

پیاده‌سازی تابع آرکتانژانت هیپربولیک بصورت سخت‌افزاری روی **FPGA** با استفاده از **IP Core** **CORDIC** آماده

عنوان درس: طراحی سیستم‌های ریزپردازنده‌ای

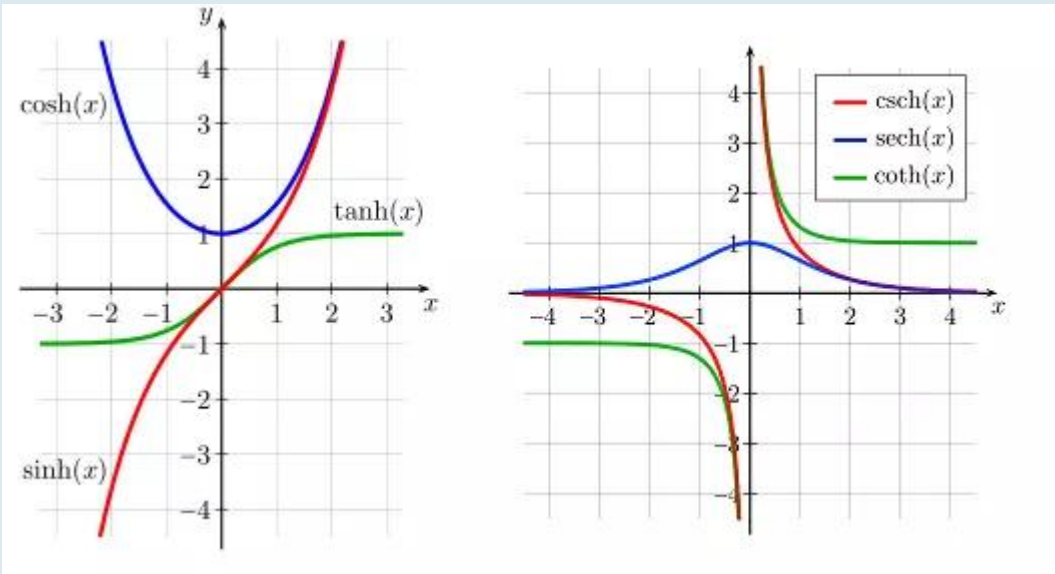
استاد درس: دکتر وحیدی شمس

نام دانشجو: مریم لیاقت

اهمیت موضوع:

توابع مثلثاتی و هیپربولیک در بسیاری از کاربردهای مهندسی DSP، ارتباطات، کنترل خودکار ضروری اند.

محاسبه سخت افزاری این توابع روی FPGA باعث افزایش سرعت و کارایی می شود.

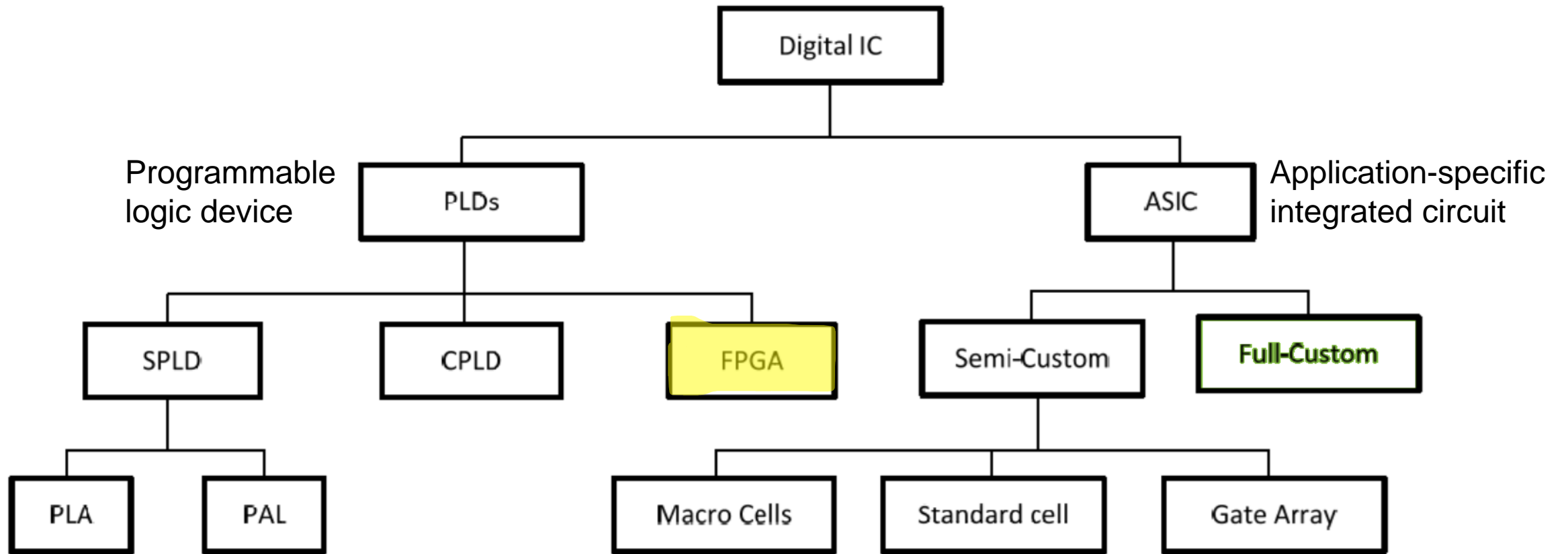


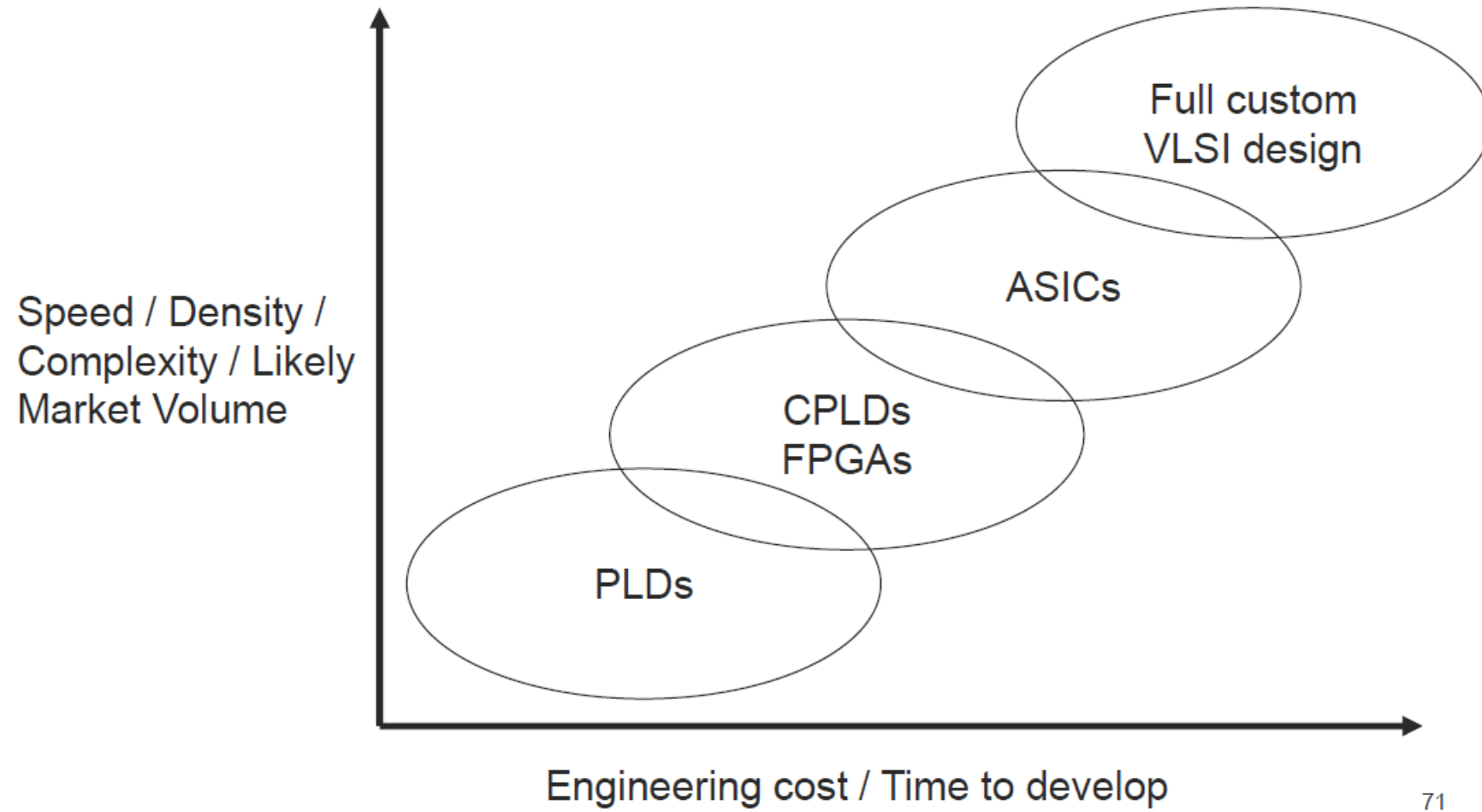
کاربردهای عملی:

۱. پردازش سیگنال: جایی که توابع هیپربولیک برای تبدیل‌های موج استفاده می‌شوند
۲. کنترل خودکار: تنظیمات پیشرفته که نیاز به محاسبه سریع توابع لاگاریتمی/هذلولی دارند
۳. ارتباطات دیجیتال: در الگوریتم‌های دینامیک لینک که محاسبات غیرخطی نیاز است
۴. شبکه‌های عصبی سخت‌افزاری: برخی لایه‌ها بر مبنای توابع

چیست؟ FPGA

Field-Programmable Gate Array





FPGA چیست؟

یک مدار مجتمع IC قابل برنامه‌ریزی در سطح گیت‌های منطقی، شامل:

بلوک‌های منطقی قابل پیکربندی CLB، بلوک‌های حافظه BRAM،

بلوک‌های DSP (ضرب‌کننده و جمع‌کننده)، و منابع مسیردهی Routing Fabric



سرعت بالا

به‌ویژه برای محاسبات عددی سنگین
سیگنال‌های زیاد، پردازش تصویر، و
کنترل‌های بلادرنگ.



اجرای هم‌زمان قابلیت Parallelism

پیاده‌سازی هزاران گیت منطقی که هم‌زمان با
هم عمل می‌کنند.

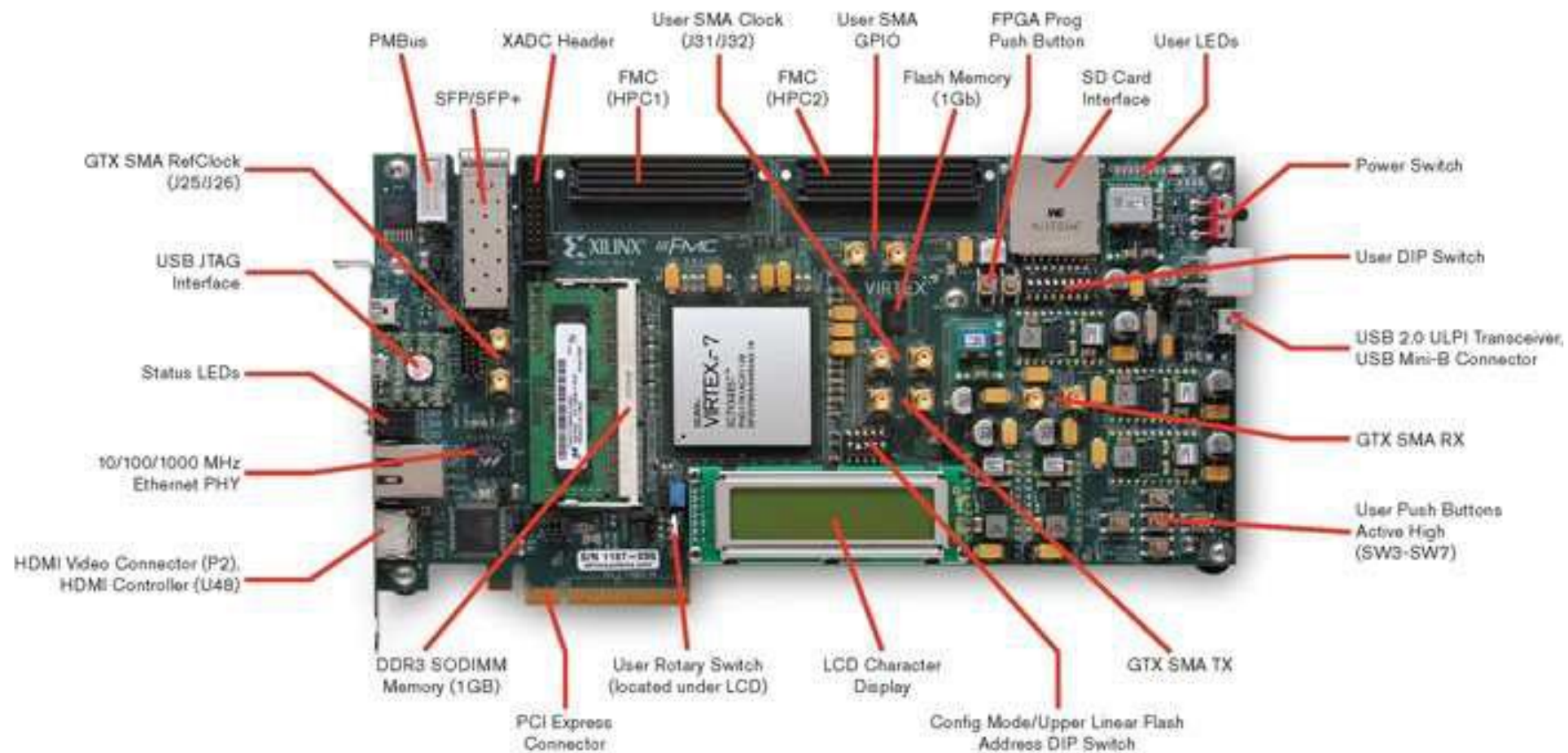


انعطاف‌پذیری بالا:

امکان تغییر سخت‌افزار پس از تولید با
بارگذاری یک فایل bitstream جدید.

کاربرد:

مخابرات (مودم‌ها، کدکننده/دکدکننده)، DSP، اتوماسیون صنعتی، رایانه‌های توکار (Embedded Systems)، شبکه‌های عصبی سخت‌افزاری FPGA-based Neural Nets



ساختار کلی FPGA

بلوک‌های اصلی:

- Configurable Logic Blocks (CLBs)

شامل (LUT) lookup table، فلیپ‌فلاپ، و منطق ترکیبی برای پیاده‌سازی گیت‌های منطقی.

- Block RAM (BRAM)

حافظه‌های داخلی قابل پیکربندی برای ذخیره داده‌ها

- DSP Slices

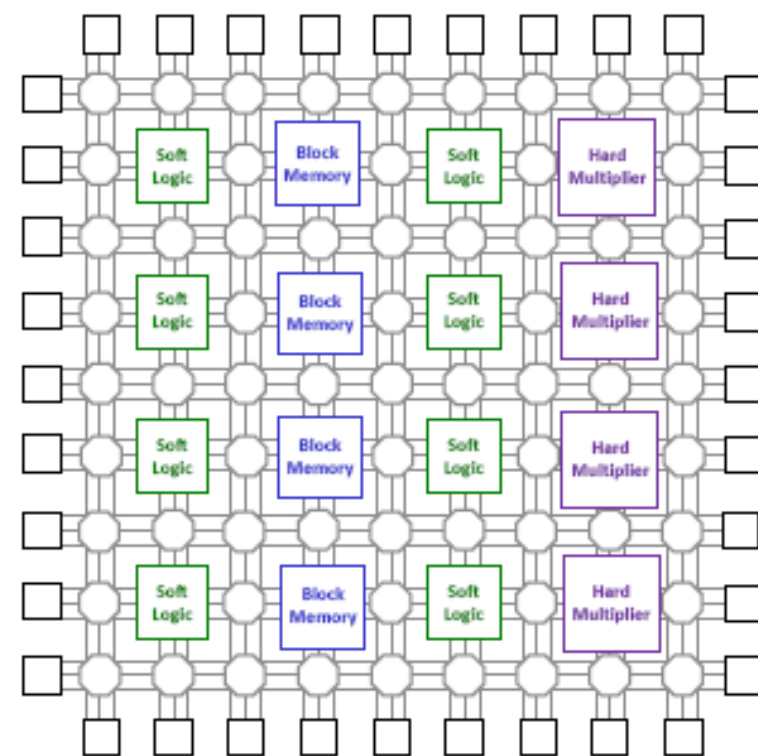
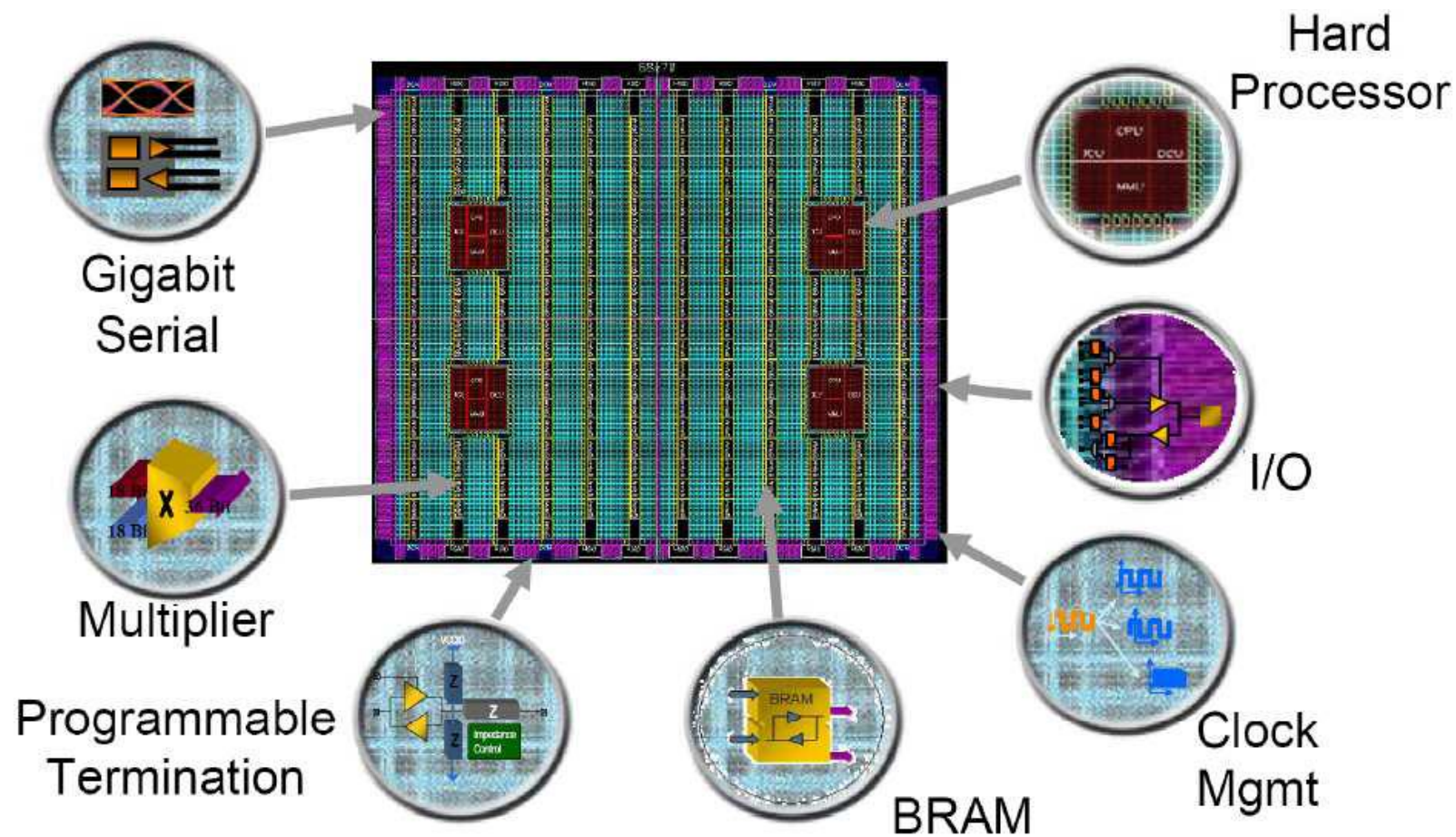
بلوک‌های سخت‌افزاری اختصاصی برای ضرب‌دهی و جمع‌کردن سریع (مورد استفاده در پیاده‌سازی فیلترها و توابع ریاضی).

- I/O Blocks

طدرگاه‌های ورودی/خروجی (برای ارتباط با دنیای بیرون مانند سنسورها، مبدل‌های آنالوگ-دیجیتال، و غیره).

- Routing Fabric

شبکه مسیردهی که پس از سنتز و پیاده‌سازی، ارتباط بین بلوک‌های مختلف را برقرار می‌کند.



فرآیند طراحی:

۱. توصیف سخت افزار HDL
با زبان هایی مثل Verilog یا VHDL

۲. سنتز Synthesis
تبدیل کد HDL به گیت های منطقی

۳. Place & Route
قرار دادن و سیم کشی

۴. تولید فایل Bitstream
فایل نهایی که روی FPGA بارگذاری می شود

Verilog چیست؟

FPGA چیست؟

یک زبان توضیح سخت افزار Hardware Description Language برای توصیف ساختار و رفتار مدارهای شمارشی و ترکیبی.



پشتیبانی از تایمینگ

پارامتر مانند #delay برای توصیف تأخیر.



توصیف رفتار و ساختار (Behavioral) (Structural)

امکان نوشتن کدی که رفتار را با استفاده از تشریح کند، assign و always بلوک های یا اتصال ماژول های زیرسیستمی instantiation.



ساختار ماژولار

هر واحد طراحی (module) می تواند ورودی (input)، خروجی (output)، و پورت های داخلی داشته باشد.

چیست؟ IP Core

IP مخفف Intellectual Property است. یک IP Core ماژولی آماده و آزمایش شده است که عملکرد خاصی (مانند پردازش DSP، کنترلر حافظه، توابع ریاضی) را ارائه می دهد.

تولیدکنندگان FPGA مثلاً Xilinx و Intel و شرکت های ثالث، IP Core های متنوعی (از ساده ترین مثل UART تا پیشرفته ترین مثل پردازش کننده های DSP) را ارائه می دهند.

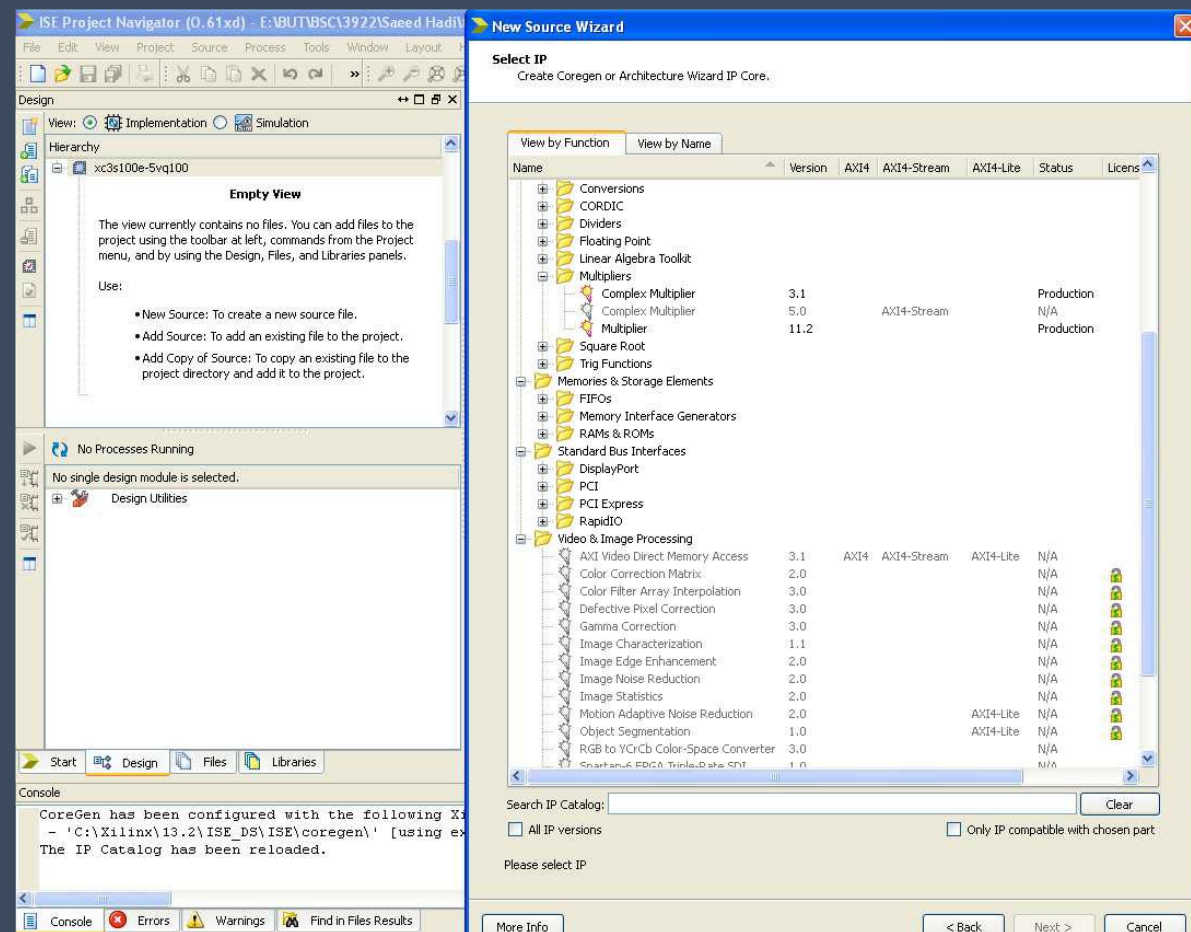
مزایای استفاده از IP Core

۱. کاهش زمان طراحی: نیازی به پیاده‌سازی و اعتبارسنجی از صفر نیست؛ از ماژول‌های آماده و تست‌شده استفاده می‌شود.
۲. بهینه‌سازی شده برای FPGA: توسط تیم تخصصی برای استفاده بهینه از منابع FPGA DSP Slices، BRAM، LUT بهینه شده است.
۳. پشتیبانی و مستندسازی قوی: معمولاً دیتاشیت، راهنمای استفاده و مثال‌های کاربردی همراه IP ارائه می‌شوند.
۴. قابلیت تنظیم Configurability: پارامترهای مختلف (مانند عرض داده، عمق FIFO، یا الگوریتم ویژه DSP) قابل تنظیم هستند.

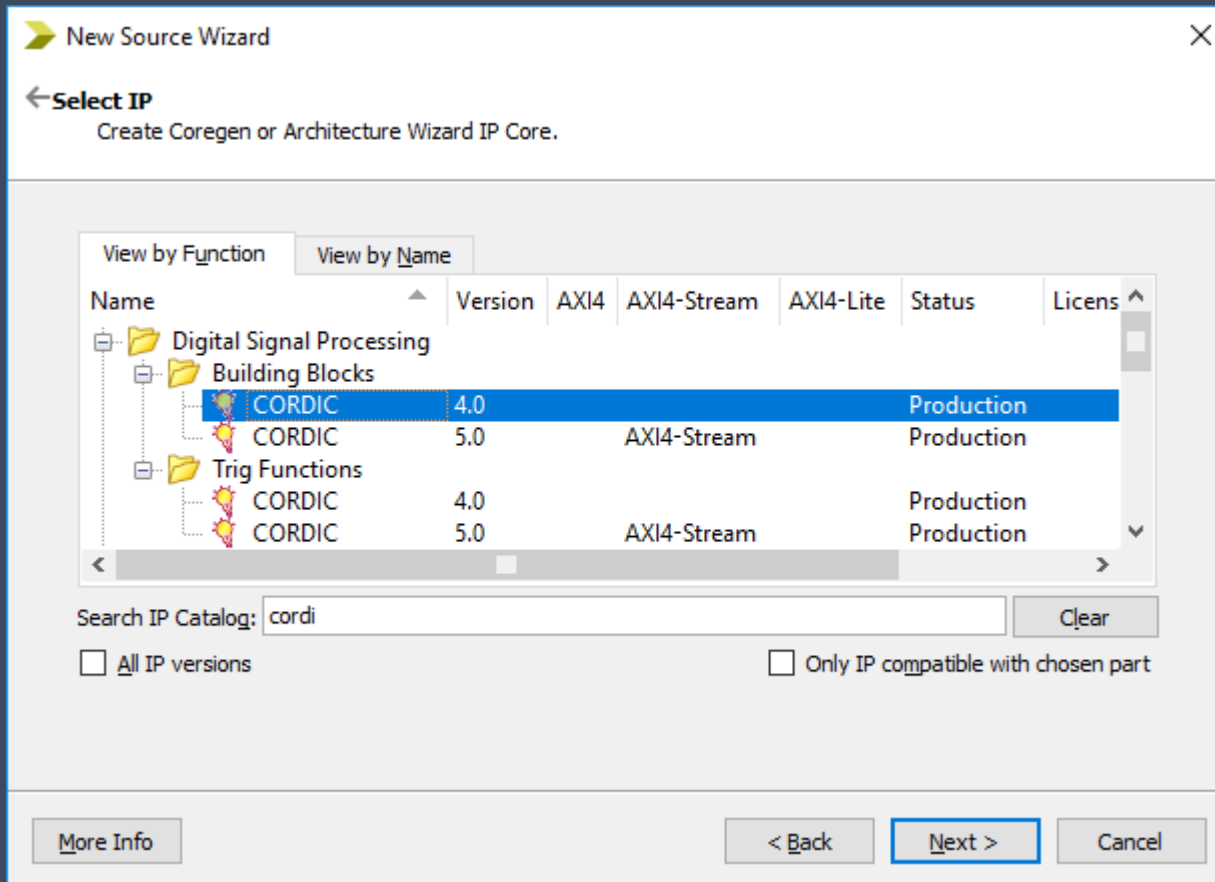
گام‌های پیاده‌سازی IP Core در محیط طراحی FPGA

ابزار طراحی Xilinx ISE Design Suite v14.7

۱. تعریف دستگاه هدف بر اساس شماره FPGA روی برد



۲. اضافه کردن IP Catalog انتخاب IP Core مربوط به الگوریتم CORDIC

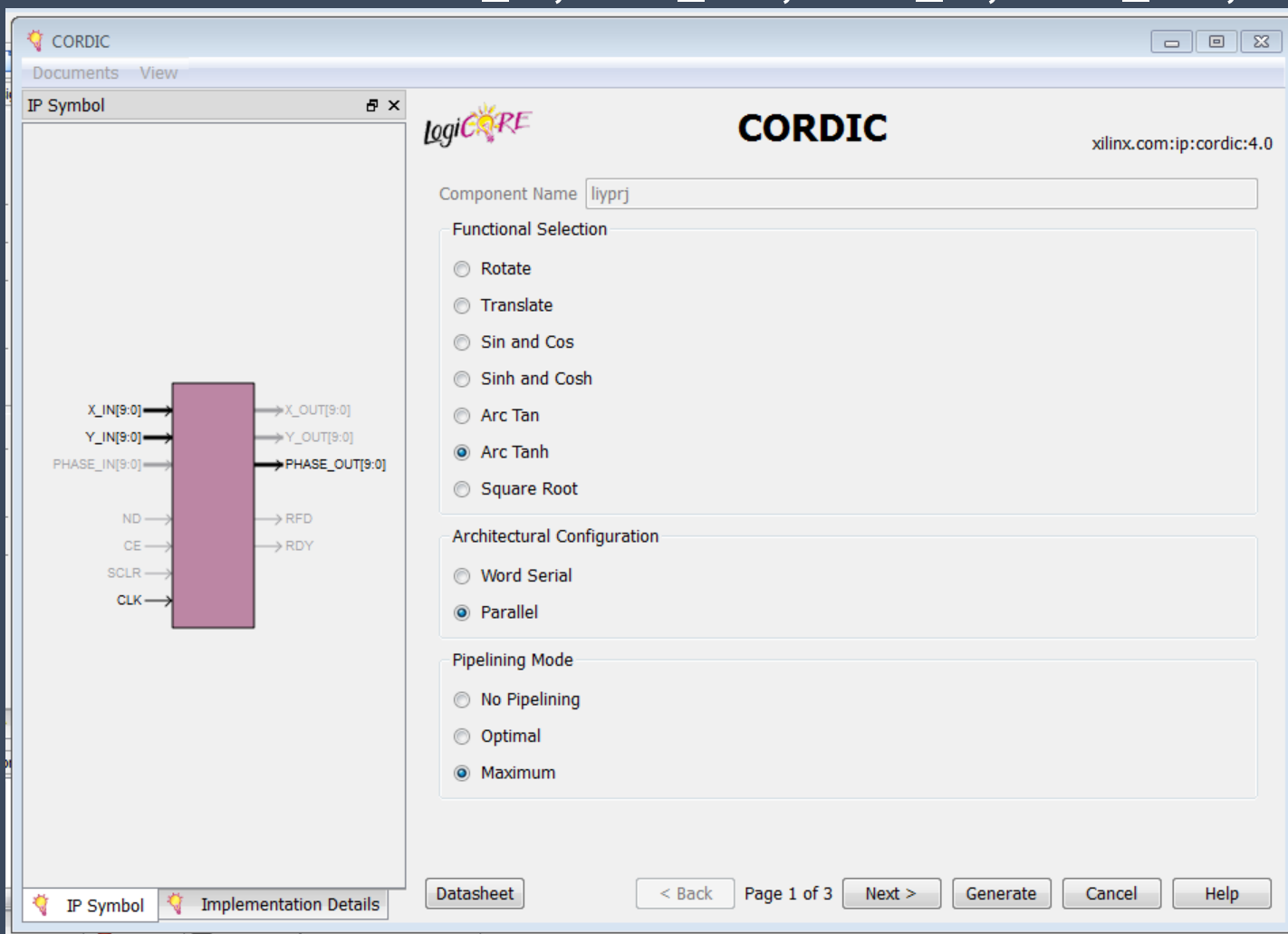


۳. پیکربندی IP Core

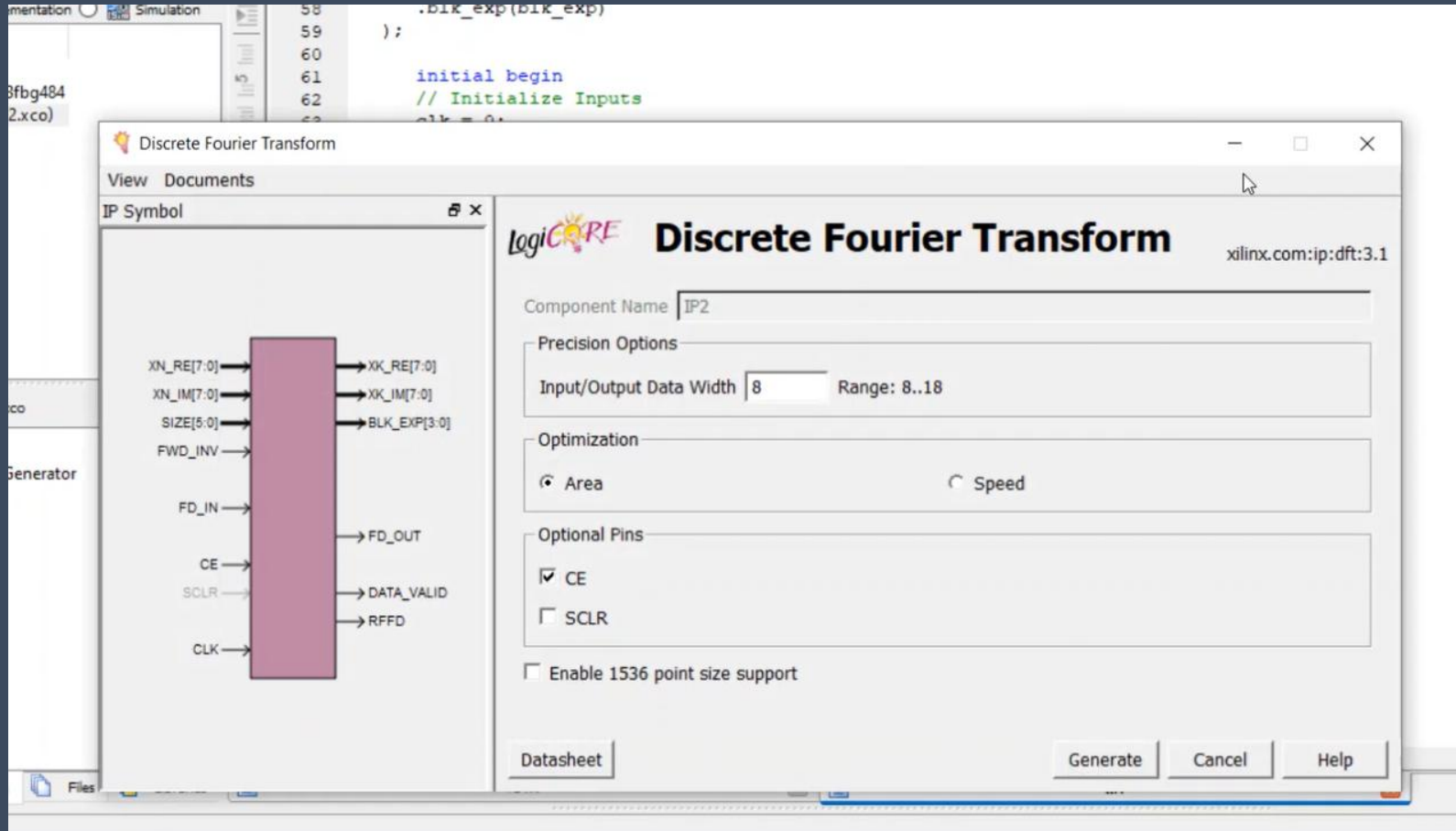
تعیین Mode مثلاً Vectoring mode برای atanh

مشخص کردن دقت (عرض بیت ورودی، تعداد چرخه‌های تکرار)
تنظیم فرمت داده‌ها (fixed-point، تعداد بیت‌ها برای نقطه اعشار)

نامگذاری پورت‌ها مثلاً data_in, data_out, valid_in, valid_out, clk, reset

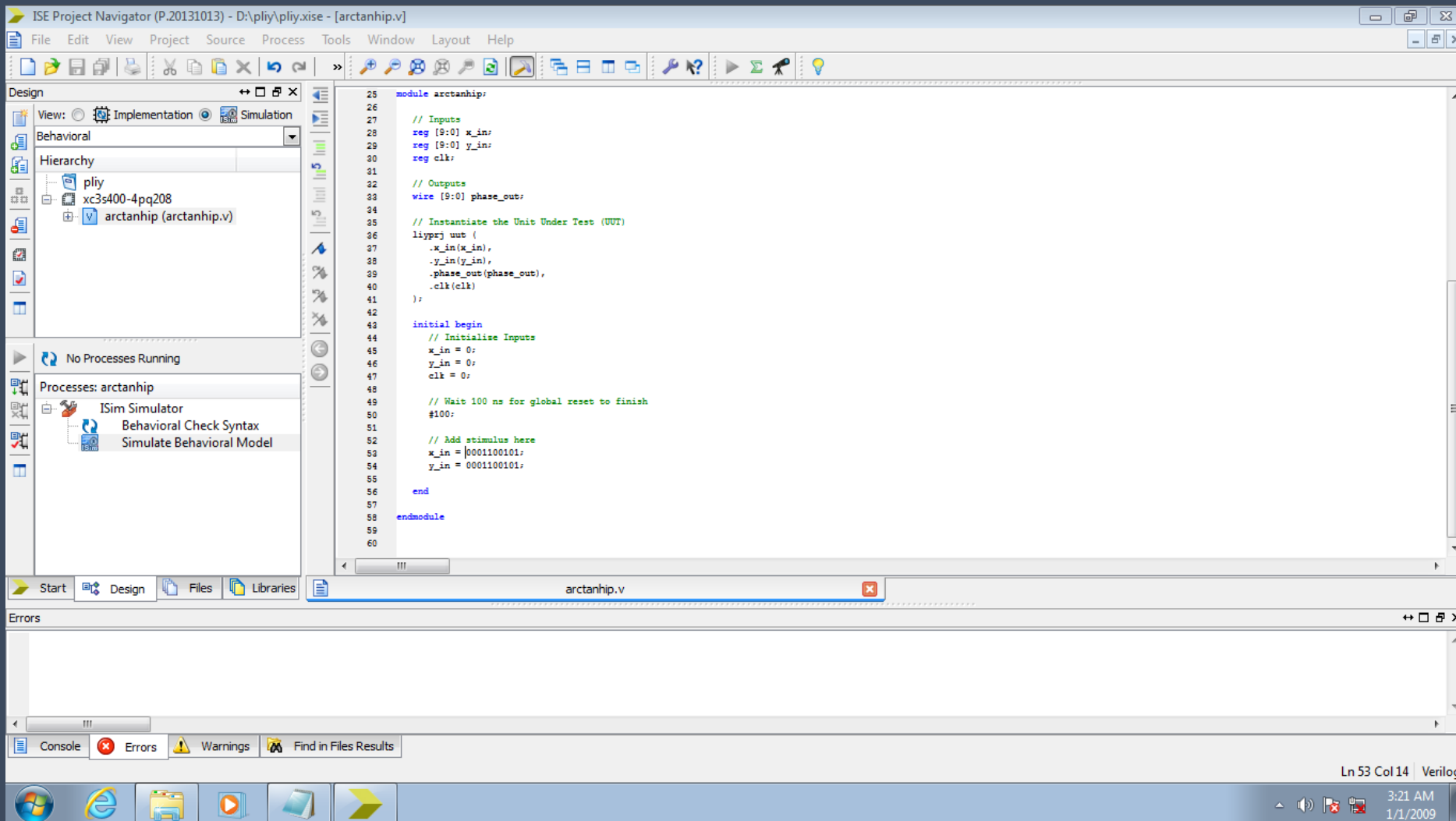


۴. تولید خروجی Generate Output Products با کلیک روی دکمه Generate



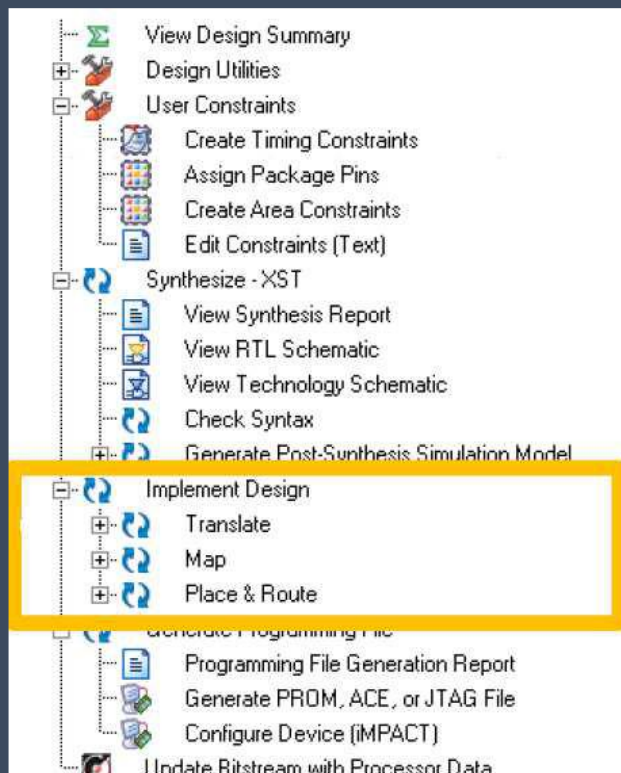
۵. اضافه کردن Testbench

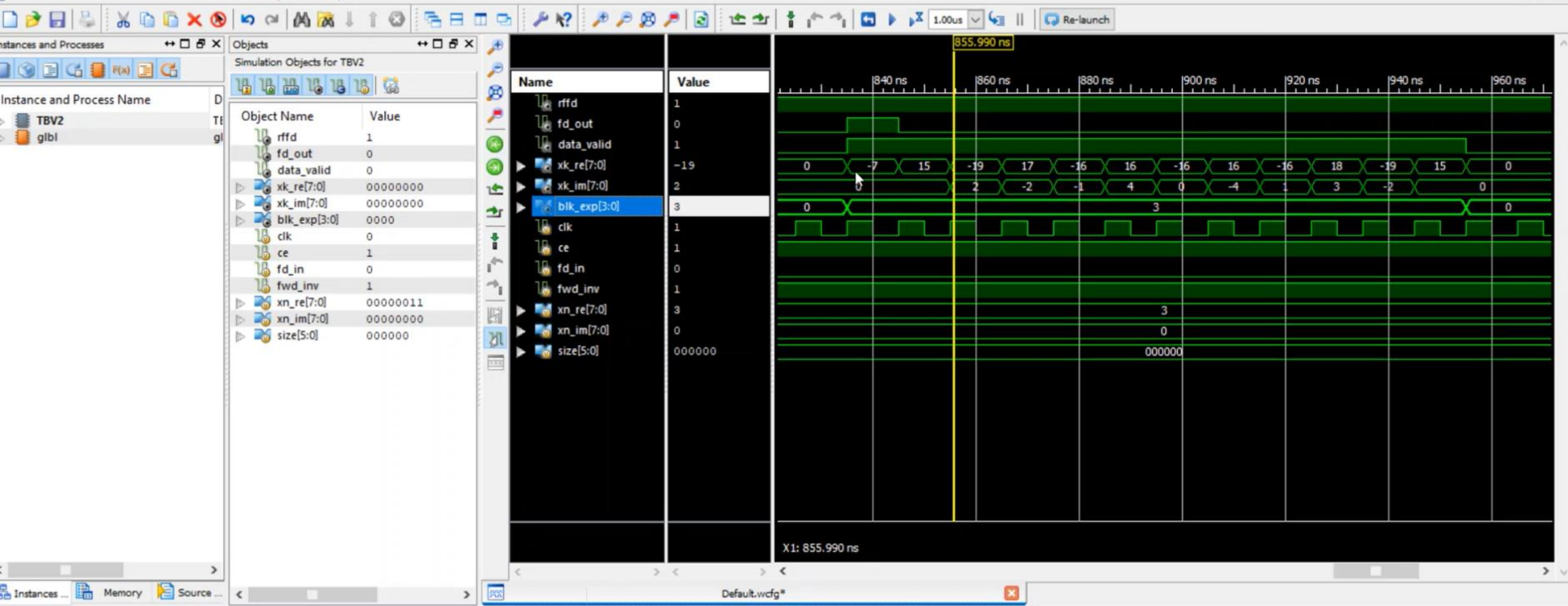
طراحی یک testbench با Verilog برای اعمال ورودی‌های تست مقادیر مختلف و بررسی خروجی‌ها.
شبیه‌سازی با استفاده از شبیه‌ساز داخلی



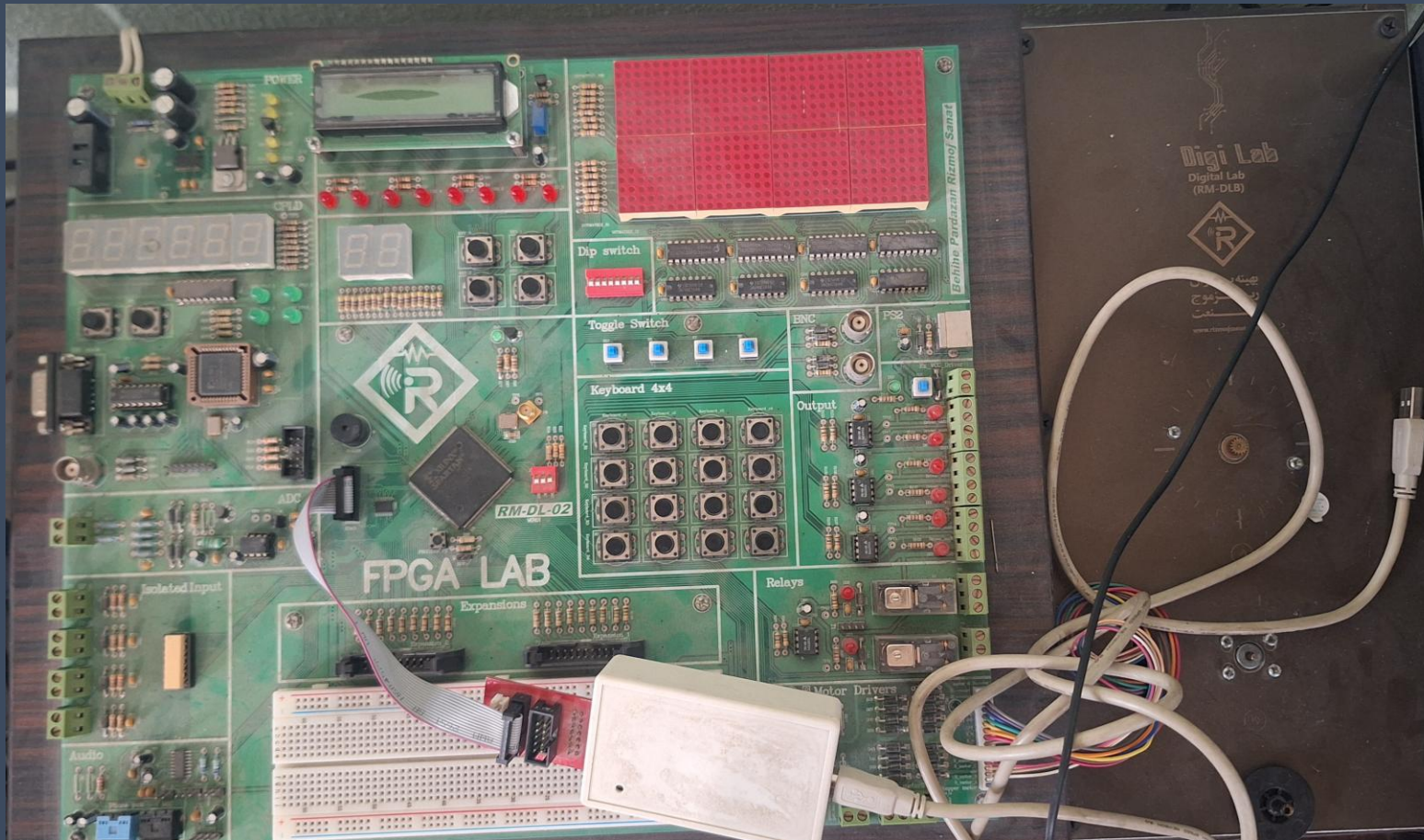
7. سنتز و پیاده‌سازی Implementation

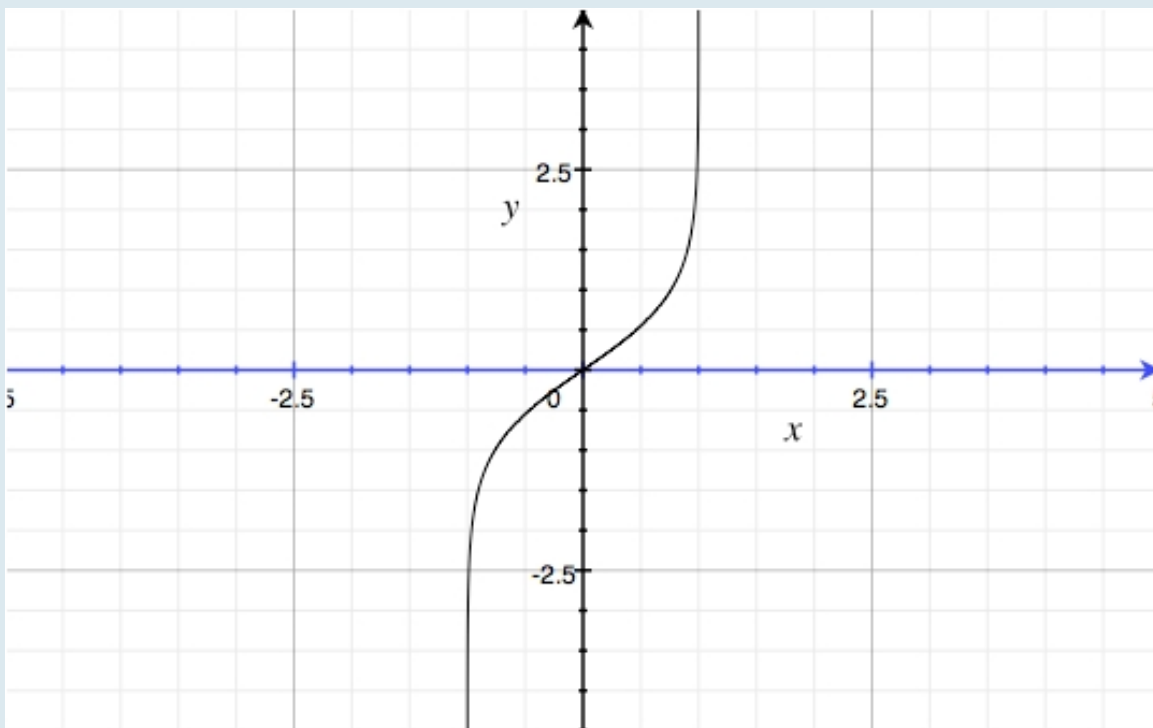
اجرای سازوکار سنتز Synthesis برای تولید netlist
اجرای Place & Route برای تخصیص منابع FPGA و تولید فایل Bitstream.





8. بارگذاری روی برد FPGA و تست سخت‌افزاری:
اتصال برد FPGA به کامپیوتر از طریق کابل USB-JTAG.
بارگذاری فایل Bitstream روی FPGA (Download Bitstream)
اعمال ورودی واقعی (مثلاً از طریق DIP switches، UART، یا USB).
مشاهده خروجی (مثلاً LEDها، سریال اُتپوت یا مانیتور اسکوپ).





با تشکر از توجه شما