

به نام خدا

گزارشکار آزمایشگاه معماری کامپیوتر

آزمایش های جلسه دوم

عناوین آزمایش ها :

BCD to 7-Segment و Full Adder

نام استاد :

استاد علی جوادی

اعضای گروه:

غزل عربعلی (97521396) - بهاره کاوسی نژاد (99431217)

آزمایش اول : Full Adder

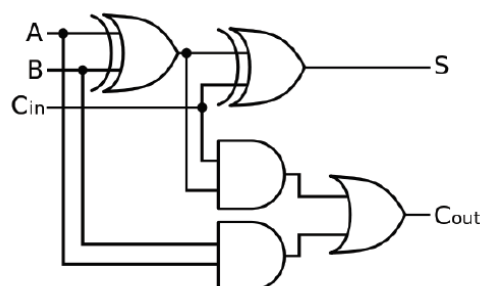
هدف آزمایش :

اعمال مدار Full Adder روی AVA Dev. board و نمایش جمع ارقام ورودی

ابزار آزمایش :

- AVA Dev. Board
- Programmer interface
- ISE Design Suite 14.7

تئوری آزمایش :



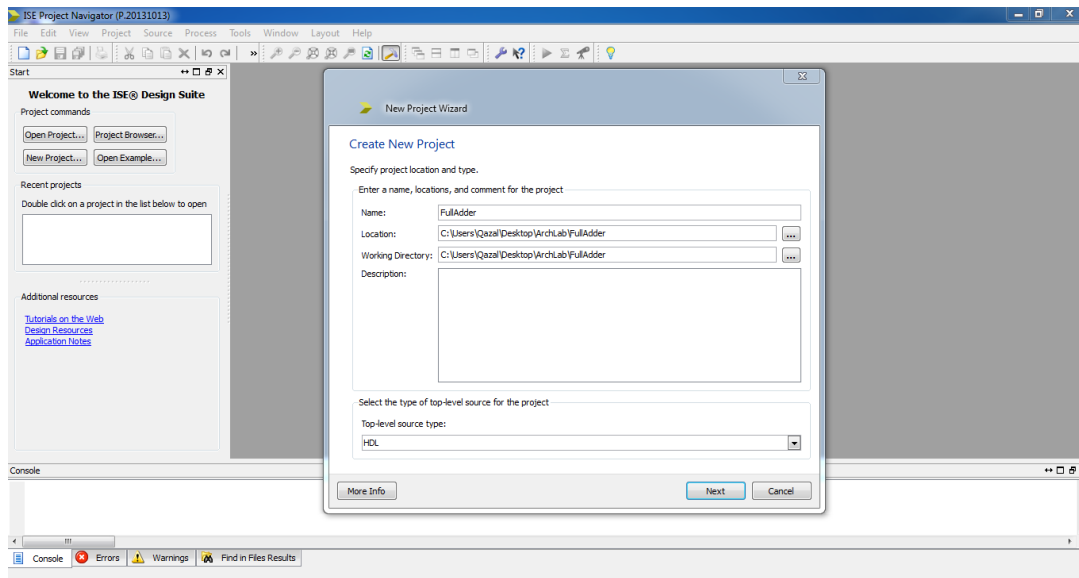
یک Full Adder سه بیت به عنوان ورودی دریافت میکند که در این شکل A ، B و CIN هستند. A و B ، عملوند ها هستند و CIN همان Carry ورودی است. خروجی های این مدار SUM و COUT هستند که SUM مقدار جمع و COUT مقدار Carry خروجی را مشخص می کند.

جدول درستی :

Inputs			Outputs	
A	B	C _{in}	S	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

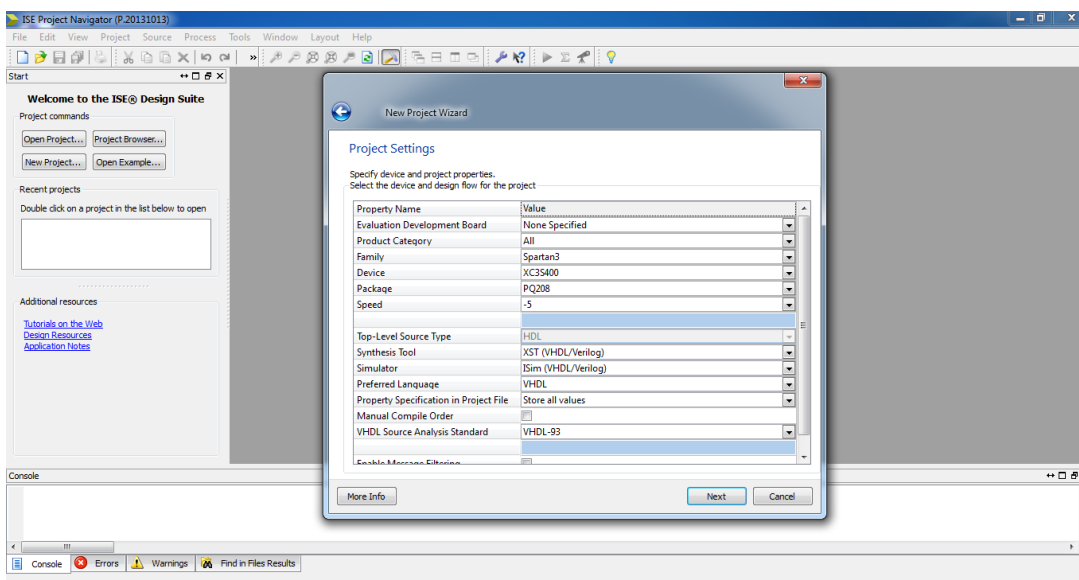
مراحل آزمایش :

1. ایجاد پروژه جدید در ISE:



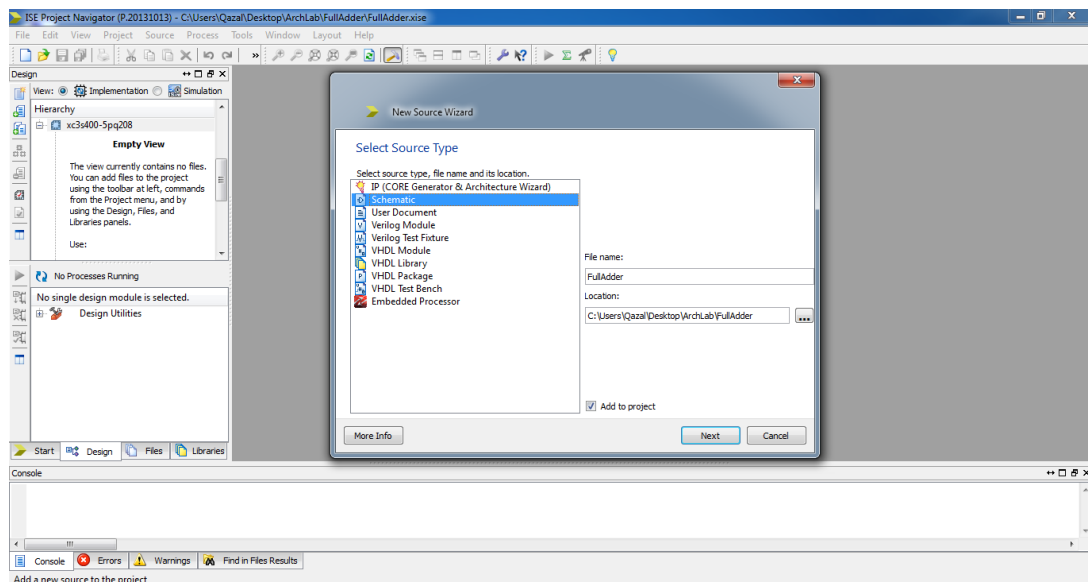
2. وارد کردن مشخصات AVA Board:

در این مرحله باید 4 مورد Family, Device, Package و Preferred Language را مانند تصویر زیر مقدار دهی کنیم. این مقادیر بر اساس FPGA موجود مشخص می شوند.

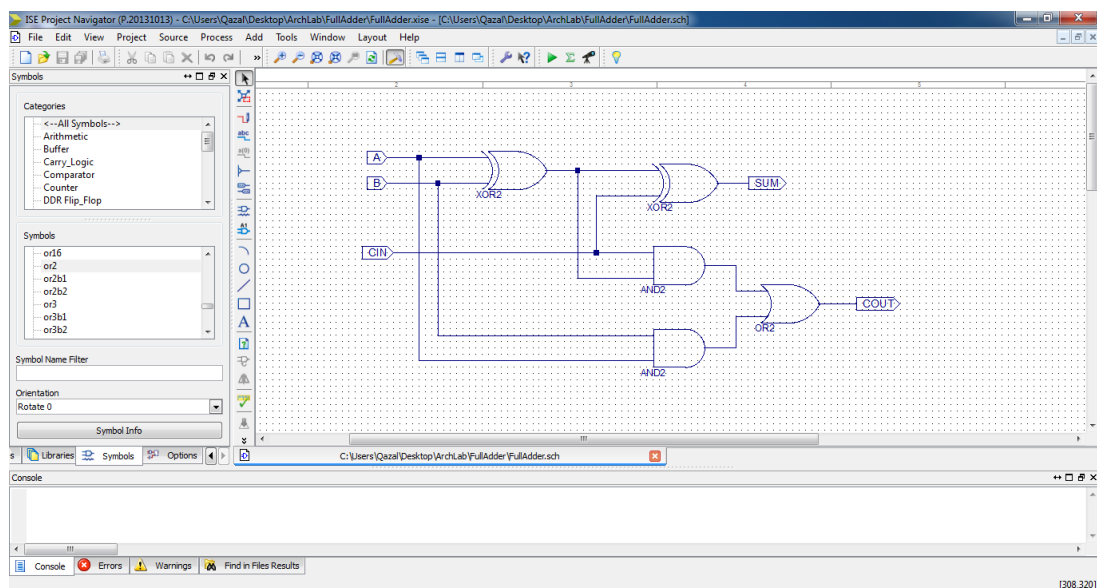


3. انتخاب نوع source:

هدف این آزمایش رسم مدار به صورت شماتیک بوده و نه با نوشتن کد به زبان VHDL؛ در نتیجه مثل عکس زیر گزینه Schematic را انتخاب می‌کنیم.



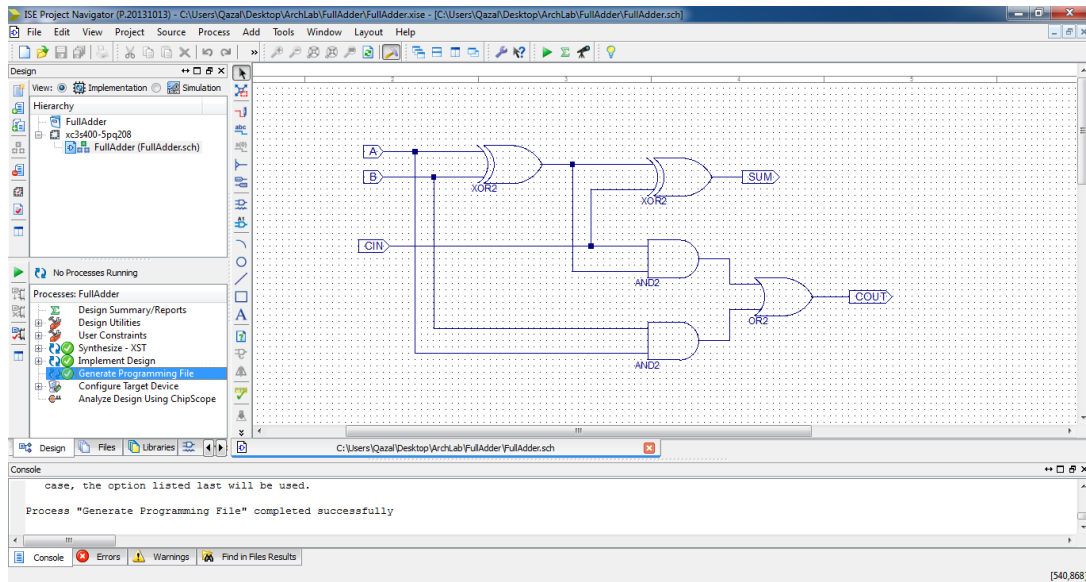
4. رسم مدار طبق جدول درستی و شکل رسم شده:



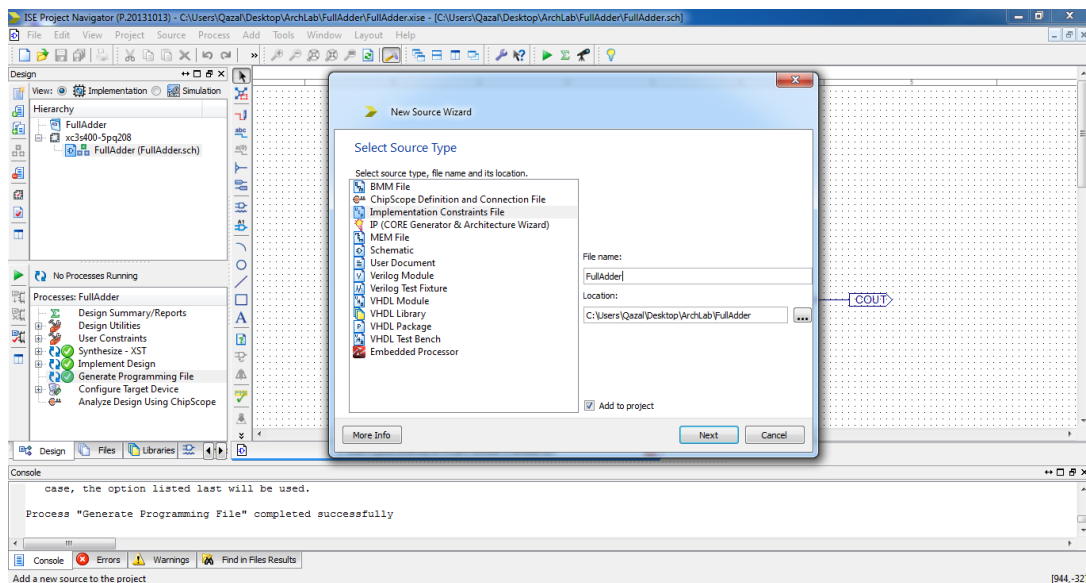
5. Build کردن پروژه در سه مرحله:

طبق عکس زیر در سه مرحله پروژه را build می کنیم که هر سه مرحله run کرده و باید از هر سه مرحله بدون خطا گذر کنیم.

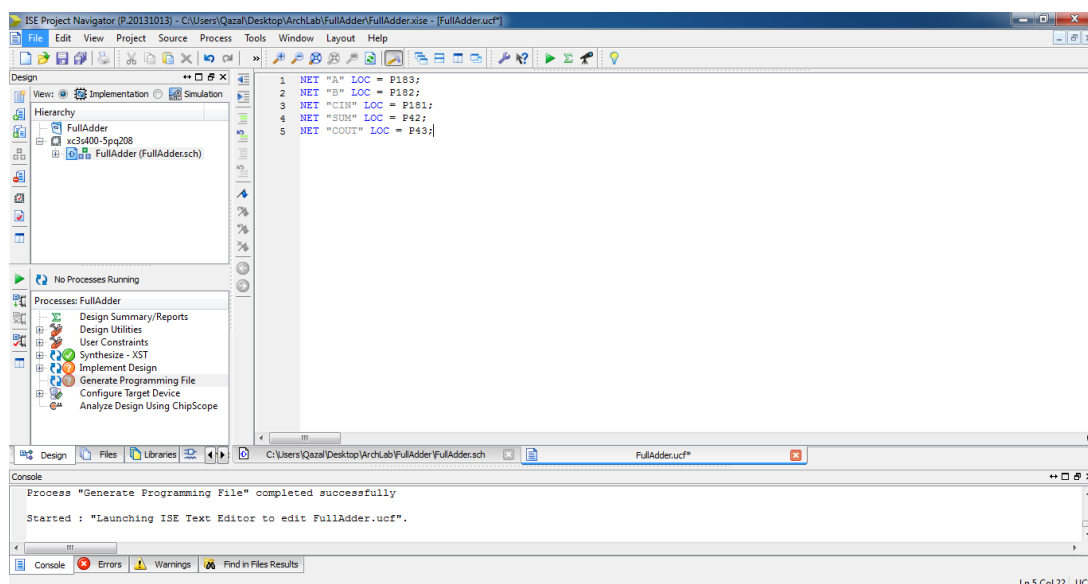
- Synthesize
- Implement Design
- Generate Programming File



6. ساخت فایل ucf برای نداشت ورودی و خروجی های مدار به پورت های موجود در Board:



7. نگاشت ورودی و خروجی های مدار با پورت های موجود در Board در فایل FullAdder.ucf



8. اتصال سیستم به Board:

با استفاده از کابل سیستم را به Programmer interface متصل می‌کنیم و آن را از طریق پورت VGA به Board وصل می‌کنیم.

9. دانلود کردن طراحی روی FPGA

با استفاده از ابزار IMPACT این مرحله را انجام می‌دهیم.

10. تست صحت طراحی

در مرحله آخر با دادن ورودی های متفاوت انتظارات خود را با خروجی های مد نظر مقایسه کرده تا از صحت کارکرد طراحی اطمینان حاصل کنیم.

بحث و نتیجه گیری :

پس از اتصال مدار به board، مشاهده کردیم که به ازای ورودی های تک بیتی، خروجی و carry متاسب توسط LED های موجود روی board به نمایش در می آمدند.

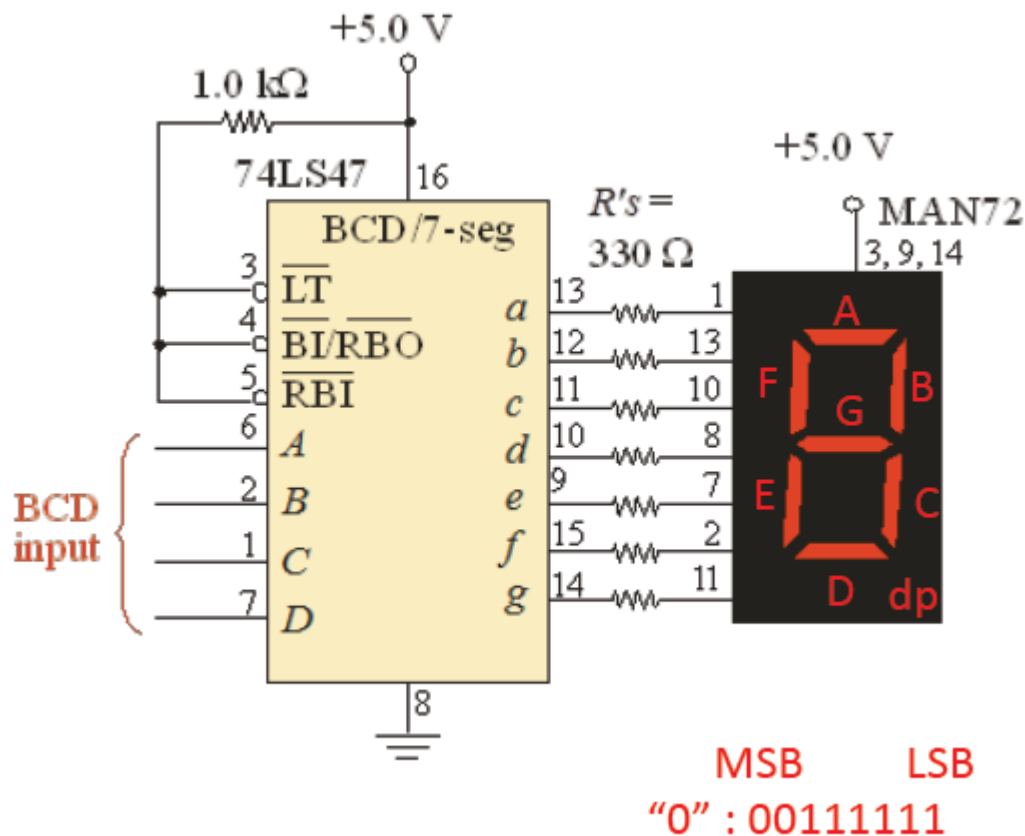
آزمایش دوم : BCD to 7-Segment

هدف آزمایش :

اعمال مدار BCD to 7-Segment روی AVA Dev. Board و نمایش ارقام ۰ تا ۹ به شکل

7-Segment

تئوری آزمایش :



در BCD یا همان Binary Coded Decimal ، هریک از ارقام ۰ تا ۹ به صورت معادل باینری آنها که ۴ بیتی است نمایش داده میشوند . 7-Segment نیز یک دستگاه الکترونیکی است که دارای هفت LED میباشد که برای نمایش اعداد Hexadecimal استفاده میشود. در این آزمایش تنها ارقام ۰ تا ۹ را نمایش میدهم.

دو مدل نمایش 7-Segment وجود دارد:

1. Common Cathode Type: در این مدل نمایش همه کاتوهای هفت LED با هم به زمین یا V_{cc} - متصل هستند و LED، رقم ها را هنگامی که یک سیگنال High به آنودها اعمال شود، نمایش می دهد.

2. Common Anode Type: در این مدل نمایش همه آنودهای هفت LED با هم به باتری یا V_{cc} + متصل هستند و LED، رقم ها را هنگامی که یک سیگنال Low به کاتودها اعمال شود، نمایش می دهد.

در ابتدا باید رقم decimal به معادل BCD آن تبدیل شود و سپس یک BCD to 7-Segment decoder، آن را به سیگنال هایی تبدیل می کند که هر بیت آن روشن یا خاموش بودن یکی از LEDها را تعیین می کند.

جدول درستی برای مدل نمایش Common Anode:

BCD Inputs				Output Logic Levels from IC 7447 to 7-segments							Decimal number display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

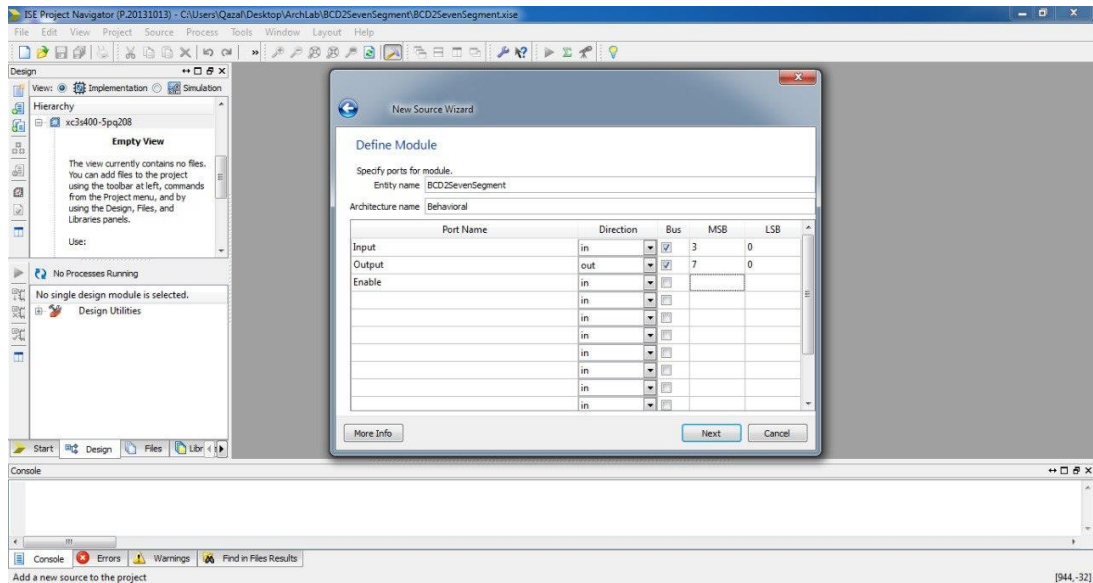
مراحل آزمایش:

1. مراحل ابتدایی:

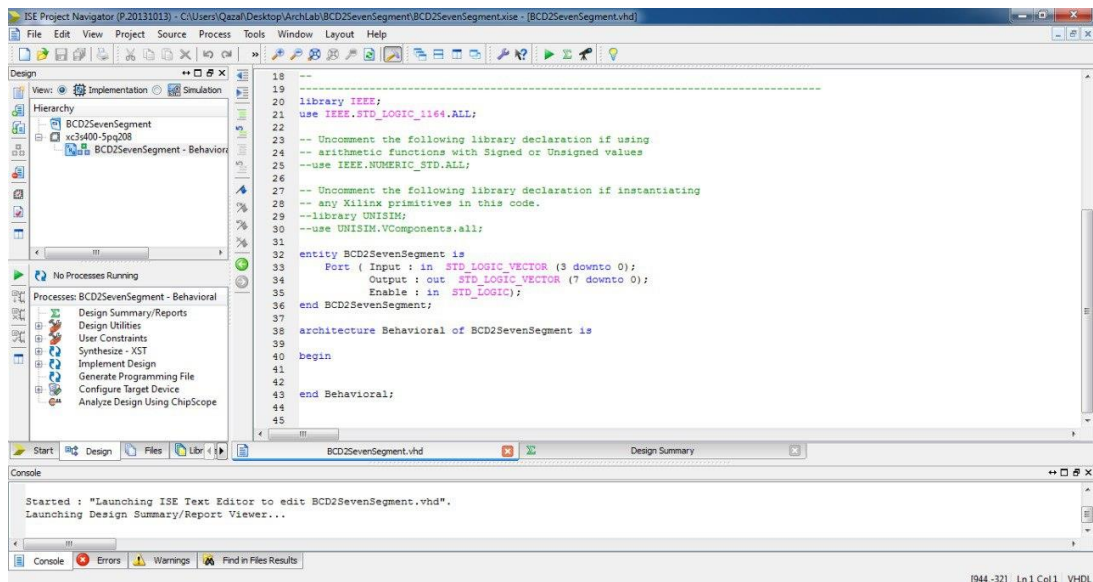
مراحل ابتدایی پروژه یعنی ساخت پروژه و وارد کردن مشخصات AVA Board مانند مثال قبل انجام می شود. در مرحله بعدی که انتخاب نوع source است، VHDL Module را انتخاب می کنیم.

2. تعیین ورودی و خروجی:

در این آزمایش از یک bus چهار بیتی به عنوان ورودی، یک bus هشت بیتی به عنوان خروجی و یک ورودی enable استفاده می کنیم. بدین منظور مانند شکل زیر عمل می کنیم:

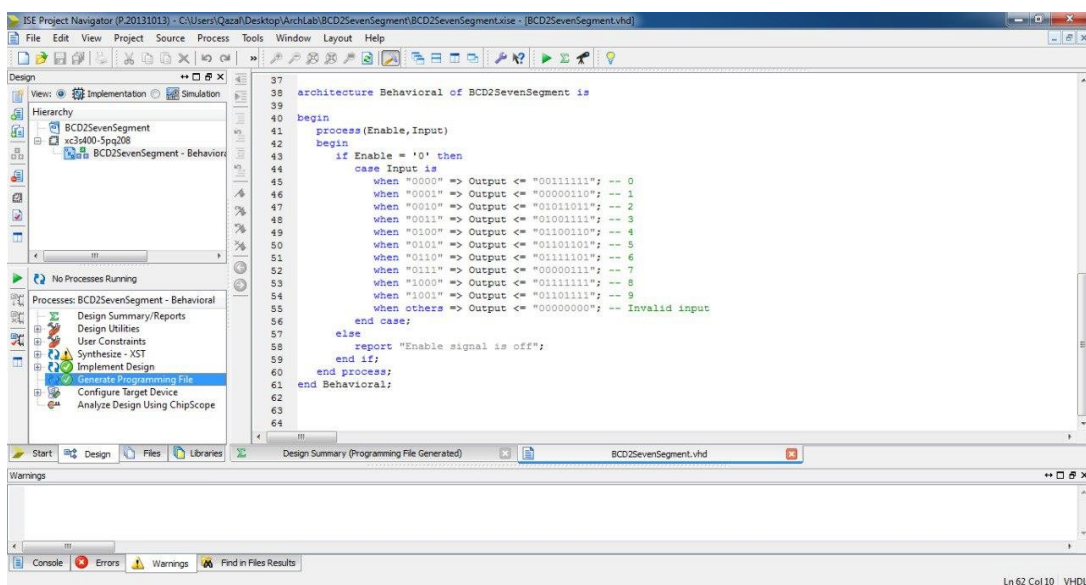


هشت بیت خروجی هریک نمایانگر یکی از LED ها از A تا G هستند و H نیز نقطه را نمایش می دهد. پس از ساخت پروژه، کد vhd زیر به صورت خودکار و بر اساس ورودی و خروجی های داده شده تولید می شود.



3. BCD to 7-Segment Functionality

منطق و کد تبدیل BCD به 7-segment را به زبان VHDL در میان begin و Behavioral end می نویسیم. فقط باید توجه کنیم که ارقام متناظر با LEDهای A تا G از راست به چپ هستند و 1 نمایانگر روشن بودن LED و 0 نمایانگر خاموش بودن آن است.

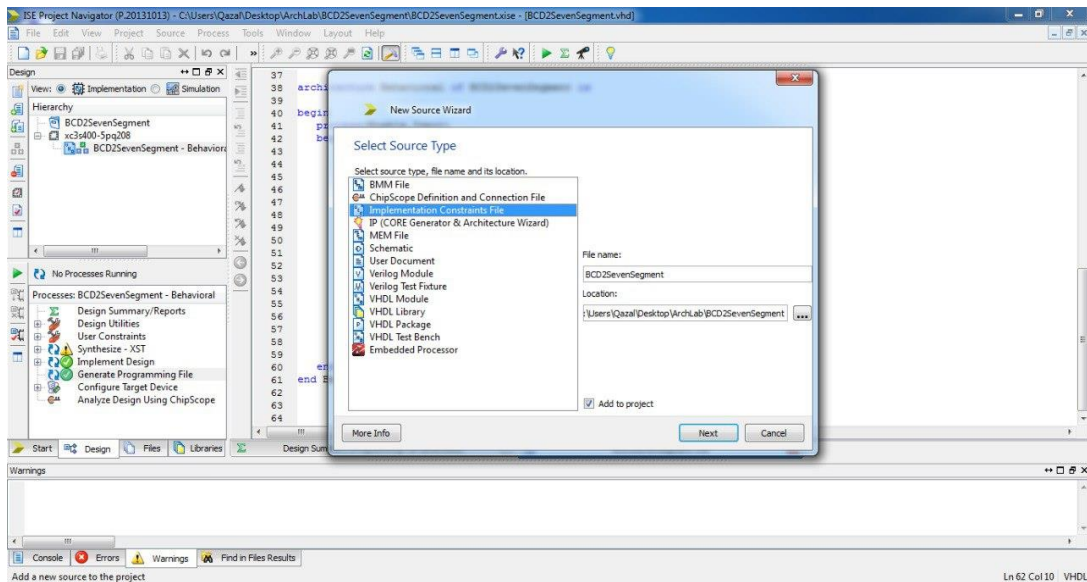


4. بیلد کردن پروژه در سه مرحله و ساخت فایل ucf

در این مرحله نیز دقیقاً مانند آزمایش قبل باید بدون خطا از سه مرحله Synthesize، Implement Design و Generate Programming File گذر کنیم و همچنین فایل ucf را می‌سازیم.

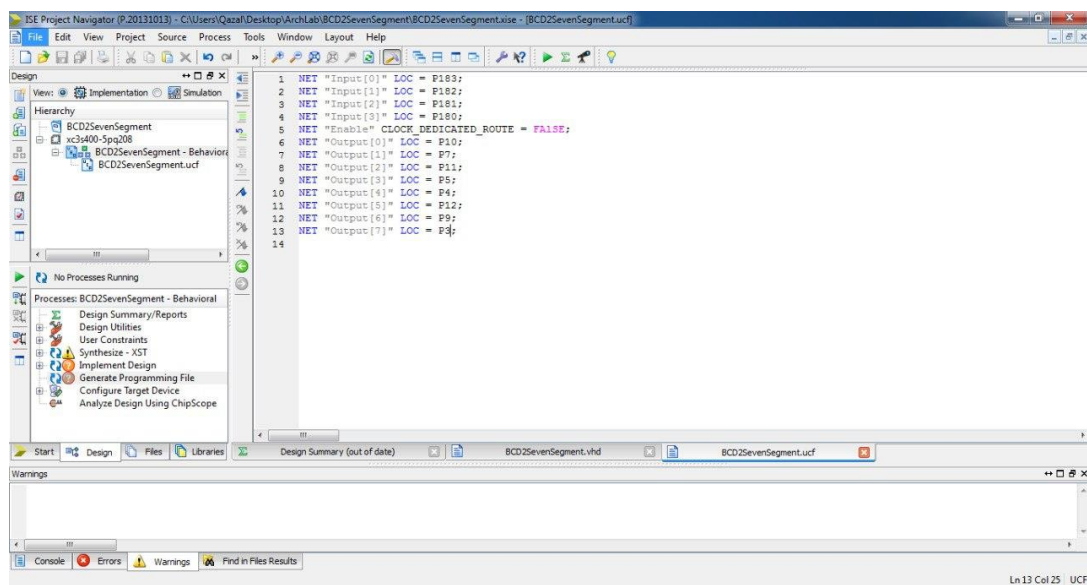
5. تعریف constraint ها:

برای تعریف constraint های پروژه مانند شکل زیر عمل می کنیم:



6. نگاشت ورودی و خروجی های مدار با پورت های موجود در Board در فایل BCD2SevenSegment.ucf:

در فایل با پسوند ucf، ورودی ها و خروجی ها را به board، assign می کنیم.



7. اتصال سیستم به Board، طراحی روی FPGA و تست:

این مراحل نیز دقیقا مانند آزمایش قبلی انجام می شوند.

بحث و نتیجه گیری:

در این آزمایش با مشخص کردن بیت های ورودی به صورت BCD روی board، ارقام در 7Segment به نمایش در می آیند.