

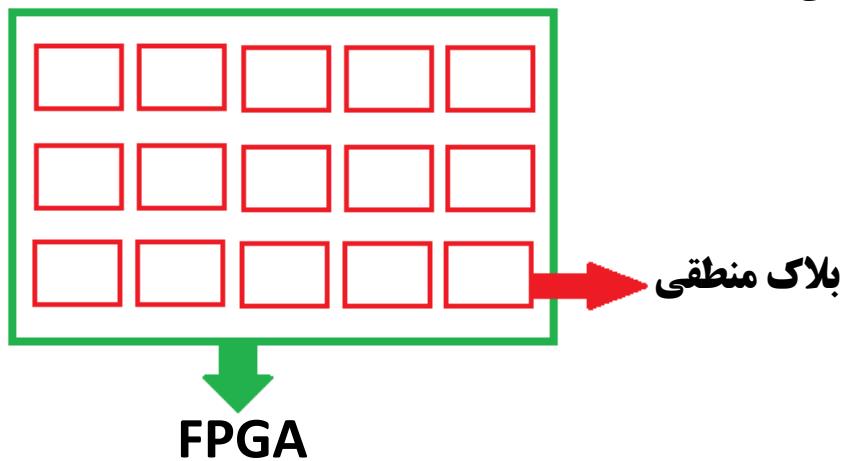
FPGA (Field Programmable Gate Array)**(مدار مجتمع دیجیتال قابل برنامه ریزی)**

* FPGA ها تراشه های نیمه رسانایی هستند که از تعداد زیادی جزء کوچک الکترونیکی به نام بلوک منطقی (Logic cell) یا سلول منطقی (Logic Block) تشکیل شده اند.

* FPGA ها بین مدارات نطقی و میکروپروسسور ها و CPU ها هستند.

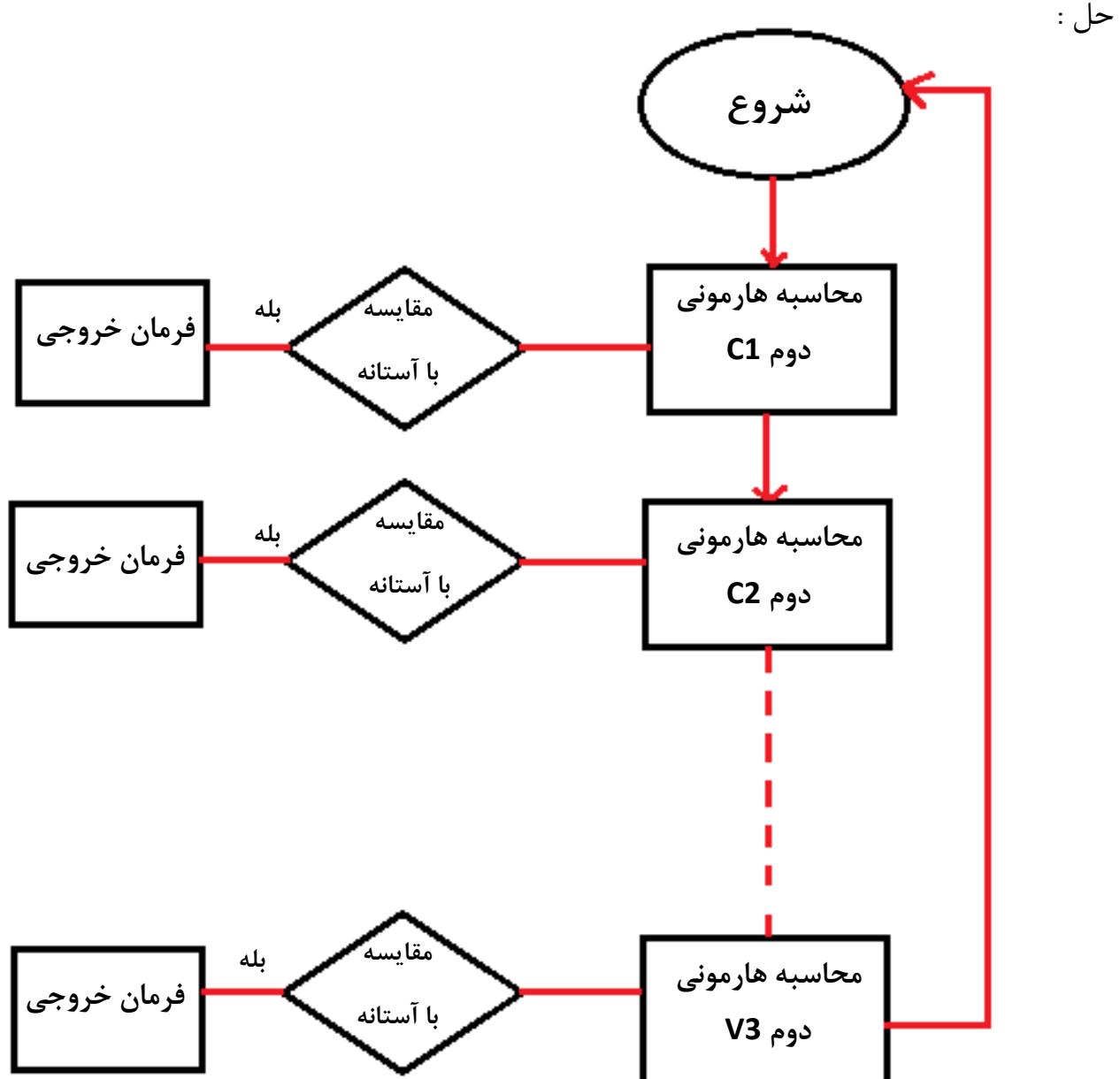
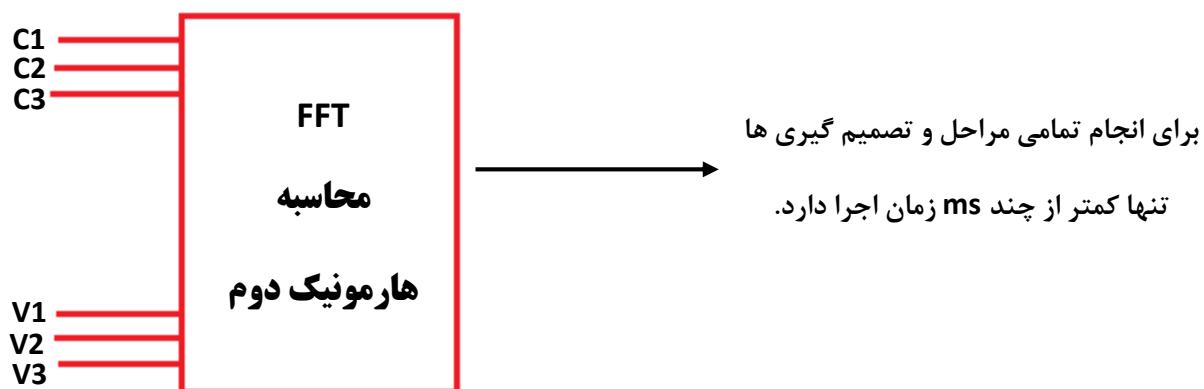
* کوچک ترین جزء سازنده CPU ها یا میکروپروسسور ها المانهای الکترونیکی به نام گیت منطقی هستند. این گیت ها ساده ترین اعمال ریاضی مانند AND و OR را انجام می دهند. از همین المان ها برای اعمال ریاضی مانند جمع و ضرب و ... استفاده می شود. در نهایت مجموعه بزرگی از این گیت ها برای کار های پیچیده استفاده می شود.

* در CPU ها و میکروپروسسور ها ارتباط بین این گیت های منطقی به طور کامل و طبق یک معماری مشخص ساخته می شود و قابل تغییر نیست.

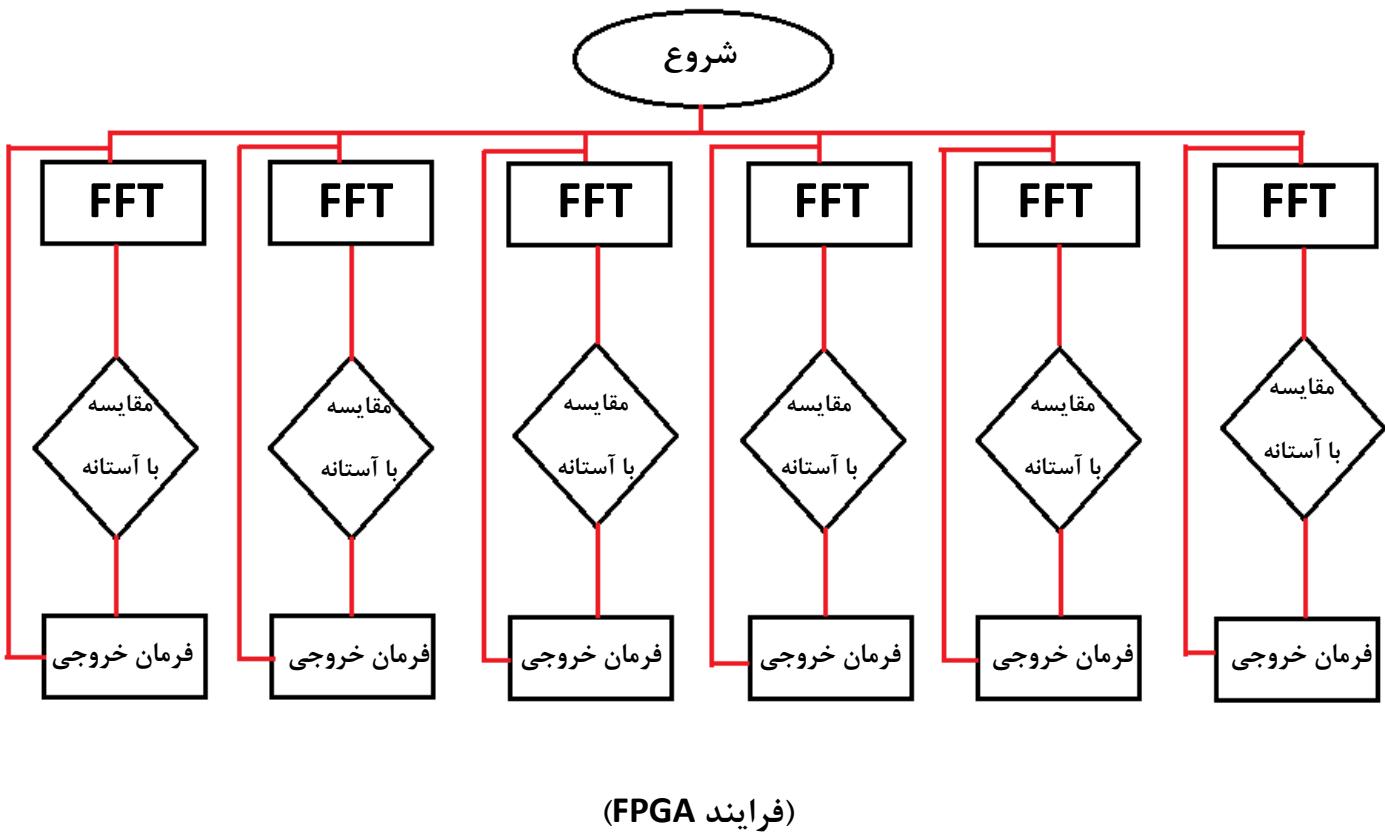


* اصلی ترین تمایز FPGA با CPU و میکرو پروسسور این است که FPGA یک تراشه با ساختار خام است. در FPGA کاربر میتواند ساختار، معماری و نحوه ارتباط بین گیت های منطقی را تعیین کند.

مثال) فرض کنید یک سیستم شامل ۶ ورودی (۳ ورودی جریان و ۳ ورودی ولتاژ) باید با استفاده از FFT مقدار هارمونیک دوم هر سیگنال استخراج شده و در صورت بالا بودن این مقدار از یک آستانه مشخص یک فرمال به خروجی ارسال کند. فرایند این کار را در یک میکرو پرسسور ARM و یک FPGA رسم کنید.



(فرایند میکرورپرسسور)



مزایای FPGA : ۱. سرعت بالا

۲. پردازش موازی

۳. انعطاف پذیر بودن

۴. هزینه و زمان توسعه متوسط

معایب FPGA : ۱. تک منظوره بودن

۲. طراحی پیچیده و زمانبر بودن

۳. قیمت بالا

: نحوه برنامه ریزی FPGA

VHDL : توسط وزارت دفاع آمریکا توسعه یافته است.

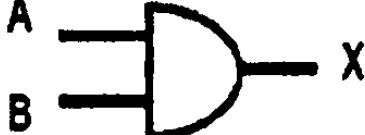
Verilog : توسط مهندسین IEEE استاندارد شده است.

نرم افزار شبیه سازی و پیاده سازی Verilog

- ❖ QUARTUS II
- ❖ Model sim

- یادآوری:

جدول زیر جهت یاد آوری گیت های منطقی که در این درس نیاز داریم.

نام	نمودار	تابع جبری	جدول درستی															
AND		$X = A \cdot B$ or $X = AB$	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$X = A + B$	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$X = A'$	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	
XOR Exclusive OR		$X = A \oplus B$ or $X = A'B + AB'$	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

* یک برد FPGA مجموعه ای از گیت های منطقی است که به نحو خاصی که ما تعیین میکنیم با هم کار میکنند.

* در FPGA می توان عنصر تشکیل دهنده را در نظر گرفت:

1. جدول صحت

2. فلیپ فلاپ

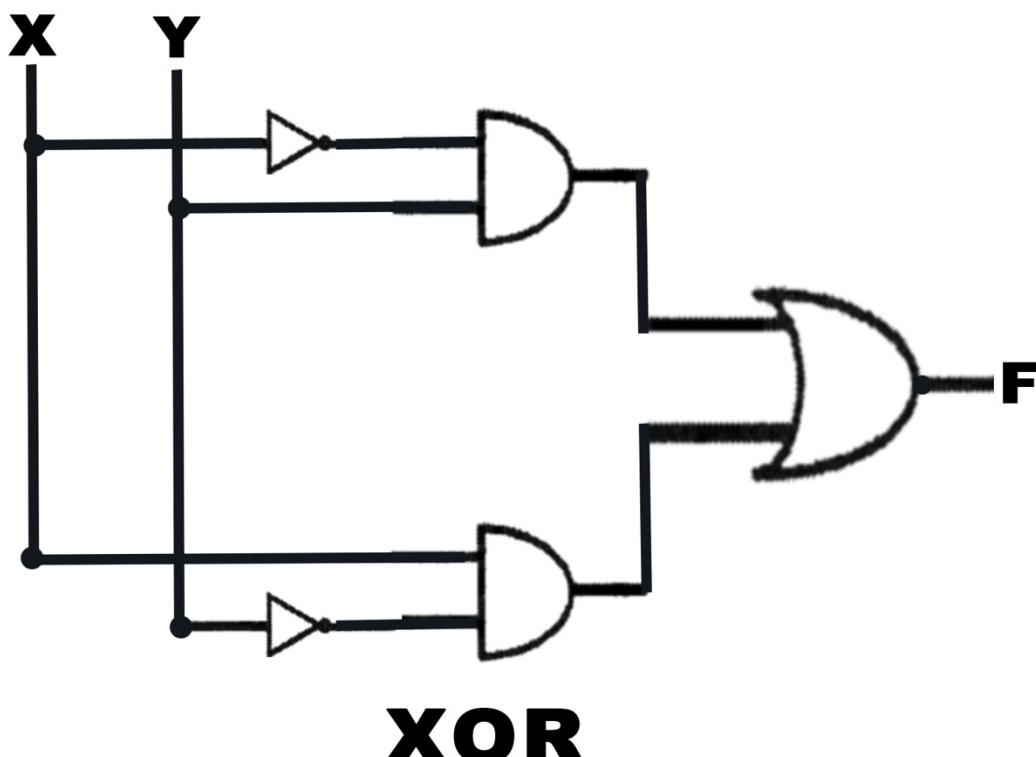
3. ماتریکس Routing

- جدول صحت:

این جدول تعیین میکند که عملکرد هر جزء FPGA چه باشد. هر LUT تعدادی ورودی و یک خروجی دارد. درون هر LUT یک حافظه وجود دارد که ارتباط بین ورودی و خروجی در آن ذخیره میشود.

* تعداد ورودی های LUT با عدد بعد از آن مشخص میشود.

مثال: یک بلاک منطقی با ۶ ورودی = LUT6



X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

: Flip-Flop -

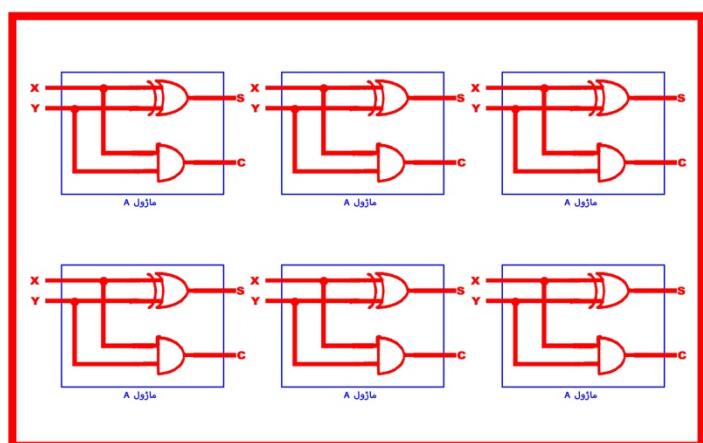
خروجی هر LUT به فلیپ-فلاب متصل میشود. برای ساختن واحد هایی مثل جمع کننده و یا ضرب کننده این کار الزامی است. به مجموعه ای از چند LUT و یک فلیپ-فلاب یک slice میگویند.

به عنوان مثال در برد sperten6 هر اسلایس از ۴ عدد LUT6 و ۸ فلیپ-فلاب تشکیل شده است. در این برد ۱۴۳۰ اسلایس و ۵۷۲۰ LUT وجود دارد.

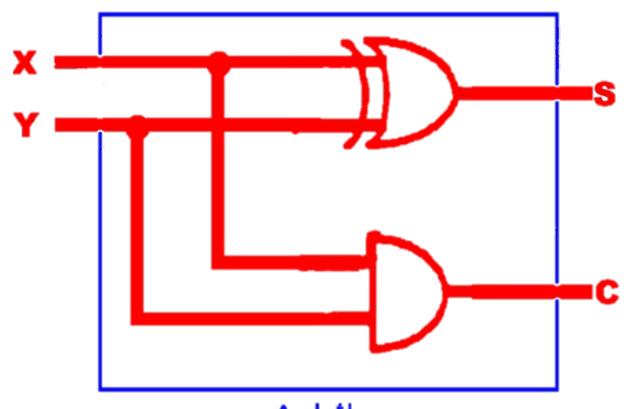
: Routing metrix -

تا حد زیادی پیچیده است که عنصر سازنده آن سیم و مالتی پلکسر است. مالتی پلکسر یک واحد الکترونیکی است که با تعدادی ورودی و یک خروجی در هر لحظه یک ورودی خاصی را به خروجی متصل میکند.

- * در اینجای بجای فاکشن در برنامه نویسی از ماذول ها استفاده میکنیم .
- * از ماذول های ایجاد شده میتوان در سایر ماذول ها استفاده کرد.

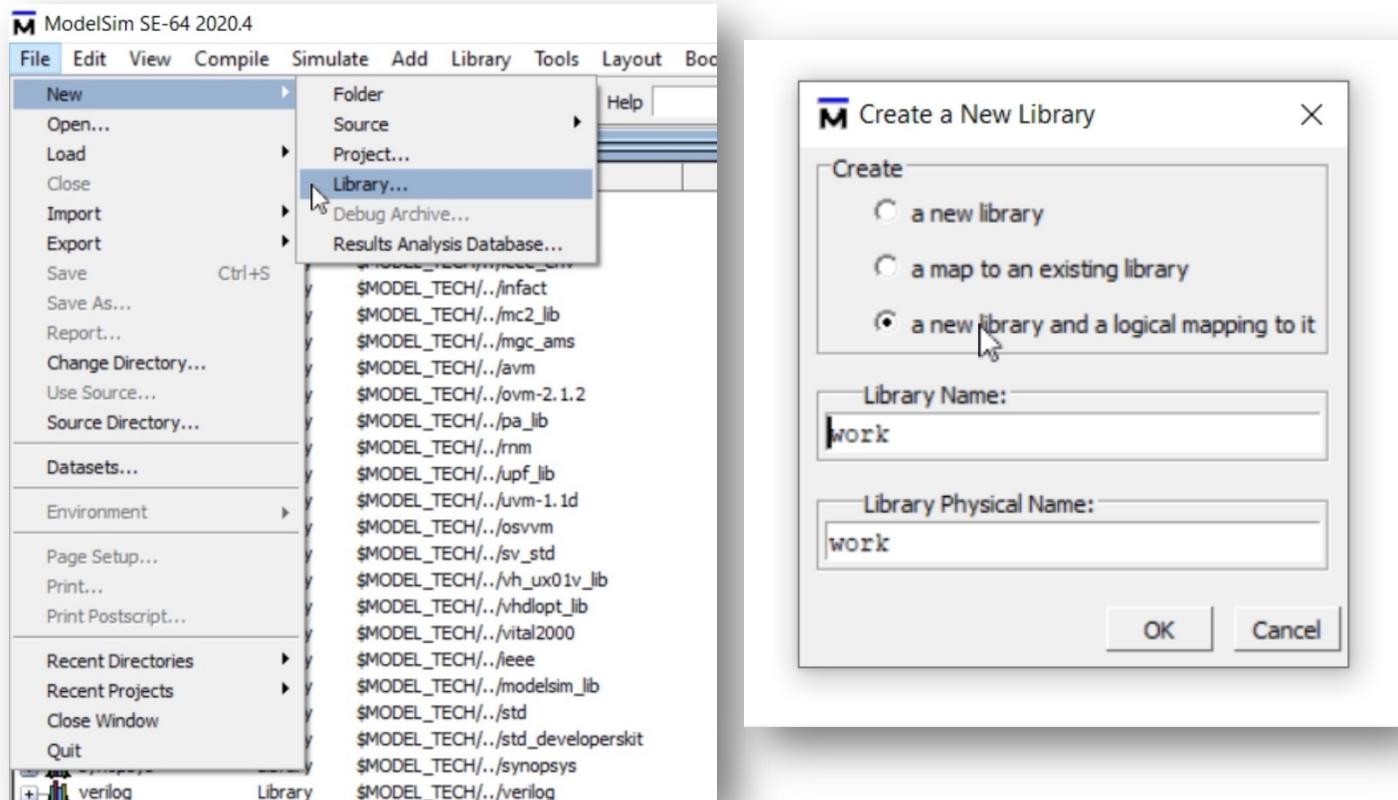


ماژول B

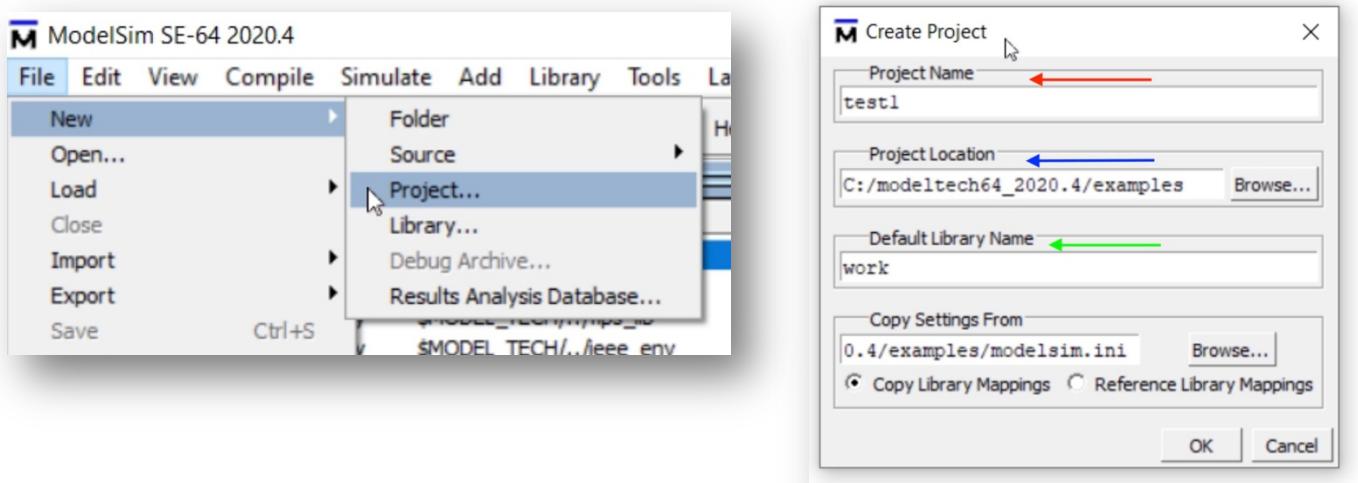


* روش کار با برنامه مدل سیم (Modelsim) :

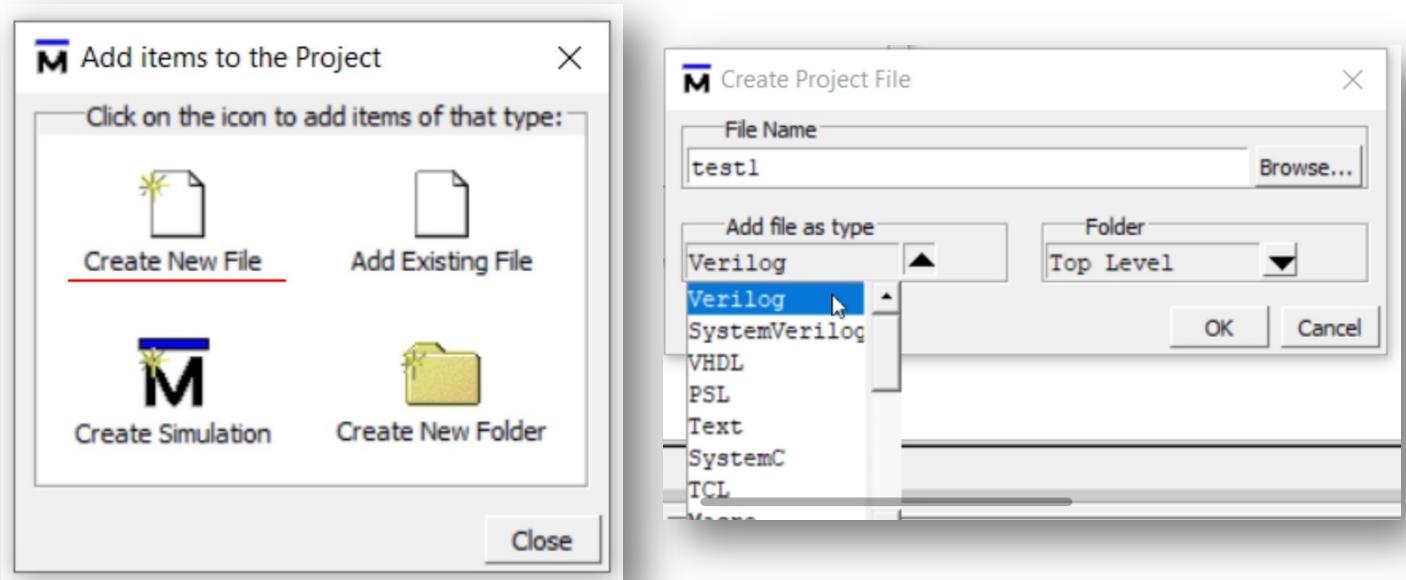
- برای ساخت لایبرری ها به منو File رفته سپس بر روی New و Library رو زده و در جفت فیلد های مشخص شده کلمه work را نوشته و اوکی را میزنیم.



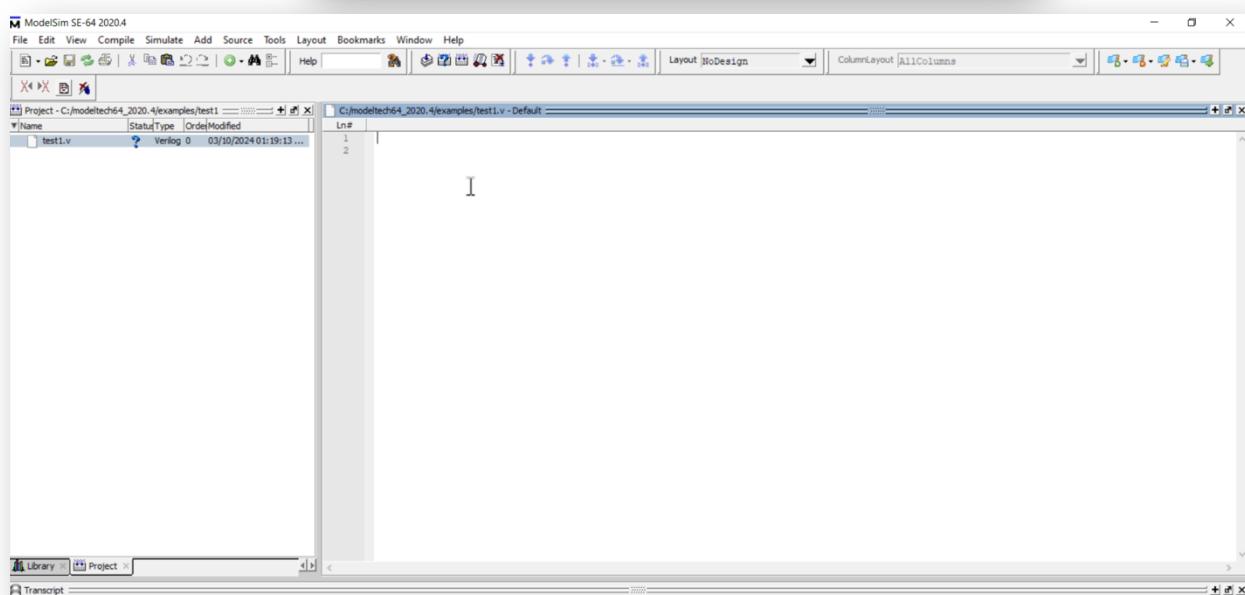
- بعد از ساخت لایبرری یک پروژه ایجاد میکنیم ، برای این کار مسیر زیر را میرویم
پس در پنجره باز شده نام پروژه (فلش قرمز) بعد مسیر پروژه(فلش آبی) و لایبرری که میخواهیم روی آن کار کنیم (فلش سبز) را انتخاب میکنیم.



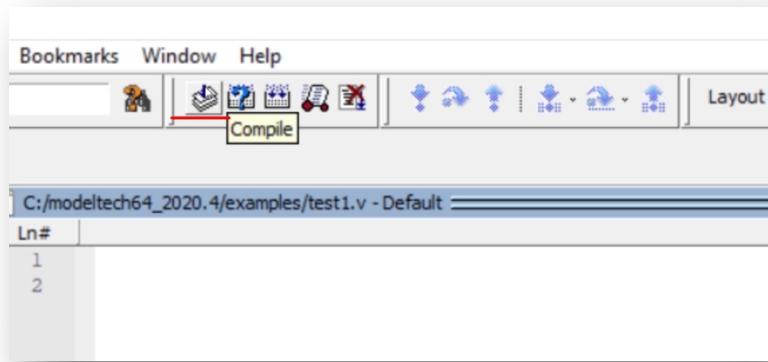
- پس از ایجاد پروژه قسمت **create new file** میز نیم در قسمت زبان را روی **Verilog** تنظیم کرده سپس اسم فایل را تنظیم میکنیم و اوکی کرده تا فایل جدید به وجود آید.



- سپس روی نام فایل ایجاد کرده کلیک کرده تا صفحه برنامه نویسی باز شود.



- از دکمه زیر برای کامپایل کردن کد نوشته شده است.



* زبان Verilog به حروف بزرگ و کوچک حساس است.

عکس کد نوشته شده در جلسه دوم : (فایل test1.v)

```
Test1.v
1 module A(input X,
2             input Y,
3                 output S,
4                 output C
5 );
6 // Inja coment bezarid
7 xor(S,X,Y);
8 and(C,X,Y);
9
10
11
12 endmodule
```

عکس شکل مدار کد بالا :

