی Sو carry output است.به عبارتی تمام جمع کننده تک بیتی سه ورودی تک بیتی A,B,C in را دریافت کرده و خروجی های sumو Cout را میدهد.در واقع این تمام تجمع کننده دو عدد تک بیتی را با هم جمع می کند.

**جدول درستی تمام جمع کننده تک بیتی:**

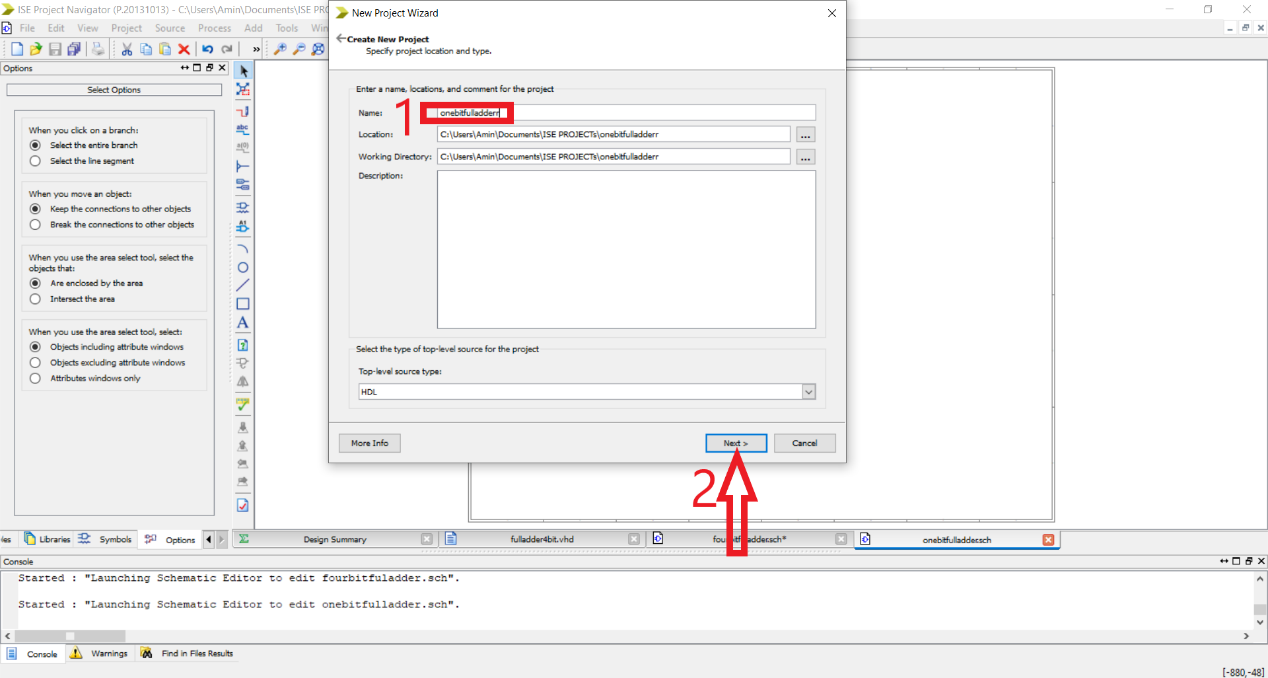
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cout** | **S** | **Cin** | **B** | **A** |
| 0 | **0** | **0** | **0** | **0** |
| 0 | **1** | **1** | **0** | **0** |
| 0 | **1** | **0** | **1** | **0** |
| 1 | **0** | **1** | **1** | **0** |
| 0 | **1** | **0** | **0** | **1** |
| 1 | **0** | **1** | **0** | **1** |
| 1 | **0** | **0** | **1** | **1** |
| 1 | **1** | **1** | **1** | **1** |

**S = A Xor B Xor Cin**

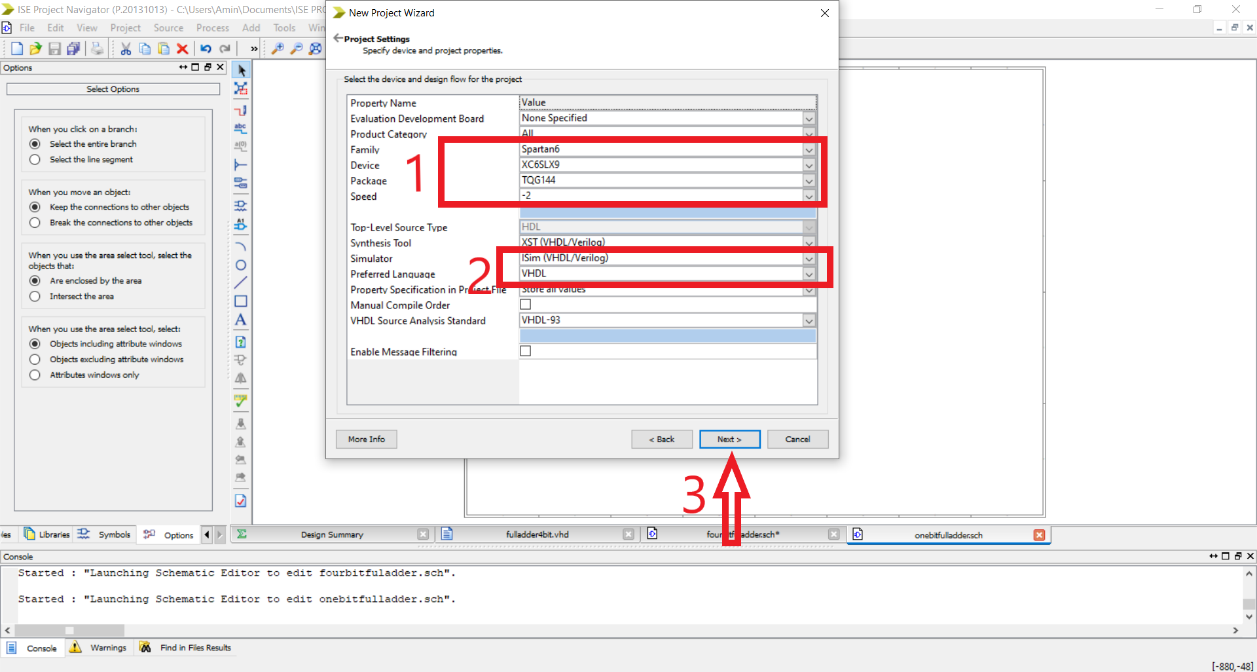
**Cout = A and B Or (A Xor B and Cin)**

**طراحی شماتیک جمع کننده تک بیتی:**

ابتدا از بخش New Project از برنامه Xilinx Ise یک پروژه جدید میسازیم و نام پروژه را مشخص میکنیم سپس تنظیمات مد نظر برنامه را اعمال میکنیم مطابق تصاویر 1و 2

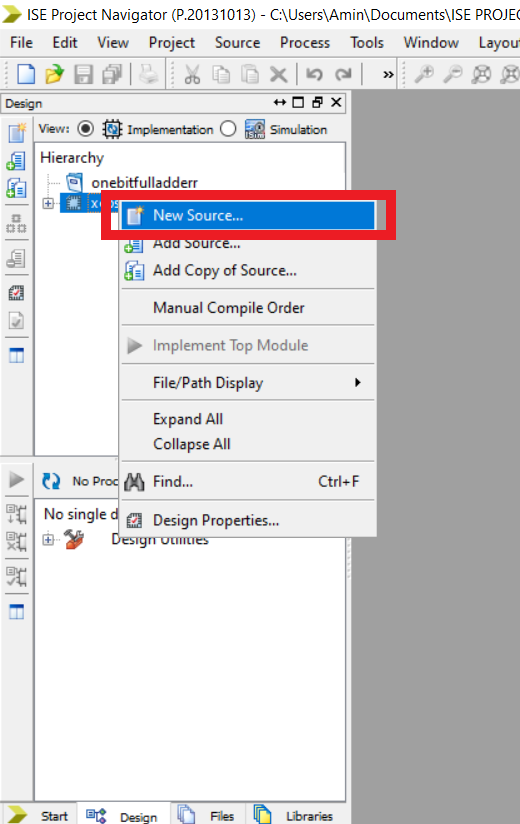


تصویر 1-ایجاد پروژه

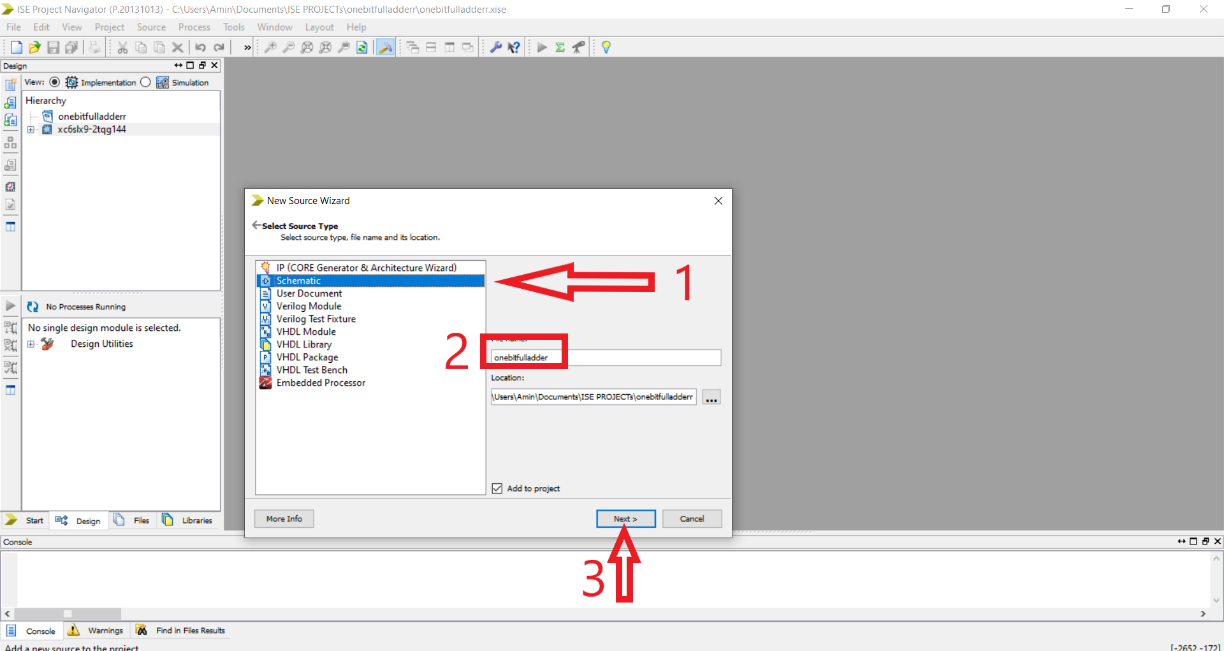


تصویر 2-تنظیمات برنامه

پس از ایجاد شدن پروژه مطابق تصاویر 3 و 4 یک فایل شماتیک ایجاد می کنیم.



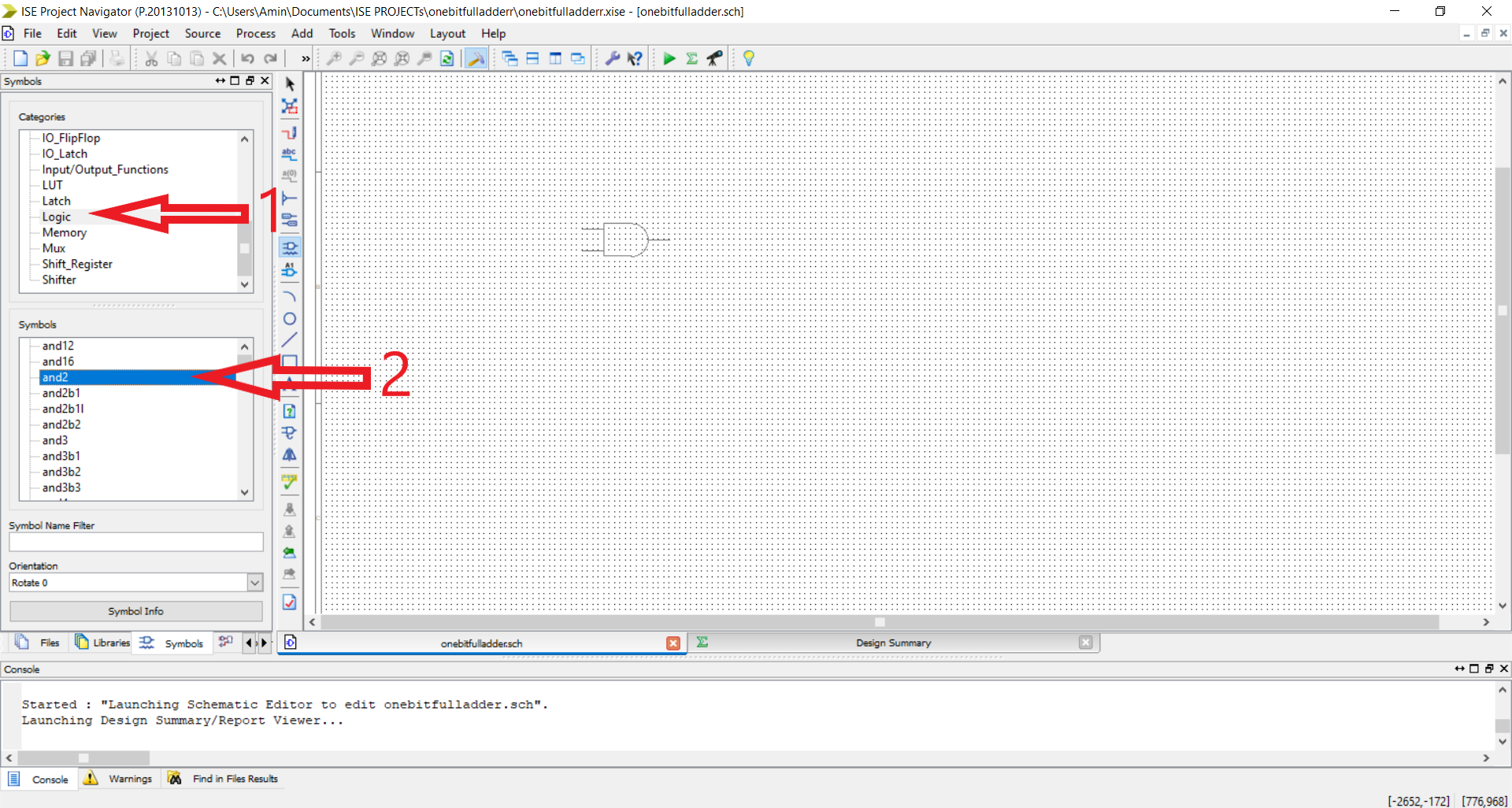
تصویر 3 - ساخت یک ماژول



تصویر 4

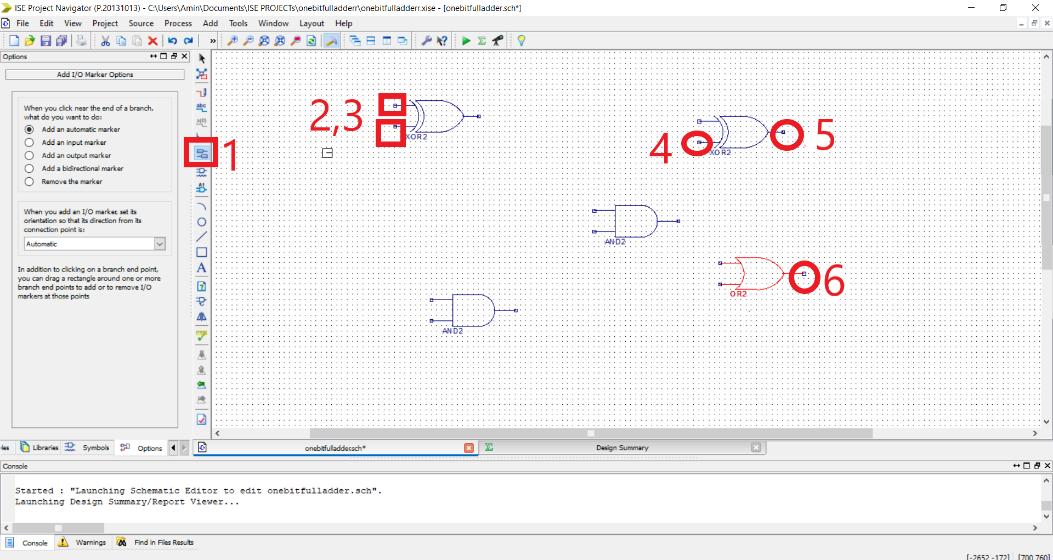
با طی این مراحل،فایل شماتیک ما ساخته شده و باید با استفاده از Symbol های از پیش تعریف شده در Xilinx،مدار خود را طراحی کنیم.(تصویر 5)

مدار تمام جمع کننده تک بیتی شامل دو گیت And، دو گیت Xor و یک گیت or است.



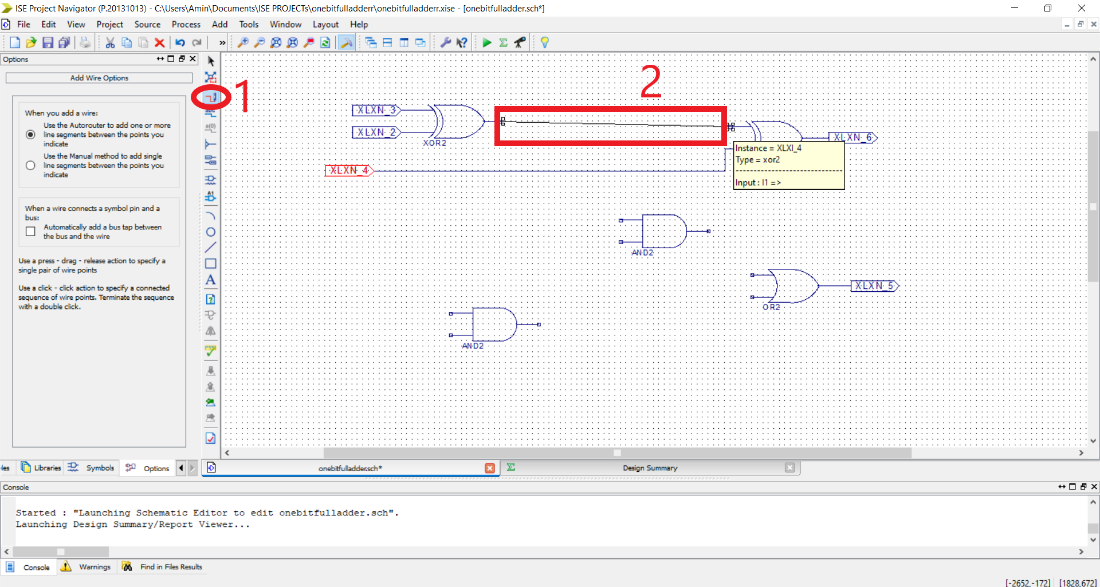
تصویر 5 -ابتدا در قسمت Categories، مدار های منطقی(logic ) را انتخاب میکنیم تا مدار های منطقی که به آن ها نیاز داریم در بخش Symbols ظاهر شوند به عنوان مثال and2 را انتخاب میکنیم که یک گیت and با دو ورودی است.سپس دو مدار Xor2 و یک مدار Or2 و یک مدار and2 نیز اضافه میکنیم.

پس از رسم گیت ها حال باید ورودی ها و خروجی ها را در مدار اعمال کنیم.



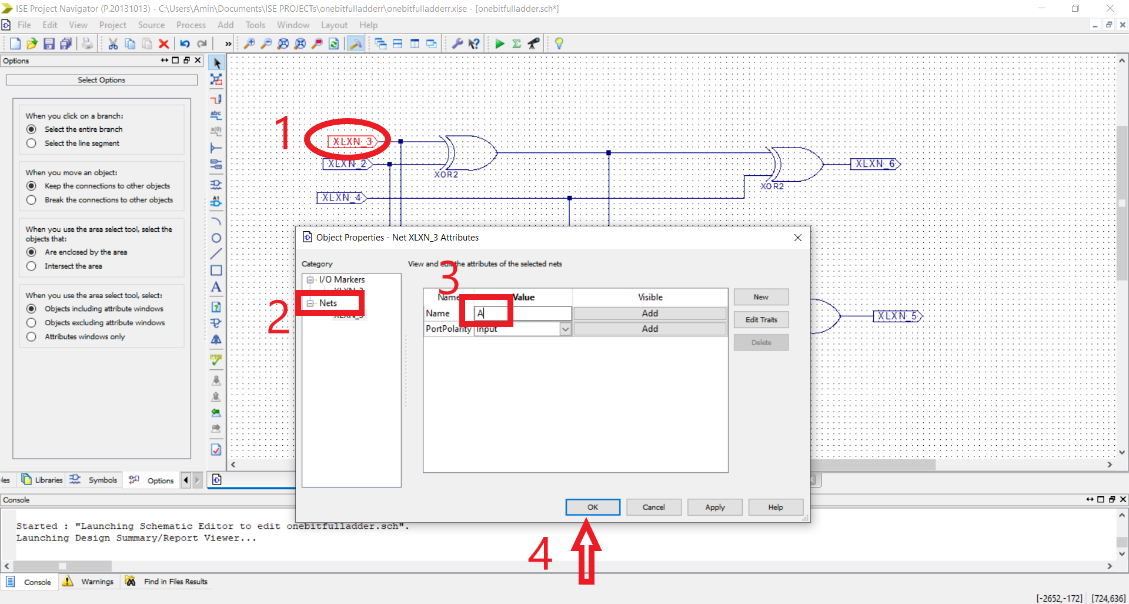
تصویر 6 - با انتخاب گزینه Add I/O marker (1) و کلیک بر ورودی های مدار Xor مورد نظر(2و3و4) و خروجی های مدار های Xor و and (5و6) ورودی و خروجی ها را اعمال میکنیم.

پس از اضافه کردن ورودی و خروجی ها به گیت های مورد نظر،باید خروجی بعضی گیت ها را به ورودی گیت های دیگر وصل کنیم به گونه ای که A با B Xor بشود و حاصل با Cin ،Xor شود و خروجی S را دهد.همچنین A با B، And شود.A,B که Xor شده بودند،حاصلشان با Cin،And شود و حاصل این دو And، با هم Or شوند .تصویر 7 نمونه ای از این عمل است.



تصویر 7 – انتخاب گزینه Add Wire (1) و وصل کردن انتهای مدار Xor به ورودی مدار Xor دیگر (2)

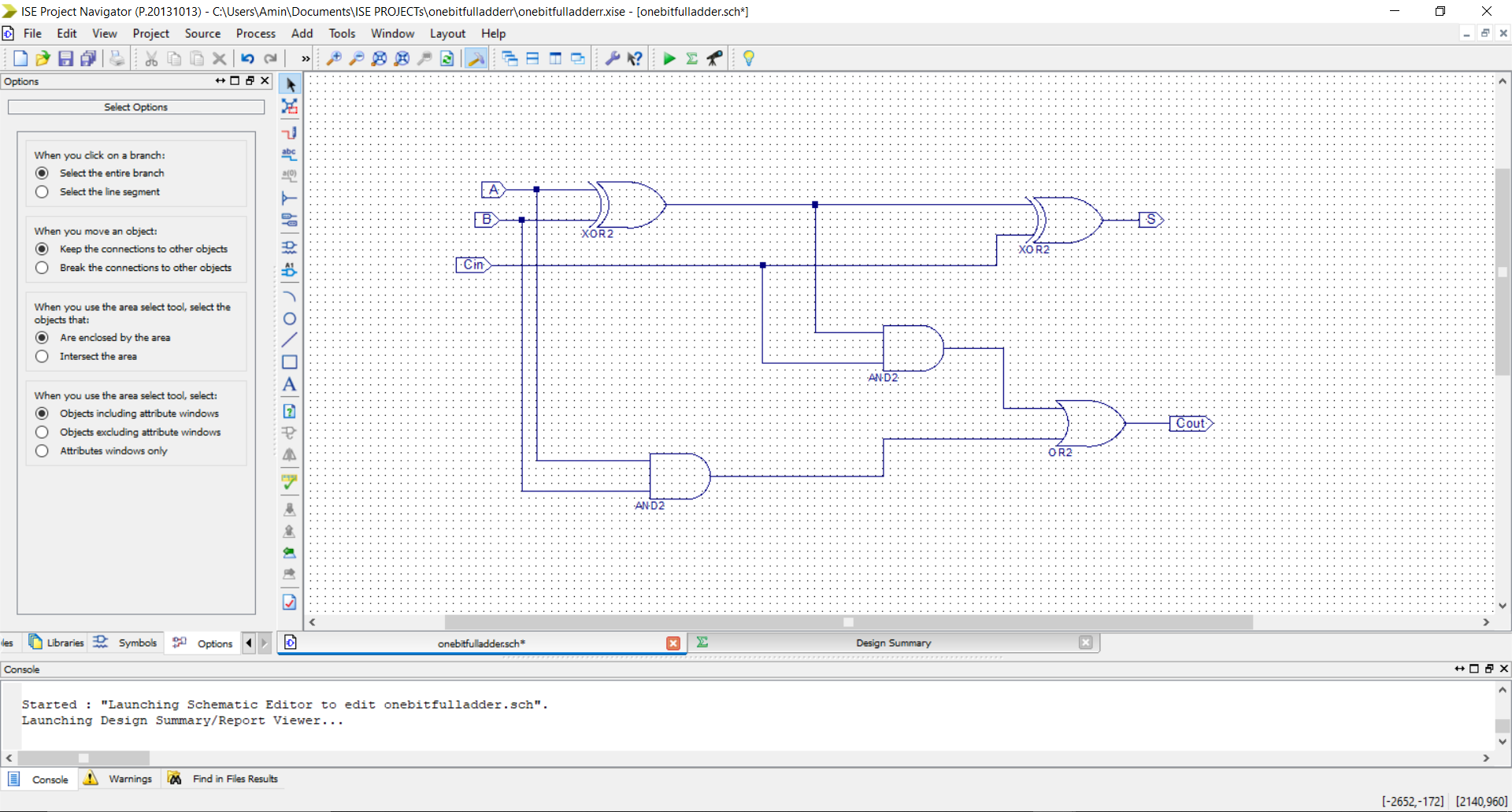
بهتر است نام ورودی ها و خروجی های مدار را تغییر دهیم برای این کار مطابق تصویر 8 عمل می کنیم.



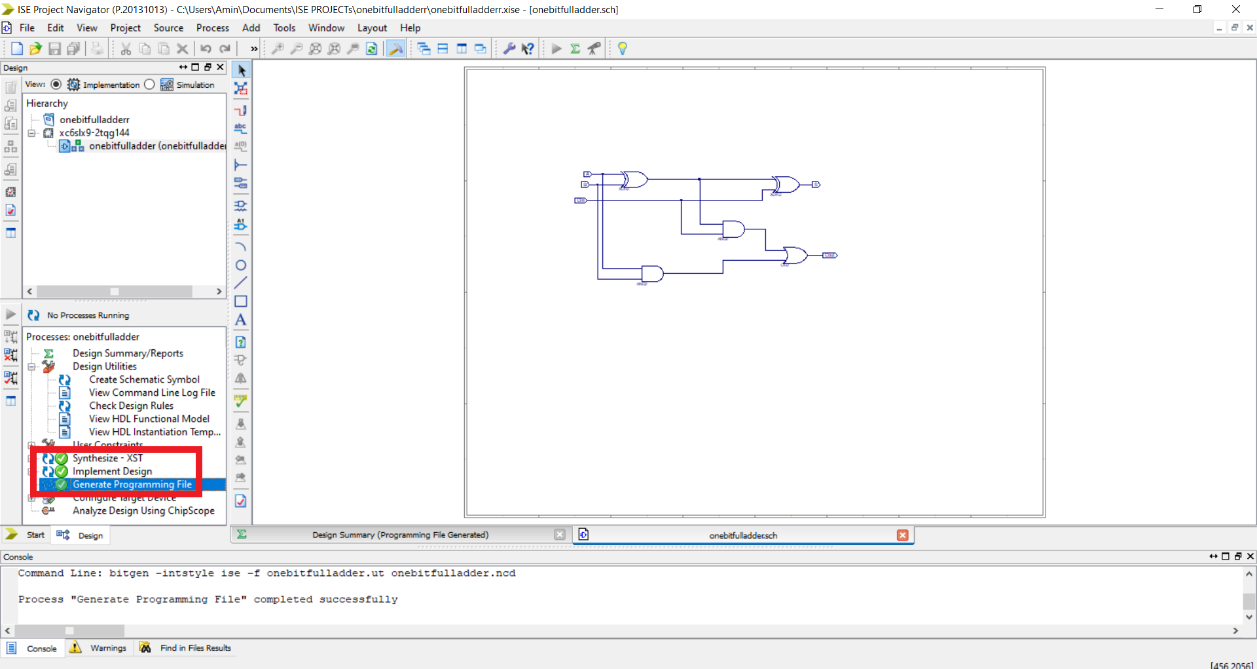
تصویر 8 –با دوبار کلیک بر روی یکی از ورودی ها(1)،به بخش Nets(2) رفته و از بخش Name(3) نام ورودی خود را به A تغییر میدهیم.

سایر ورودی ها را نیز به Bو Cin تغییر میدهیم خروجی ها را نیز به Cout , sum تغییر میدهیم.

مدار تمام جمع کننده تک بیتی ما در نهایت به شکل زیر خواهد بود.(تصویر 9)



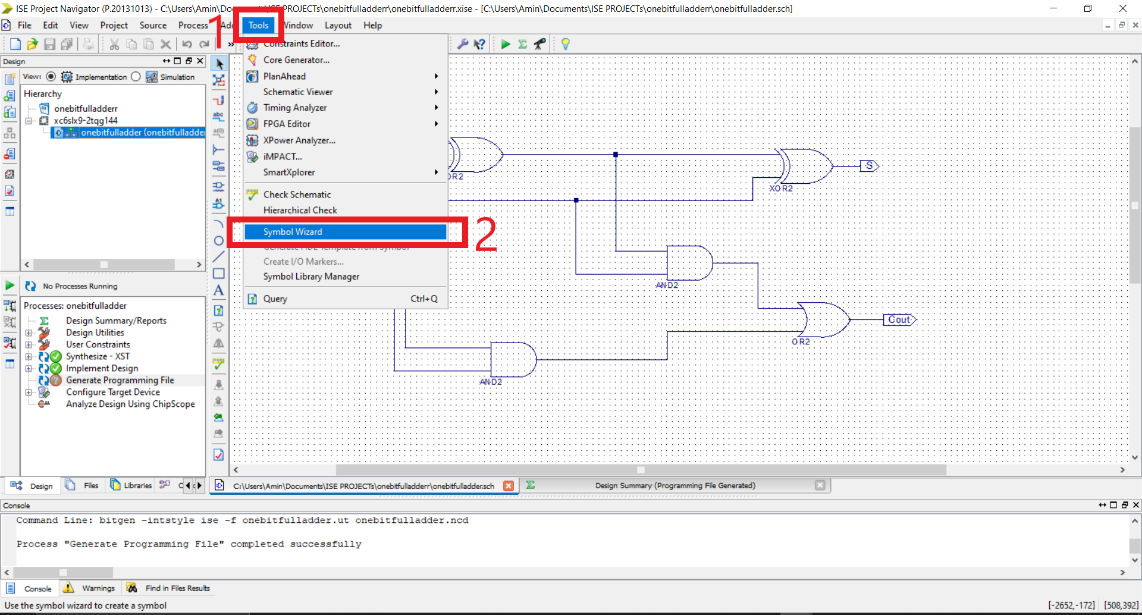
تصویر 9 – شماتیک مدار تمام جمع کننده تک بیتی



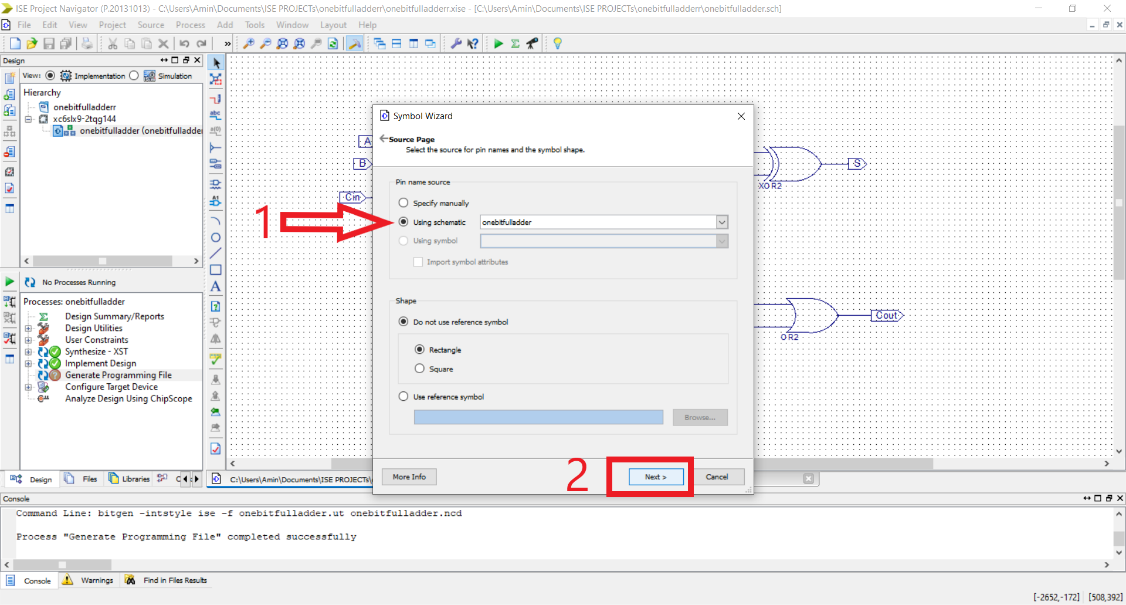
تصویر 10- سنتز و ساخت فایل پروگرام شماتیک تمام جمع کننده تک بیتی

**تمام جمع کننده 4 بیتی:** این جمع کننده، دو عدد 4 بیتی را با هم جمع می کند و ساختار آن به این صورت است که از 4 جمع کننده تک بیتی ساخته شده است به صورتی که Cin بعضی جمع کننده های تک بیتی، Coutجمع کننده قبلی است . شماتیک تمام جمع کننده 4 بیتی به صورت زیر است.

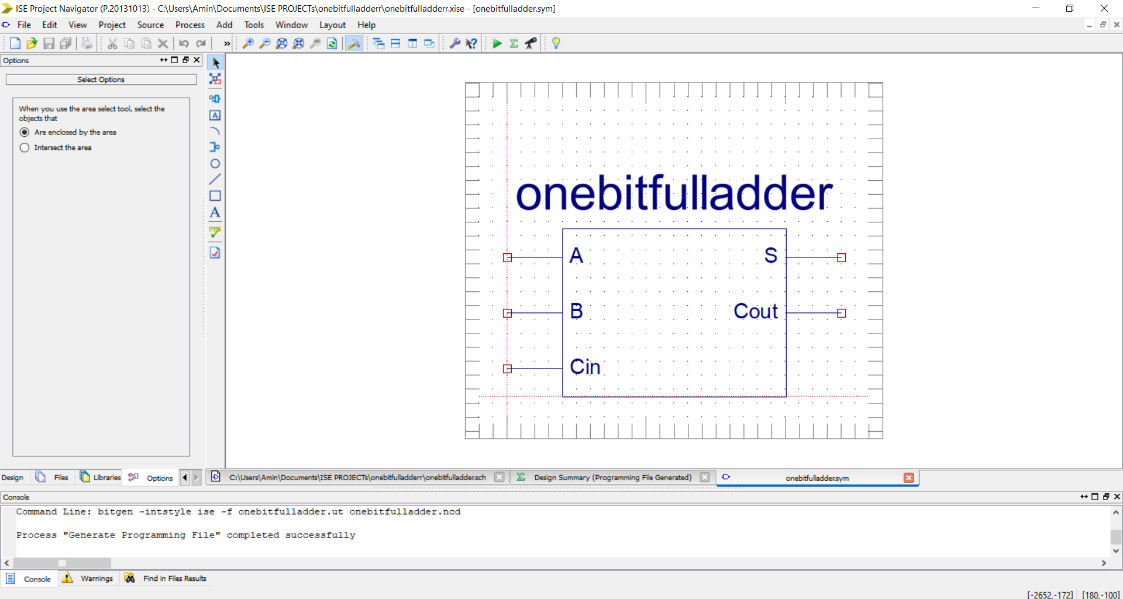
**نحوه ساخت شماتیک تمام جمع کننده 4 بیتی:** شماتیک مدار تمام جمع کننده تک بیتی را به عنوان یک ماژول در کتابخانه Xilinx اضافه میکنیم.مطابق تصاویر 11و12



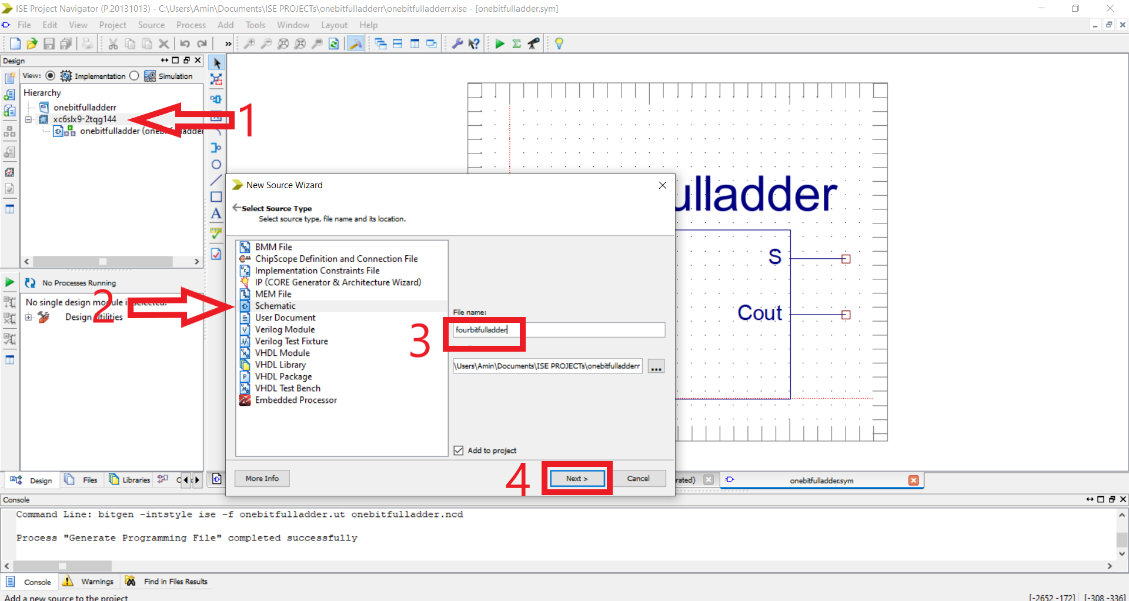
تصویر 11



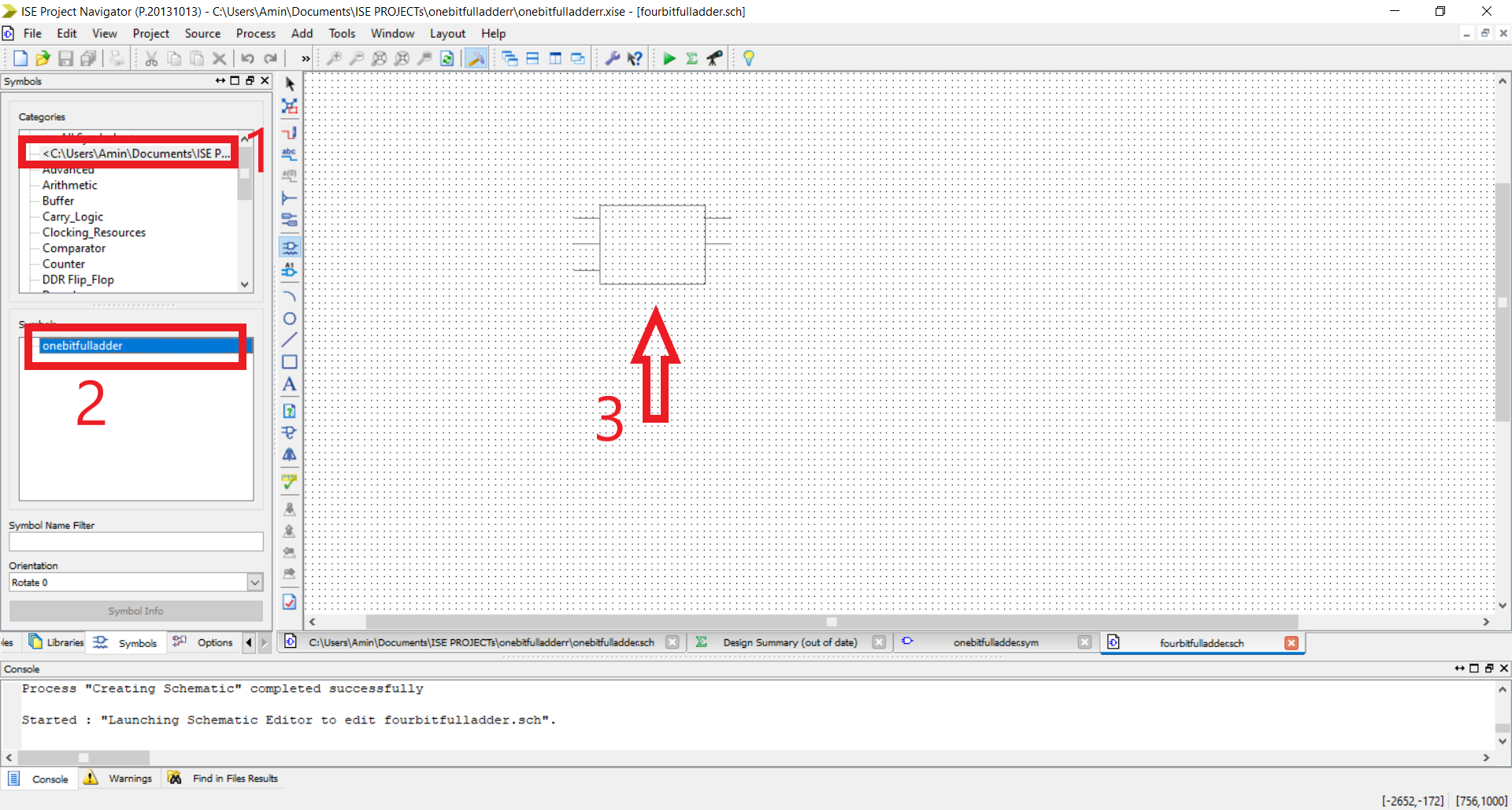
تصویر 12



تصویر 13-ماژول تمام جمع کننده تک بیتی ذخیره شده به عنوان یک کتابخانه در Xilinix

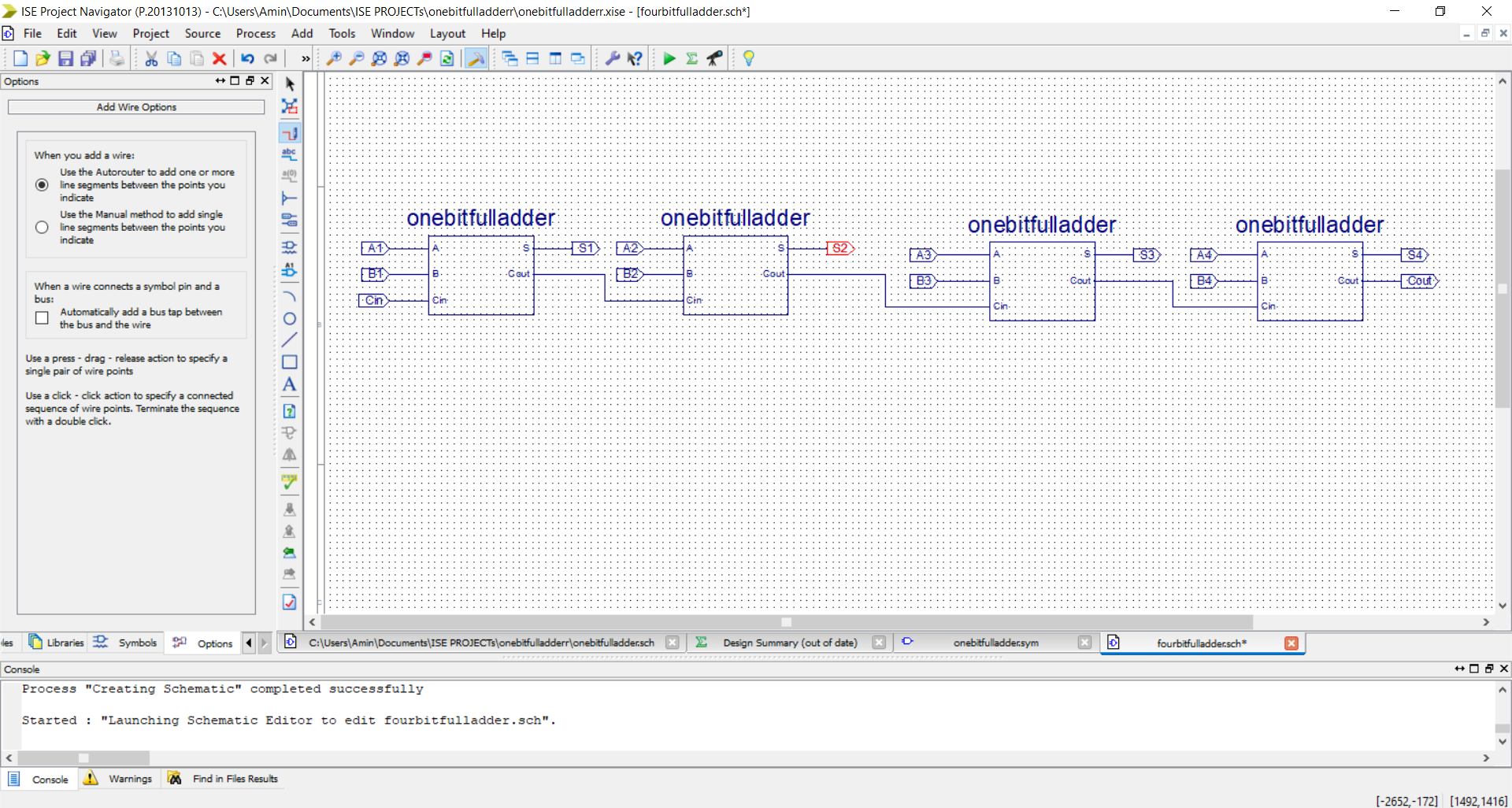


تصویر 14-با کلیک راست برروی (1) و گزینه New Source گزینه شماتیک را انتخاب کرده و نام ماژول جدید خود را تعیین و به مرحله بعد می رویم

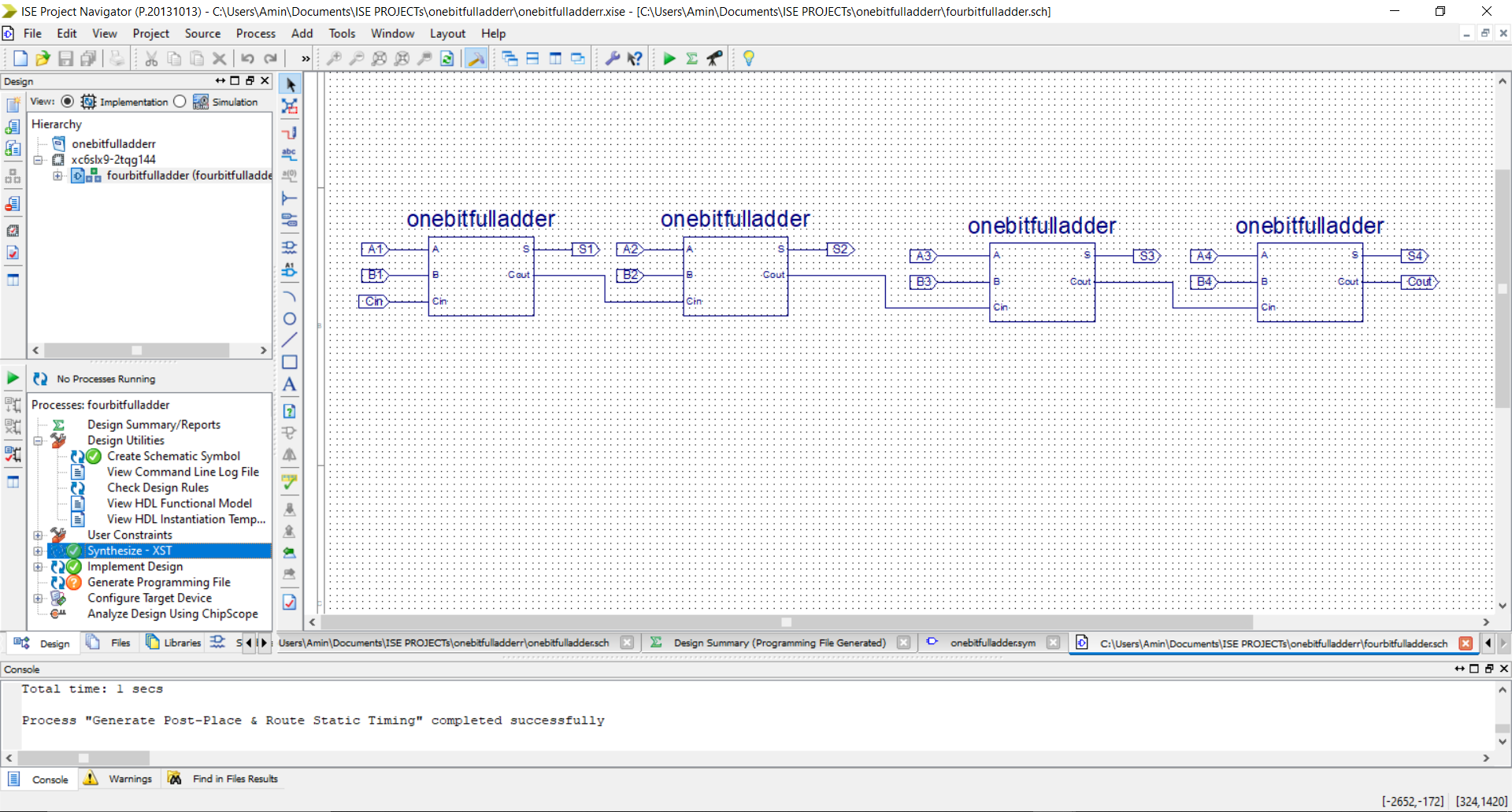


تصویر 15 – انتخاب تمام جمع کننده تک بیتی از کتابخانه زایلینکس

با استفاده از 4 تمام جمع کننده تک بیتی،یک تمام جمع کننده 4 بیتی ساخته می شود. شماتیک نهایی تمام جمع کننده 4 بیتی به صورت زیر است.



تصویر 16



تصویر 17 – سنتز شماتیک تمام جمع کننده 4 بیتی

پاسخ به سوالات)

1) ذخیره کردن داده ها در ماژول Mojo-V3:

به طور کلی،برای پروگرام کردن ماژول Mojo از یک پورت USB و یک میکروکنترلری که در خود ماژول تعبیه شده است استفاده میشود.اما FPGA ها به طور معمول نمیتوانند مقدار سوئیچ ها را در خود ذخیره کنند. برای رفع این مشکل یک حافظه فلش در ماژول قرار داده شده است و بعد از هر بار روشن و خاموش شدن FPGA مقادیر مورد نیاز را لود می کند.در برنامه Mojoloader نیز با انتخاب گزینه Store to flash میتوان داده را در حافظه فلش ذخیره کرد.

2) ساختار دو ترانزیستور برنامه ریزی FPGA و CPLD:

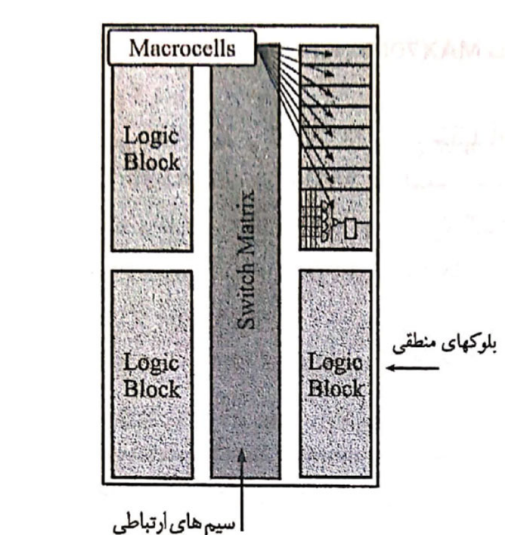
FPGA سه بلوک اصلی دارد و از مجموعه ای از بلوک های منطقی و فلیپ فلاپ ها تشکیل شده است.

الف)بلوک های منطقی قابل برنامه ریزی CLB: در هر کدام از این بلوک ها میتوان هر تابع منطقی پیاده سازی کرد.

ب) بلوک های ورودی و خروجی(IOB): برای فراهم کردن دسترسی ورودی و خروجی به یکی از خروجی های FPGA استفاده می شود.

پ)سیم های اتصال برای اتصالات: این بخش شامل سیم ها و سوئیچ های قابل برنامه ریزی است.

CPLD ها دارای بخش های متعددی مانند آرایه های تماما برنامه پذیر و گیت های AND,OR و بلوک های ماکروسل،بلوک های کنترل IO و حافظه میباشد.مهم ترین بلوک CPLD ها ماکروسل است که امکان ایجاد انواع مدار های ترکیبی و ترتیبی و فیدبک های بین آن ها را از طریق اتصالات میانی بوجود می آورد.



معماری FPGA ها مبنی بر اساس تعداد بسیار زیادی CLB است.به دلیل این اختلاف در معماری ها،CPLD دارای سرعت بالاتر و تاخیر قابل پیش بینی هستند.

تعداد بلوک های I/O و گیت های تشکیل دهنده FPGA به مراتب بیشتر از CPLD است.

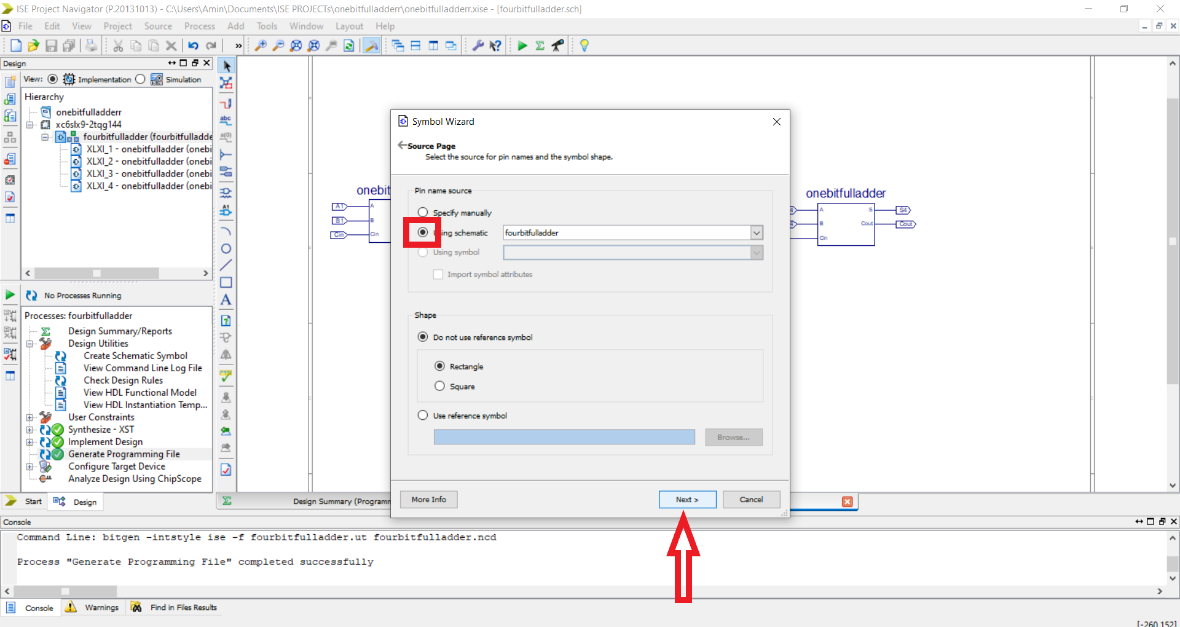
حافظه برنامه ریزی CPLD ها اکثرا براساس EEPROMو non-Volatile است بنابراین با یک بار برنامه ریزی،با قطع برق دستگاه ،نیاز به برنامه ریزی مجدد نیست.

FPGA ها بر اساس RAMهستند در نتیجه با هر بار قطع برق،نیاز به برنامه ریزی مجدد دارند و در نتیجه زمان بوت آنها بیشتر از CPLD است

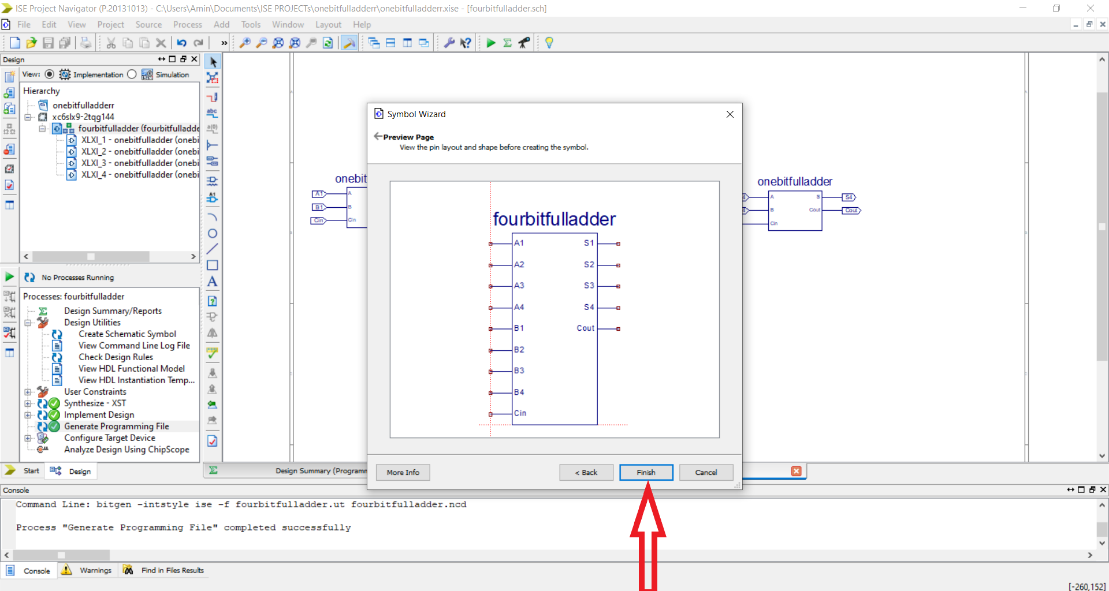
3)نحوه اضافه کردن شماتیک تک بیتی و 4 بیتی به کتابخانه زایلینکس:

برای اضافه کردن مدار طراحی شده به کتابخانه زایلینکس باید از تب tools و بخش Symbol Wizards اقدام کرد. اضافه شدن مدار تمام جمع کننده تک بیتی در تصاویر 11 تا 13 انجام شده است.

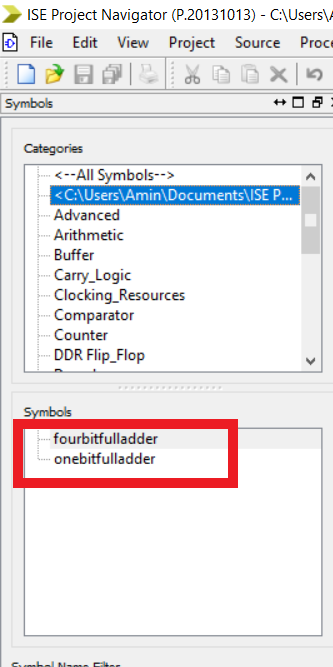
برای اضافه کردن مدار تمام جمع کننده 4 بیتی مانند تمام جمع کنده تک بیتی عمل میکنیم.



تصویر 18



تصویر 19



تصویر 20