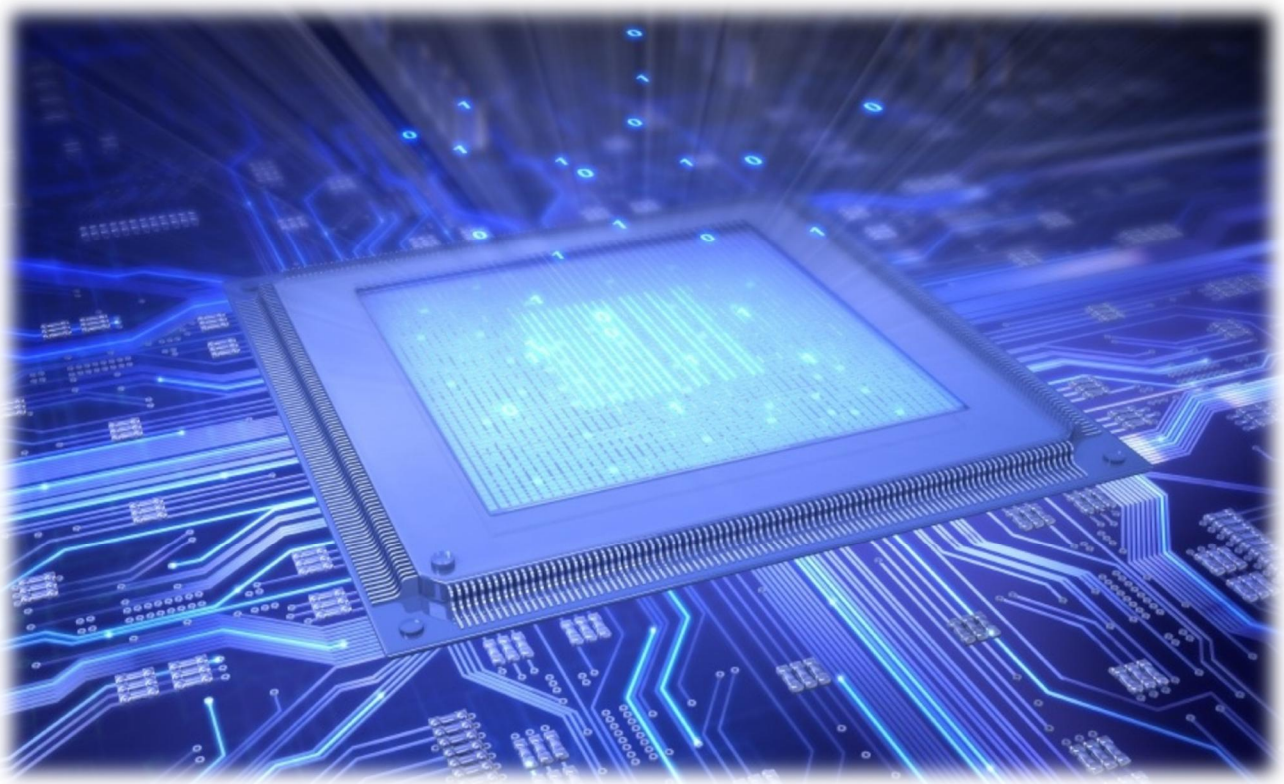
دانشگاه ارومیه

دانشکده فنی و مهندسی

گروه مهندسی برق



دستورکار آزمایشگاه معماری کامپیوتر

فهرست مطالب

پیشگفتار

بخش اول آشنایی با نرم افزار ISE

بخش دوم: موضوعات آزمایشگاه معماری کامپیوتر

آزمایش اول: راه‌اندازی و کار با نمایشگر هفت قسمتی

آزمایش دوم: جمع کننده دهدهی

آزمایش سوم: شمارنده ها

آزمایش چهارم: ثبات۱

آزمایش پنجم: ثبات۲

آزمایش ششم: واحد محاسبه و منطق

آزمایش هفتم: گذرگاه داده

پیشگفتار

تراشه های قابل برنامه ریزی با توجه به کارآیی و توانایی بالا در سیستم‌های صنعتی و تحقیقاتی، باعث تحول فوق العاده ای در صنعت میکرو الکترونیک شده اند. اما به دلیل فقدان آموزش و ابزار مورد نیاز در دانشگاههای کشورمان، همواره شکاف عمیقی میان محیط دانشگاهی و محیط صنعتی ملموس و محسوس بوده است ;آزمایشگاه معماری کامپیوتر با هدف ایجاد زیرساخت های لازم جهت اتوماسیون طراحی های الکترونیکی، آسان نمودن طراحی های پیشرفته دیجیتال و ساخت نمونه های آزمایشگاهی ارائه شده است. در این میان طراح نیز باید نسبت به مفاهیم پایه آگاهی و دانش کامل داشته باشد. به همین منظور در این آزمایشگاه سعی شده است تا این مفاهیم با دیدی آموزشی به بهترین نحو ممکن انتقال داده شود. این مفاهیم عبارتند از :

آشنایی با ساختار تراشه های قابل برنامه ریزی FPGA

روند طراحی با یک تراشه قابل برنامه ریزی FPGA

آشنایی و برنامه نویسی با زبان توصیف سخت افزار Verilog یا VHDL

آشنایی و طراحی با نرم افزار ISE

طراحی، پیاده سازی و تست انواع مدارهای دیجیتال بر روی FPGA

طراحی و پیاده سازی مدار ارتباط با **نمایشگر** های مختلف از جمله مجموعه LED ها ، نمایشگر های هفت قسمتی، نمایشگرهای LCD

طراحی و پیاده سازی انواع پروتکل های سری و ارتباط سری با کامپیوتر از طریق FPGA

طراحی و پیاده سازی مدار ارتباط با حافظه SRAM از طریق FPGA

طراحی و پیاده سازی مدار ارتباط با ADC و DAC ها از طریق FPGA

آموزش نرم‌افزار ISE

فهرست مطالب

۱- آموزش نرم‌افزار ISE Design Suite

۱-۱- ایجاد یک پروژه جدید

۱-۲- ایجاد یک فایل شماتیک جدید

۱-۳- ایجاد یک فایل VHDL جدید

۱-۴- بررسی نگارش برنامه VHDL و ساخت بلوک شماتیک

۱-۵- طراحی مدار درصفحه شماتیک

۱-۶- اتصال ورودی و خروجیهای طرح به پین‌های FPGA

۱-۷- شبیه‌سازی طرح

۱-۷-۱- شبیه‌سازی رفتاری

۱-۸- پیاده سازی طرح

۱-۹- ریختن برنامه روی FPGA

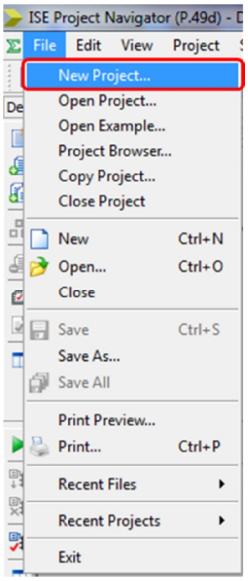
۱- آموزش نرم‌افزار ISE Design Suite

نرم‌افزار ISE مخصوص FPGA های شرکت Xilinx می‌باشد و می‌تواند برنامه نوشته شده را کامپایل، سنتز شبیه‌سازی و پیاده‌سازی کند.

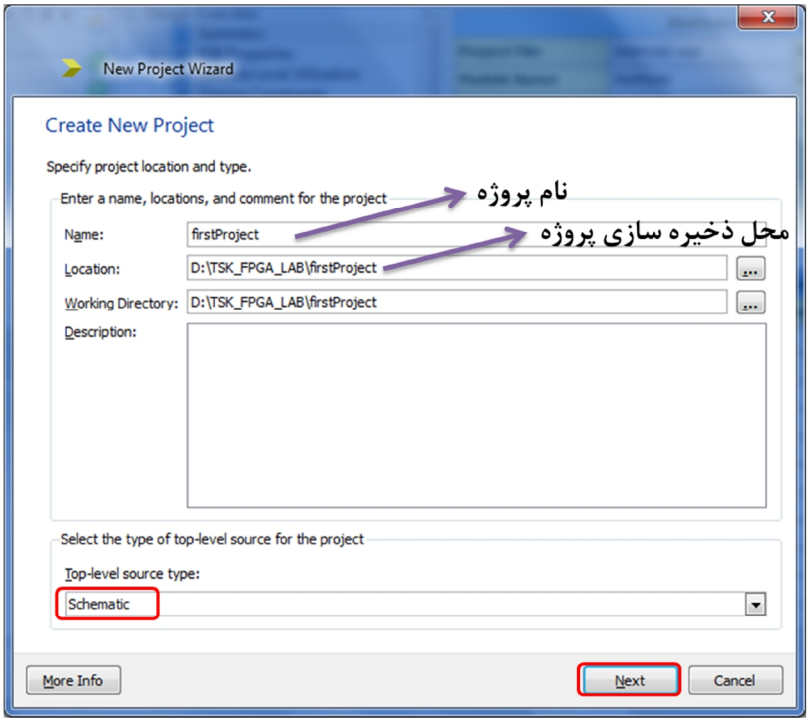
۱-۱- ایجاد یک پروژه جدید

ابتدا با کلیک روی آیکون

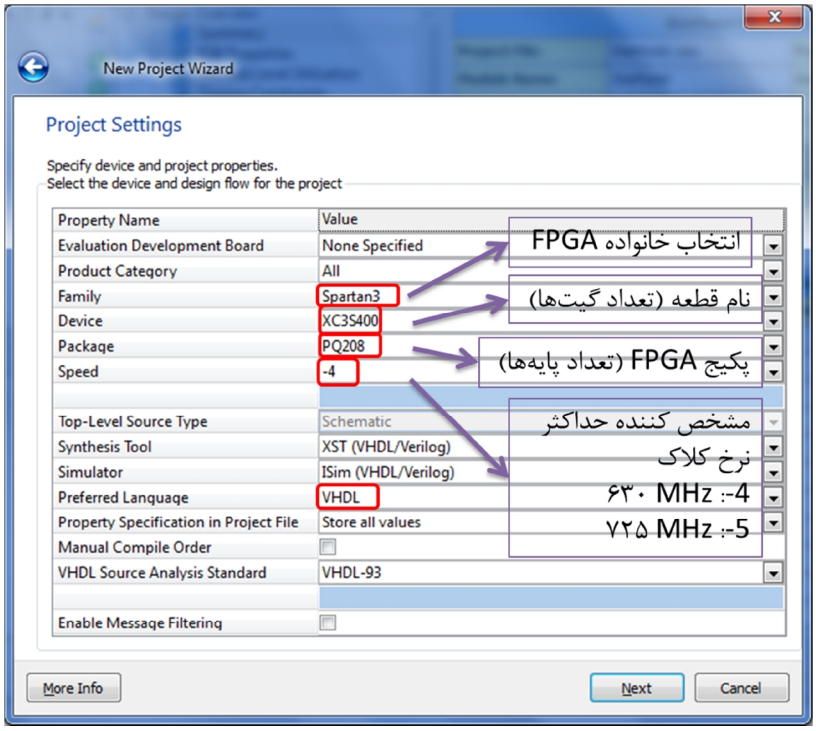
در desktop برنامه زیر را باز کنید به قسمت File، New Project بروید:



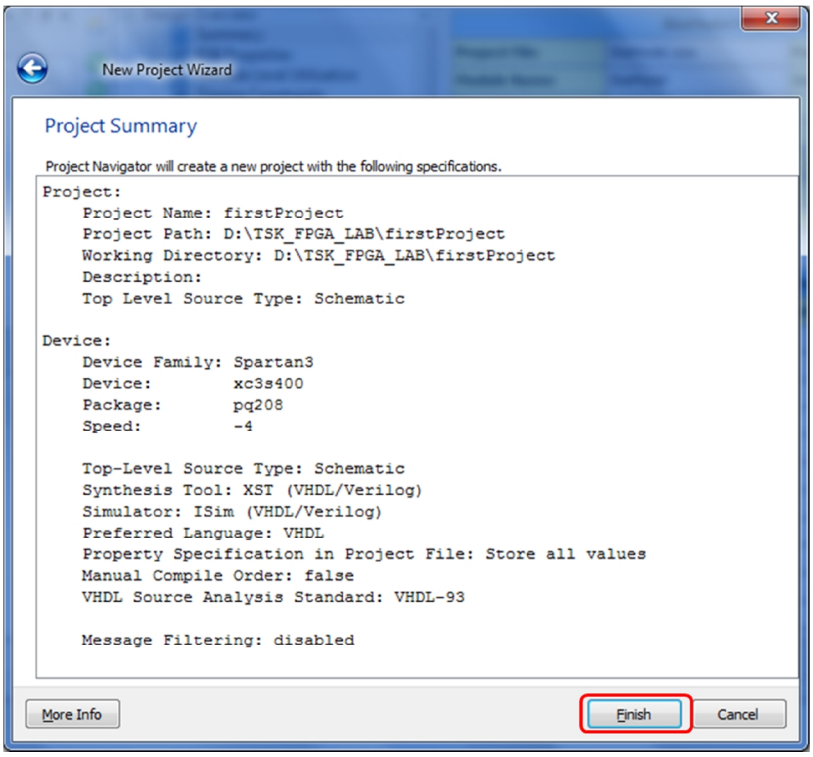
پنجره زیر باز می‌شود. نام پروژه و محل ذخیره‌سازی پروژه را وارد نمایید:



در قسمت Top-level source type، گزینه Schematic را انتخاب کنید. برنامه Top-level مانند برنامه main.c در زبان C می‌باشد. Top-level برنامه اصلی طرح می‌باشد که شامل برنامه های دیگر می‌شود. ورودی خروجی‌های این برنامه با دنیای خارج ارتباط برقرار می‌کنند و به پایه‌های FPGA متصل می‌شوند. دکمه Next را فشار دهید. در پنجره باز شده مشخصات آی سی FPGA که روی برد آزمایشگاه قرار دارد را به صورت شکل زیر وارد کنید:



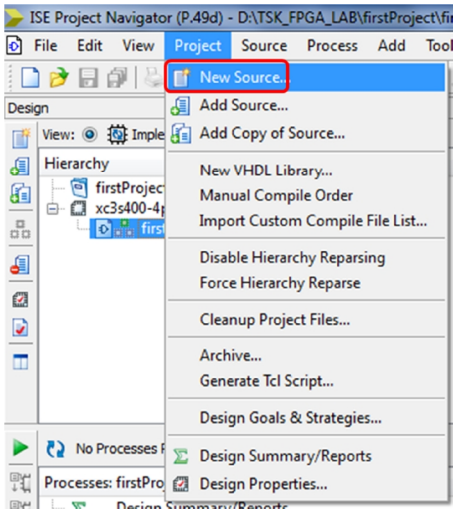
کلیک Next را بزنید. خلاصه‌ای از مشخصات پروژه‌ای که ایجاد کرده‌اید نشان داده می‌شود:



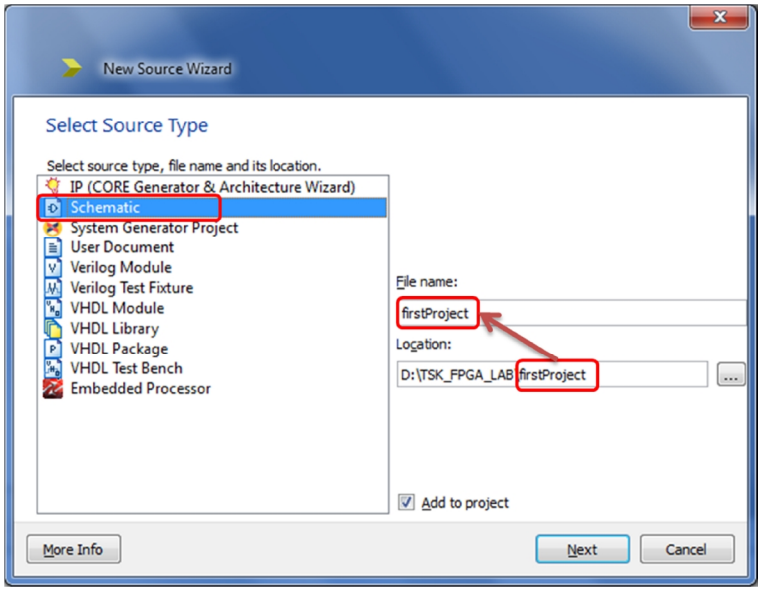
روی Finish کلیک کنید تا پروژه ایجاد شود.

۱-۲- ایجاد یک فایل شماتیک جدید

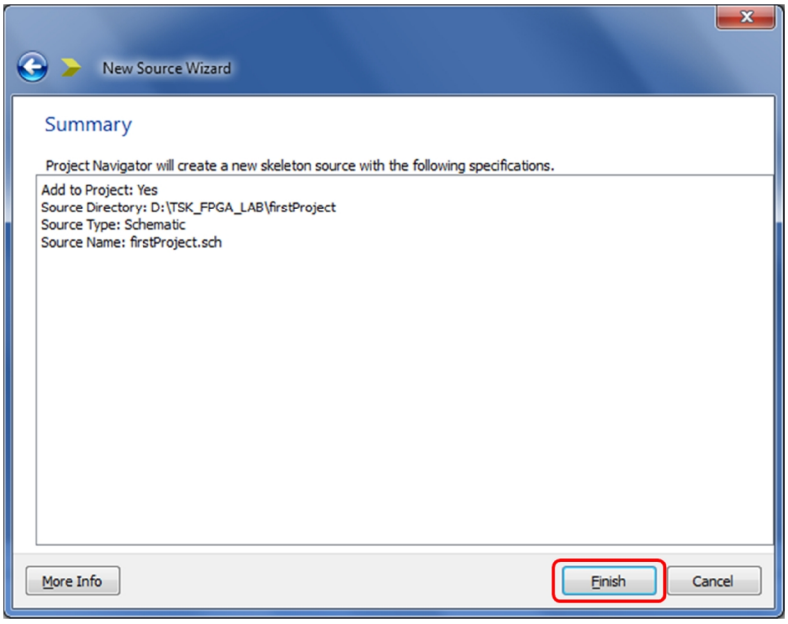
در قسمت Project روی ..New Source کلیک کنید:



در قسمت Select Source Type، Schematic را انتخاب کنید. نام فایل را همان نام پروژه تایپ کنید.



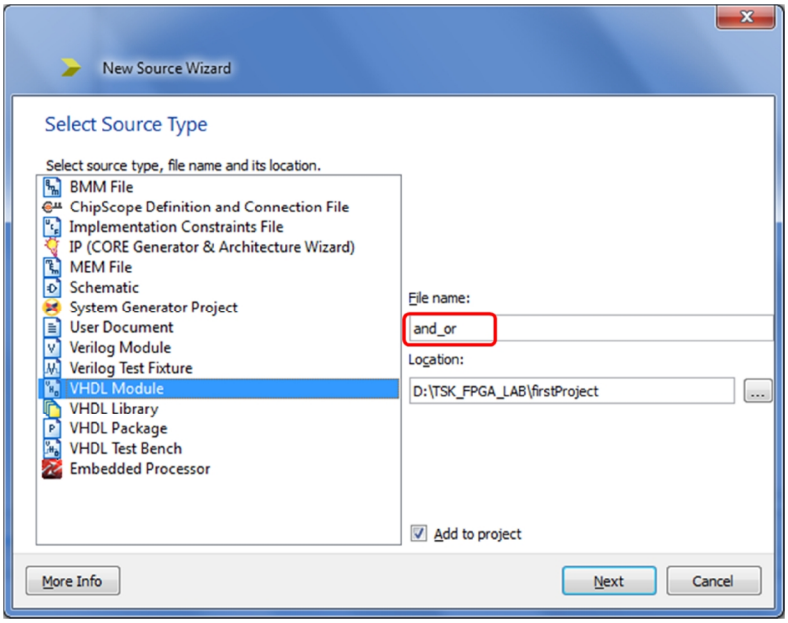
کلیک Next را بزنید.



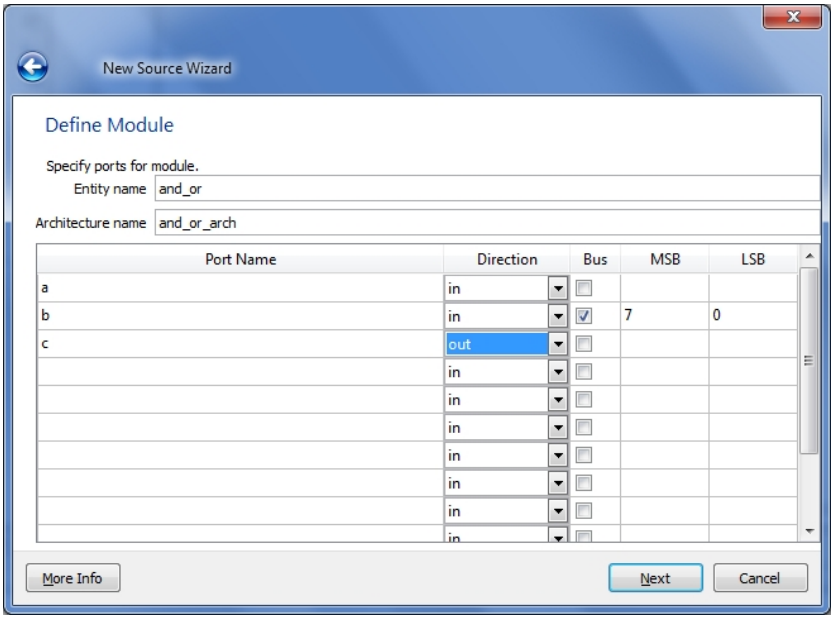
روی Finish کلیک کنید تا فایل جدید ایجاد شود.

۱-۳- ایجاد یک فایل VHDL جدید

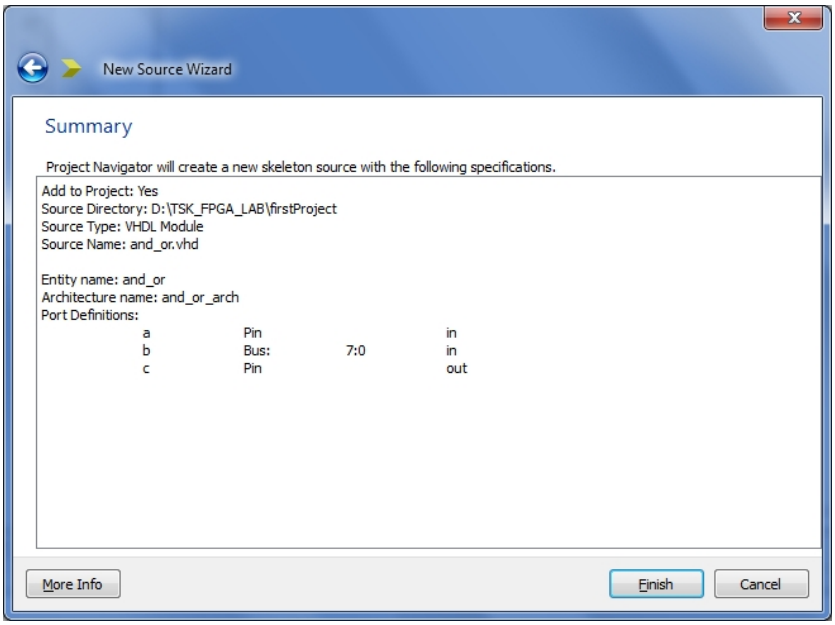
مانند قسمت قبل عمل کنید. در قسمت Select Source Type، VHDL Module را انتخاب کنید. دقت کنید نام فایل باید با نام entity یکسان باشد.



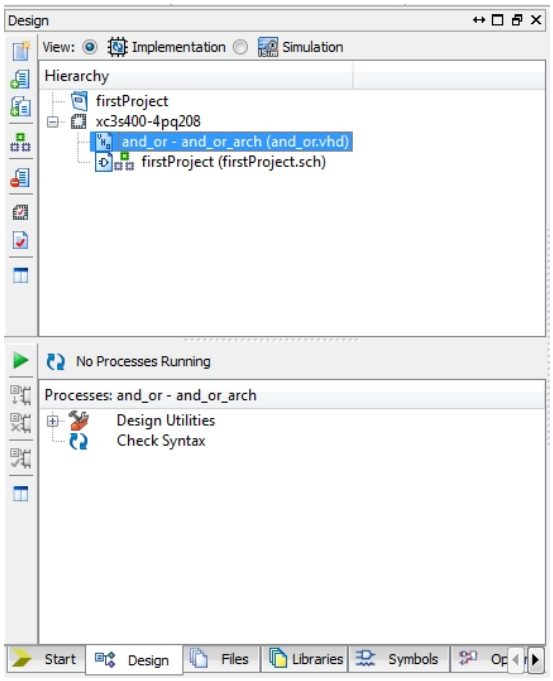
کلیک Next را بزنید. در پنجره بعدی مطابق شکل زیر عمل کنید:



کلیک Next را بزنید.



روی Finish کلیک کنید تا فایل جدید ایجاد شود. همان طور که مشاهده می‌کنید در قسمت Design فایل and_or.vhd و firstProject.sch به پروژه اضافه شده است. با دو بار کلیک روی and_or.vhd آن را باز کنید:



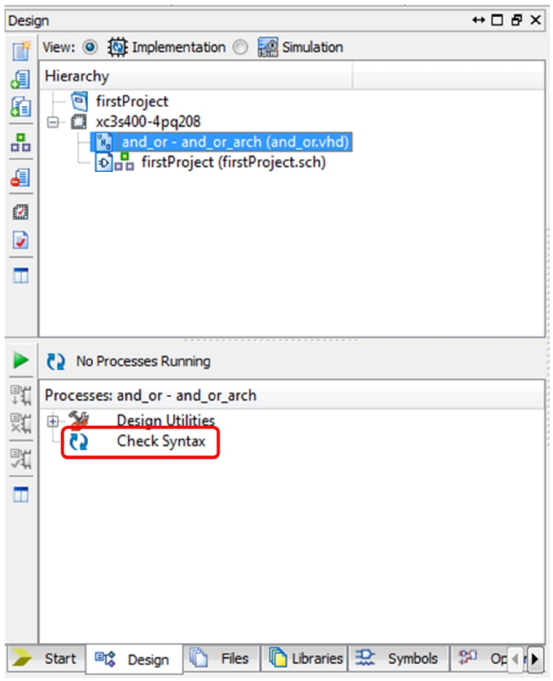
برنامه را به صورت زیر کامل کنید:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if
instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity and_or is
    Port ( a : in  STD_LOGIC;
```

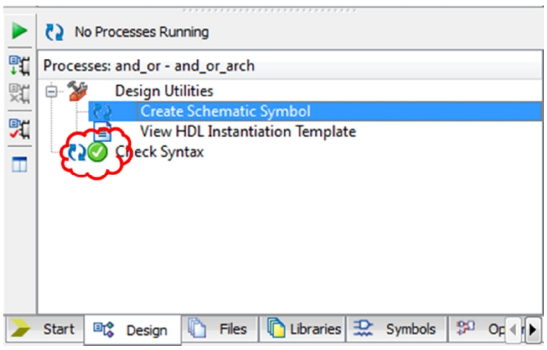
```
            b : in  STD_LOGIC_VECTOR (7 downto 0);
            c : out  STD_LOGIC);
end and_or;

architecture and_or_arch of and_or is
     signal d: STD_LOGIC;
begin
     process(a,b)
     begin
          d <= a or b(0);
          C <= d and b(7);
     end process;
end and_or_arch;
```

۱-۴- بررسی نگارش برنامه VHDL و ساخت بلوک شماتیک

پس از این که برنامه را تایپ کردید باید مطمئن شوید که برنامه خطای نگارشی ندارد برای این منظور ابتدا مطابق شکل در قسمت Design فایل را انتخاب کنید، سپس در قسمت Processes روی Check Syntax کلیک کنید.



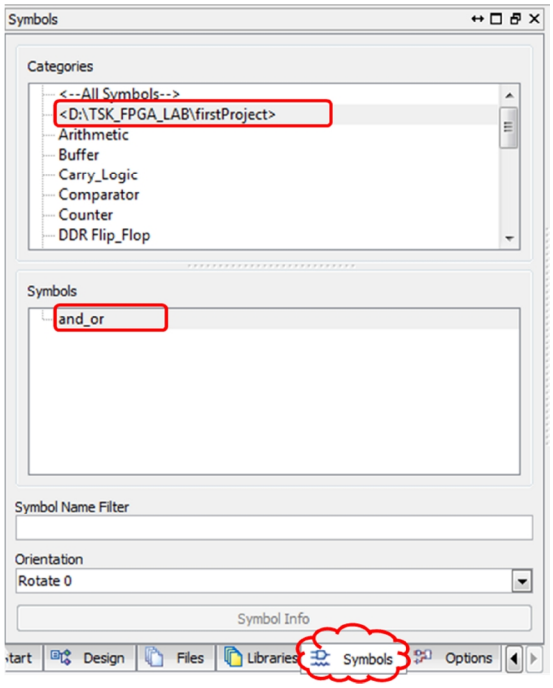
اگر کنار Check Syntax تیک سبز قرار گرفت یعنی برنامه مشکل نگارشی ندارد. در غیر این صورت خطاها را برطرف کنید و سپس دوباره روی Check Syntax کلیک کنید.



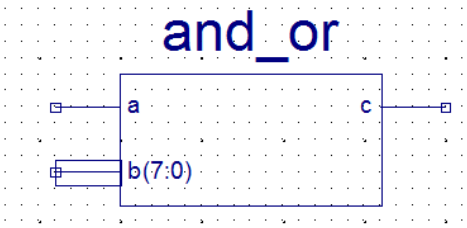
در قسمت Processes روی Create Schematic Symbol کلیک کنید تا یک بلوک شماتیک از فایل VHDL ساخته شود.

۱-۵- طراحی مدار درصفحه شماتیک

فایل firstProject.sch را باز کنید. به قسمت Symbols بروید. در این قسمت بلوک‌های زیادی که هر یک بیانگر یک IC می‌باشند قرار داده شده است و می‌توانید از آن‌ها در شماتیک طرح خود استفاده کنید. برای این که بتوانید از نمودار بلوکی برنامه VHDL نوشته شده در طرح شماتیک استفاده کنید، در بخش Categories، گزینه دوم را انتخاب کنید تا بلوک‌های ساخته شده در پروژه نمایش داده شود. در بخش Symbols، and_or را انتخاب کنید.

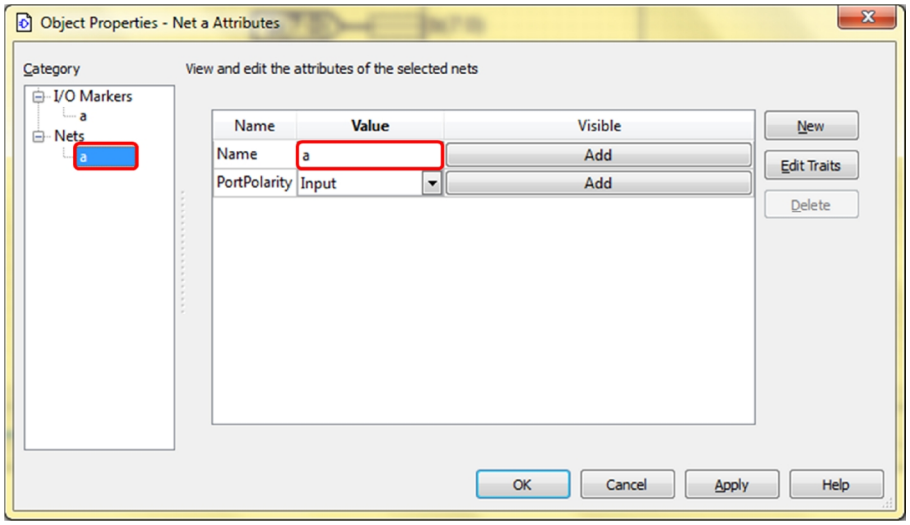


ماوس را به صفحه شماتیک ببرید و بلوک را در محل مناسبی بگذارید.

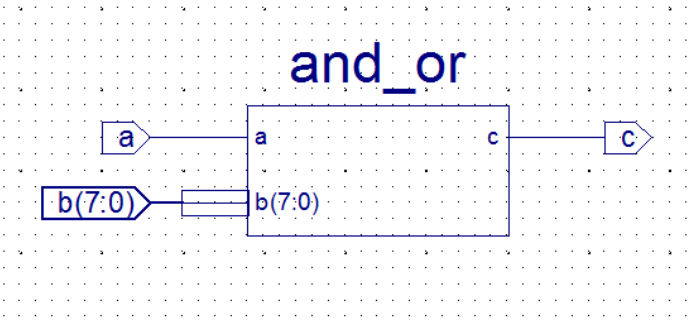


برای این که به شماتیک ورودی و خروجی اضافه کنید، روی Add I/O Marker کلیک کنید:

سپس به پین‌های نمودار بلوکی برچست ورودی/خروجی را اضافه کنید و نام آن‌ها به طور دلخواه تغییر دهید. برای تغییر نام روی برچست ورودی/خروجی دوبار کلیک کنید تا پنجره شکل ❋ باز شود. در قسمت Name نام دلخواهی وارد کنید.

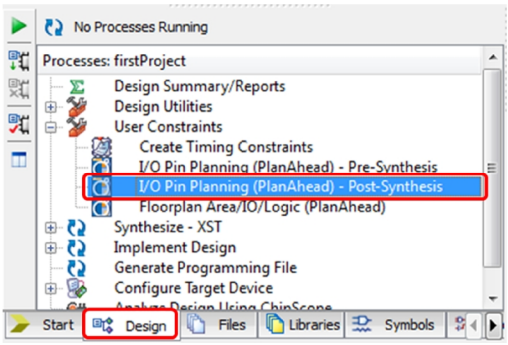


در نهایت مدار شماتیک، مانند شکل * می‌شود.

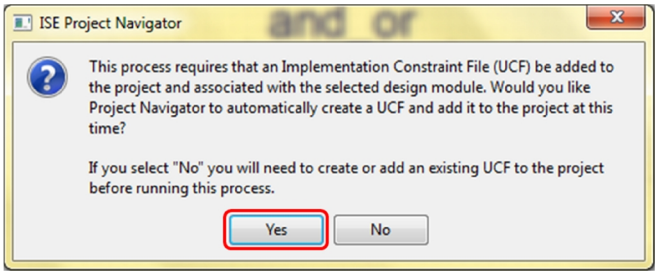


۱-۶- اتصال ورودی و خروجی‌های طرح به پین‌های FPGA

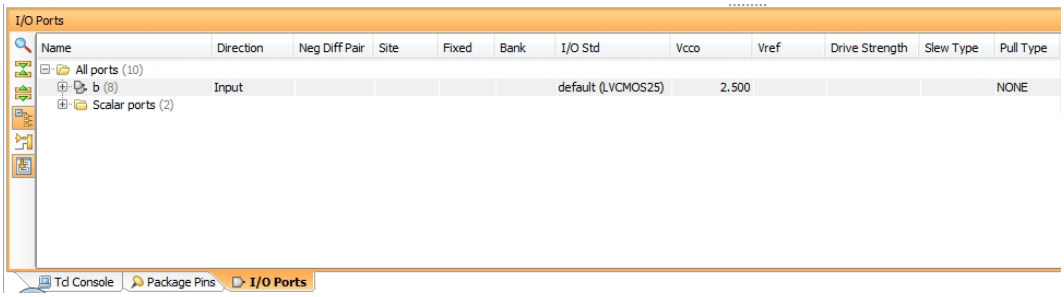
اکنون نوبت این است تا برچسب‌های ورودی/خروجی را به پین‌های FPGA مرتبط کنیم. برای این منظور از نرم‌افزار PlanAhead استفاده می‌کنیم که همراه ISE نصب می‌شود. به قسمت Design بروید در بخش User Constrants روی I/O Pin Planning (PlanAhead) – Post-Synthesis کلیک کنید.



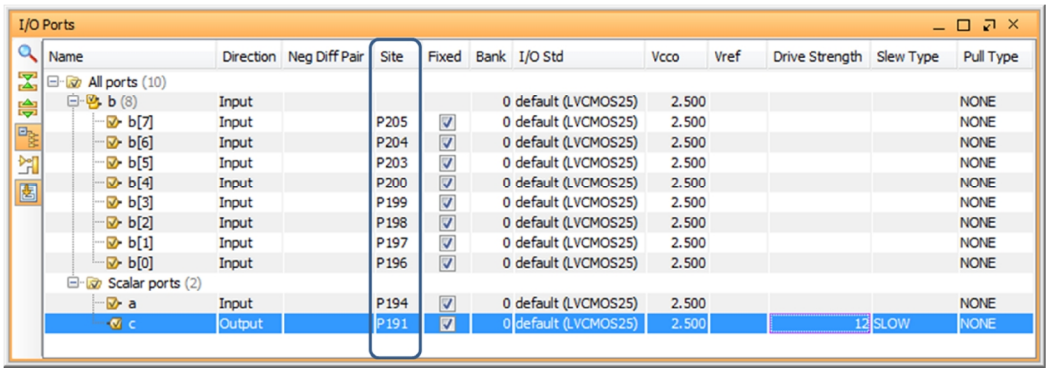
چند دقیقه صبر کنید تا نرم‌افزار PlanAhead باز شود. هنگامی که پنجره زیر باشد شد، روی Yes کلیک کنید:



در پنجره I/O پورت که در پایین صفحه قرار دارد می‌توانید مشخص کنید که هر پورت طرح به چه پایه‌ای از FPGA متصل شود، حداکثر ولتاژ آن چقدر باشد، مقاومت بالاکش یا پایین کش داشته باشد و ... .

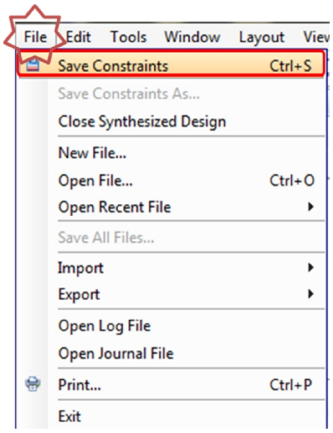


پنجره I/O ports را مطابق شکل زیر تنظیم کنید. در این صورت پورت‌های طرح به پایه‌هایی از FPGA متصل می‌شوند که به کانکتور گسترش‌دهنده می‌روند.

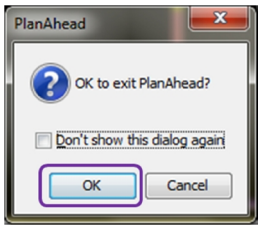


پس از این که وضعیت تمام پورت‌های طرح را مشخص کردید، باید فایل ایجاد شده، که با پسوند ucf.[1] می‌باشد را ذخیره کنید. برای این منظور به منوی فایل بروید و روی گذینه Save Constraints کلیک کنید.

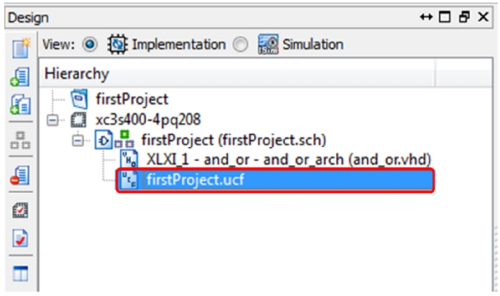
[1] User Constrains File



پنجره PlanAhead را ببندید. پیغام زیر نمایش داده می‌شود، OK را بزنید.



همان طور که در پنجره Design مشاهده می‌کنید، فایل firstProject.usf به طرح اضافه شده است. با دوبار کلیک روی این فایل آن را باز کنید و محتویات آن را مشاهده کنید.

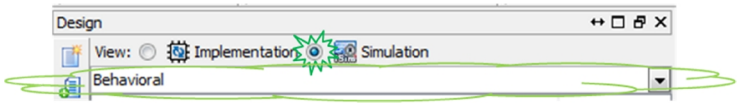


۱-۷- شبیه‌سازی طرح

برای شبیه سازی طرح، در قسمت Design، Simulation را انتخاب کنید. در مد شبیه‌سازی Behavioral فقط عملکرد منطقی طرح مورد شبیه‌سازی قرار می‌گیرد و تأخیرهای زمانی که در عمل وجود دارند و ممکن است سبب شوند طرح به درستی کار نکند، شبیه‌سازی نمی‌شوند. برای استفاده از شبیه‌سازی زمانی که اطلاعات دقیق‌تری را در اختیار طراح قرار می‌دهد، مد post-Place and Route را انتخاب کنید. در این مد، از اطلاعات تأخیر بلوک‌ها و سیم‌کشی مدار استفاده می‌شود در نتیجه باید بعد از پیاده‌سازی و سیم‌کشی طرح انجام شود.

۱-۷-۱- شبیه‌سازی رفتاری

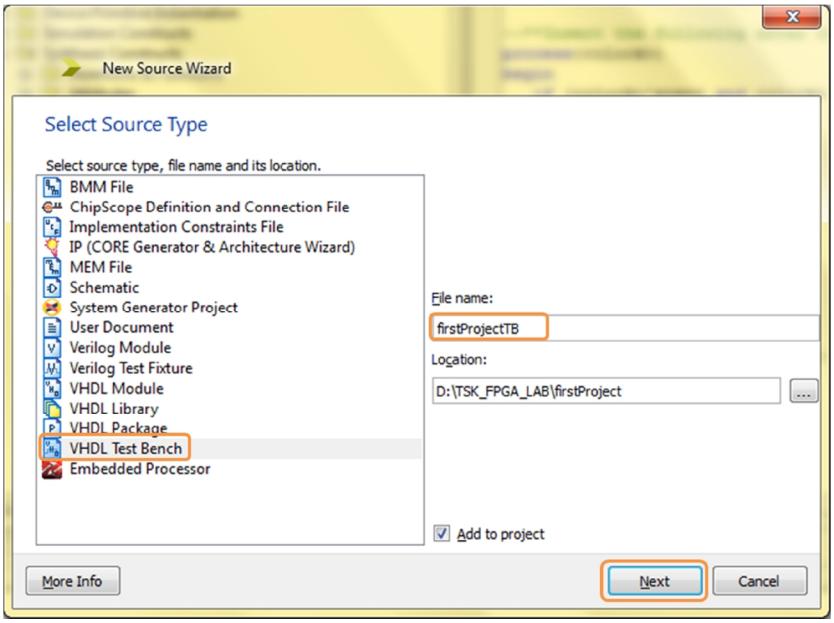
مد شبیه‌سازی را Behavioral انتخاب کنید.



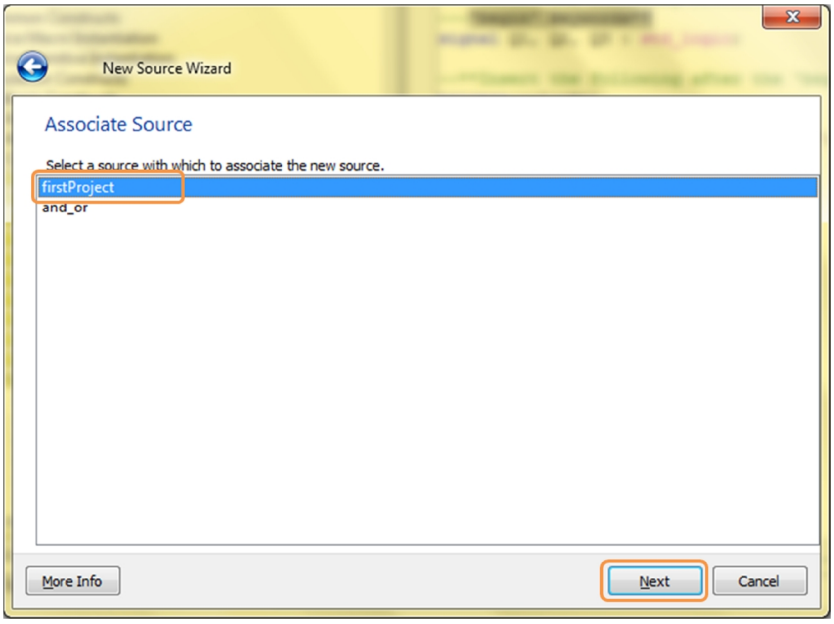
برای شبیه‌سازی هر طرحی، نیاز به برنامه تست (Test Bench) می‌باشد. این برنامه مانند یک مدار خارجی می‌باشد که سیگنال‌های ورودی لازم برای تست طرح را فراهم می‌آورد.

۱-۷-۱-۱- ایجاد برنامه تست

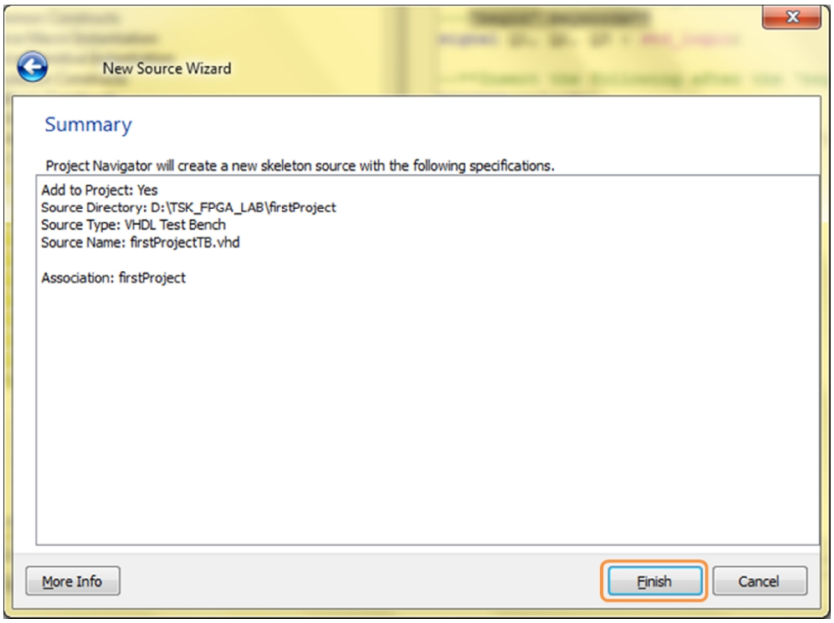
برای ایجاد برنامه تست، از منوی Project، گزینه New Source را انتخاب کنید، پنجره New Source Wizard باز می‌شود، VHDL Test Bench را انتخاب کنید و نام مناسبی را وارد کنید. Next را فشار دهید.



در قسمت Associate Source، Top Module را که در این پروژه firstProject می‌باشد را انتخاب کنید.



Next و سپس Finish را بزنید.



از قسمت Design، برنامه تست را که ایجاد کرده‌اید باز کنید. برنامه را به صورت زیر کامل کنید:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
ENTITY firstProjectTB IS
END firstProjectTB;
```

```
ARCHITECTURE behavioral OF firstProjectTB IS

   COMPONENT firstProject
   PORT( a     :    IN   STD_LOGIC;
         b     :    IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
         c     :    OUT  STD_LOGIC);
   END COMPONENT;

   SIGNAL a    :    STD_LOGIC;
   SIGNAL b    :    STD_LOGIC_VECTOR (7 DOWNTO 0);
   SIGNAL c    :    STD_LOGIC;

BEGIN

   UUT: firstProject PORT MAP(
         a => a,
         b => b,
         c => c
   );

-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN

            -- -------------  Current Time:  50ns
            WAIT FOR 50 ns;
            a <= '0';
            b <= "00000001";
            -- -------------------------------------
            -- -------------  Current Time:  100ns
            WAIT FOR 50 ns;
            a <= '0';
            b <= "10000001";
            -- -------------------------------------
            -- -------------  Current Time:  150ns
            WAIT FOR 50 ns;
            a <= '1';
            b <= "10000001";
            -- -------------------------------------
            -- -------------  Current Time:  200ns
            WAIT FOR 50 ns;
            a <= '1';
            b <= "00000000";
            -- -------------------------------------
```

```
            -- -------------  Current Time:  250ns
            WAIT FOR 50 ns;
            a <= '1';
            b <= "10000000";
             -----------------------------------
             -- -------------  Current Time:  300ns
   END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```

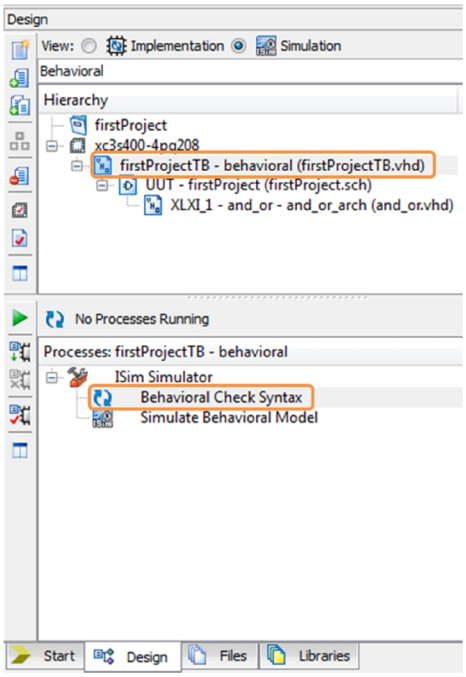
۱-۷-۱-۲- بررسی درست بودن برنامه تست از نظر دستوری

از قسمت Design برنامه تست را انتخاب کنید. در قسمت Processes در بخش Isim Simulator دو گزینه وجود دارد:

Behavioral Check Syntax: این پروسه، وجود خطاهای گرامری در برنامه تست را بررسی می‌کند.

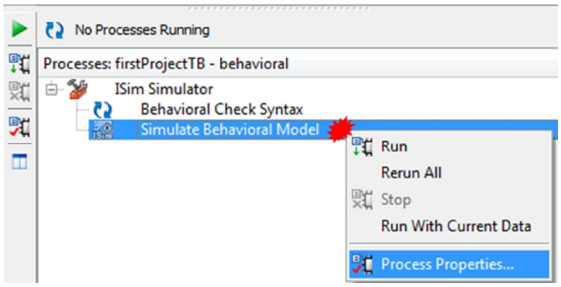
Simulate Behavioral Model: این پروسه باعث آغاز فرایند شبیه‌سازی می‌شود.

با دوبار کلیک روی Behavioral Check Syntax بودن برنامه تست از نظر دستوری بررسی می‌شود. اگر خطایی وجو داشت آن را رفع کنید.

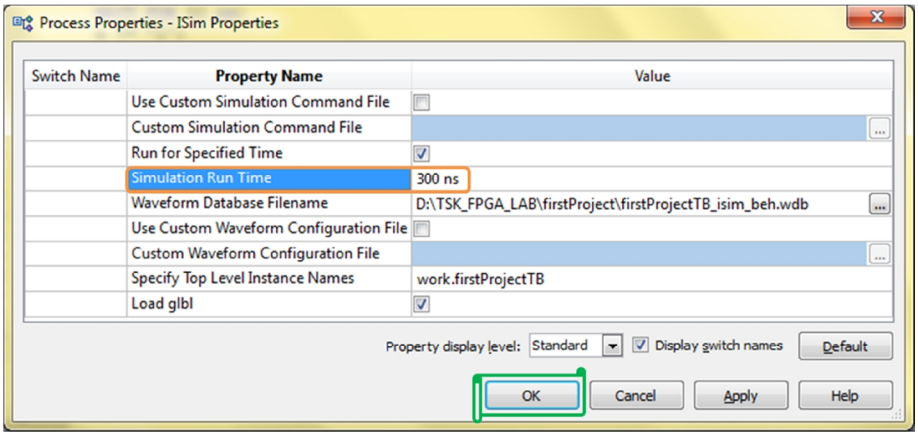


۱-۷-۱-۳- شبیه‌سازی طرح

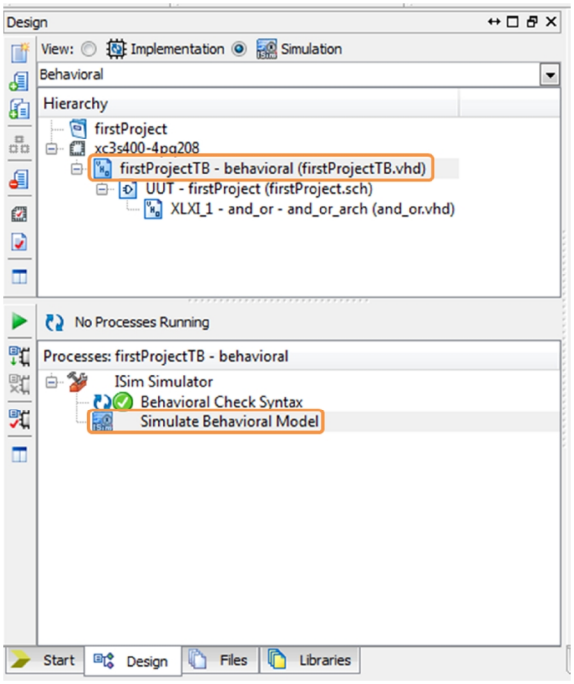
برای شبیه‌سازی باید تعیین کنید تا چه مدت طرح شبیه‌سازی شود. چون برنامه تست ۳۰۰ نانو ثانیه طول می‌کشد، زمان شبیه‌سازی را همین مقدار انتخاب می‌کنیم. برای این منظور روی Simulate Behavioral Model کلیک راست کنید و Process Properties… را انتخاب کنید.



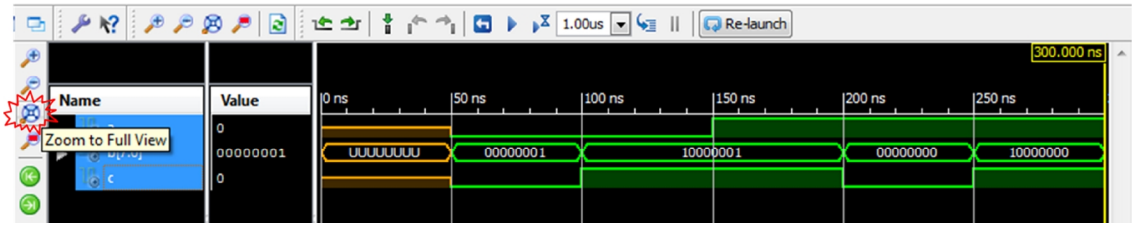
در پنجره باز شده Simulation Run Time را به ۳۰۰ ns تغییر دهید. OK را بزنید.



اکنون طرح آماده شبیه سازی می‌باشد. برای شبیه‌سازی روی Simulate Behavioral Model دوبار کلیک کنید.

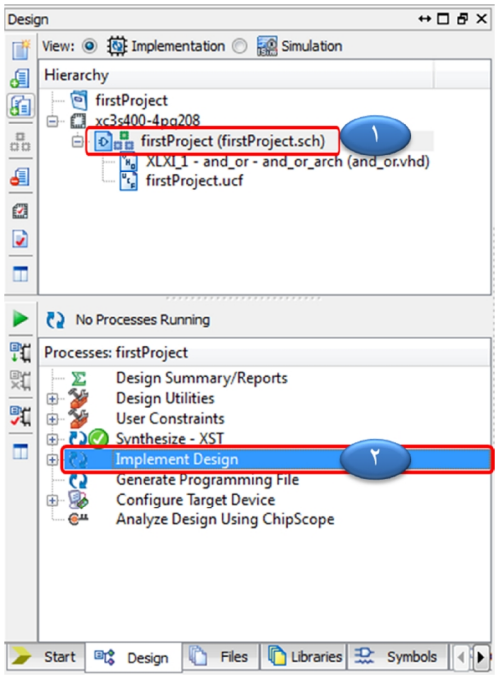


برنامه Isim باز می‌شود. برای مشاهده شکل موج‌های ورودی و خروجی مدار از منوی سمت چپ، Zoom to Full View را انتخاب کنید.



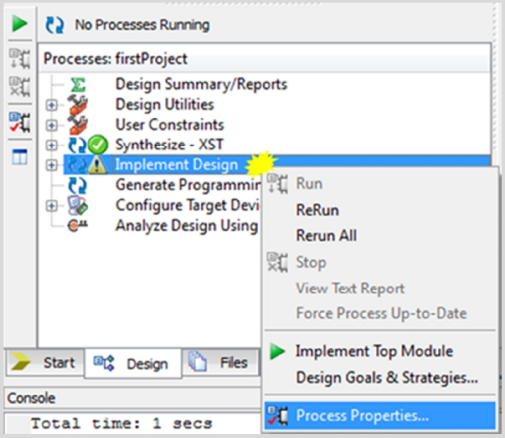
۱-۸- پیاده سازی طرح

اکنون نوبت پیاده‌سازی طرح می‌باشد. ابتدا Top Module، که در این طرح firstProject.sch می‌باشد را انتخاب کنید، سپس در قسمت Processes روی Implement Design دوبار کلیک کنید.

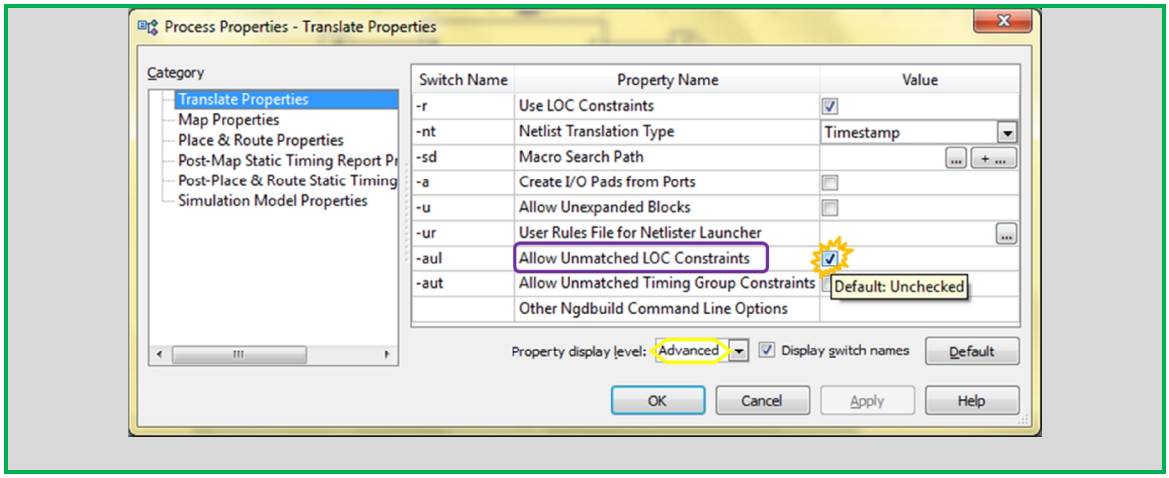


قسمت Errors را بررسی کنید، اگر خطایی بود، آن را بر طرف کنید و دوباره مراحل بالا را انجام دهید.

☼ توجه: اگر از فایل FPGA_LAB_FPGA.ucf برد استفاده می‌کنید در صورت پاک نکردن تمام پین‌های اضافی، پیام خطا تولید می‌شود. اگر نمی‌خواهید تمام پین‌ها را پاک کنید، برای این منظور، ابتدا در قسمت Design فایل Top Module را انتخاب کنید، در قسمت Processes روی Implement Design کلیک راست کنید و Process Properties... را انتخاب کنید.



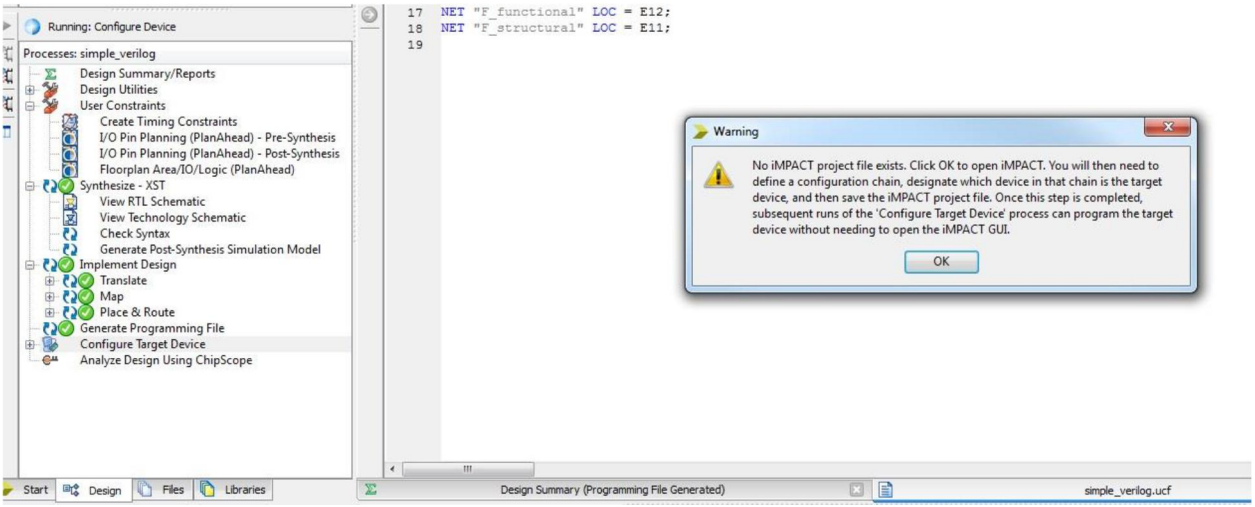
در پنجره باز شده تیک گزینه Allow Unmatched LOC Constraints را بنزید تا نرم‌افزار به شما اجازه دهد در فایل ucf. پین‌های استفاده نشده داشته باشید.


Process Properties - Translate Properties
Category
Translate Properties
Map Properties
Place & Route Properties
Post-Map Static Timing Report Pr
Post-Place & Route Static Timing
Simulation Model Properties
Switch Name
Property Name
Value
-r
Use LOC Constraints
-nt
Netlist Translation Type
Timestamp
-sd
Macro Search Path
-a
Create I/O Pads from Ports
-u
Allow Unexpanded Blocks
-ur
User Rules File for Netlister Launcher
-aul
Allow Unmatched LOC Constraints
-aut
Allow Unmatched Timing Group Constraints
Default: Unchecked
Other Ngdbuild Command Line Options
Property display level:
Advanced
Display switch names
Default
OK
Cancel
Apply
Help

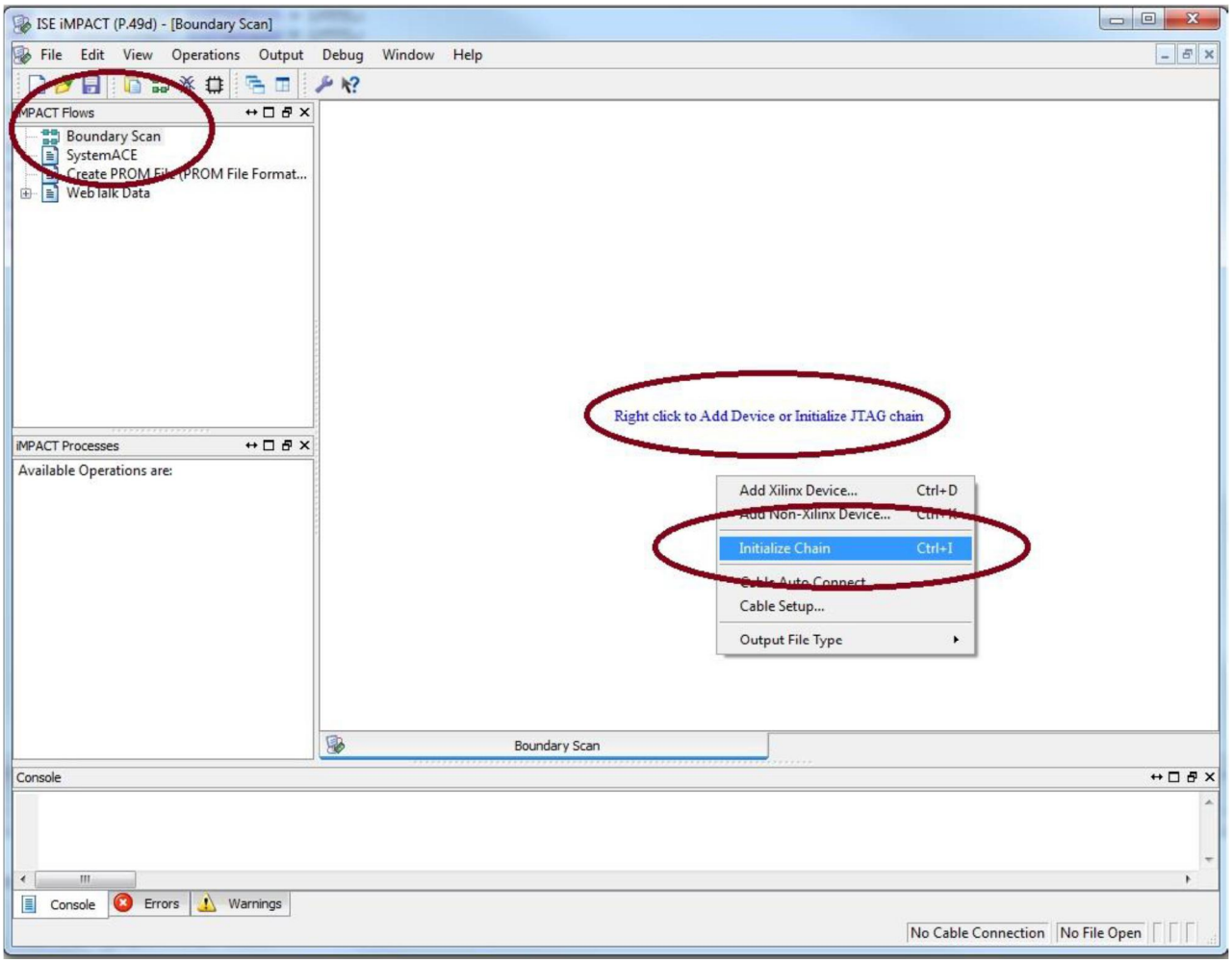
۱-۹- ریختن برنامه روی FPGA

با دو بار کلیک روی Configure Target Device می‌توانید پروسه پروگرام کردن را شروع کنید، با اینکار ISE به محیط جدیدی با نام iMpact وارد می‌شود.

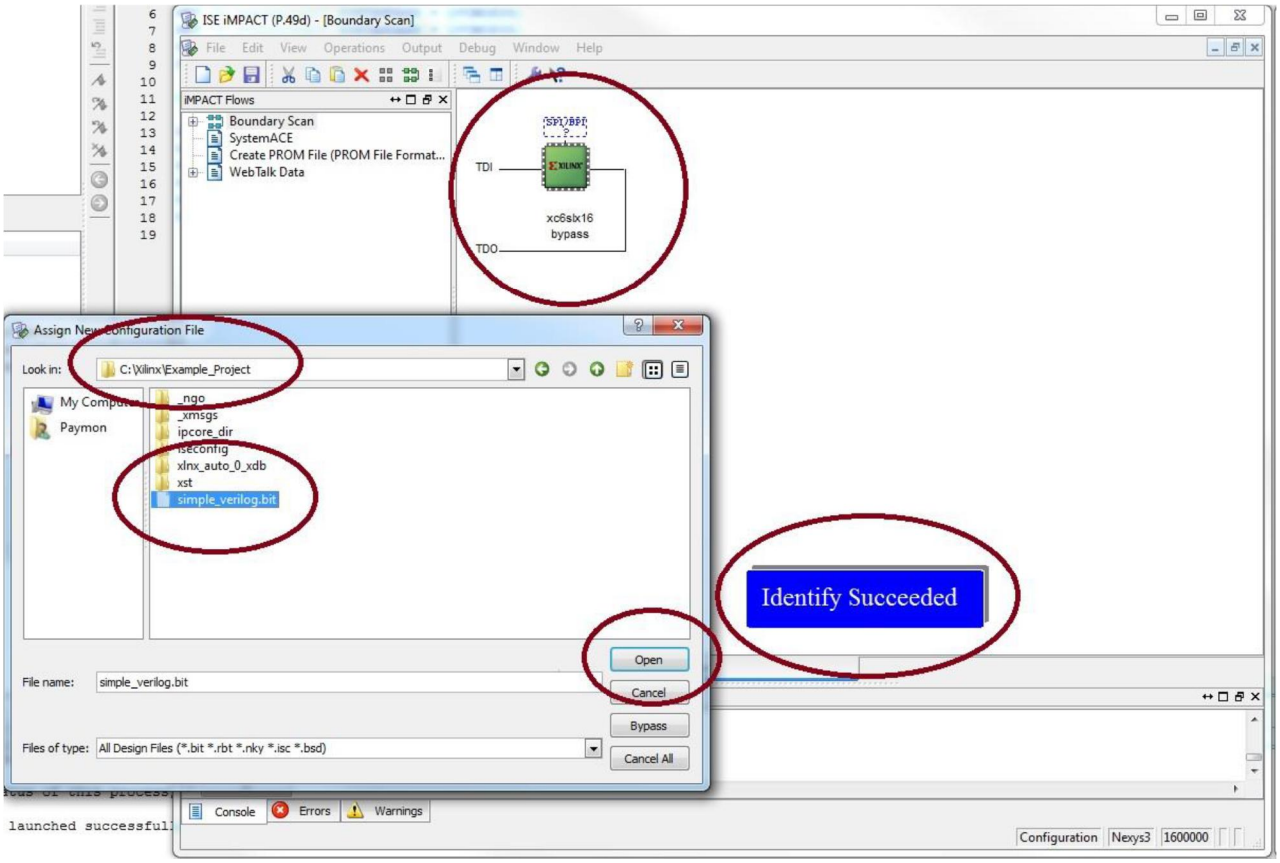
یک پیغام هشدار با عنوان "No iMpact project file exists…" ظاهر می‌شود، روی OK کلیک کنید.



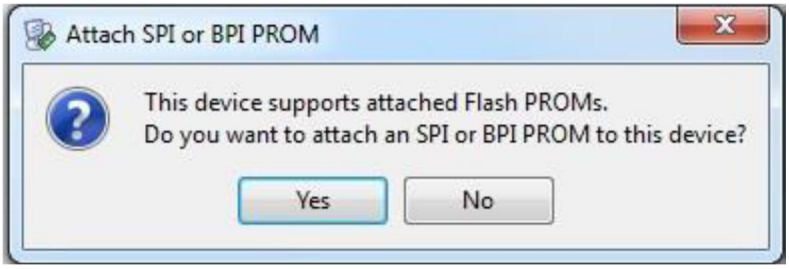
در صفحه ISE iMpact در گوشه بالا سمت چپ روی گزینه Boundary Scan دو بار کلیک کنید. در صفحه اصلی Boundary Scan در جایی که پیغام " Right click to Add Device or Initialize chain" وجود دارد راست کلیک کنید و گزینه Initialize Chain را انتخاب کنید. قبل از اینکار مطمئن شوید که کابل واسط برد به کامپیوتر وصل شده و تغذیه برد نیز روشن است.



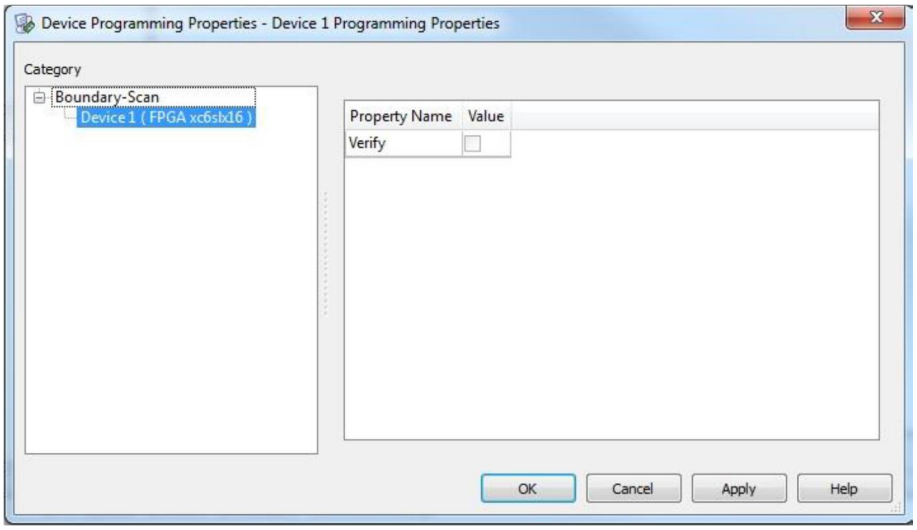
بعد از اینکه iMpact اتصال کابل را با پیام "identify succeeded" تایید کرد باید فایلی با پسوند bit. و با نام top module تعریف شده(نام پروژه اصلی) را open کنید. در صورتی که Boundary Scan دو IC را شناسایی کند فایل bit. را open و دیگری را bypass نمایید.



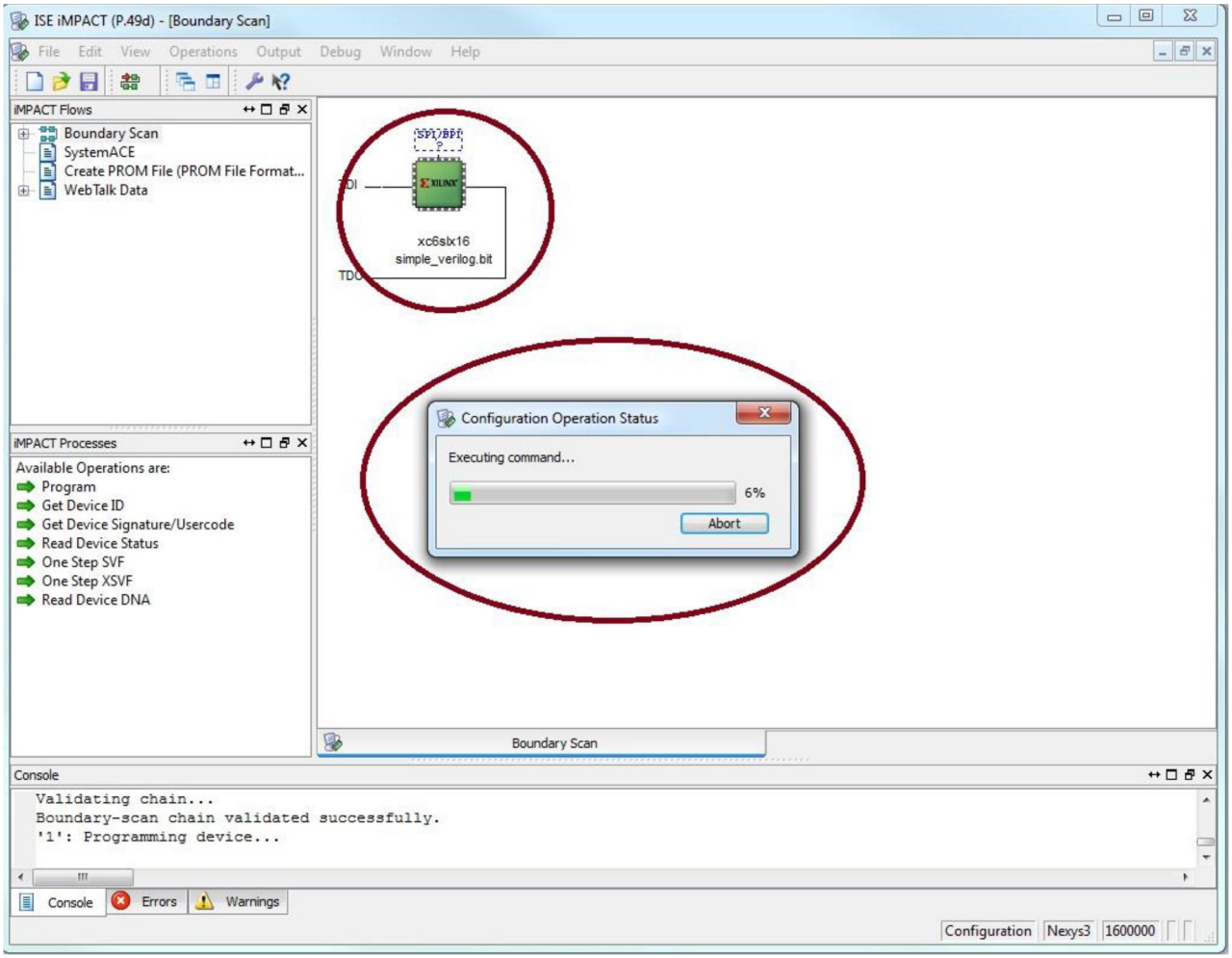
بعد از تعریف فایل bit. برای برد iMpact از شما میخواهد که در صورت تمایل یک PROM Controller نیز اضافه کنید، این مرحله برای زمانی است که شما یک فایل PROM آماده کرده‌اید و میخواهید که FPGA را یکبار برای پروگرام کرده و با قطع کابل واسط برنامه شما روی FPGA همچنان باقی بماند و پاک نشود. ولی چون در آزمایشگاه این مورد مطلوب ما نیست روی NO کلیک کنید(در حالت فعلی بعد از قطع تغذیه برد برنامه پروگرام شده از روی FPGA پاک می‌شود.



در پنجره بعدی iMpact از شما میخواهد که device مورد نظر خود را verify نمایید ولی چون ما فقط چیپ FPGA روی برد داریم **بدون** زدن تیک verify روی OK کلیک و آماده‌سازی برای پروگرام کردن را به انتها برسانید.



در آخرین مرحله روی چیپی که به رنگ سبز درآمده راست کلیک کرده و گزینه Program را بزنید. حال پروگرام کردن FPGA به اتمام رسیده و طراحی شما روی FPGA پیاده‌سازی شده است.



بخش دوم:

موضوعات آزمایشگاه معماری کامپیوتر

آزمایش اول

راه‌اندازی و کار با نمایشگر هفت قسمتی

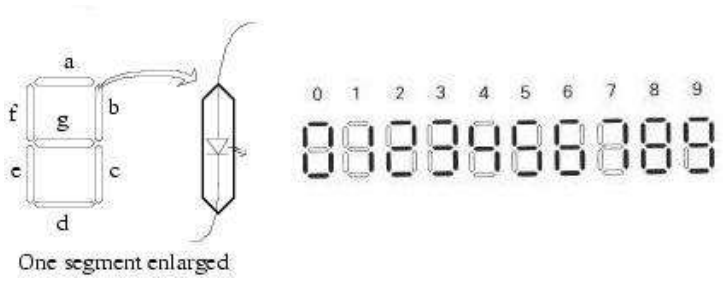
هدف: در این آزمایش اهداف زیر دنبال می شوند :

آشنایی با نرم‌افزار ISE

کسب مهارت در انجام مراحل طراحی، تحلیل، تست و پیاده‌سازی

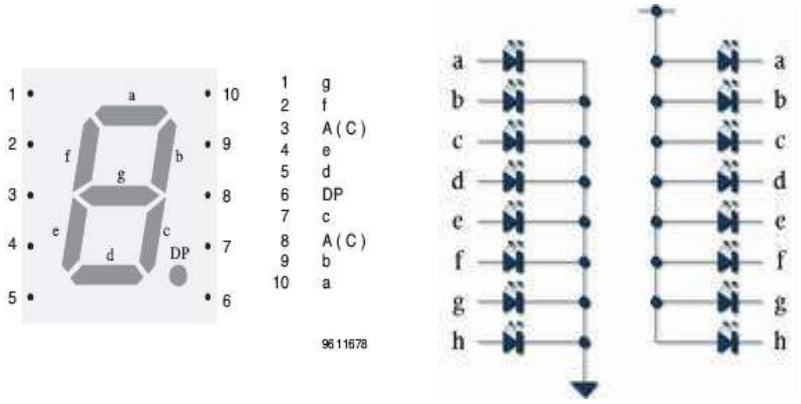
تئوری آزمایش

نمایشگر هفت قسمتی یکی از قطعات بسیار مهم در آزمایشگاه است که مشاهده نتایج آزمایش‌های صورت گرفته را ممکن می‌سازد. نمایشگر هفت قسمتی در واقع وسیله‌ای است که از هفت قطعه فتوالکتریک مستقل، مانند LED یا کریستال مایع، تشکیل شده است. قطعات مزبور به فرمی که در شکل زیر نمایش داده شده است در کنار هم قرار گرفته و امکان نمایش اعداد را مهیا می‌نمایند. همان طور که در شکل نیز مشاهده می‌نمایید با فعال نمودن گروه خاصی از این LED ها اعداد مورد نظر نمایش داده می‌شود.



شکل ۱-۱

Binary State	INPUTS RBI	A_3	A_2	A_1	A_0	OUTPUTS P_1	P_2	P_3	P_4	P_5	P_6	P_7	RBO	Display
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Blank
0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
1	x	0	0	0	1	1	0	0	1	1	1	1	1	1
2	x	0	0	1	0	0	0	1	0	0	1	0	1	2
3	x	0	0	1	1	0	0	0	0	1	1	0	1	3
4	x	0	1	0	0	1	0	0	1	1	0	0	1	4
5	x	0	1	0	1	0	1	0	0	1	0	0	1	5
6	x	0	1	1	0	0	1	0	0	0	0	0	1	6
7	x	0	1	1	1	0	0	0	1	1	1	1	1	7
8	x	1	0	0	0	0	0	0	0	0	0	0	1	8
9	x	1	0	0	1	0	0	0	1	1	0	0	1	9
10	x	1	0	1	0	0	0	0	1	0	0	0	1	A
11	x	1	0	1	1	1	1	0	0	0	0	0	1	b
12	x	1	1	0	0	0	1	1	0	0	0	1	1	C
13	x	1	1	0	1	1	0	0	0	0	1	0	1	d
14	x	1	1	1	0	0	1	1	0	0	0	0	1	E
15	x	1	1	1	1	0	1	1	1	0	0	0	1	F



شکل ۱-۲

به طور کلی نمایشگرهای هفت قسمتی به دو نوع کاتد مشترک و آند مشترک تقسیم می‌شوند. در نمایشگر هفت قسمتی کاتد مشترک، کاتد تمامی LED ها به هم متصل شده و به منظور روشن نمودن LED ها مورد نظر می‌بایست ولتاژ آند آنها را نسبت به کاتد مشترک افزایش داد. در نمایشگر هفت قسمتی آند مشترک، کاتد تمامی LED ها به هم متصل شده و به منظور روشن نمودن LED ها مورد نظر می‌بایست ولتاژ کاتد آنها را نسبت به آند مشترک کاهش داد. پس از آشنایی با ساختار نمایشگر هفت قسمتی و نحوه عملکرد آن به چگونگی راه‌اندازی و کار با آن می‌پردازیم. به منظور استفاده از نمایشگر هفت قسمتی می‌بایست اعداد BCD را به کدهایی تبدیل کرد که LED های متناسب با عدد مورد نظر را روشن نمایند. با استفاده از جدولی که در شکل ۱-۲ آورده شده است بسادگی می‌توان مدار ترکیبی راه‌اندازی نمایشگر هفت قسمتی را طراحی نمود.

تکالیف پیش از آزمایش

پیش از ورود به آزمایشگاه و آغاز آزمایش مطالب این بخش را مطالعه، موارد خواسته شده را انجام و به سؤالات مطرح شده پاسخ دهید و نتایج بدست آمده را در گزارش خود ثبت نمایید.

۱- با استفاده از گیت های NOT ، OR و AND مدار مبدل BCD به نمایشگر هفت قسمتی را که تنها قادر به نمایش اعداد یک رقمی باشد طراحی نمایید. سپس توسط نرم افزار ISE آن را پیاده‌سازی کرده و خروجی را پس از شبیه سازی با تئوری مقایسه نمایید. آیا جواب بدست آمده درست است ؟

به منظور نمایش اعداد چند رقمی، که به عنوان مثال علاوه بر رقم یکان دارای رقم دهگان نیز می‌باشند چه راه حل و مداری را پیشنهاد می‌نمایید. مدار مورد نظر را با استفاده از گیت‌های منطقی طراحی کرده و در انتها کلیه موارد فوق را برای این مدار نیز اجرا کنید.

تکالیف داخل آزمایشگاه

برنامه مدار مبدل BCD تک رقمی به نمایشگر هفت قسمتی را توسط نرم افزار مورد نظر اجرا و مقدمات پیاده سازی برروی برد را مهیا نمایید.

مراحل پیاده سازی طرح مورد نظر برروی برد را اجرا و نتایج حاصل را برروی نمایشگر هفت قسمتی مشاهده کنید.

بدین منظور با استفاده از DIP SWITCH هایی که در روی برد قرار دارد عدد BCD مورد نظر را وارد نمایید و عدد دسیمال متناظر با آن را بر روی نمایشگر هفت قسمتی مشاهده نمایید.

نتایج حاصل از اجرای برنامه برروی برد را با آنچه قبلاً بدست آورده اید، مقایسه نمایید. مدار مبدل اعداد دو رقمی بر روی نمایشگر هفت قسمتی را برروی FPGA پیاده سازی و نتایج حاصل از اجرای برنامه برروی برد را مشاهده و با آنچه قبلاً بدست آورده اید، مقایسه کنید.

نکته به منظور دستیابی به پایه‌های نمایشگر هفت قسمتی استفاده شده بر روی برد و کسب اطلاعات لازم در مورد ترتیب پایه‌های آن به پیوست مراجعه شود.

آزمایش دوم

جمع کننده ها

هدف در این آزمایش اهداف زیر دنبال می‌شوند :

طراحی و شبیه سازی نیم- جمع کننده، تمام جمع‌کننده و جمع‌کننده ٤- بیتی و مقایسه آن با تئوری.

پیاده‌سازی مدارهای طراحی شده برروی FPGA.

تست مدار پیاده‌سازی شده و بررسی خروجی‌های بدست آمده.

تئوری آزمایش

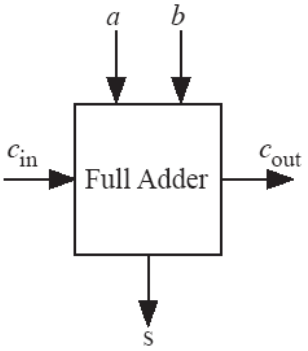
همان طور که از اهداف آزمایش نیز مشخص است، هدف نهایی در این آزمایش طراحی و پیاده سازی جمع کننده ٤ بیتی و ١٦ بیتی می باشد. اما از آنجایی که ساده ترین و پایه ای ترین مدار در طبقه‌بندی مدارهای جمع کننده، مدار نیم- جمع کننده[1] است، لذا در ابتدا به طراحی مدار نیم-جمع‌کننده می‌پردازیم. پس از شبیه‌سازی و پیاده‌سازی این مدار و بررسی خروجی آن به طراحی مدار تمام- جمع‌کننده خواهیم پرداخت.در بخش بعدی با بهره‌گیری از مدار تمام- جمع‌کننده طراحی شده به طراحی مدار جمع‌کننده ٤- بیتی خواهیم پرداخت. لازم به ذکر است که در این بخش نیز با توجه به اینکه خروجی نهایی مدار بر روی نمایشگر هفت قسمتی قابل مشاهده می‌باشد، لذا بکارگیری و استفاده از دیکد کننده دودویی به هفت قسمتی که در آزمایش اول طراحی نمودید را نیز تجربه خواهید نمود.

پیش از انجام آزمایش مختصری به تئوری مربوط به مدارهای جمع‌کننده می پردازیم. همان طور که می‌دانید مدارنیم- جمع‌کننده مداری است که دو عدد تک بیتی را با هم جمع و حاصل را به همراه بیت نقلی تولید می‌نماید. مدار تمام-جمع کننده[2] تک بیتی نیز مداری است که دو عدد تک بیتی را با بیت نقلی ورودی جمع و مجدداً حاصل را به همراه بیت نقلی تولید می‌نماید (شکل١-٣).

---

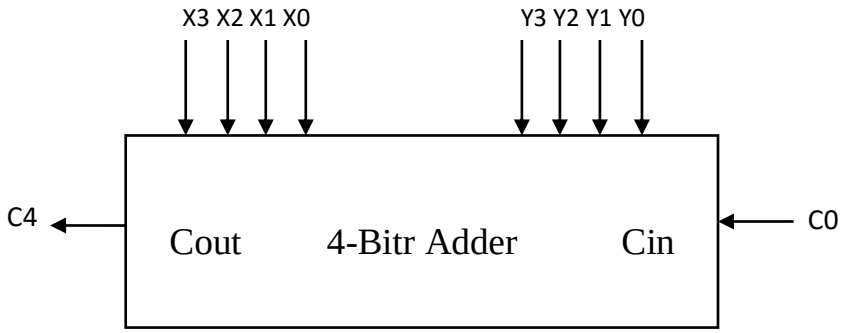
[1] Half-Adder

[2] Full-Adder



شکل ۱-۳ تمام- جمع کننده تک بیتی

تمام-جمع کننده را می‌توان به عنوان یک بلوک پایه در طراحی جمع کننده‌های n-بیتی مورد استفاده قرار داد. به عنوان مثال در طراحی جمع کننده ٤-بیتی از ٤ تمام-جمع کننده استفاده می‌شود. در این جمع کننده دو عدد ٤-بیتی با احتساب بیت نقلی ورودی با هم جمع و حاصل که یک عدد ٤-بیتی و بیت نقلی خروجی است نتیجه می‌شود.



شکل۱-٤ جمع کننده ٤بیتی

تکالیف پیش از آزمایش

پیش از ورود به آزمایشگاه و شروع آزمایش، مطالب این بخش را مطالعه نموده، موارد خواسته شده را انجام داده و به سؤالات مطرح شده پاسخ دهید و نتایج بدست آمده را در گزارش خود ثبت نمایید.

۱- نیم -جمع کننده

۱-۱) با استفاده از گیت های NOT، OR و AND مدار یک نیم- جمع کننده را طراحی نموده ، سپس کد Verilog آن را نوشته و توسط نرم افزار ISE آن را اجرا نمایید. همچنین خروجی آن را پس از شبیه سازی با تئوری مقایسه نمایید. آیا جواب بدست آمده درست است ؟

۲- تمام –جمع کننده

۱-۲) مدار یک تمام- جمع کننده را طراحی نموده و توسط نرم افزار ISE آن را پیاده‌سازی و خروجی حاصل از شبیه سازی را با تئوری مقایسه نمایید، آیا جواب بدست آمده درست است؟

۳- جمع کننده ٤ –بیتی

۱-۳) با استفاده از مدار تمام–جمع کننده یک جمع‌کننده ٤- بیتی طراحی نمایید. ورودی و خروجی های S_0 تا S_3، رقم نقلی خروجی و ارتباط داخلی مابین تمام جمع‌کننده‌ها را مشخص نمایید.

تکالیف داخل آزمایشگاه

۱) نیم جمع کننده

۱-۱) برنامه نیم- جمع‌کننده را توسط نرم افزار ISE اجرا و مقدمات پیاده‌سازی برروی برد را مهیا نمایید.

۱-۲) مراحل پیاده‌سازی طرح مورد نظر بر روی FPGA را اجرا نمایید. بدین منظور با استفاده از DIP SWITCH هایی که در روی برد قرار دارد. خروجی را بر روی LED ها مشاهده نمایید.

۱-۳) نتایج حاصل از اجرای برنامه برروی برد را با آنچه قبلاً بدست آورده‌اید، مقایسه نمایید.

۲) تمام جمع کننده

۲-۱) برنامه تمام– جمع‌کننده را توسط نرم افزار ISE اجرا و مقدمات پیاده‌سازی برروی برد را مهیا نمایید.

۲-۲) مراحل پیاده‌سازی طرح مورد نظر بر روی FPGA را اجرا نمایید. بدین منظور با استفاده از DIP SWITCH هایی که در روی برد قرار دارد ورودی را اعمال و خروجی را بر روی LED ها مشاهده نمایید.

۲-۳) نتایج حاصل از اجرای برنامه برروی برد را با آنچه قبلاً بدست آورده اید، مقایسه نمایید.

۳) جمع کننده ٤ بیتی

۳-۱) برنامه جمع کننده ٤–بیتی را توسط نرم افزار ISE اجرا و مقدمات پیاده سازی برروی برد را مهیا نمایید.

۳-۲) مراحل پیاده‌سازی طرح مورد نظر بر روی FPGA را اجرا نمایید. بدین منظور با استفاده از DIP SWITCH هایی که در روی برد قرار دارد یک عدد باینری مورد نظر را وارد نمایید و عدد دوم را به صورت ثابت و از طریق vcc و gnd وارد کنید. خروجی را بر روی LED ها مشاهده نمایید

به منظور نمایش خروجی جمع کننده ٤–بیتی برروی نمایشگر هفت قسمتی، از دیکد کننده‌ای که در آزمایش اول طراحی نمودید استفاده نمایید.

آزمایش سوم

شمارنده ها

هدف : در این آزمایش اهداف زیر دنبال می‌شوند :

آشنایی با مفاهیم طراحی مدارهای ترتیبی

طراحی و پیاده سازی شمارنده‌ها

تئوری آزمایش

شمارنده ها ، در طراحی سیستم های دیجیتال به طور عام و بویژه در پردازشگرها بکار می روند. این مدارها برای تولید مقادیر زمانی به منظور تنظیم و کنترل عملیات در یک دستگاه دیجیتال مفید هستند. یک شمارنده ضرورتاً یک ثبات است که به ازای پالسهای ورودی به آن، دنباله‌ای از حالتهای از پیش تعیین شده را ایجاد می‌کند. شمارنده‌ها را می توان به دو دسته سنکرون و آسنکرون (موج گونه) طبقه بندی نمود.

اما نکته بسیار مهم در مدارهای ترتیبی، نحوه طراحی آنها است. در طراحی مدارهای ترتیبی برخلاف مدار ترکیبی که با ترسیم یک جدول درستی طراحی مدار کاملاً مشخص می‌شود، به جدول و دیاگرام حالت نیز نیاز است. طراحی مدار ترتیبی شامل تعیین فلیپ فلاپ و سپس یافتن ساختار مدار ترکیبی است که همراه با فلیپ فلاپ ها نیازهای تعیین شده را برآورده می سازد. روش طراحی مطرح شده شامل مراحل زیر است:

۱- توصیف عملکرد مدار توسط دیاگرام حالت یا دیاگرام زمانی.

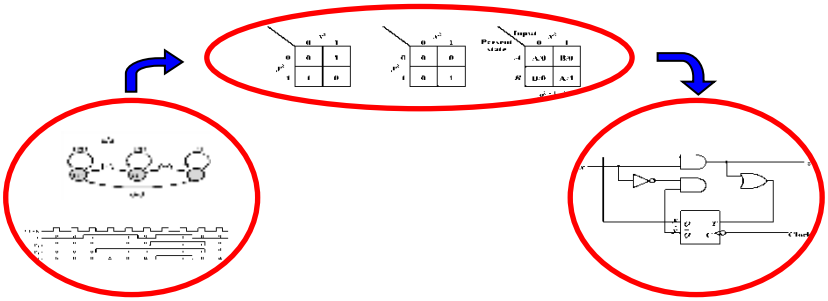
۲- تهیه جدول حالات با توجه به اطلاعات مفروض.

۳- در صورت امکان جدول حالات را ساده نمائید.

٤- تعیین تعداد و نوع فلیپ فلاپ‌های مورد نیاز و اختصاص یک سمبل به هر کدام.

۵- با بکارگیری روش نقشه یا هر روش ساده سازی دیگر، توابع خروجی مدار و توابع ورودی فلیپ فلاپ ها را بدست آورید.

٦- دیاگرام منطقی مدار را ترسیم نمایید.



شکل ۱-۵

تکالیف پیش از آزمایش

پیش از ورود به آزمایشگاه و شروع آزمایش مطالب این بخش را مطالعه، موارد خواسته شده را انجام و به سؤالات مطرح شده پاسخ دهید و نتایج بدست آمده را در گزارش خود ثبت نمایید.

۱) طراحی شمارنده ٤ – بیتی سنکرون (۳– افزا)

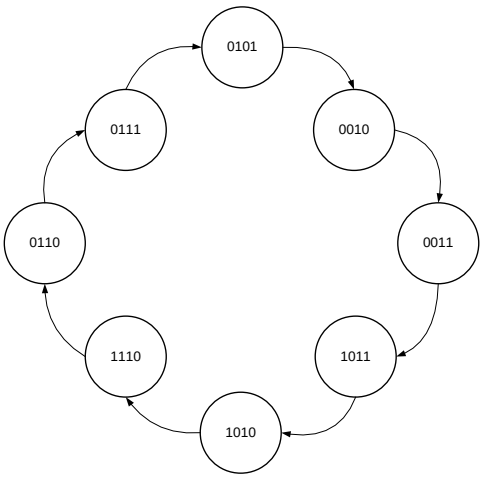
۱–۱) با استفاده از فلیپ فلاپ های نوع JK شمارنده ٤ – بیتی سنکرونی طراحی نمایید که دارای یک ورودی X باشد. این ورودی جهت شمارش رو به بالا (X=1) و رو به پایین (X=0) را تعیین می‌نماید. شمارنده مورد نظر ۳– افزا است . بدین معنی که باید اعداد ۰ تا ۱۵ را به طوری که اختلاف دو عدد متوالی همواره ۳ باشد، بشمارد.

۱–۲) برنامه شمارنده مورد نظر را نوشته و تست نمایید.

۱–۳) برنامه را به منظور نمایش خروجی برروی نمایشگر هفت قسمتی تکمیل نمایید.

۲) طراحی شمارنده اختصاصی (شبیه کد گری)

۲–۱) مدار شمارنده‌ای را طراحی نمایید که دیاگرام حالت آن به صورت زیر باشد. در طراحی خود از فلیپ فلاپ های نوع D استفاده کنید. مراحل طراحی خود را در گزارش ثبت نمایید.



شکل ۱-٦

۲-۲) برنامه مدار طراحی شده را نوشته و خروجی مدار را توسط شبیه سازی مشاهده و تست نمایید.

۲-۳) برنامه را به منظور نمایش خروجی برروی نمایشگر هفت قسمتی تکمیل نمایید.

راهنمایی: مدار مورد نظر یک شمارنده دو دویی می‌باشد که تنها اعداد ۵ , ۷ , ٦ , ١٤ , ١٠ , ١١ , ٣ , ٢ را می شمارد . این مدار نوعی کد کننده گری[۳] است.

تکالیف داخل آزمایشگاه

نکات : دانشجویان عزیز برای مشاهده عملکرد مدارهای طراحی شده باید مقدمات زیر را فراهم نمایند.

۱) برای قابل رویت شدن نتایج از یکی از push button های موجود در برد بعنوان پالس ساعت استفاده کنید

۲) برروی برد LED ها و نمایشگرهای هفت قسمتی وجود دارند. از این LED ها و نمایشگرها به منظور نمایش خروجی شمارنده ها استفاده نمایید.

۳) از سوئیچ‌های موجود بر روی برد به منظور فرمان دادن یا بار نمودن شمارنده‌ها استفاده نمایید.

۱- شمارنده ۳- افزا طراحی شده را برروی برد پیاده سازی نموده و با استفاده از سوئیچ های موجود برروی برد نحوه شمارش (بالا شمار - پایین شمار) را مشخص و نتایج حاصل را برروی نمایشگرهای هفت قسمتی مشاهده نمایید.

۲- شمارنده اختصاصی کد گری طراحی شده را برروی برد پیاده سازی نموده و نتایج حاصل را برروی نمایشگرهای هفت قسمتی مشاهده نمایید.

---

[۳] Gray Coder

آزمایش چهارم

ثبات۱

هدف

در این آزمایش اهداف زیر دنبال می‌شوند :

طراحی و پیاده سازی یک ثبات ۱۶ بیتی که به شکل زیر باشد :

با قابلیت متمم یک

با قابلیت متمم دو

با قابلیت افزایشی یک

با قابلیت کاهشی یک

تئوری آزمایش

در این آزمایش شما با کاربردهای دیگری از ثبات‌ها آشنا خواهید شد. این کاربردها که در بالا آمده است در قسمت‌های مختلف یک سیستم دیجیتال مورد استفاده قرار می‌گیرند. به‌طور مثال در سیستم کامپیوتر پایه برای انجام عملیات افزایش یک یا کاهش یک به راحتی در یک سیکل با سیگنال‌های INC یا DEC انجام خواهندشد. طراحی چنین ثباتی که دارای قابلیت‌های فوق است، در سیکل‌های اجرائی بسیاری از دستورالعمل‌ها صرفه‌جویی خواهدنمود.

شکل ۷-۱

تکالیف پیش از آزمایش

۱) در این آزمایش ابتدا یک ثبات ٤ بیتی با قابلیت افزایشی یک با استفاده از فلیپ فلاپ D طراحی نمایید.

۲) مرحله ۱ را به‌طور کامل برای سه کاربرد دیگر ، طراحی و تست می‌نمایید.

یک ثبات طراحی و تست نمایید که دارای هر چهار قابلیت فوق یعنی متمم یک، متمم ۲ ، افزایشی یک و کاهشی یک باشد.

تکالیف داخل آزمایشگاه

نکات : دانشجویان عزیز برای مشاهده عملکرد مدارهای طراحی شده باید مقدمات زیر را فراهم نمایند.

برروی برد، از LED ها به عنوان بیت‌های خروجی ثبات‌ها استفاده نمایید.

از سوئیچ‌های موجود برروی برد به عنوان سیگنالهای کنترلی استفاده نمایید.

برنامه ثبات با قابلیت های فوق را برروی برد پیاده سازی و تست نمایید. نتایج بدست آمده را در گزارش خود ثبت نمایید.

آزمایش پنجم

ثبات۲

هدف

در این آزمایش اهداف زیر دنبال می‌شوند :

طراحی و پیاده سازی یک ثبات ٤ بیتی که به شکل زیر باشد :

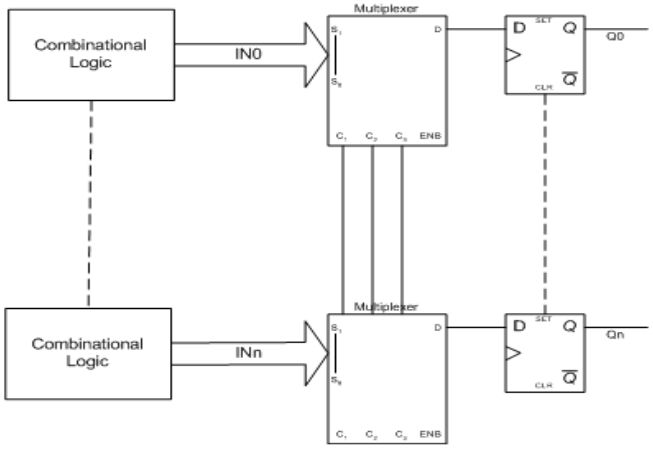
Circulate Right shift, Circulate Left shift
Arithmatic Right shift, Arithmatic Left shift
One’s complement, Two’s complement
INC, DEC
Clear

تئوری آزمایش

در این آزمایش شما می‌توانید یک ثبات با قابلیت‌های فوق طراحی نمایید. همچنانکه در آزمایش قبل ذکر شد، داشتن چنین ثباتی باعث کاهش سیکل های اجرائی یک دستورالعمل در کامپیوتر می‌گردد. به عبارت دیگر ریز دستورالعمل‌های مورد نیاز برای یک دستورالعمل کاهش پیدا می‌کند.

تکالیف پیش از آزمایش

در این آزمایش با استفاده از Multiplexer ورودی های فلیپ فلاپ D مشخص می‌گردد. شکل ۸-۱ بلوک دیاگرام کلی مدارات مورد نظر را نمایش می دهد. مدار مورد نظر را ترسیم ، برنامه آن را نوشته و خروجی را تحلیل نمایید.



شکل۸-۱ بلوک دیاگرام کلی مدارات مورد نظر

تکالیف داخل آزمایشگاه

نکات : دانشجویان عزیز برای مشاهده عملکرد مدارهای طراحی شده باید مقدمات زیر را فراهم نمایند.

برروی برد، از چهار LED به عنوان بیت‌های خروجی ثبات‌ها استفاده کرده و ثبات را ٤ بیتی طراحی کنید. از سوئیچ-های موجود برروی برد به عنوان سیگنالهای کنترلی Multiplexer استفاده نمایید.

به علت محدودیت DIPSWITCH برای بار کردن ثبات ها پین ورودی در نظر نگیرید، با INC و COMPLEMENT می‌توان ثبات‌ها را مقداردهی کرد.

برنامه ثبات با قابلیت های فوق را برروی برد پیاده سازی و تست نموده و نتایج بدست آمده را در گزارش خود ثبت کنید .

آزمایش ششم

واحد محاسبه و منطق[4]

هدف

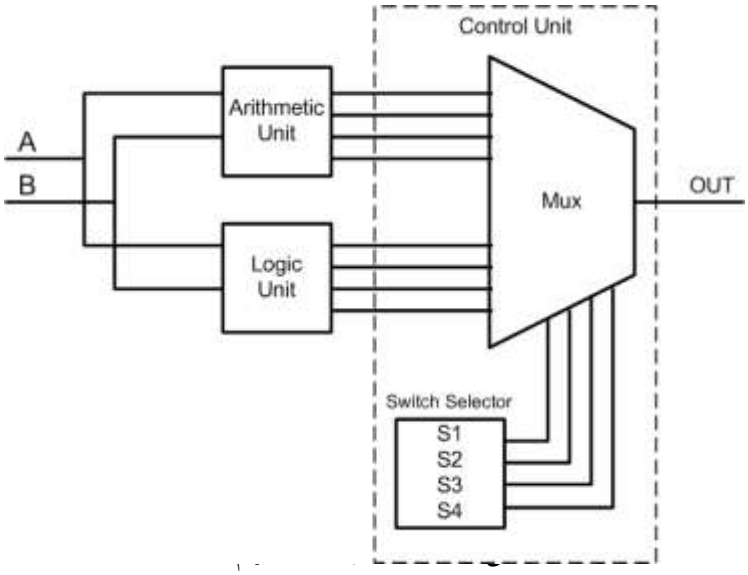
در این آزمایش اهداف زیر دنبال می‌شوند :

آشنایی با ALU و ساختار آن

طراحی و پیاده سازی مدارهای محاسباتی ، منطقی و ثباتی یک CPU

تئوری آزمایش

بخش محاسبات و منطق یکی از مهمترین قسمت های یک CPU می‌باشد. به طور کلی ALU از سه قسمت محاسبات، منطق و کنترل تشکیل شده است. هر بخش با توجه به وظایفی که دارد یک سری عملکردهایی را انجام می‌دهد. قسمت محاسباتی اعمالی نظیر جمع، تفریق، ضرب و تقسیم را بر عهده دارد. در قسمت منطقی عملیاتی نظیر NOT ,XOR ,OR ,AND صورت می‌پذیرد و در نهایت قسمت کنترل نیز وظیفه تعیین واحد عملیاتی و عملیات مورد نظر را بعهده دارد. در این آزمایش هدف پیاده سازی یک ALU بسیار ساده می‌باشد. شکل ۹-۱ بلوک دیاگرام سیستم مورد نظر را نمایش می‌دهد.



شکل**۹-۱ دیاگرام واحد محاسبه و منطق**

[4] ALU

پیش از آغاز آزمایش لازم می‌دانیم تا کمی بیشتر در مورد بلوک‌های این سیستم به بحث بپردازیم. در بخش محاسباتی دانشجویان عزیز می‌بایست از بلوک جمع کننده‌ای که در آزمایش۲ طراحی شد به عنوان جمع کننده ٤ – بیتی استفاده نمایند. با استفاده از همان مدار نیز می‌توانند افزایش به اندازه یک واحد را نیز پیاده سازی نمایند. اما مدار تفریق کننده و کاهش به اندازه یک واحد را که در آزمایش‌های گذشته به آنها نپرداخته‌اند می‌بایست از ابتدا طراحی نمایند. البته طراحی تفریق کننده ٤ – بیتی نیز بر پایه تمام جمع کننده امکان پذیر است و از اینرو با جمع کننده ٤ – بیتی اختلاف بسیار کوچکی دارد.

به منظور سادگی سیستم بخش کنترلی تنها به عنوان یک مالتی‌پلکسر عمل می می‌نماید. ورودی این بلوک حاصل کلیه اعمال مورد نظر (محاسباتی: جمع، تفریق، ... و منطقی: AND، OR، ...) است و با توجه به سیگنال کنترلی، ورودی مورد نظر به خروجی منتقل می‌شود.

جدول زیر کدهای عملیاتی را که توسط بخش محاسبات و منطق انجام می‌شود نمایش می‌دهد.

دانشجویان با استفاده از سوئیچ‌های موجود برروی برد عملکرد مورد نظر را مشخص می‌نمایند. لازم به ذکر است که تمام عملیات های مورد نظر ٤ بیتی می‌باشند

operation	S3	S2	S1	Carry in
Add	0	0	0	0
Add with carry	0	0	0	1
Sub with borrow	0	0	1	0
Sub	0	0	1	1
A	0	1	0	0
INC A	0	1	0	1
DEC A	0	1	1	0
A	0	1	1	1
AND	1	0	0	X
OR	1	0	1	X
XOR	1	1	0	X
NOT A	1	1	1	X

تکالیف پیش از آزمایش

تا کنون کلیه مدارهای مورد نیاز در این بخش در آزمایش‌های قبل پیاده سازی شده‌اند، بجز تفریق کننده ٤ – بیتی که می‌بایست برنامه آن را پیاده سازی نمایید.

تکالیف داخل آزمایشگاه

نکات : دانشجویان عزیز برای مشاهده عملکرد مدارهای طراحی شده باید مقدمات زیر را فراهم نمایند.

برروی برد از چهار عدد LED و نمایشگر هفت قسمتی به منظور نمایش خروجی ALU استفاده نمایید.

از سوئیچ‌های موجود بر روی برد به عنوان سیگنال‌های کنترلی و رقم نقلی ورودی استفاده نمایید، دو عدد ٤ بیتی A و B را به صورت ثابت و از طریق vcc و gnd وارد کنید.

برنامه مدار ترسیم شده را نوشته و برروی FPGA برنامه ریزی نمایید.

تست های لازم را انجام و نتایج را در گزارش خود ثبت نمایید.

آزمایش هفتم

گذرگاه داده[۵]

هدف

در این آزمایش اهداف زیر دنبال می شوند :

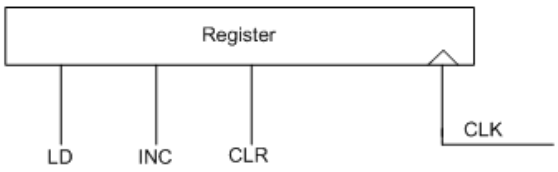
آشنایی با BUS و ساختار آن

طراحی و پیاده سازی مدارهای محاسباتی ، منطقی و ثباتی یک CPU.

تئوری آزمایش

BUS یکی از وا سط‌های بسیار مهم در سیستم‌های دیجیتالی است. در این سیستم‌ها برای ارسال و دریافت داده از BUS به عنوان محیط انتقال استفاده می‌گردد. همانطور که در معماری کامپیوتر با آن آشنا شده‌اید به دو طریق BUS ها قابل پیاده‌سازی هستند. آن دو راه به کارگیری Multiplexer و استفاده از بافر tri-state می‌باشد. در کامپیوترها به علت دوطرفه بودن BUS امکان استفاده از Multiplexer برای یک خط که خواندن و نوشتن از روی آن صورت می‌گیرد امکان‌پذیر نمی‌باشد. بنابراین این عمل از طریق tri-state انجام خواهدشد.

درفصل‌های قبل با نحوه طراحی ثبات ها آشنا شدید. در سیستم کامپیوتر پایه یک ثبات ممکن است دارای قابلیت‌های مختلفی باشد. اما در کامپیوتر پایه‌ای که برای این آزمایشگاه طراحی می شود یک ثبات باید قابلیت های Increment، LD ، Clear و Read را دارا باشد (شکل۱۰-۱) .



شکل۱۰-۱ ثبات داخلی کامپیوتر پایه‌ای

لازم به ذکر است که ثباتهای مذکور ثبات های داخلی کامپیوتر و مورد استفاده در قسمت کنترلی و BUS خواهدبود.

---

[۵] BUS

نکته بسیار مهم نحوه ارتباط این ثبات ها با BUS سیستم میباشد. همانطور که می‌دانید BUS داده به علت دو طرفه بودن، خواندن و نوشتن از یک مسیر صورت خواهد گرفت. بنابراین هر ثبات باید در زمان مورد نیاز BUS را در اختیار گیرد و سپس بعد از اتمام ارسال یا دریافت آن را رها نماید. با توجه به مطالب فوق در قسمت ورودی هر ثبات با سیگنال کلاک و LD مشکل وارد شدن داده جدید به ثبات حل می‌شود.

اما در قسمت خروجی باید در مسیر هر بیت خروجی یک tri-state قرار گیرد که با سیگنال Read هر ثبات فعال شده و بعد از خواندن، خروجی ثبات به صورت امپدانس بالا درآید. با طراحی چنین ثباتی به راحتی می‌توان تعداد زیاد ثبات را که از یک BUS واحد استفاده می‌نمایند به یکدیگر متصل نمود.

تکالیف پیش از آزمایش

۱- ثباتی ۱۶ بیتی مطابق با شکل۱۰-۱ طراحی نمایید. در این ثبات از فلیپ فلاپ‌های D استفاده میشود و در قسمت خروجی باید در مسیر هر بیت خروجی یک tri-state قرار گیرد که با سیگنال Read فعال شده و بعد از خواندن، خروجی ثبات به صورت امپدانس بالا درآید.

تکالیف داخل آزمایشگاه

۱- ۱۶ ثبات را با استفاده از ثباتی که در بند اول طراحی نموده‌اید، پیاده‌سازی نمایید و از طریق BUS طراحی شده ارتباط بین ثبات ها را برقرار نمایید. به دلیل محدودیت DIP SWITCH، سیستم مورد نظر را در simulator شبیه سازی کنید.

ضمیمه

در این بخش اطلاعات لازم در مورد آدرس پین‌های FPGA ارائه می‌شود. برد موجود یک LCD و دو 7segment ۲بیتی جهت نمایش اطلاعات خروجی در اختیار کاربر قرار می‌دهد. نکته حائز اهمیت این است LCD و 7segment همزمان با هم در مدار نیستند و در هر برنامه‌ای یکی از این دو را در اختیار خواهید داشت. نحوه انتخاب این دو از طریق Jumper-JP5 به شکل زیر است.

Jumper Position	Selected Mode
1-2 & 3-4	LCD Display Mode
5-6 & 7-8	7-Segment Display Mode

Jumper settings for display mode select (JP5)

در برد ۸ عددLED برای نمایش باینری اطلاعات وجود دارد. آدرس پین‌های آنها در جدول زیر آمده است:

FPGA Pin #	FPGA Pin Name		FPGA Pin #	FPGA Pin Name
LED0	P183		LED4	P189
LED1	P184		LED5	P190
LED2	P185		LED6	P191
LED3	P187		LED7	P194

LED Pin-out

همچنین دو ماژول 7segment دو بیتی آند مشترک نیز وجود دارد که آدرس پین‌های آنها در جدول زیر آمده است. چند نکته در این بین لازم به ذکر است:

- چون 7segment ها آند مشترک هستند برای روشن کردن هر segment باید خروجی پین متناظر **0** نوشته شود.
- به دلیل صرفه جویی در استفاده از پین های FPGA از یک باس مشترک برای هر ۴ مجموعه 7segment ها استفاده شده لذا نمی‌توان همه آنها را با هم مقداردهی کرد. برای فعال کردن هر رقم باید common مربوطه را فعال کرد. در صورت لزوم به استفاده همزمان از هر ۴ رقم باید از multiplexing زمانی بهره جست.

FPGA Pin #	FPGA Pin Name	7-Segment Pin Name
P95	DISP1	A
P94	DISP2	F
P85	DISP3	E
P86	DISP4	D
P87	DISP5	C
P90	DISP6	B
P81	DISP7	G
P80	DISP8	Dot Point
P97	DISP9	Common_2
P100	DISP10	Common_1
P93	DISP11	Common_4
P96	DISP12	Common_3

Seven-Segment Module Pin-out

در صورت استفاده از LCD نیز pinout آن بصورت زیر است.

FPGA Pin #	FPGA Pin Name	LCD Pin #	LCD Pin Name
P95	DISP1	7	DB0
P94	DISP2	8	DB1
P85	DISP3	12	DB5
P86	DISP4	13	DB6
P87	DISP5	14	DB7
P90	DISP6	9	DB2
P81	DISP7	11	DB4
P80	DISP8	10	DB3
P97	DISP9	6	E
P100	DISP10	4	RS
---	---	1	GND
---	---	2	VCC
---	---	3	VO
---	---	5	R/W

LCD Module Pin-out

و در انتها pinout برای ورودی‌های DIPSwitch و PUSH Button های برد به فرم زیر خواهد بود:

FPGA Pin #	DIP Switch #
P196	DIP1
P197	DIP2
P198	DIP3
P199	DIP4

FPGA pin-out for DIP Switches

FPGA Pin #	Push-Button #
P200	PB1
P203	PB2
P204	PB3
P205	PB4

FPGA pin-out for Push-Buttons