

بسمه تعالیٰ

# دُفْرِجَه راهنمای فارسی

**HMZF3S400 (RV\_1.1)**  
**XILINX SPARTAN XC3S400**



WWW.DSPCore.ir  
info@dspcore.ir

شرکت رهپویان علم و صنعت آوا

## جهت استفاده از برد و عدم بروز مشکلات به نکات زیر

### توجه نمایید:

- ✓ ولتاژ اعمالی به پورت‌ها نباید بیشتر از 3.3 ولت باشد، در غیر اینصورت FPGA ممکن است فواهد شد.
- ✓ برای تغذیه برد متن الامکان از آداتپر همراه با برد استفاده شود. در صورت استفاده از تغذیه‌ی دیگر از یک منبع AC یا DC در محدوده 7 الی 12 ولت استفاده نمایید.
- ✓ در صورتی که از پروگرامرهای دیگر استفاده می‌کنید ابتدا پایه‌های JTAG را بروزی و از استاندارد بودن آن اطمینان حاصل نمایید و در صورت نیاز پایه‌ها را به صورت دستی تطبیق دهید.
- ✓ جهت دریافت ویرایش‌های جدید دفترچه راهنمایی به صورت PDF و نمونه برنامه‌ها به سایت مراجعه نمایید. همچنین می‌توانید محصولات جدید را در سایت مشاهده کنید.
- ✓ در فضوی مسائل فنی و پشتیبانی با شماره‌های ۰۹۱۲۷۰۱۰۶۳۸ و ۰۹۱۲۷۷۲۲۹۹۵۶ تلفکس ۰۹۱۲۷۰۱۰۶۳۸ و ایمیل [dspcore@yahoo.com](mailto:dspcore@yahoo.com) مکاتبه نمایید.

تیم پشتیبانی فنی شرکت رهپویان علم و صنعت آوا

WWW.DSPCore.ir

## فهرست مطالب ..... شماره صفحه

۴	..... معرفی
۶	..... مقدمه
۹	..... آشنایی با برد SPARTAN XC3S400
۱۰	..... مشخصات تراشه XC3S400
۱۱	..... امکانات ساخت افزار
۱۱	..... وسایل و امکانات جانبی
۱۲	..... ماثول های مرتبط
۱۲	..... حوزه های کاربرد
۱۳	..... شمای کلی
۱۴	..... بررسی قسمت های مختلف برد به همراه شکل
۲۶	..... روند کلی طراحی ، توصیف و پیاده سازی مدارهای منطقی در XILINX FPGA
۲۷	..... راهنمای استفاده از نرم افزار ISE
۲۷	..... مراحل ایجاد پروژه جدید
۲۹	..... ایجاد سورس فایل
۳۲	..... نمونه کد ساخت شمارنده (چشمک زن)
۳۵	..... Checking The Syntax

۳۶	اختصاص ورودی ها و خروجی ها به شماره پایه های مورد نظر تراشه (Assign Package Pins) XC3S400
۴۱	تولید فایل مورد نیاز برای پروگرام کردن حافظه‌ی FPGA
۴۲	تولید فایل مورد نیاز برای پروگرام کردن حافظه‌ی XCF02S
۴۳	نحوه برنامه‌ریزی کردن (Program) تراشه یا حافظه‌ی XCF02S با استفاده از نرم افزار iMPACT
۴۷	پشتیبانی فنی و خدمات

**معرفی:****شرکت رهپویان علم و صنعت آوا (شماره ثبت ۳۸۴۳۸۵)**

حوزه فعالیت های این شرکت در زمینه طراحی مدارات الکترونیکی و کنترل هوشمند، اجرای پروژه های سخت افزاری در زمینه اتوماسیون صنعتی، تولید انواع برد های صنعتی و وسایل آزمایشگاهی و همچنین واردات و صادرات کلیه ی تجهیزات الکترونیکی و مخابراتی می باشد.

از دیگر فعالیت های این شرکت، طراحی ساخت افزارهای مبتنی بر تراشه های **FPGA**، **ARM**، **DSP** و میکروکنترلر به منظور تجهیز آزمایشگاه های برق و کامپیوتر مراکز آموزشی می باشد. همچنین این شرکت در راستای پشتیبانی از محصولات خود و سهولت به کارگیری آنها، اقدام به برگزاری کارگاه ها و دوره های آموزشی نموده است که بنا به درخواست مراکز آموزشی در محل آنها برگزار میگردد. علاوه بر آموزش های مرتبط با ساخت افزارها، دوره های تخصصی نیز به منظور تسلط بیشتر در پیاده سازی نرم افزارها و الگوریتم های پردازشی برگزار میشود. محدودیت هایی که تا کنون در کشور ما وجود داشته عمدتا در دسترس نبودن ساخت افزارهای مناسب و ارزان قیمت و عدم پشتیبانی است و این عاملی بوده که بسیاری از افراد را از شروع به کار در این زمینه منصرف میکند. همچنین آنچنان که باید حوزه های کاربردی تراشه های **DSP**

**Programmable Gate Array FPGA و Digital Signal Processor**

**ARM** و **Field** علم به آن برای افراد به خوبی معرفی نشده است.

هر چند که مطالب آنقدر وسیع است که قطعا نمیتوان به همه آنها پرداخت ولی حتی الامکان سعی میشود که با ارائه ساخت افزارهای مناسب و پشتیبانی علمی از افراد زمینه برای فعالیت و پیشرفت فراهم شود.

فعالیتهای صورت گرفته در چند سال اخیر عمدتاً مربوط به **FPGA** های شرکت **Instrument TI** و **DSP** های سری ۲۰۰۰، ۵۰۰۰ و ۶۰۰۰ شرکت **XILINX Texas** بوده و قصد بر آن است که حیطه های کاری را گسترش داده تا حوزه های کاربردی بیشتری را شامل شود. خواهشمند است نظرات و پیشنهادات خود را از طریق ایمیل به اطلاع ما برسانید.

[dspcore@yahoo.com](mailto:dspcore@yahoo.com) & [info@dspcore.ir](mailto:info@dspcore.ir)

مدیر تحقیق و توسعه شرکت رهپویان علم و صنعت آوا

**مقدمه**

امروزه علاوه بر مدارهای مجتمع (IC)<sup>۱</sup> استاندارد، مدارهای مجتمع خاص با آرایه ای از گیت ها که ASIC<sup>۲</sup> نام دارند نیز بر اساس نیاز کاربر ساخته می شود این نوع مدارها را مدارهای قابل برنامه ریزی می گویند . اولین مدارهای منطقی قابل برنامه ریزی PLA<sup>۳</sup> نامیده شدند و بعد از آن مدارهای قابل اندازه گیری PAL<sup>۴</sup> و PLD<sup>۵</sup> نیز به بازار عرضه شدند و بالاخره مدارهای پیچیده تر قابل برنامه ریزی CPLD<sup>۶</sup> به بازار آمدند .

با توجه به پیشرفت دنیای دیجیتال و همچنین عدم پاسخ گویی مدارات مجتمع معمولی برای پیاده سازی توابع پیچیده ، به مرور تراشه های جدیدی طراحی شده اند. اخیرا یک نوع مدار مجتمع قابل برنامه ریزی که به دنیا عرضه شده FPGA<sup>۷</sup> نامیده می شود که یک مدار مجتمع قابل برنامه ریزی توسط کاربر، با ظرفیت حدود ۲۰۰۰۰ تا چند میلیون گیت می باشد و با تکنولوژی CMOS<sup>۸</sup> ساخته شده است.

<sup>1</sup> Integrated Circuits

<sup>2</sup> Application Specific Integrated Circuits

<sup>3</sup> Programmable Logic Array

<sup>4</sup> Programmable Array Logic

<sup>5</sup> Programmable Logic Device

<sup>6</sup> Complex Programmable Logic Device

<sup>7</sup> Field Programmable Gate Array

<sup>8</sup> Complementary Metal Oxide Semiconductor

FPGA بر اساس سلولهای منطقی برنامه ریزی طراحی شده است که ارتباط بین این سلولها نیز قابل برنامه ریزی می باشد و برای ساختن نمونه ای از مدار و طراحی سریع سیستمهای دیجیتال، بسیار مناسب می باشد و چون قابل برنامه ریزی است ، بنابراین به آسانی می توان متناسب با نیاز کاربر تغییرات لازم را در طراحی انجام داد و FPGA را مجدا برنامه ریزی نمود .

#### FPGA دارای محسن زیر است :

- ❖ کاربرد اصلی FPGA در ایجاد هسته های پردازشی می باشد.
- ❖ مدارهای دیجیتال پیچیده به آسانی در آنها پیاده سازی می شوند.
- ❖ تست مدار سریع است.
- ❖ برای تولیدات با تیراژ پایین ارزان تمام می شود .
- ❖ متناسب با نیاز، تغییرات لازم را در طراحی می توان انجام داد و مجدا FPGA را با ساختار جدید برنامه ریزی نمود.
- ❖ قابل برنامه ریزی توسط کاربر است.
- ❖ می توان چند تا هسته پردازشی داخل یک FPGA تعریف کرد تا در یک زمان واحد چندتا کار را باهم انجام بدهد مثلا شما می توانید با یک FPGA معمولی حدود ۲۰۰ تا هسته atmega32 تعریف کنید و ۲۰۰ تا کار را همزمان انجام بدهید.
- ❖ میکروکنترلر ها و DSP ها و میکروپروسسورها به صورت سریال دستورالعمل اجرا می کنند و قابلیت پردازش بصورت موازی در آنها وجود ندارد ، اما در FPGA قابلیت پردازش موازی وجود دارد و قابلیت انجام عملیات بصورت همزمان را دارد.
- ❖ کاربردهای FPGA خیلی تخصصی می باشد و در اکثر موارد به عنوان پردازشگر در مدارات پردازشی استفاده می شود . سرعت بالای FPGA ها آنها را مساعد

کارهای پردازشی سنگین مثل پردازش تصویر ، پردازش صدا می کند و سرعت این پردازش نسبت به سیستم های دیگر خیلی بالاتر است .

❖ FPGA ها در پیاده سازی توابع پیچیده دیجیتالی به کار می روند که نیاز به سرعت پردازش بالایی دارد .

❖ کاهش سخت افزار مورد نیاز و همچنین برنامه نویسی ساده و استاندارد نیز از دیگر مزیت های استفاده از FPGA است .

❖ سرعت اجرای توابع منطقی در FPGA ها بسیار بالا و در حد نانو ثانیه است .

سازندگان FPGA شامل شرکت های بزرگی همچون Xilinx و Altra و Actel

می باشند که سری های مختلفی ، از جمله Xilinx شامل : Spartan و

Stratix و Flex10k و Cyclone و Altra شامل : Actel و Altera و .... را به بازار عرضه کرده اند .

در کشور ما معمولاً با CPLD های شرکت های مذکور کار میشود که

از نظر محبوبیت به صورت زیر هستند

Actel -۳ Altera -۲ Xilinx -۱

## آشنایی با برد HMZ3S400



برد HMZ3S400 براساس تراشه‌ی Spartan- XC3S400 شرکت Xilinx طراحی شده است و هدف از آن به کارگیری امکانات این تراشه و پیاده‌سازی الگوریتم‌های دیجیتالی و پردازشی توسط زبان‌های برنامه‌نویسی HDL بر روی آن می‌باشد.

قسمت های اصلی این برد شامل تراشه Spartan- XC3S400 و حافظه های XCF02S به همراه پروگرامر می باشد. از ویژگیهای خاص این برد، قرار دادن حافظه های XCF02S در کنار تراشه های اصلی جهت ذخیره کردن دائمی برنامه و در دسترس بودن I/O ها به صورت تفکیک شده و همچنین استفاده از DIP SWITCH ها کلیدهای فشاری و نمایشگرها (LED) جهت تست برنامه های کاربر، می باشد.

پکیج کامل و منحصراً بفرد این برد (شامل: برد اصلی، پروگرامر، آداتور، ماثول ها، نرم افزارها و نمونه کدهای خاص) باعث راه اندازی آسان و و کابر پسند گردیده، به طوری که کاربر بطور کامل تمام مراحل پروژه خود را طراحی و ساخته و خروجی کار خود را مشاهده می نماید.

#### • خلاصه های مشخصات تراشه های استفاده شده (Spartan- XC3S400)

#### Summary of Spartan- XC3S400 FPGA Attributes

System Gates	400K	
Equivalent Logic Cells	8,064	
LB Array (One CLB) =	Rows	32
	Columns	28
Four Slices	Total CLBs	896
Distributed Ram Bits (K=1024)	56K	
Block RAM Bits (K=1024)	288K	
Dedicated Multipliers	16	
DCMs	4	
Maximum User I/O	264	
Maximum Differential I/O Pairs	116	

## امکانات سخت افزار:

- دارای تراشهی SPARTAN3 XC3S400-PQG208
- دارای حافظهی XCF02S ظرفیت (2Mbit)
- قراردادن هر بانک در یک کانکتور، جهت در دسترس بودن کلیهی پایهی ها
- امکان انتخاب حالت های مختلف برنامه ریزی
- دو عدد DIP SWITCH هشت تایی جهت اعمال DATA به تراشه
- ۸ عدد LED مجهری به بافر جهت نمایش داده های خروجی
- اسیلاتور 40Mhz روی برد و امکان راه اندازی تراشه با فرکانس کاری 280Mhz
- ولتاژ تغذیه ۷-۱۲ ولت DC یا AC
- قابلیت برنامه ریزی تراشه و حافظه، با استفاده از پورت USB
- در دسترس بودن ولتاژ های 5V, 3.3V, 2.5V, 1.2V
- محافظت در برابر اتصال تغذیهی معکوس

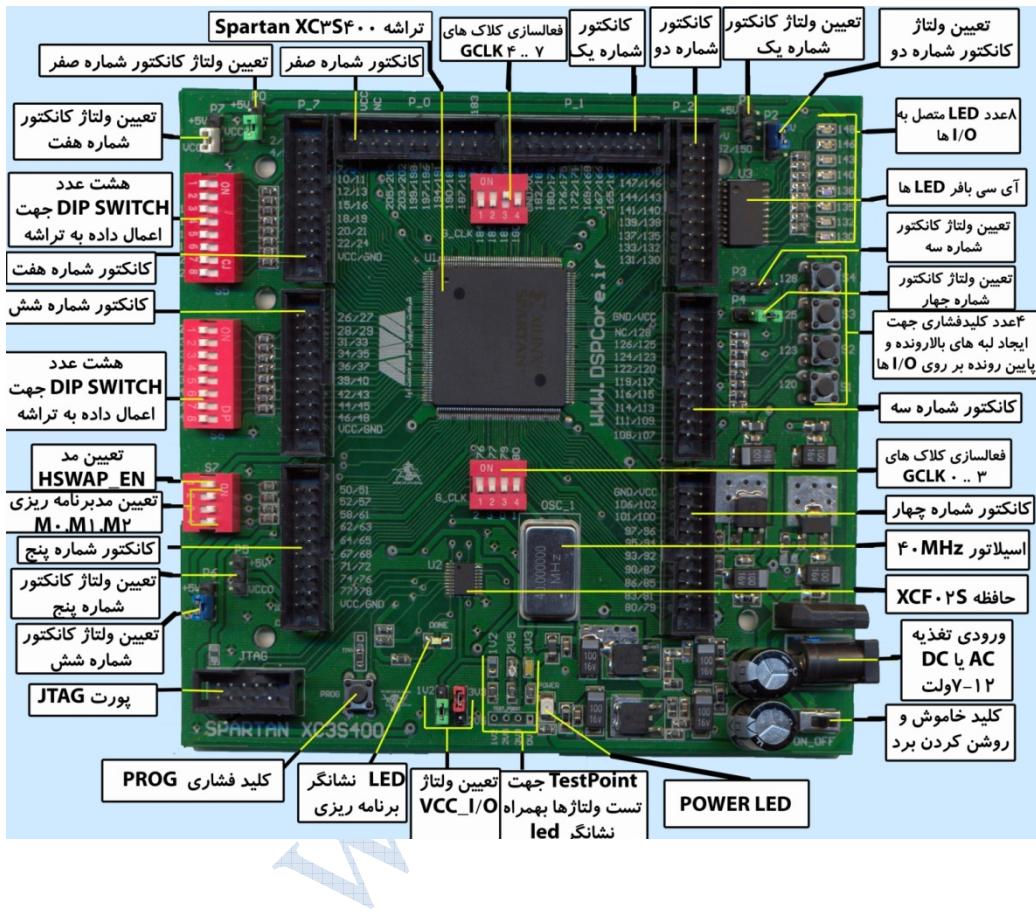
## وسایل و امکانات جانبی:

- جدیدترین ویرایش نرم افزار قدرتمندو کاربردی ISE: بستهی کامل XILINX ISE\_11.1 در دو عدد DVD به همراه نمونه مثال ها و کدها
- نرم افزار ISE 9.2i
- آداپتور (منبع تغذیه) قابل حمل
- مازول های چند منظوره جانبی مانند: DIP SWITCH, LED, 7Segment
- LCD و ....
- پروگرامر مخصوص برد های CPLD و FPGA که قابلیت کار با ISE و Altium Designer را دارد.

**ماژول های مرتبط :****SDRAM** ماژول •**SRAM** ماژول •**NOR FLASH** ماژول •**SPI FLASH** ماژول •**ADC-DAC** ماژول •**USB** ماژول •**VGA** ماژول •**مثال هایی از حوزه های کاربردی:**

- ایجاد هسته های پردازشی گوناگون
- پیاده سازی مدارهای دیجیتال پیچیده
- پیاده سازی طرح های تحقیقاتی که نیاز به ساخت تراشه دارند. (ASIC)
- پیاده سازی الگوریتم های پیچیده مخابراتی
- پردازش سیگنال و پردازش موازی و بلا درنگ
- تجهیز آزمایشگاه های معماری کامپیوتر و پردازش سیگنال به این سخت افزار
- ..... •

## شماي گلى برد:

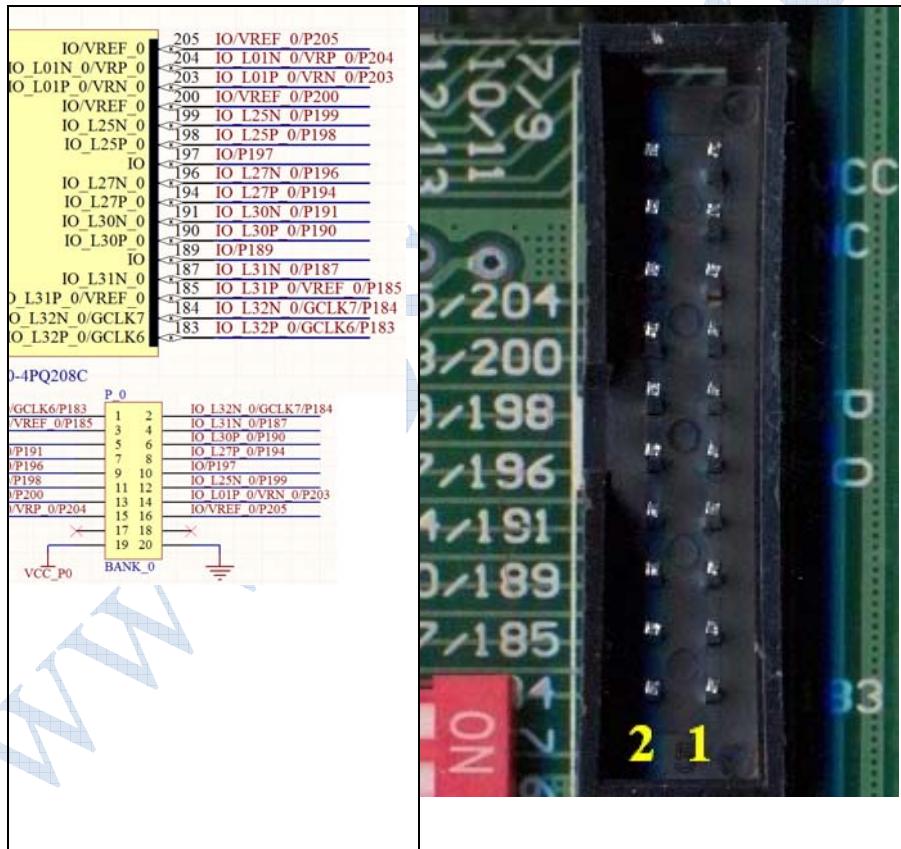


## بررسی قسمت های مختلف برد به همراه شکل

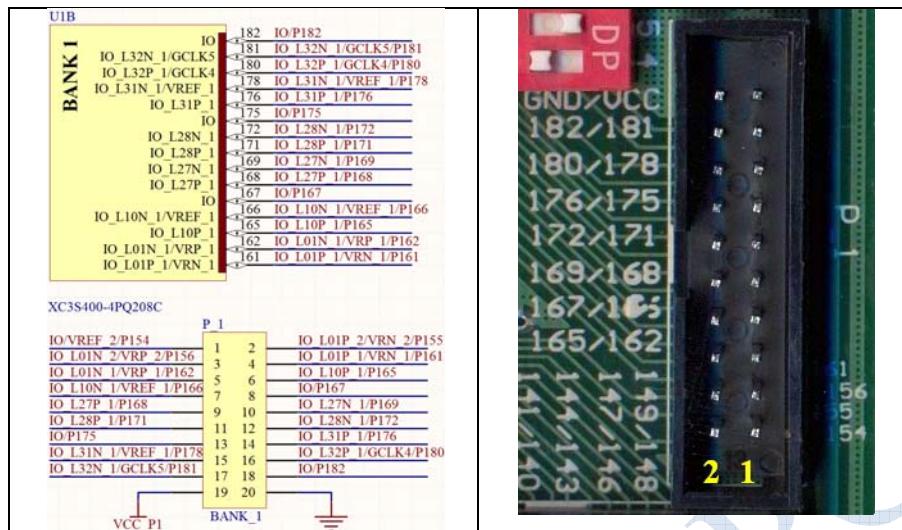
- بانک ها:

بانکهای Spartan XC3S400 به هشت قسمت تقسیم شده است که هر بانک تعدادی I/O را شامل می شود که در یک کانکتور قرار داده شده است. در زیر هر یک از آنها تفکیک و شماره I/O آنها مشخص شده است.

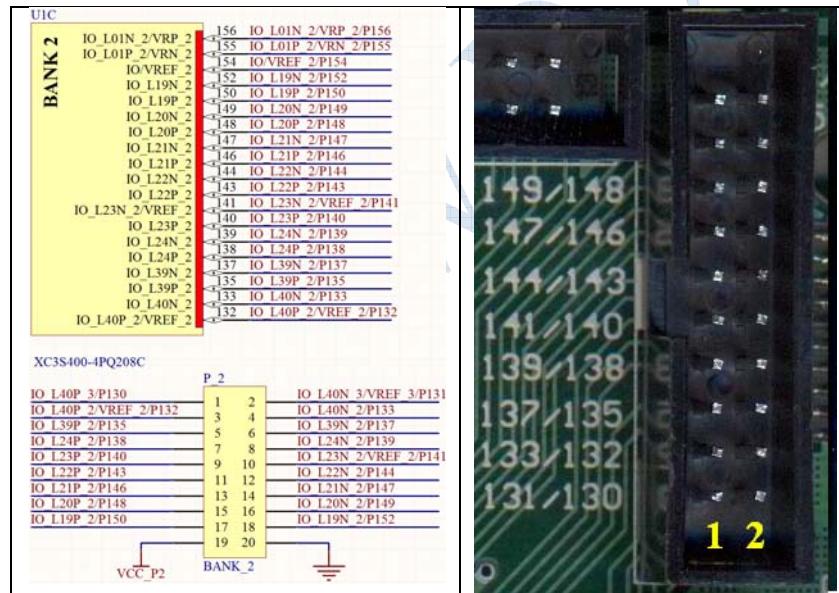
بانک شماره صفر:



بانک شماره یک:

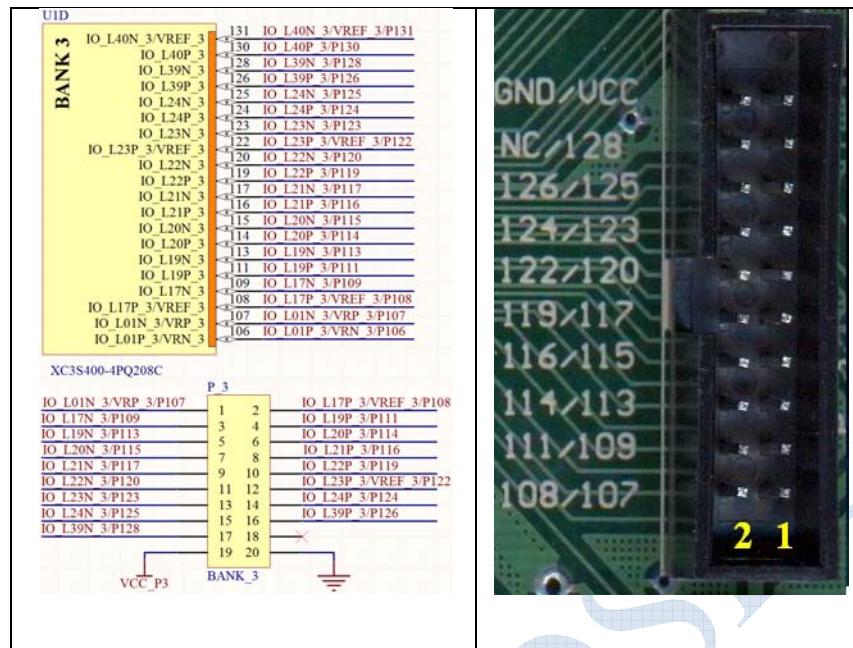


بانک شماره دو:

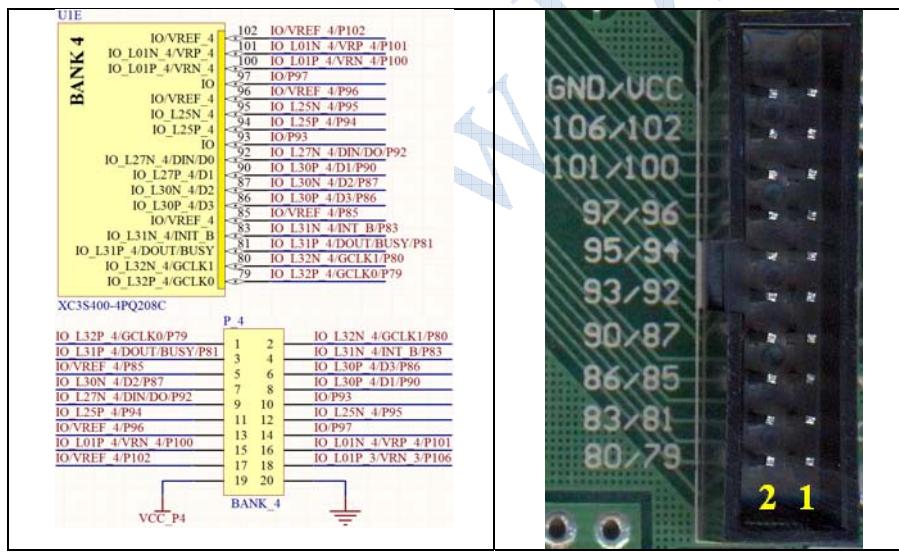


شرکت رهپویان علم و صنعت آوا

بانک شماره سه:

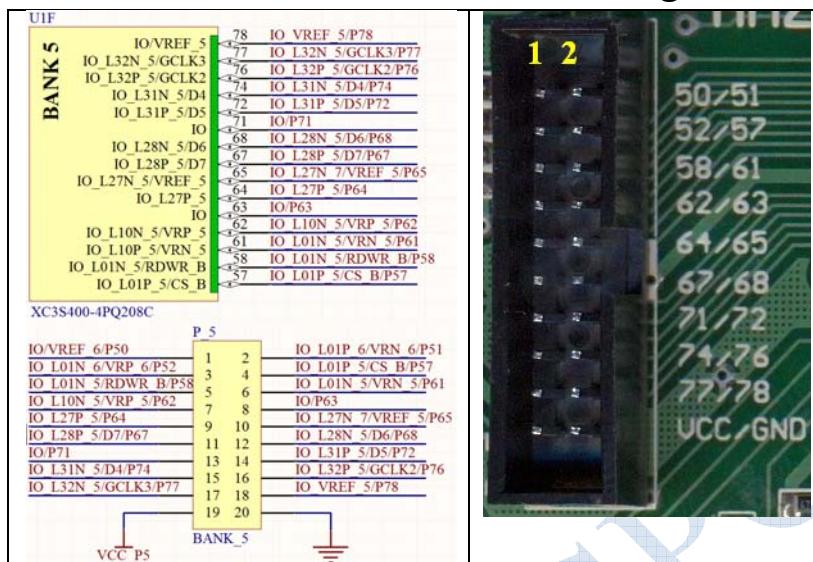


بانک شماره چهار:

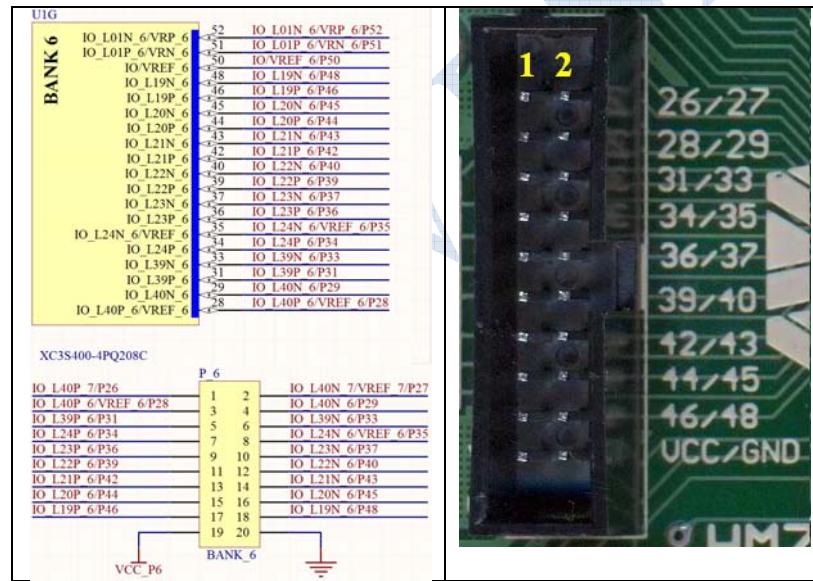


شرکت رهپویان علم و صنعت آوا

بانک شماره پنج:

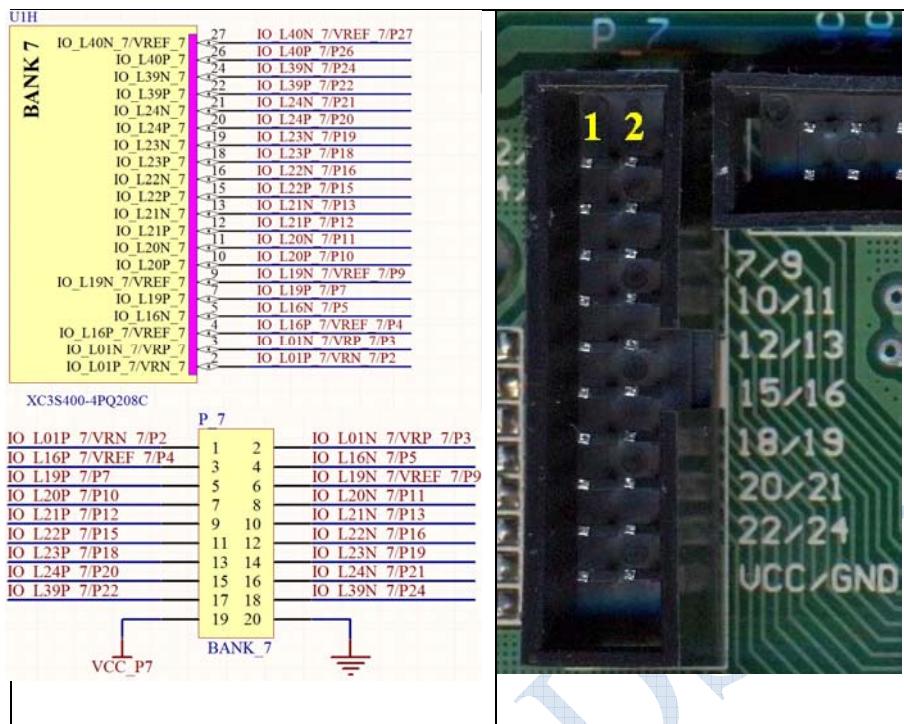


بانک شماره شش:

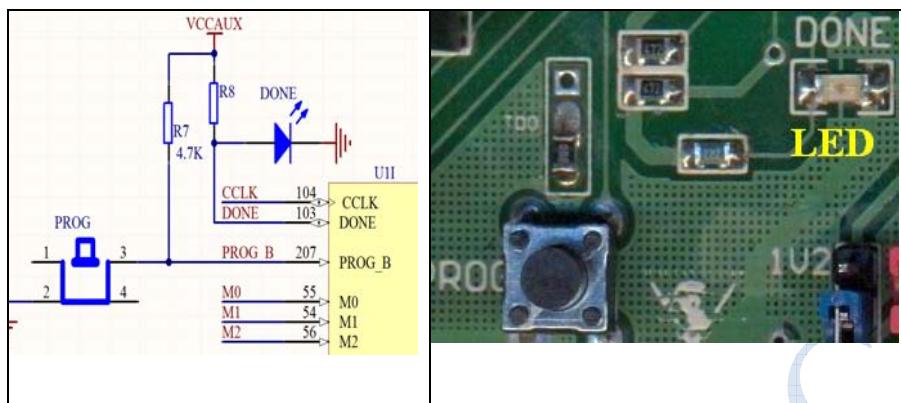


شرکت رهپویان علم و صنعت آوا

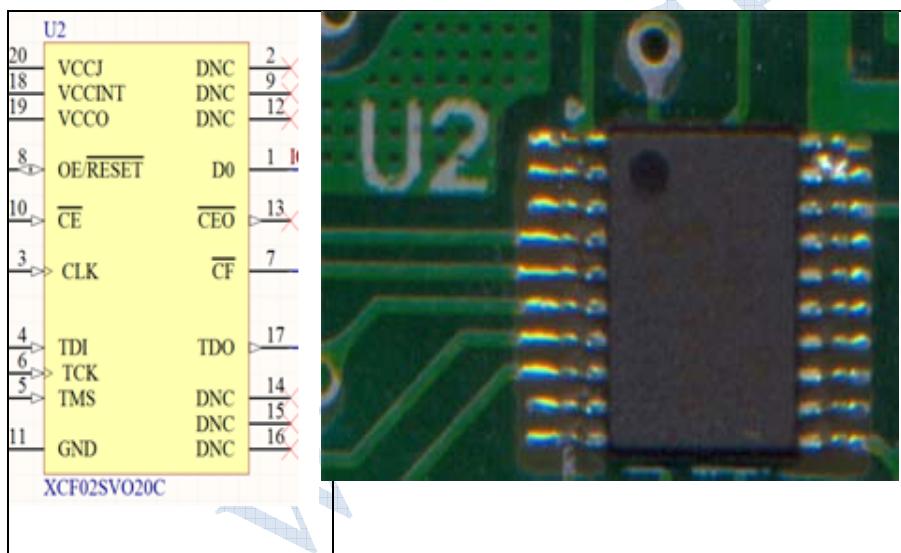
بانک شماره هفت:



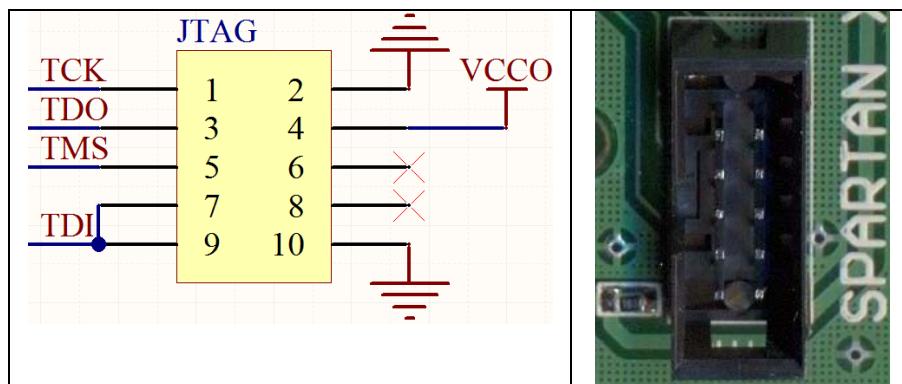
- کلید PROG : با فشردن این کلید، تراشه FPGA با برنامه ای که در حافظه می شود، برنامه ریزی می شود.



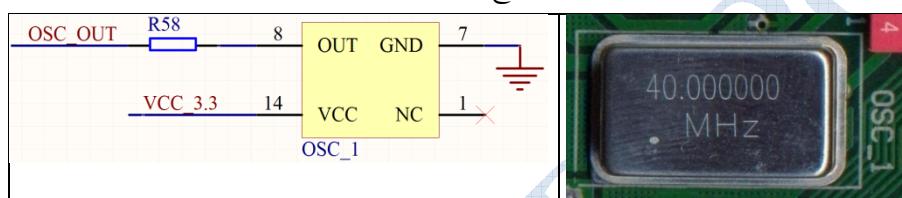
- حافظه می XCF02S : که دارای 2Mbit حافظه می باشد



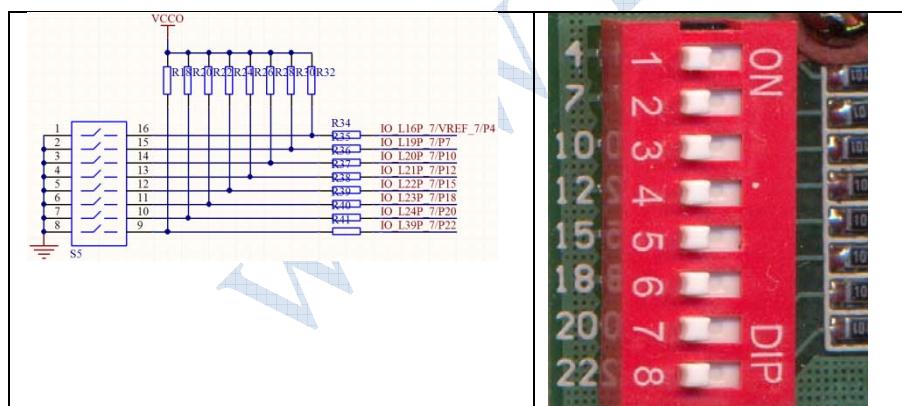
- پورت JTAG: جهت ارتباط برد با کامپیوتر و برنامه ریزی تراشه استفاده می شود.



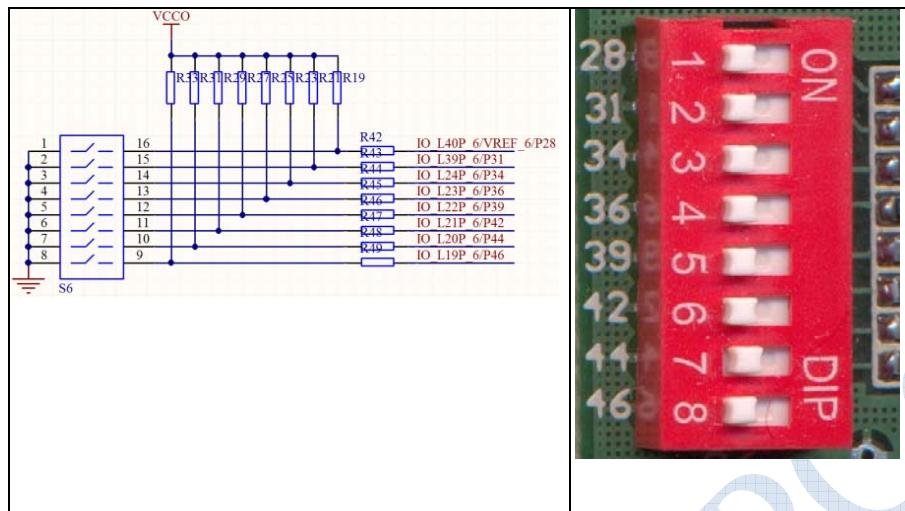
- اسیلاتور 40MHZ: به عنوان منبع کلک بکار می رود.



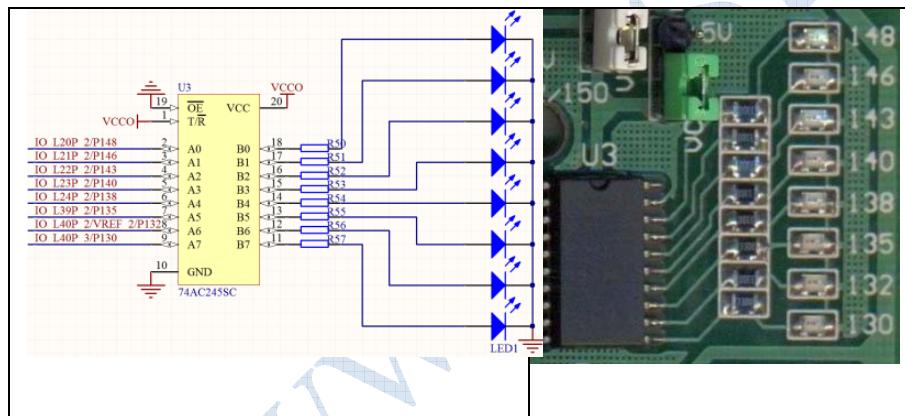
- جهت اعمال داده به ورودی تراشه DIP SWITCH (S5)



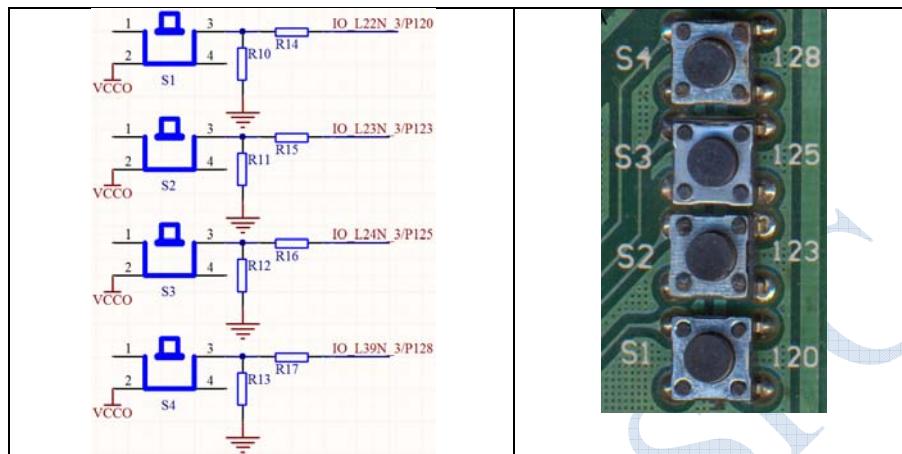
## جهت اعمال داده به ورودی تراشه DIP SWITCH (S6) •



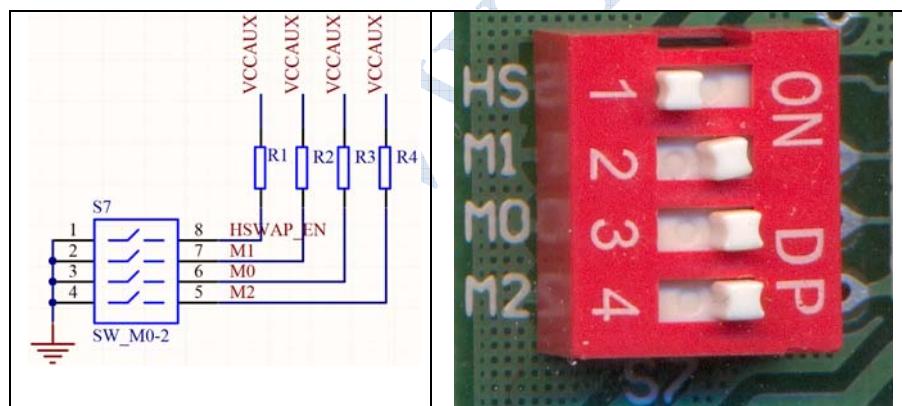
## جهت اعمال داده به ورودی تراشه LED •



- جهار عدد کلید فشاری جهت ایجاد لبه های بالا رونده و پایین رونده بر روی **I/O**ها استفاده شده است و پایه های شماره ۱۲۰، ۱۲۳، ۱۲۵ و ۱۲۸ از FPGA به آنها اختصاص یافته است.



- کلیدهای M0, M1, M2 DIP SWITCH (S7) جهت تعیین نحوه برنامه ریزی FPGA با توجه به جدول زیر استفاده می شود.



***M2, M1, M0: Configuration Mode Selection***

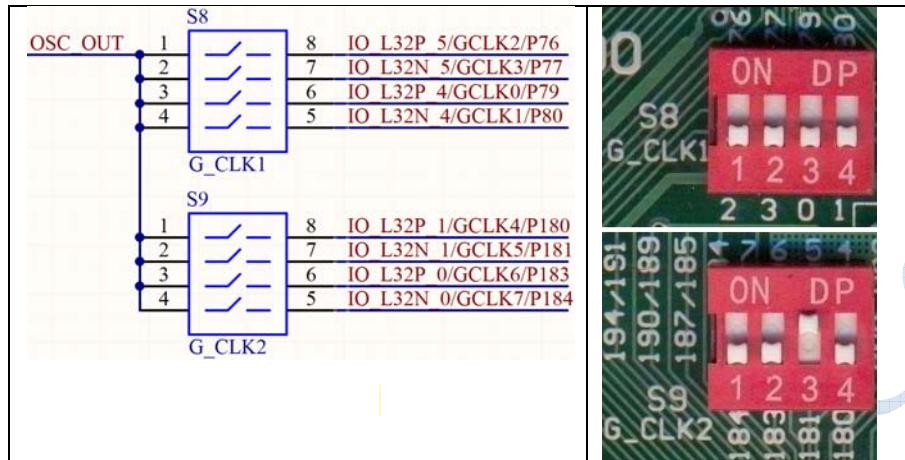
Configuration Mode	M0	M1	M2	Synchronizing Clock	Data Width	Serial DOUT
Master Serial	0	0	0	CCLK Output	1	Yes
Slave Serial	1	1	1	CCLK Input	1	Yes
Master Parallel	1	1	0	CCLK Output	8	No
Slave Parallel	0	1	1	CCLK Input	8	No
JTAG	1	0	1	TCK Input	1	No

- کلید HSWAP\_EN : با صفر کردن این کلید، O/Iهای تراشه، از طریق مقاومت . وصل می شوند

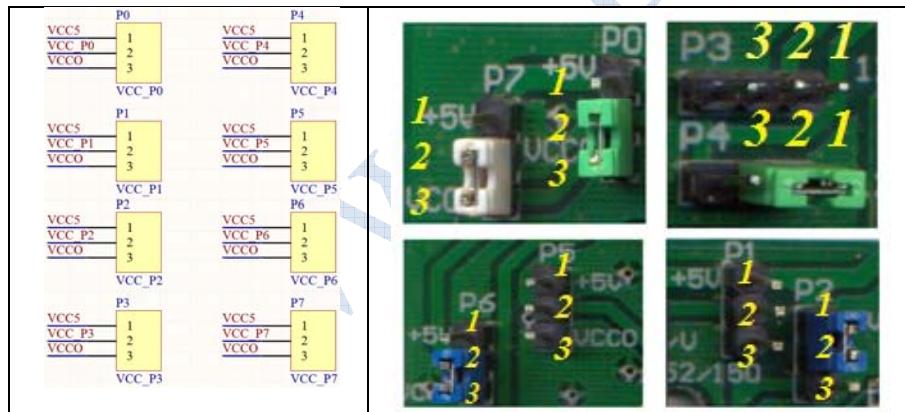
***HSWAP\_EN: Disable Pull-up Resistors During Configuration*** •***HSWAP\_EN Encoding***

HSWAP_EN	Function
<b>During Configuration</b>	
•	Enable pull-up resistors on all pins not actively involved in the configuration process. Pull-ups are only active until configuration completes. See
1	No pull-up resistors during configuration.
<b>After Configuration, User Mode</b>	
X	This pin has no function except during device configuration.

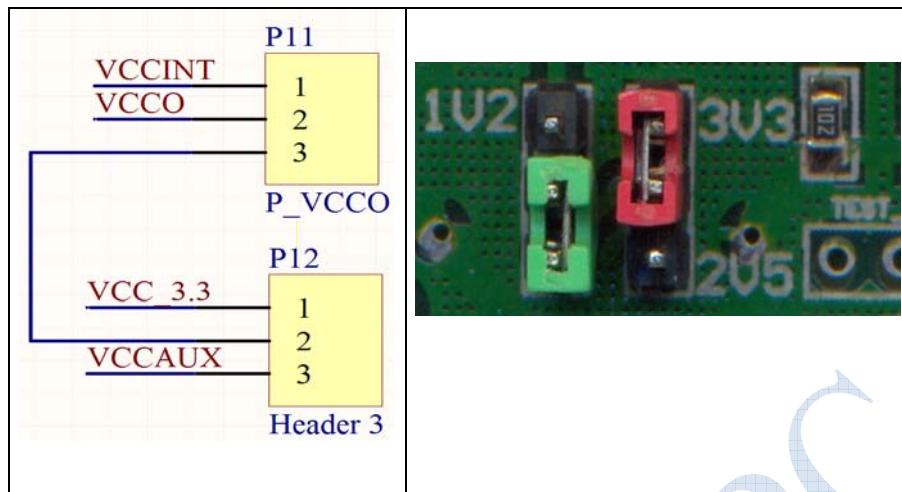
- جهت فعالسازی کلک های عمومی در سرتاسر تراشه DIP SWITCH GCLK استفاده می شوند که در کنار هر پایه شماره کلک درج شده است. در این تراشه هشت ورودی کلک طبق شکل با پایه های مشخص شده وجود دارد.



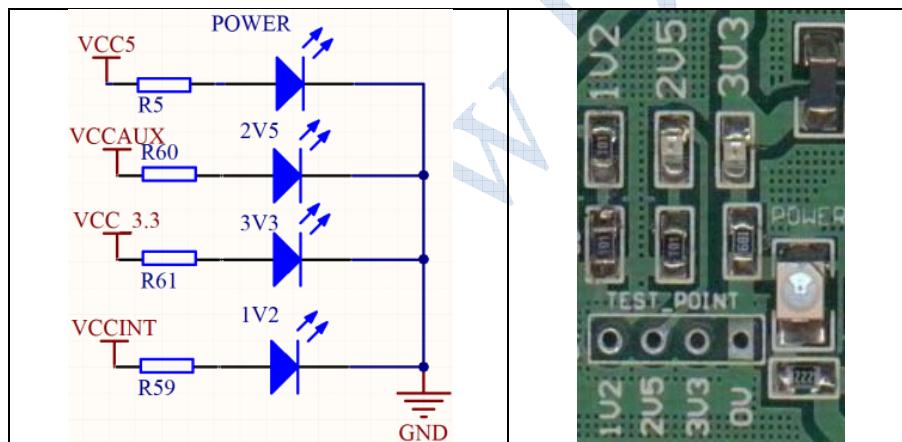
- جامپرهای تعیین ولتاژ کانکتورها: هر جامپر مربوط به یک کانکتور می باشد که مقدار ولتاژ آن کانکتور را باتوجه به نیاز کاربر و یا نوع مازول مورد استفاده، تعیین می کند.



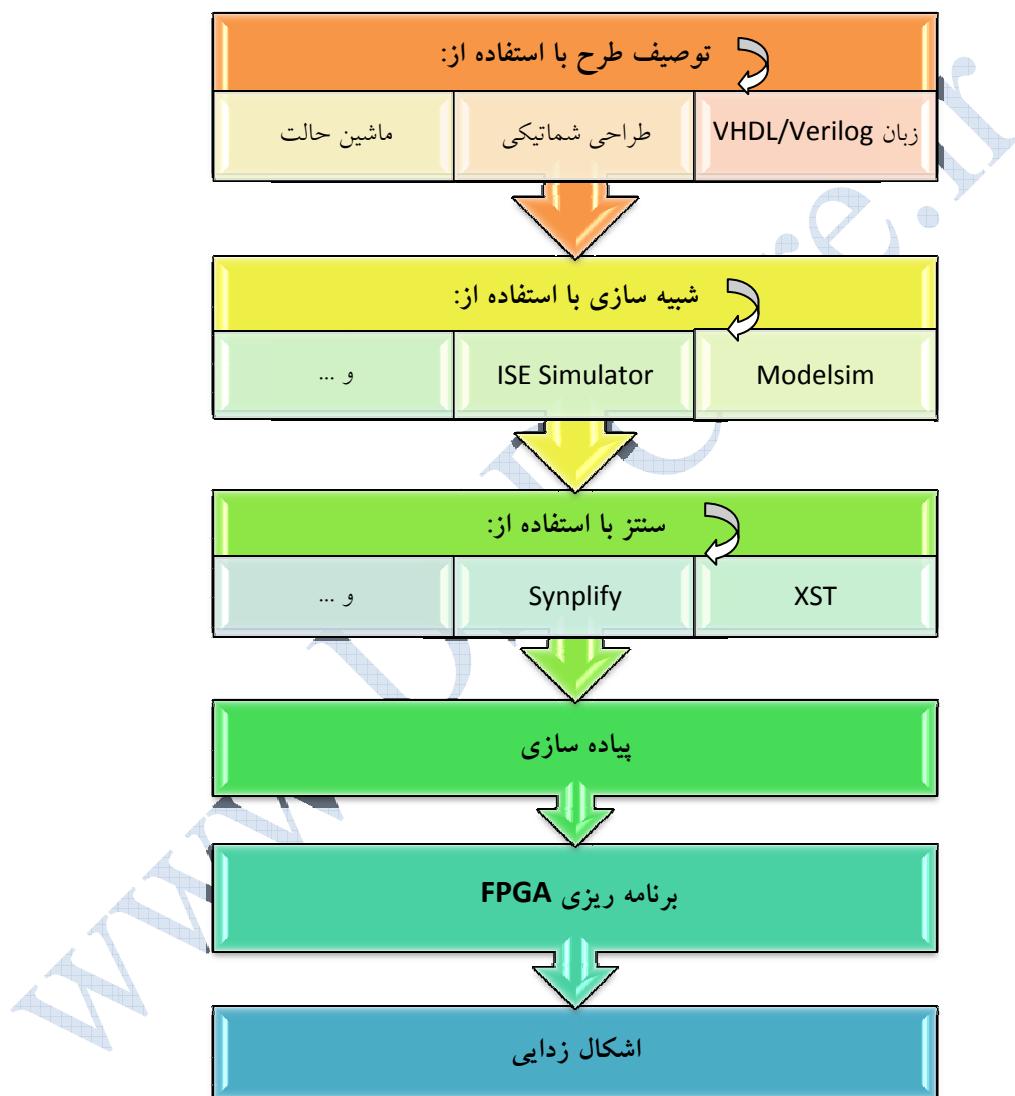
- جامپر تعیین ولتاژ VCC\_I/O : جهت تعیین مقدار ولتاژ I/O ها به کار می روند.



- LED TEST POINT: برای تست هر یک از ولتاژهای موجود در برد، یک عدد LED درنظر گرفته شده است. همچنین یک آبی رنگ طبق شکل به عنوان POWER دستگاه درنظر گرفته شده است.



## روند کلی طراحی، توصیف و پیاده سازی مدارهای منطقی در XILINX FPGA



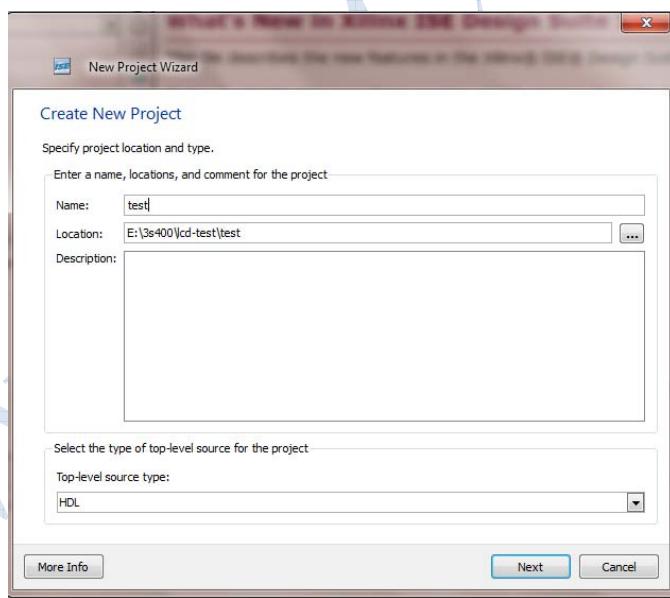
## راهنمای استفاده از نرم افزار : ISE

در این دفترچه راهنمای نحوه کار با نرم افزار Xilinx\_ISE را، با ایجاد یک پروژه ساده شرح می دهیم.

برای شروع برنامه را از مسیر Start/All Program/Xilinx ISE/Project Navigator اجرا کنید.

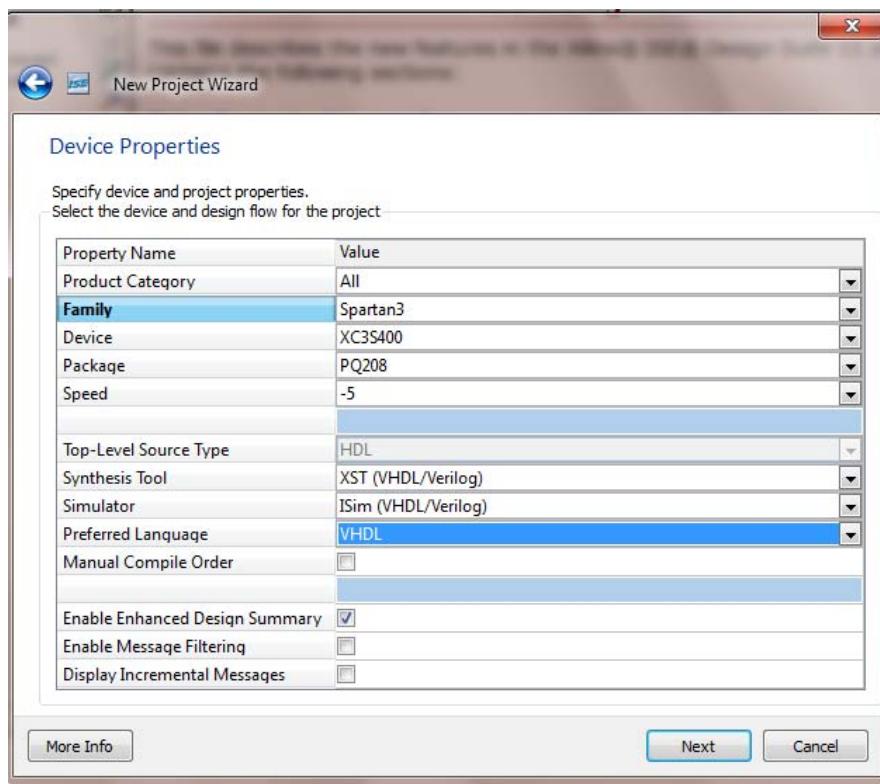
### مراحل ایجاد پروژه جدید:

- ۱- از منوی File گزینه **NewProject...** را انتخاب کنید . پنجره **NewProjectWizard** ظاهر می شود



- ۲- در قسمت **Project Name Field** نام پروژه را تایپ کنید. (مثلا **test**)
- ۳- مسیر مورد نظر را برای ذخیره پروژه جدید انتخاب کنید.
- ۴- در قسمت **Top-LevelSourceType** گزینه **HDL** را انتخاب کنید

- ۵- Device Properties را بزنید تا وارد صفحه Next شوید.
- ۶- مشخصات تراشه مورد نظر خود را وارد کنید . مثلا جهت انتخاب Spartan- XC3S400 می توانیم اطلاعات را طبق شکل زیر وارد کنیم.
- گزینه Enable Enhanced Design Summary را فعال کنید و بقیه گزینه ها را در حالت پیش فرض رها کنید.



### ایجاد سورس فایل:

بر روی گزینه NEXT کلید کنید تا وارد پنجره CreateNewSource شوید.

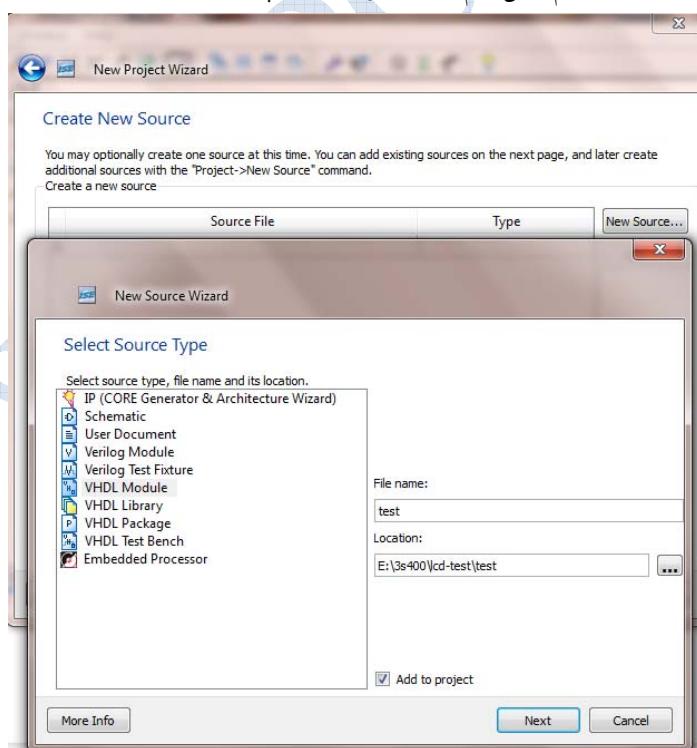
در این قسمت یک فایل HDL سطح بالا ( Top-Level ) برای پروژه جدیدمان ایجاد می کنیم. برای این کار باید زبان توصیف سخت افزاری که می خواهیم با آن فایل مورد نظرمان را ایجاد کنیم. ما در اینجا زبان VHDL را انتخاب می کنیم.

برای ایجاد سورس فایل VHDL طبق روال زیر عمل می کنیم:

1. در ادامه قسمت قبل، در پنجره NewProjectWizard روی گزینه NewSource کلیک کنید.

2. در پنجره NewSource نوع فایل سورس را VHDL Module انتخاب کنید.

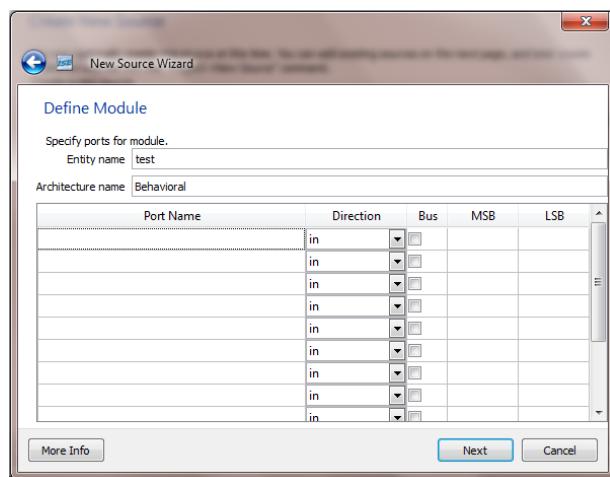
3. در قسمت نام فایل، نام مورد نظرتان را تایپ کنید ( مثلا در اینجا test )



4. گزینه Add to Project را نیز انتخاب کنید.

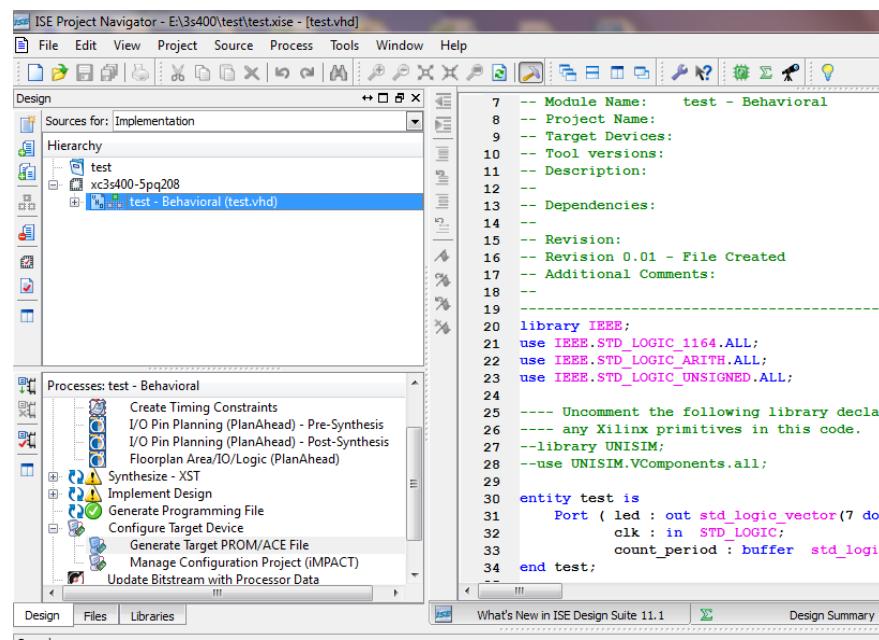
5. بر روی Next کلیک کنید.

6. سپس مانند شکل زیر پورت های طرحتان را تعریف کنید.



7. در ادامه برای تکمیل تنظیمات سورس فایل در پنجره New Source بر روی Next کلیک کنید و سپس Finish کلیک کنید.

8. در صفحه اصلی بر روی Next ، دوباره Next و سپس Finish کلیک کنید.  
بعد از انجام مراحل بالا صفحه workspace مانند شکل زیر ایجاد می شود.



در اینجا باید با توجه به نوع الگوریتمی که مدنظر داریم ، کدنویسی را شروع کنیم. لازم است از قبل اطلاعاتی در رابطه با ساختار زبان VHDL داشته باشیم. قدم بعدی در مسیر ایجاد یک پروژه ، اضافه کردن کدهای توصیف کننده طرح با توجه به الگوریتم مورد نظر ، به فایل VHDL می باشد، که در اینجا یک شمارنده است. خوب شنخته در ISE تعدادی از کدهای پرکاربرد به صورت آماده تحت عنوان Language Templates وجود دارد که بعضی موقع کدنویسی را خیلی ساده تر می کند .

قالب کلی برنامه نویسی با زبان سخت افزاری VHDL به شکل زیر است:

```
entity test_s3 is
  -- تعریف ورودی ها و خروجی ها
end test_s3;

architecture Behavioral of test_s3 is
  -- تعریف Signal ها
begin
  -- کد برنامه
end Behavioral;
```

### نمونه کد ساخت شمارنده (چشمک زن) :

در زمان خرید برد بر روی حافظه‌ی XCF02S این کد برنامه ریزی شده است

```
1 -----
-----
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date: 19:01:32 04/10/2010
6 -- Design Name:
7 -- Module Name: test_s3 - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
```

```
19 -----
-----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25 ---- Uncomment the following library
26 declaration if instantiating
27 ---- any Xilinx primitives in this code.
28 --library UNISIM;
29 --use UNISIM.VComponents.all;
30
30 entity test_s3 is
31 Port ( led : out std_logic_vector(7 downto
0) ;
32 clk : in STD_LOGIC;
33 count_period : buffer std_logic_vector(7
downto 0) );
34 end test_s3;
35
36 architecture Behavioral of test_s3 is
37 signal clk_2 : std_logic ; -----for
clk_div-----
38 --signal count_period : integer range 0 to
255 ;
39 begin
40 process(clk)
41 variable count_div : integer range 0 to
1000000 :=0;
42 begin
43
44 if (rising_edge(clk)) then
45
46 if count_div < 1000000 then
```

```
47 count_div := count_div + 1 ;
48 if count_div < 500000 then
49 clk_2 <= '1';
50 else
51 clk_2 <= '0';
52 end if;
53 else count_div := 0;
54 end if;
55 end if;
56 end process;
57 -----
58 process(clk_2)
59 begin
60 if rising_edge (clk_2) then
61 count_period <= (count_period) + '1' ;
62 end if;
63
64 led <= count_period;
65
66 end process;
67
68 end Behavioral;
```

پس از اینکه کدنویسی به پایان رسید ابتدا درستی برنامه را باید تست کرد.

### : Checking The Syntax

برای پیدا کردن خطاهای syntax موجود در طراحی مراحل زیر را انجام دهید:

۱. Sources را در لیست بالای پنجره Synthesis/Implementation در بالا و سمت

چپ پنجره برنامه ISE انتخاب کنید.

۲. در پنجره Sources فایل counter را انتخاب کنید تا پروسس های مربوطه در پنجره (

Processes درپایین و سمت چپ پنجره برنامه ISE ) نمایش داده شود.

۳. بر روی "+" کنار پروسس Synthesize-XST کلیک کنید تا زیر مجموعه های آن

نمایش داده شود.

۴. بر روی پروسس CheckSyntax دو بار کلیک کنید.

۵. پس از پایان این مرحله خطاهای موجود در syntax برنامه مشخص می شود و در

قسمت پایین پنجره ISE نشان داده می شود.

توجه کنید که در این مرحله باید تمام خطاهای موجود را برطرف کنید تا بتوانید طراحی

خود را شبیه سازی و سنتز کنید.

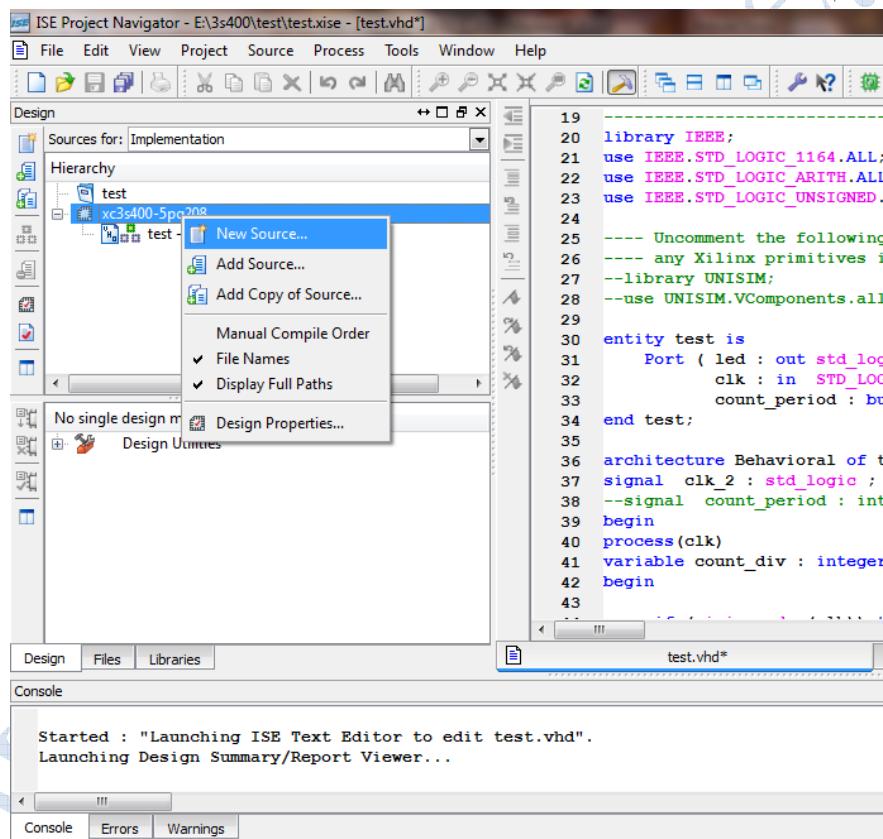
پس از رفع خطاهای احتمالی برنامه را شبیه سازی و سنتز می کنیم تا فایل های مورد نیاز

جهت برنامه ریزی بر روی برد ایجاد شوند.

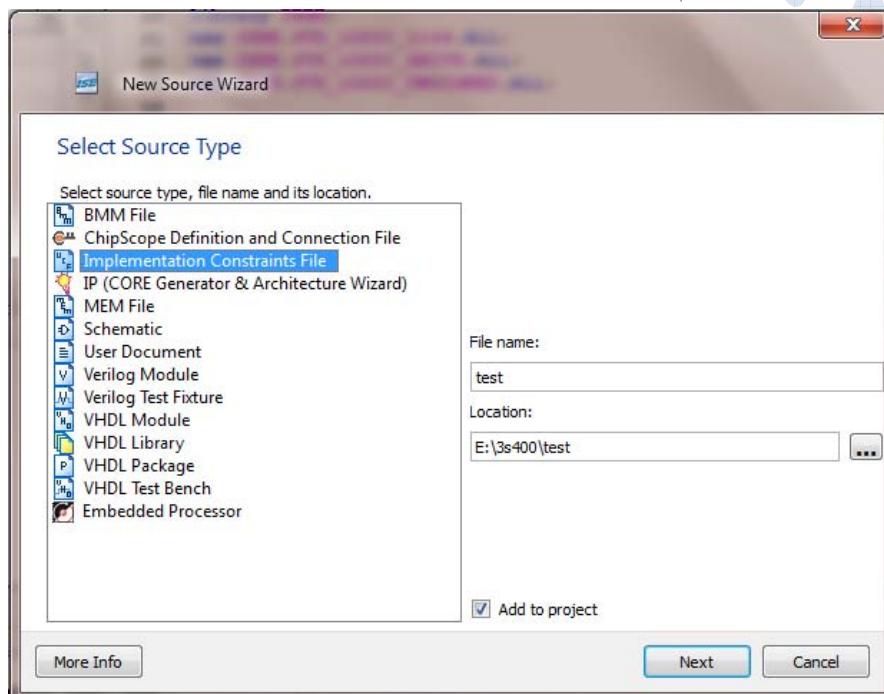
## اختصاص ورودی ها و خروجی ها به شماره پایه های مورد نظر تراشه XC3S400 (Assign Package Pins)

بعد از رفع خطاهای احتمالی و نوشتن برنامه، باید فایل با پسوند `.ucf` جهت اختصاص پایه های مورد نظر را به شکل زیر ایجاد کرد.

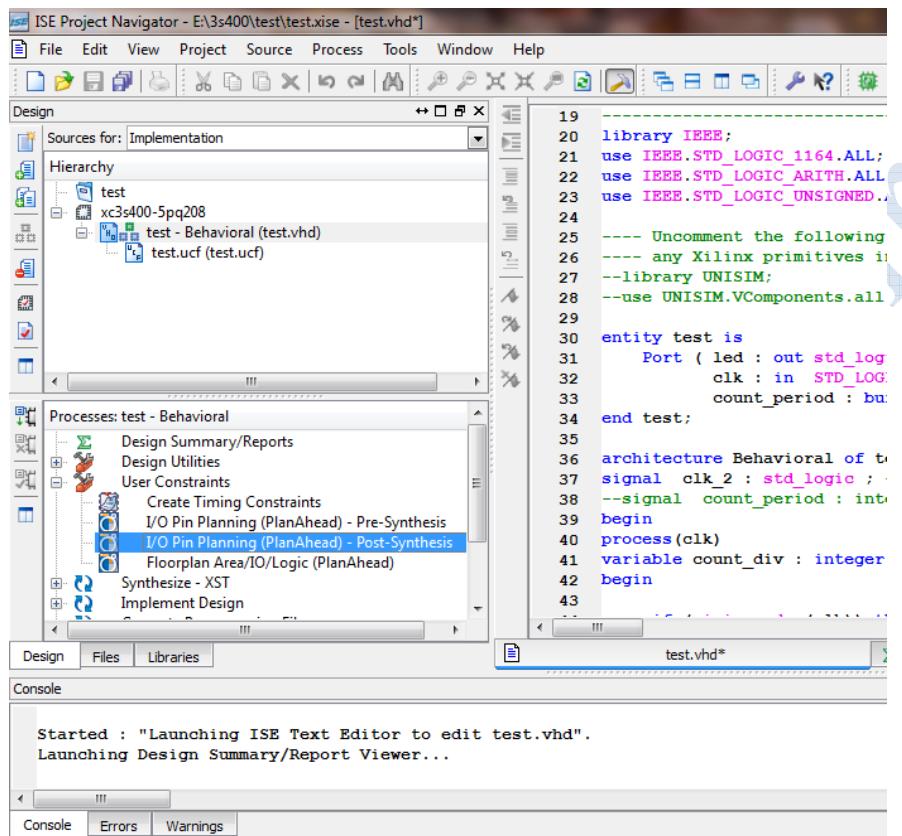
ابتدا بروی پروژه کلیک راست کرده و مطابق شکل گزینه **New Source** را انتخاب می نماییم.



سپس طبق شکل زیر گزینه Implementation Constrains File را انتخاب و در قسمت File name اسم فایل دلخواه را وارد می کنیم و بر روی Next کلیک و سپس Finish را می زنیم.



در این حالت فایل مورد نظر با پسوند **.ucf** ایجاد می گردد.(مثلا **.test.ucf**). مطابق شکل زیر به قسمت User Constraints I/O Pin Planing رفته و زیر شاخه را کلیک می کنیم تا باز شود.



در این قسمت هر ورودی یا خروجی طرح را مطابق میل خودمان به هریک از پین های FPGA اختصاص می دهیم، که این کار را می توان به دو روش مطابق شکل زیر انجام داد.

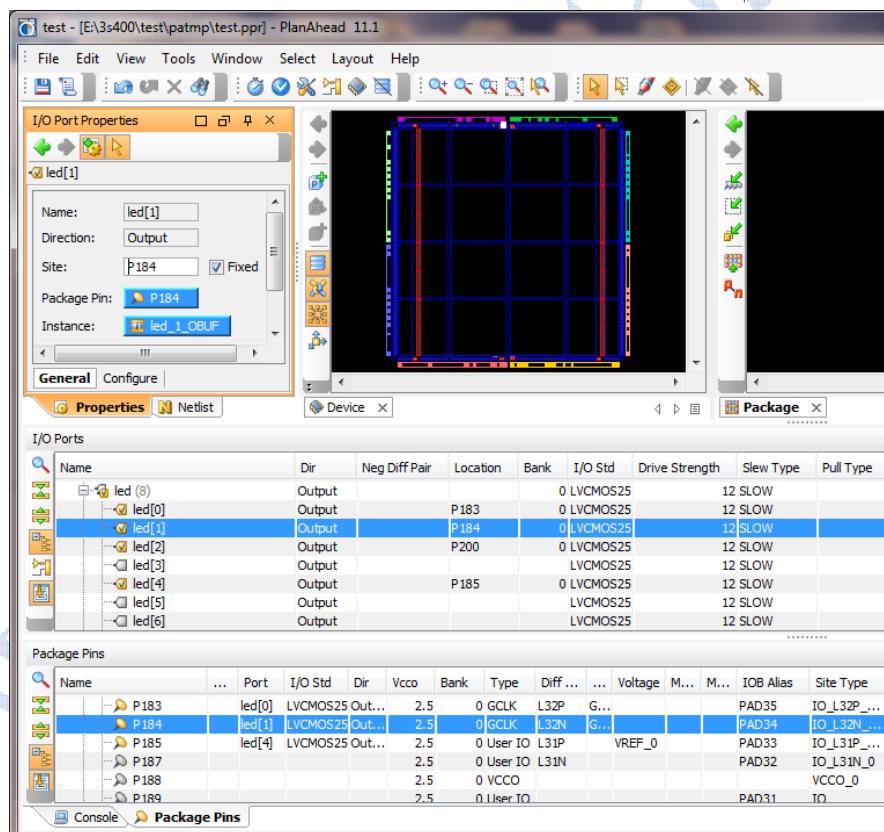
## ۱- عمل کشیدن (Drag)

## ۲- اختصاص دادن شماره پین به صورت دستی

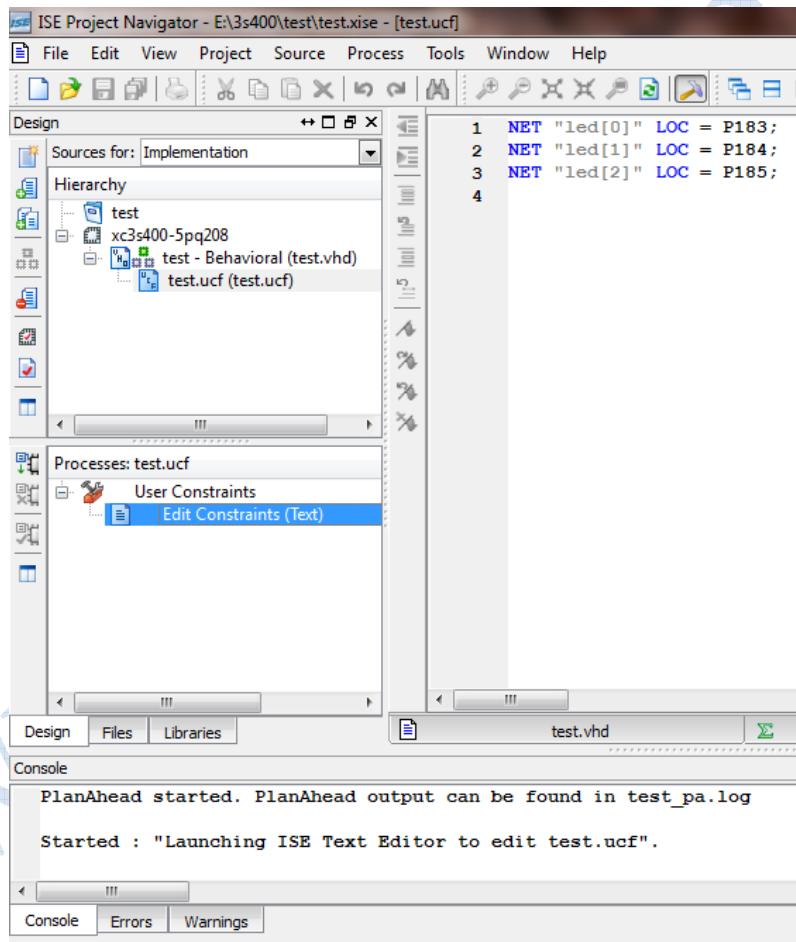
آوا صنعت و علم رهپویان کمپانی

۱- عمل کشیدن (Drag) : ابتدا پورت مورد نظر را از قسمت(سربرگ) I/O Ports انتخاب و آن را به قسمت(سربرگ) Package Pins کشیده و بر روی شماره پین مورد نظر در FPGA ، رها می کنیم. بدین ترتیب پورت انتخاب شده به پین مورد نظر اختصاص داده می شود.

۲- اختصاص دادن شماره پین به صورت دستی : ابتدا در قسمت(سربرگ) I/O Ports بر روی پورت مورد نظر کلیک چپ می کنیم تا انتخاب شود، سپس در قسمت(سربرگ) I/O Port Properties شماره پین مورد نظر در FPGA را وارد می کنیم.



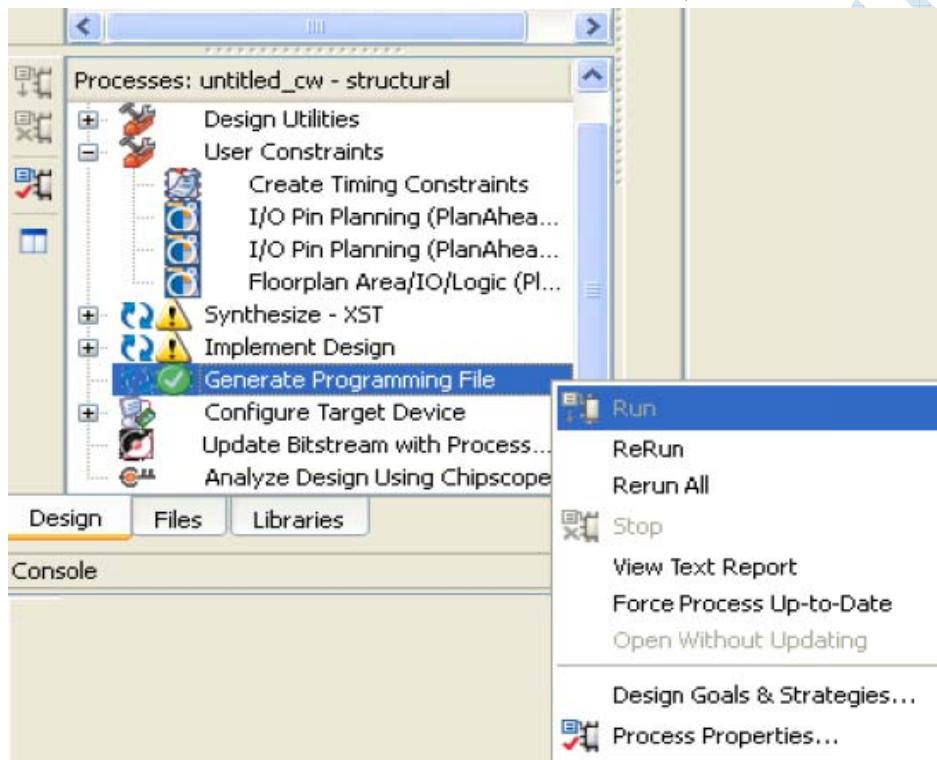
سپس صفحه را ذخیره کرده و می بندیم. بدین ترتیب ورودی و خروجی ها به پین های مورد نظر در FPGA اختصاص داده می شود. یکی دیگر از روش های دستی جهت اختصاص پین ها اینست که ابتدا فایل ساخته شده با پسوند .ucf را انتخاب و در قسمت پایین مطابق شکل زیر به قسمت User Constraints و زیرشاخه Edit Constraints(text) را کلیک می کنیم تا محتویات آن باز شود، در قسمت متنی باز شده می توان شماره پایه ها را به شکل دلخواه به صورت دستی تغییر و یا اختصاص داد.



شرکت رهپویان علم و صنعت آوا

### تولید فایل مورد نیاز برای پروگرام کردن : FPGA

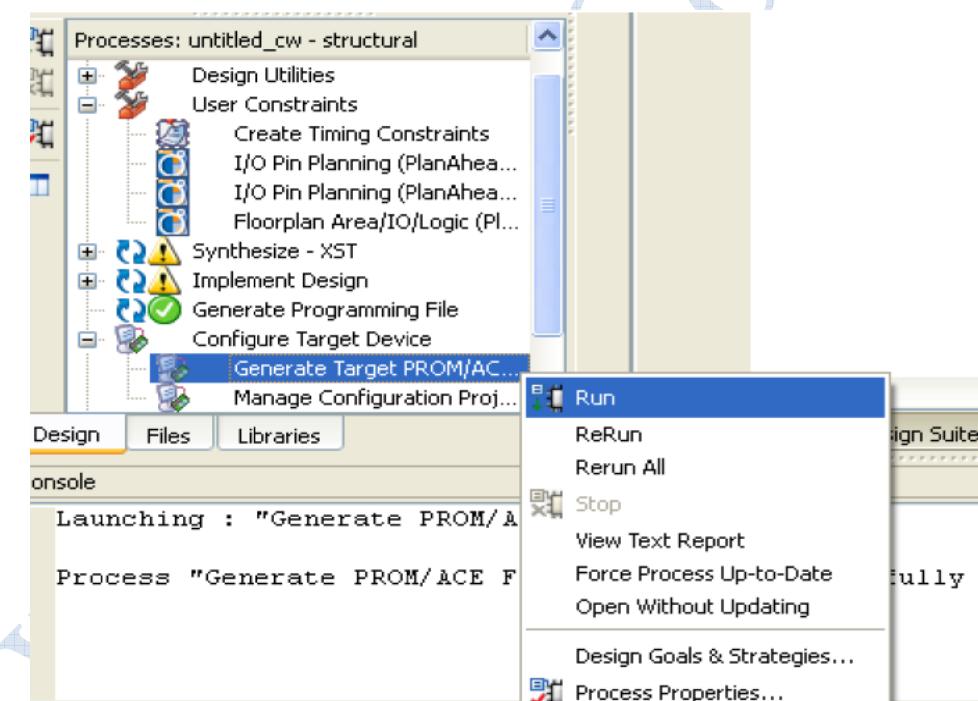
بعد از اینکه اختصاص پایه ها به صورت دستی انجام گرفت مطابق شکل زیر بر روی کلیک راست کرده و Run را انتخاب می کنیم تا فایل مورد نیاز برای پروگرام کردن FPGA تولید شود. پسوند فایل ایجاد شده .bit می باشد.



## تولید فایل مورد نیاز برای پروگرام کردن حافظه‌ی XCF02S

به دلیل اینکه ساختار FPGA بر اساس RAM می‌باشد و با قطع برق اطلاعات آن از بین می‌رود برای پروژه‌هایی که نهایی می‌شوند در کنار FPGA حافظه‌ی XCF02S قرار داده شده و برنامه را در این حافظه قرار میدهند و هنگام Power up برنامه‌ی حافظه‌ی XCF02S در FPGA بارگذاری می‌شود.

حال برای تولید فایل مورد نیاز برای پروگرام کردن حافظه‌ی XCF02S بایستی مطابق شکل زیر بر روی Generate Target PROM کلیک راست کرده و Run را انتخاب می‌کنیم تا فایل مورد نیاز برای پروگرام کردن حافظه‌ی XCF02S تولید شود.



فایل ایجاد شده برای پروگرام کردن حافظه‌ی XCF02S با پسوند .mcs، .hex، .bin و ... می‌باشد.

## نحوه برنامه ریزی کردن (Programmig) تراشه یا حافظه ی XCF02S با استفاده از نرم افزار iMPACT

بعد از اینکه برنامه نوشته شده را بدون خطا سنتز کردیم به ترتیب زیر عمل می کنیم:

۱- قبل از شروع برنامه ریزی FPGA و یا حافظه ی XCF02S توسط کامپیوتر ، بایستی وضعیت سوییچ های M0 M1 M2 طبق جدول زیر بر روی JTAG تنظیم شده باشد. یعنی M1 در وضعیت ON و بقیه در وضعیت OFF باشند. توجه شود که وضعیت ON به معنای اعمال صفر منطقی می باشد.

Configuration Mode	M0	M1	M2	Synchronizing Clock	Data Width	Serial DOUT
Master Serial	0	0	0	CCLK Output	1	Yes
Slave Serial	1	1	1	CCLK Input	1	Yes
Master Parallel	1	1	0	CCLK Output	8	No
Slave Parallel	0	1	1	CCLK Input	8	No
JTAG	1	0	1	TCK Input	1	No

۲- ابتدا باید از ایجاد فایل با پسوند **.bit** مطمئن باشیم: (برای برنامه ریزی کردن حافظه XCF02S باید فایل با پسوندهای دیگری مانند **.mcs** . ایجاد کرد ) جهت ایجاد فایل ابتدا باید بر روی **Generate Programming File** دوبار کلیک تا فایل ها ایجاد شوند.

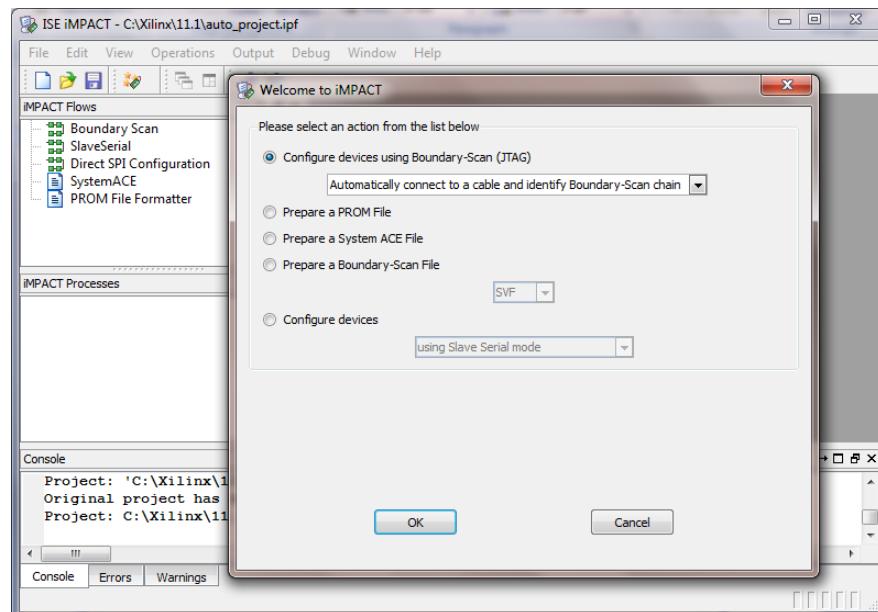
۳- پروگرامر را به پورت پریتر کامپیوتر وصل کرده و طرف دیگر آنرا به پورت JTAG ساخته شده بر روی برد وصل می کنیم.

۴- سپس برد را روشن می کنیم.

۵- در این مرحله برنامه iMPACT را لمسیم Start/All Program/Xilinx ISE/accessories/iMPACT اجرا می کنیم.

۶- باز می شود و جعبه محاوره ConfigureDevices نمایش داده می شود

۶- گزینه **Configure devices using Boundary Scan (JTAG)** را انتخاب کنید



۷- مطمئن شوید

۸- انتخاب **Automatically connect to a cable and identify Boundary-Scan Chain**

شده است.

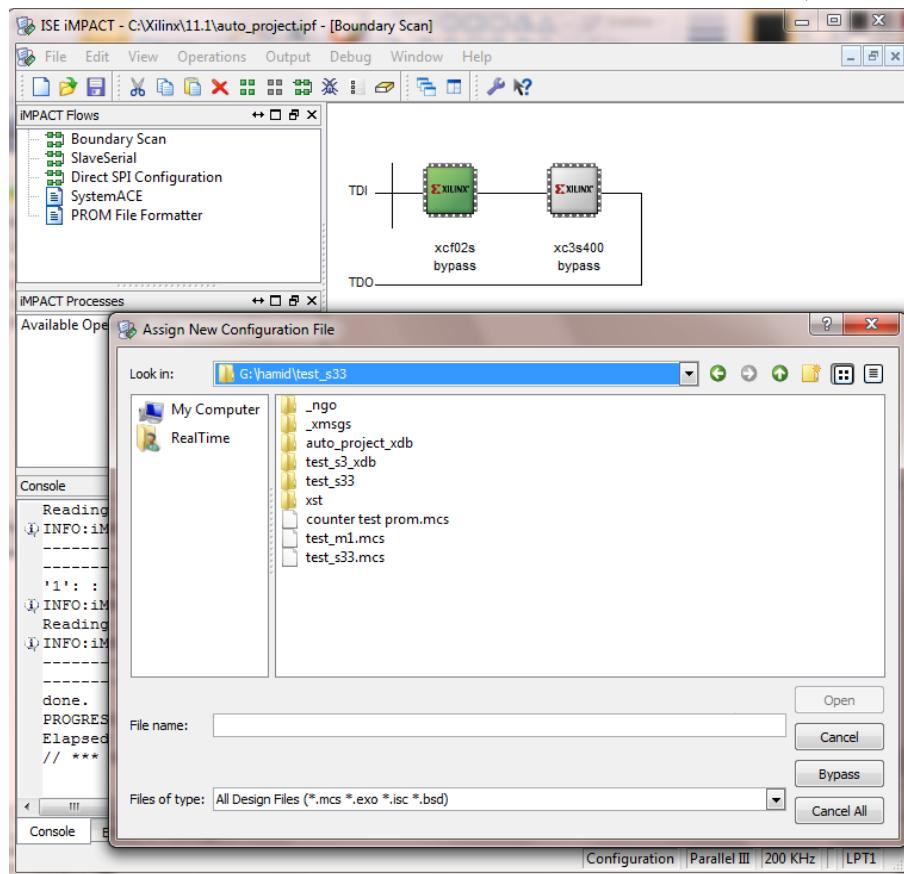
۹- بر روی **Finish** کلیک کنید.

۱۰- در صورت دریافت پیغام وجود دو **device** بر روی **OK** کلیک کنید  
 در پنجره iMPACT windows تمام **device** های متصل به زنجیره **JTAGChain** ( XCF02S ) ، شامل تراشه های اصلی و حافظه **JTAG** با مشخصات دقیق نمایش داده می شود

۱۱- با کلیک بر روی هر کدام از **device** ها، قابلیتها و کارهای قابل اعمال بر روی آن در سمت چپ صفحه ظاهر می شود.

۱۱- در این مرحله بر روی device مورد نظر کلیک راست کرده و گزینه AssignNewConfigurationFile را انتخاب می کنیم.

۱۲- در این حالت طبق شکل باید آدرس برنامه ای که ایجاد کرده ایم را قرار بدهیم. در صورتی که تراشه Spartan را انتخاب کنیم نوع فایل باید با پسوند .bit (در مراحل قبلی .mcs) ایجاد شده است) انتخاب کنیم و اگر حافظه‌ی XCF02S را انتخاب کنیم پسوند .mcs (در مراحل قبلی ایجاد شده است) یا پسوندهای خاص حافظه‌ی XCF02S را انتخاب می کنیم.



شرکت رهپویان علم و صنعت آوا

۱۳-پس از انتساب فایل مورد نظر به تراشه Spartan-xc3s400 یا حافظه‌ی XCF02S، ابتدا حافظه‌ی XCF02S را انتخاب کرده و گزینه... program... در سمت چپ Configuration operation صفحه را کلیک کرده تا برنامه ریزی شود، سپس پنجره status باز می‌شود و برنامه شروع به پروگرام کردن می‌کند. در پایان پیغامی مبنی بر program succeeded به شکل زیر ظاهر می‌شود.

## Program Succeeded

تراشه Spartan-xc3s400 را نیز به همین ترتیب می‌توان برنامه ریزی کرد. تفاوتی در اولویت انتخاب تراشه Spartan-xc3s400 یا حافظه‌ی XCF02S جهت برنامه ریزی وجود ندارد.

در اینجا برد ما برنامه ریزی شده و می‌توان آنرا از کامپیوتر جدا کرد. اگر فقط تراشه Spartan را برنامه ریزی کرده باشیم، با قطع تغذیه برد، برنامه‌ی پروگرام شده بر روی تراشه پاک می‌شود. در صورتی که حافظه‌ی XCF02S را برنامه ریزی کرده باشیم با قطع تغذیه برنامه پروگرام شده بر روی آن از بین نمی‌رود.

نکته: برای اینکه هنگام روشن کردن مجدد برد، برنامه‌ی ذخیره شده در حافظه‌ی XCF02S به صورت اتوماتیک بر روی FPGA بارگذاری شود باستی وضعیت سوییچ های M0 M1 M2 بر روی Master Serial تنظیم شده باشد.

در صورتی که این کار به درستی انجام شود LED DONE نشانگر بر روی برد روشن می‌شود. می‌توان این عملیات را در هنگام روشن بودن برد به صورت دستی با فشردن کلید PROG انجام داد.

**پشتیبانی فنی و خدمات:**

جهت دریافت نسخه های جدید راهنمای استفاده ، نمونه کدهای کاربردی و یا کسب اطلاعات بیشتر می توانید به آدرس [www.dspcore.ir](http://www.dspcore.ir) مراجعه فرمایید. و در صورت هرگونه سوال، پیشنهاد و انتقاد با ایمیل [info@dspcore.ir](mailto:info@dspcore.ir) و [dspcore@yahoo.com](mailto: dspcore@yahoo.com) و یا شماره های ۰۹۱۲۷۰۱۰۶۳۸ - ۰۲۱۷۷۲۲۹۲۵۶ - ۰۹۱۲۶۸۳۱۶۳۶ تماس حاصل فرمایید.

**با آرزوی موفقیت**

**شرکت رهپویان علم و صنعت آوا**

تیم پشتیبانی فنی پاییز ۱۳۸۹