

به نام خدا

آموزش کار با نرم افزار Xilinx ISE Design Suite

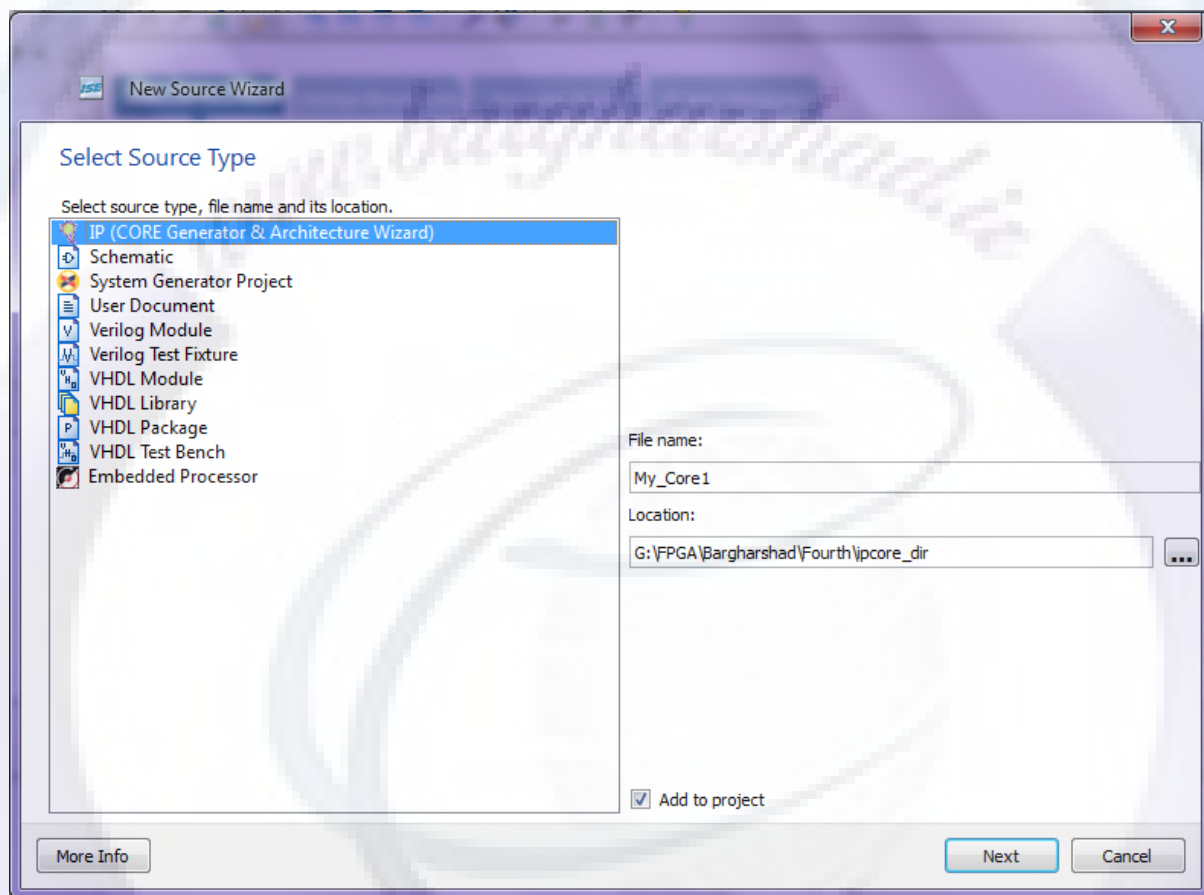
**قسمت چهارم : آشنایی با هسته های نرم افزاری ( Cores )**

در نرم افزار Xilinx ISE و همچنین نرم افزار های دیگر مربوط به کارخانه های دیگر تولید FPGA بلوک های از پیش ساخته شده ای وجود دارد که برای کار های مشخصی ساخته شده اند و بستگی به مدل FPGA قابلیت های بیشتر یا کمتری دارند. این بلوک ها را افراد خبره در بهترین حالت ساخته اند تا کسانی که می خواهند با FPGA کار کنند نروند از اول آن ها را بسازند و بهینه هم نباشد. این بلوک ها که در نرم افزار به آنها Core گفته می شود برای مثال DSP Blocks, FIFO, blockRAM, FFT, CORDIC از انواع این هسته های نرم افزاری هستند .

در این قسمت از آموزش قصد داریم تا شما را با روند کار با Core ها آشنا کنیم و کار با یکی از هسته های نرم افزاری ساده را بیان کنیم در قسمت های بعدی آموزش (مربوط به هسته های نرم افزاری) کار با چند هسته ی نرم افزاری پر کاربرد را برای شما شرح می دهیم .

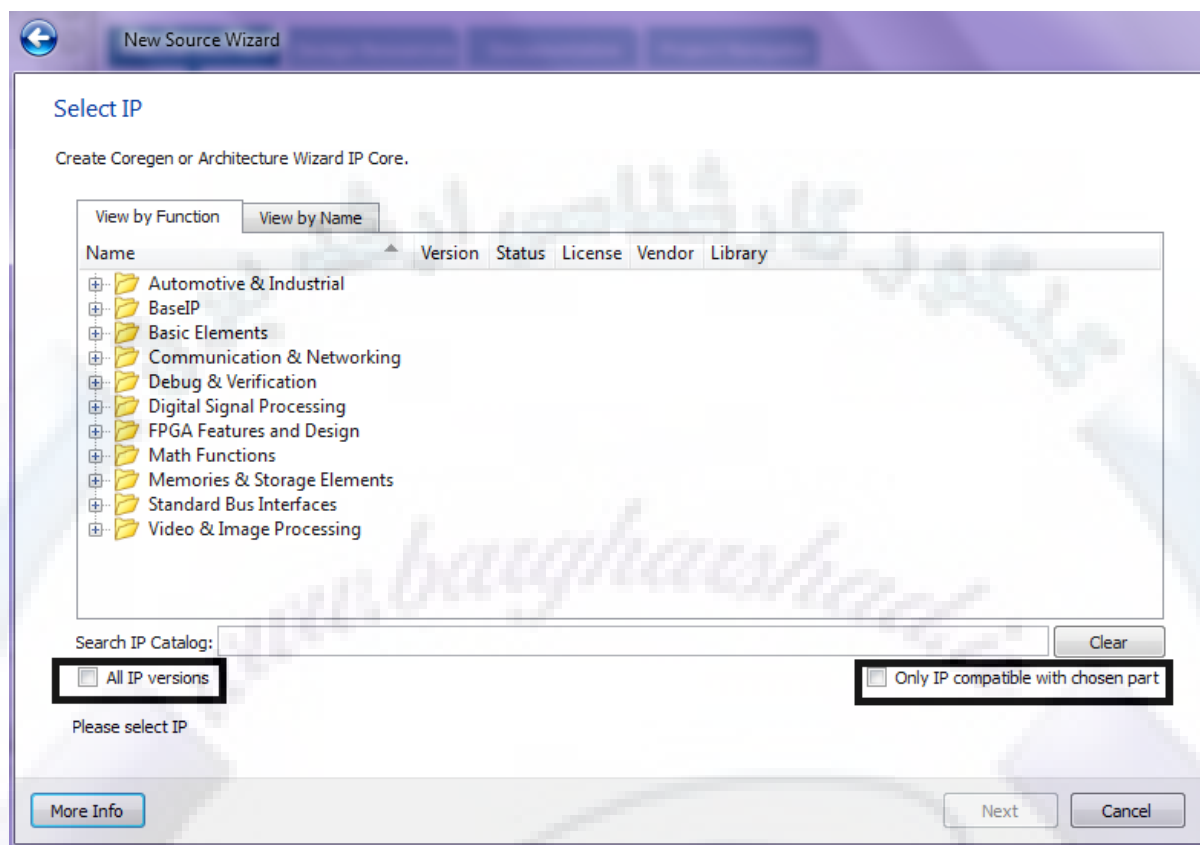
برای شروع شما باید یک پروژه ی بسازید ،به این نکته توجه داشته باشید که FPGA ای که انتخاب می کنید بر این که از چه هسته هایی می توانید استفاده کنید تاثیر دارد ، در این جا شما همان Spartan 3 را انتخاب کنید و مدل XC3S50 را انتخاب کنید تا در ادامه به شما نشان دهم که چه هسته هایی را می توانیم استفاده کنیم .

حال در پروژه ای که ساختید یک New Source ایجاد کنید و از پنجره ی باز شده گزینه ی IP را طبق شکل ۱ انتخاب کنید( در اینجا IP مخفف Intellectual Property است) :



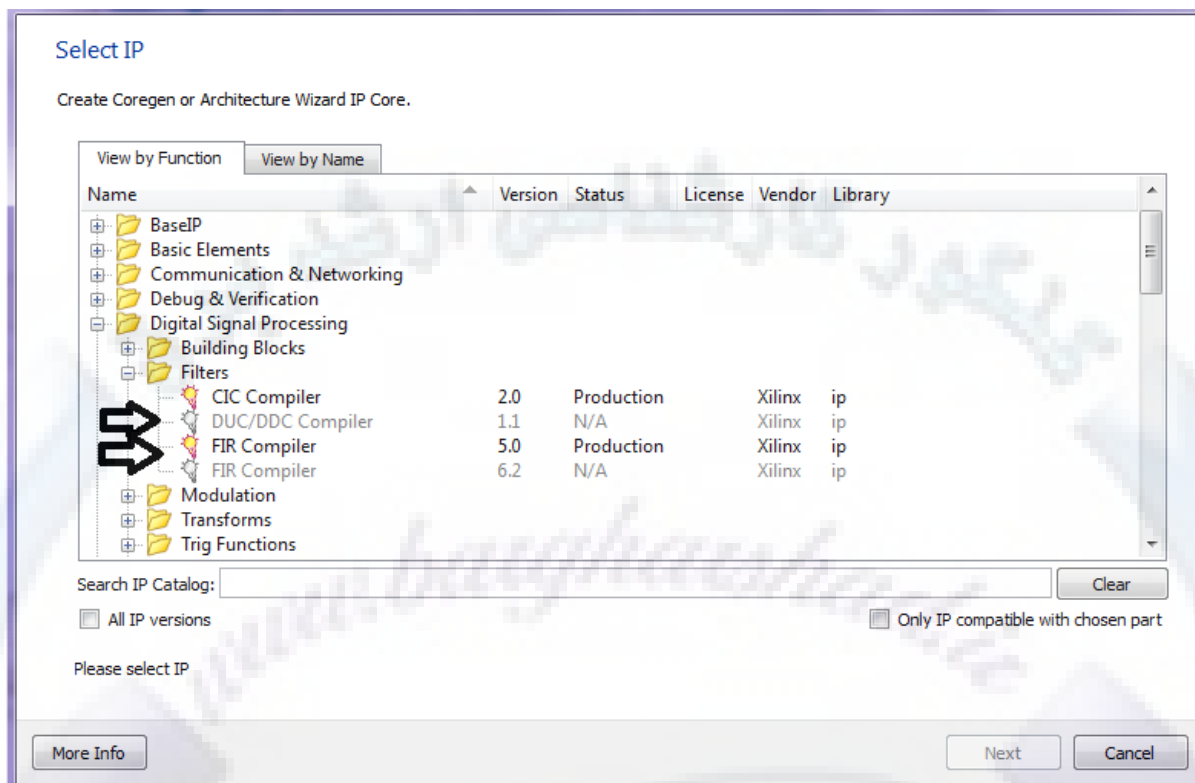
شکل ۱

پس از زدن گزینه ی Next باید کمی صبر کنید تا هسته های نرم افزاری load شوند و لیست زیر (شکل ۲) برای شما نمایش داده شود :



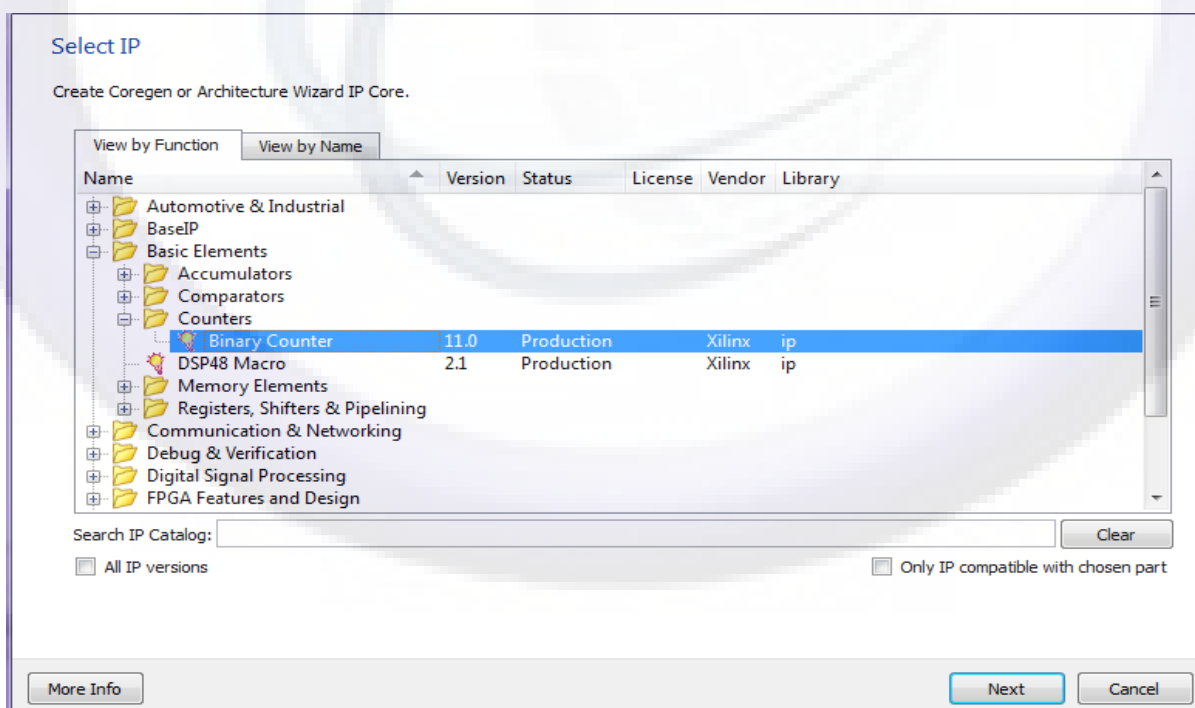
( شکل ۲ )

دو گزینه در صفحه ی فوق (شکل ۲) مشخص شده اند که گزینه ی **ALL IP versions** برای شما تمامی ورژن های موجود هسته های نرم افزاری را برای شما نمایش می دهد و گزینه ی **Only IP Compatible with chosen part** فقط آن هسته های نرم افزاری سازگار با آی سی **FPGA** ای که شما ابتدا انتخاب کردید را نمایش می دهد ، برای مثال شما می بینید که اگر مثل شکل زیر به قسمت **Digital Signal Processign** بروید و قسمت **Filter** را انتخاب کنید می بینید که برخی از هسته ها برای شما قابل استفاده نیست چون با مدل آی سی شما سازگار نیست قطعاً شما اگر از مدل های بالاتر **FPGA** استفاده کنید می توانید از هسته های بیشتری استفاده کنید. در واقع یکی از راه های تبلیغ یک شرکتی مثل **Xilinx** برای فروش محصولات جدیدش این است که می گوید مثلاً این آی سی جدید دارای هسته های نرم افزاری برای پردازش تصویر است و برای کار های پردازش تصویر امکانات ویژه ای در اختیار کاربر قرار می دهد .



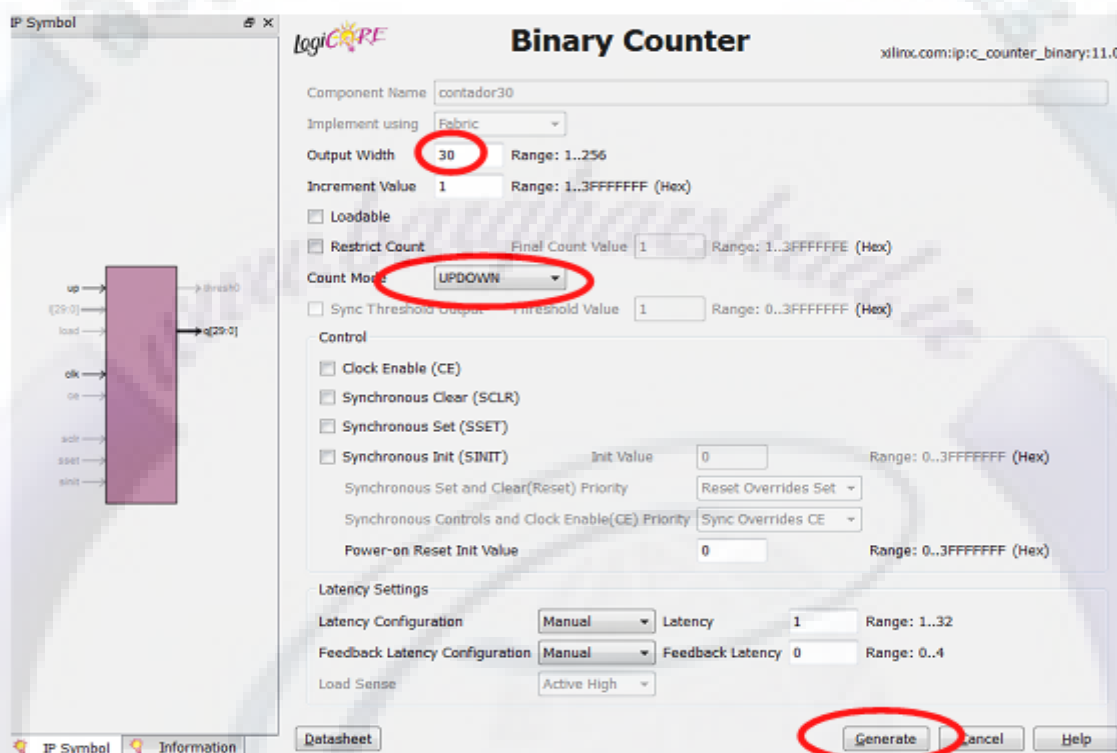
(شکل ۳)

حال می خواهیم به عنوان مثال شما را با هسته ی نرم افزاری شمارنده ی باینری آشنا کنیم از لیست هسته های نرم افزاری گزینه ی Basic Elements را انتخاب کنید از زیر مجموعه ی آن قسمت Counters و از زیر مجموعه ی آن Binary Counter را مطابق شکل ۳ انتخاب کنید :



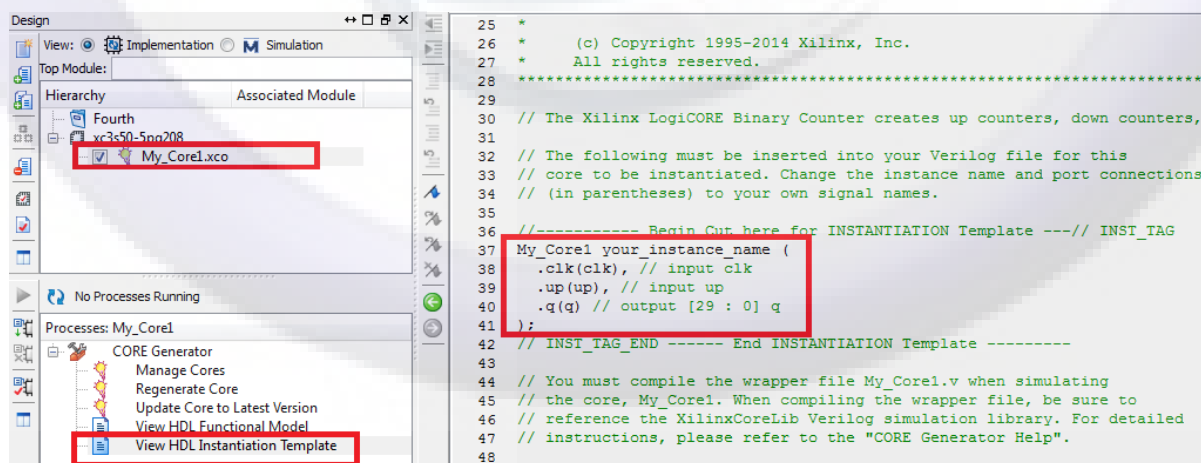
(شکل ۳)

با زدن Next امکانات Wizard مربوط به این هسته ی نرم افزاری برای شما باز می شود. طبق شکل ۴ گزینه ها را پر کنید تا ما یک شمارنده ی بالا/پایین شمار با خروجی ۳۰ بیتی باشیم سپس Generate را بزنید تا هسته برای شما ساخته شود.



(شکل ۴)

پس از ساخته شدن Core باید کد فراخوانی آن را کپی کنید و در Testbench خود استفاده کنید



(شکل ۵)

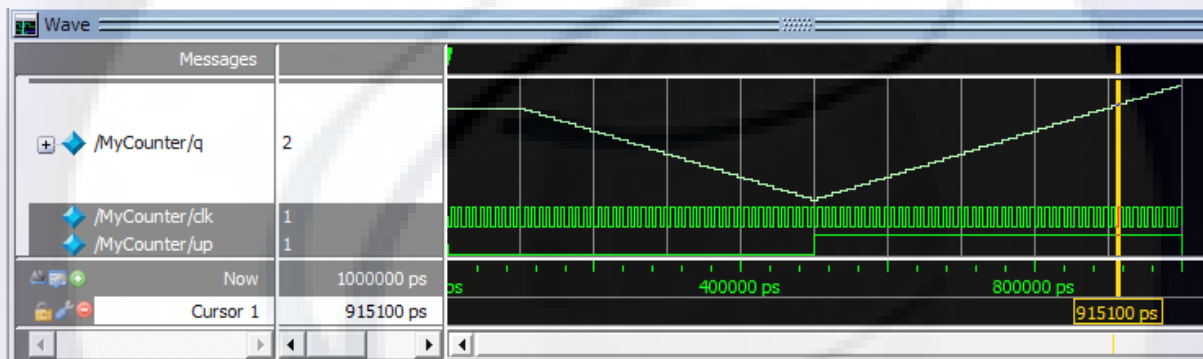
کد فراخوانی Core ساخته شده را مطابق شکل ۵ می توانید مشاهده کنید .

سپس می توانید مشابه کد زیر Testbench مناسبی بنویسید و آن را شبیه سازی کنید و نتایج را مطابق شکل ۶ مشاهده کنید.

```
module MyCounter(output [29 : 0]q);
reg clk=0,up=0;
  My_Core1 Counter1 (
    .clk(clk), // input clk
    .up(up), // input up
    .q(q) // output [29 : 0] q
  );
always
#5 clk=~clk;

initial
begin
#500 up=1;
#500 up=0;
end

endmodule
```



شکل ۶

ملاحظه می کنید که در لحظه ای که Up یک شده است شمارنده شروع به بالا شمردن کرده است.

در محیط Modelsim برای اینکه بتوانید شکل خروجی را مثل شکل فوق به صورت آنالوگ مشاهده کنید باید روی اسم متغیر راست کلیک کنید از قسمت Format گزینه ی Analog(automatic) را انتخاب کنید و همچنین برای اینکه عدد روی شکل را به صورت دهدهی نمایش دهد باید روی متغیر راست کلیک کنید و از قسمت Radix گزینه ی Decimal را انتخاب کنید.

در قسمتهای بعدی آموزش بیشتر با هسته های نرم افزاری آشنا می شویم ....