بسم الله الرحمن الرحيم

دانشکده فنی و مهندسی

پایان نامه کارشناسی
رشته مهندسی کامپیوتر گرایش سخت افزار

عنوان پایان نامه :
برد آموزشی FPGA

استاد راهنما : آقای دکتر دلیلی

نگارش : محمد اسمعیل دوست

تابستان ۸۵

تقدیم به پدر و مادرم

چکیده:

در این پروژه پایه های I/O بر روی پایه هایی قرار گرفته اند تا هر وسیله جانبی را بتوان به آن متصل نمود . پایه هایی برای برنامه ریزی FPGA به روش های JTAG وSerial Slave تعبیه شده است . این بورد به گونه ای طراحی شده است که پایه های I/O به وسیله خاصی متصل نمی باشد و بعد از طراحی برنامه توسط کاربر چه به صورت شماتیک در Protel DXP و چه بصورت نوشتن کد VHDL ، I/O های تعریف شده را به پایه های FPGA اختصاص می دهیم که این کار را با ساخت Constraint file در Protel DXP و ISE می توانیم انجام می دهیم.

فهرست:

۱- فصل ۱: ساختار کلی FPLD ۱

۱-۱ مقدمه ای بر تراشه FPGA..... ۲

۱-۲ ساختار IC های قابل برنامه ریزی FPGA..... ۴

۱-۲-۱ بلوک های قابل برنامه ریزی CLB..... ۶

۱-۲-۱-۱ بلوک های ترکیبی منطقی..... ۷

۱-۲-۱-۲ مالتی پلکسر..... ۸

۱-۲-۱-۳ فلیپ فلاپ ها..... ۸

۱-۲-۲ سیم ها و بلوک های ارتبا طی..... ۹

۱-۲-۳ بلوک های ورودی – خروجی یا IOB..... ۱۱

۱-۳ ساختار IC های قابل برنامه ریزی CPLD..... ۱۱

۱-۳-۱ حجم گیت..... ۱۱

۱-۳-۲ نوع برنامه ریزی و نگهداری برنامه..... ۱۱

۱-۴ توضیح قسمت های مختلف CPLD به طور خلاصه..... ۱۲
۱-۴-۱ JTAG Controller..... ۱۲
۱-۴-۲ I/O..... ۱۳
۲- فصل ۲: توصیف معماری..... ۱۵
۲-۱ بلوک های IO..... ۱۶
۲-۱-۱ مسیر ورودی..... ۱۹
۲-۱-۲ مسیر خروجی..... ۱۹
۲-۱-۳ بانک های I/O..... ۲۰
۲-۲ configurable logic Block (CLB)..... ۲۲
۲-۲-۱ جدول جستجو..... ۲۲
۲-۲-۲ عناصر ذخیره ساز..... ۲۲
۲-۲-۳ دیگر منطق ها..... ۲۳
۲-۲-۴ منطق های ریاضی..... ۲۳
۲-۲-۵ بافرها..... ۲۳
۲-۳ بلوک حافظه (RAM)..... ۲۴
۲-۴ ماتریس مسیر یابی قابل برنامه ریزی..... ۲۵
۲-۴ مسیر یابی محلی..... ۲۵
۲-۵ مسیر یابی اصلی..... ۲۶

۲-۶ توزیع کلاک ۲۶

۲-۶ توسعه سیستم ۲۸

۲-۷ پیکره بندی FPGA ۲۸

۲-۷-۱ فایل پیکره بندی ۲۹

۲-۷-۲ مدل های پیکره بندی ۲۹

۲-۷-۲-۱ سیگنال ها ۳۰

۲-۷-۳ فرایند پیکره بندی ۳۱

۲-۷-۳-۱ آغاز پیکره بندی ۳۲

۲-۷-۳-۲ پیکره بندی حافظه ۳۳

۲-۷-۳-۳ بار کردن داده های پیکره بندی ۳۴

۲-۷-۳-۳-۱ چک کردن خطاهای CRC ۳۴

۲-۷-۳-۴ فاز شروع ۳۵

۲-۷-۴ مد سریال ۳۶

۲-۷-۴-۱ Serial slave mode ۳۷

۲-۷-۴-۲ مد اسکن مرزی : (JTAG) ۳۸

۲-۷-۵ read black : (باز خوانی) ۴۰

۲-۸ بکار گیری حلقه های delay –locked ۴۰

۲-۸-۱ نماد های کتاب خانه DLL ۴۱

۲-۸-۱-۱ توصیف پایه های BUFGDLL.....۴۳

۲-۸-۱-۱-۱ منبع کلاک ورودی – I.....۴۳

۲-۸-۱-۱-۲ کلاک خروجی – O.....۴۴

۲-۸-۲ توصیف پایه های اولیه CLKDLL.....۴۴

۲-۸-۲-۱ کلاک منبع ورودی CLKIN.....۴۴

۲-۸-۲-۲ فیدبک کلاک ورودی CLKDFB.....۴۴

۲-۸-۲-۳ ورودی Reset.....۴۵

۲-۸-۲-۴ کلاک خروجی CLK2X-2X.....۴۵

۲-۸-۲-۵ خروجی تقسیم کننده کلاک CLKDV.....۴۵

۲-۸-۲-۶ خروجی های کلاک CLK[0/90/180/270] ۴۵

۲-۸-۳ محدودیت مکانی DLL.....۴۶

۲-۸-۴ کلاک ورودی.....۴۷

۲-۹ مدهای عملیاتی ۴۷

۲-۹-۱ Read through.....۴۸

۲-۹-۲ Write Back.....۴۸

۲-۱۰ کتابخانه اصلی.....۴۸

۲-۱۱ بکار گیری I/O های همه منظوره.....۴۹

۲-۱۱-۱ LVTTL (TTL ولتاژ پایین) ۵۱

۲-۱۱-۲ CMOS -LVCMOS2 ولتاژ پائین برای ۲/۵ ولت..... ۵۱

۲-۱۱-۳ PCI (واسط اجزاء جانبی)..... ۵۱

۳-فصل سوم : توضیح ساختار نرم افزار سنتز xilinx (ISE)..... ۵۳

۳-۱ نرم افزارهای مورد نیاز..... ۵۴

۳-۲ شروع افزار نرم ISE..... ۵۵

۳-۳ طراحی به زبان VHDL..... ۵۵

۳-۴ ایجاد پروژه جدید..... ۵۵

۳-۵ تغییر ماژول شمارنده با الگوی شمارنده..... ۵۷

۳-۶ ایجاد یک منبع Test bench ۶۱

۳-۷ مقدار دهی ورودی های شما رنده..... ۶۲

۳-۸ تولید مقادیر خروجی..... ۶۳

۳-۹ وارد کردن طراحی..... ۶۴

۳-۹-۱ ایجاد یک طراحی سطح بالا..... ۶۴

۳-۱۰ سیم بندی طرح..... ۶۵

۳-۱۱ انتساب نام به ورودی ها و خروجی ها..... ۶۶

۳-۱۲ اضافه کردن نشانه های I/O..... ۶۷

۴-فصل ۴ : طراحی برد..... ۶۹

۴-۱ مقدمه..... ۷۰

۴-۲ فاز ۱ : طراحی برد اصلی..... ۷۰

۴-۳ فاز دوم : اتصال قطعات جانبی..... ۷۲

۴-۴ فاز سوم: راه اندازی قطعات جانبی..... ۷۲

۴-۵ فاز چهارم : برقراری ارتباط با نرم افزار و تست برد..... ۷۴

۵-فصل ۵ : پیشنهادات..... ۷۶

فصل ۱

۱- ساختار کلی FPLD

۱- ۱ مقدمه ای بر تراشه FPGA :

در این بخش سعی در بررسی اجمالی ساختار IC قابل برنامه ریزی ظرفیت بالای FPLD موجود در بازار می باشد در ابتدا در مورد IC های FPLD صحبت خواهیم کرد . سه اصطلاح مشتق شده از کلمه FPLD عبارتند از SPLD,CPLD , FPGA (مجموعه ارایه ای از گیت های قابل برنامه ریزی).

بعد از ان هر کدام از قطعات را به طور تفصیلی بررسی خواهیم کرد.بلا فاصله پس از ارایه و توسعه قطعات جدید وپیشرفته FPLD یا مجموعه المان های قابل برنامه ریزی ، پروسه طراحی مدارات سخت افزاری دیجیتال ، دچار تغییر غیر باوری گردید.

بر خلاف تکنولوژی گذشته که در موارد طراحی شده از مجموعه قطعات گیتی جدا از هم (مانند TTL و CMOS) استفاده می شد ، در حال حاضر محصولات دیجیتالی دارای قطعات بسیار پر ظرفیت می باشند . این فرایند به کار گرفته شده ، تنها شامل قطعاتی نظیر میکرو پروسسور ها و حافظه نیست ، بلکه به تبع ان در مورد استفاده از مدارات منطقی نظیر دیاگرام حالت ، شمارنده ها ، رجیستر و دیکودر نیز می باشد.

از طرفی استفاده از قطعات منطقی طراحی شده نیز برای موارد خاص (ASIC) بسیار گران تمام می شود وعلاوه بر زمان بسیار زیادی که برای طراحی این گونه قطعات ، تا رسیدن به محصول مصرف می شود ، برای تولید انبوه به هیچ وجه مقرون به صرفه نمی باشد .

به همین دلیل بسیاری از مدارات طراحی شده به صورت نمونه از این تکنولوژی بهره می برند ضمن اینکه محصولات تولید انبوه نیز در حال حاضر به وسیله FPLD طراحی می شوند . مزیت های اصلی در مجبور نمودن طراح برای استفاده از FPLD عبارتند از: شتاب برای رسیدن به چرخه تولید ، هزینه بسیار پایین شروع و آزمایش ، ریسک کم مالی و سهولت تغییر محصول نهایی پس از تحویل به مشتری .

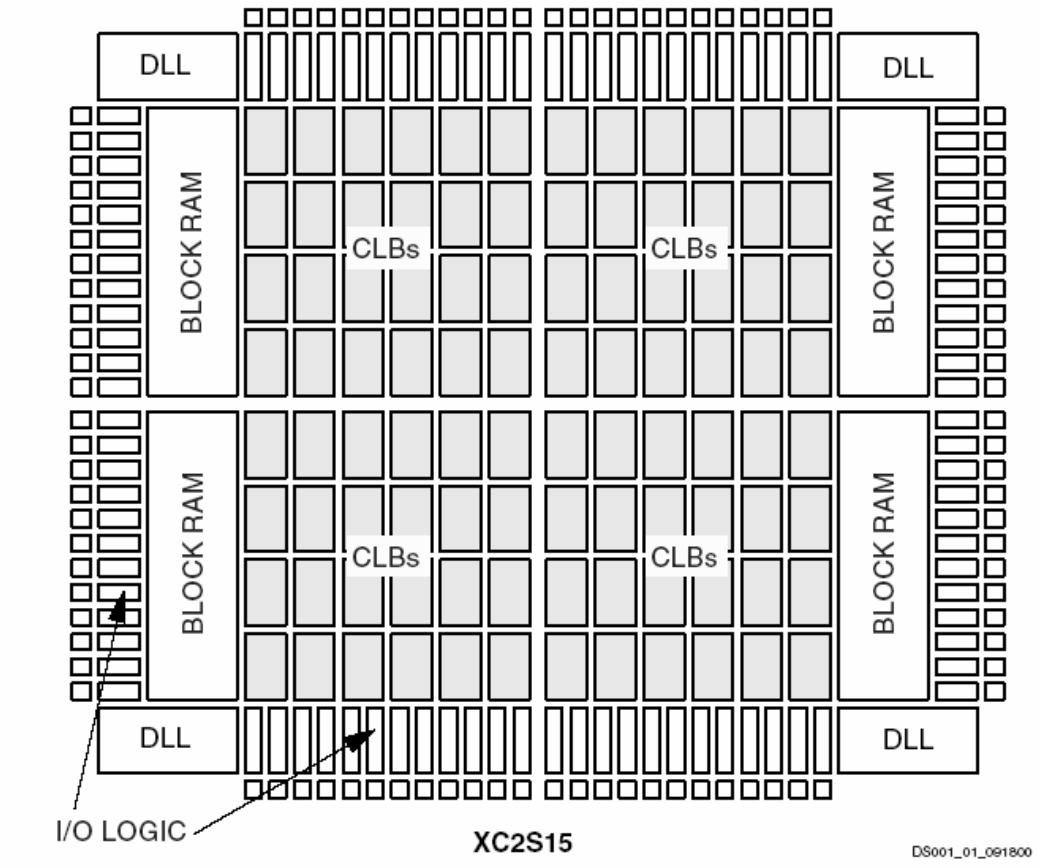
دانستن این که بهترین قطعات برای طراحی کدامند و انتخاب یک کارخانه تولید قطعات قابل برنامه ریزی و همچنین فراگیری استفاده از نرم افزار های پیچیده مختلف ، شروط اولیه یک طراحی جامع و مناسب می باشد .

هدف از این بخش شرحی در مورد ساختمان داخلی انواع مختلف FPLD ها است و تاکید انجام گرفته بر روی قطعات پر ظرفیت منطقی است چرا که زمینه ساز بحث بر روی مهم ترین قطعات الکترونیکی خواهد بود.

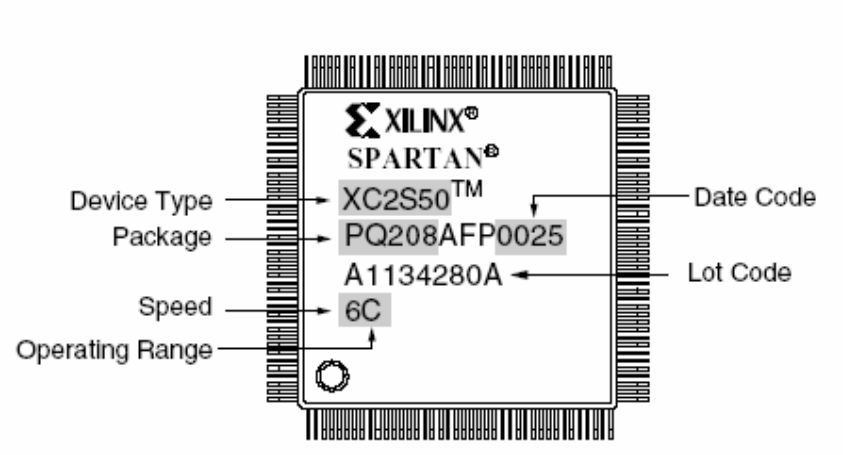
مزیت اصلی ان نسبت به ASIC ، قابل تغییر بودن الگوریتم و مدار داخلی این تراشه میباشد.

۱-۲ ساختار IC های قابل برنامه ریزی FPGA :

FPGA به معنی مجموعه ای از ارایه ای (ماتریسی) از گیت های قابل برنامه ریزی است . شکل زیر ساختار SPARTAN II که یک خانواده از FPGA می باشد را تشان می دهد.



شکل ۱: بلوک دیاگرام یک FPGA



شکل ۲

FPGA از سه جزء تشکیل شده است:

۱) بلوک قابل برنامه ریزی به صورت ماتریس n*n که در ان کلیه اعضای این ماتریس شبیه به هم بـوده وبه نام CLB (Combinational Logic block) معروف هستند.

مدار داخلی CLB برحسب کاربرد می تواند برنامه ریزی شود تا با استفاده از CLB های مجاور و ترکیـب با آنها ، المان هایی مانند : شمارنده ، جمع کننده ، تفریق کننده ، اکومولاتور ، رجیستر ، شیفت رجیستر ، دیکودر ، مقایسه کننده و RAM را بوجود اورد.

۲) سیم ها وبلوک های ارتباطی (Programmable Interconnect) بین CLB ها هستند که بـرای ترکیب CLB های متعدد و اتصال خروجی ها و ورودی ها ی آنها به یکدیگر بکار میروند.

فراوانی این اتصالات چنان زیاد است که امکان تحقق هر ترکیب پیچیده را فراهم می آورند.

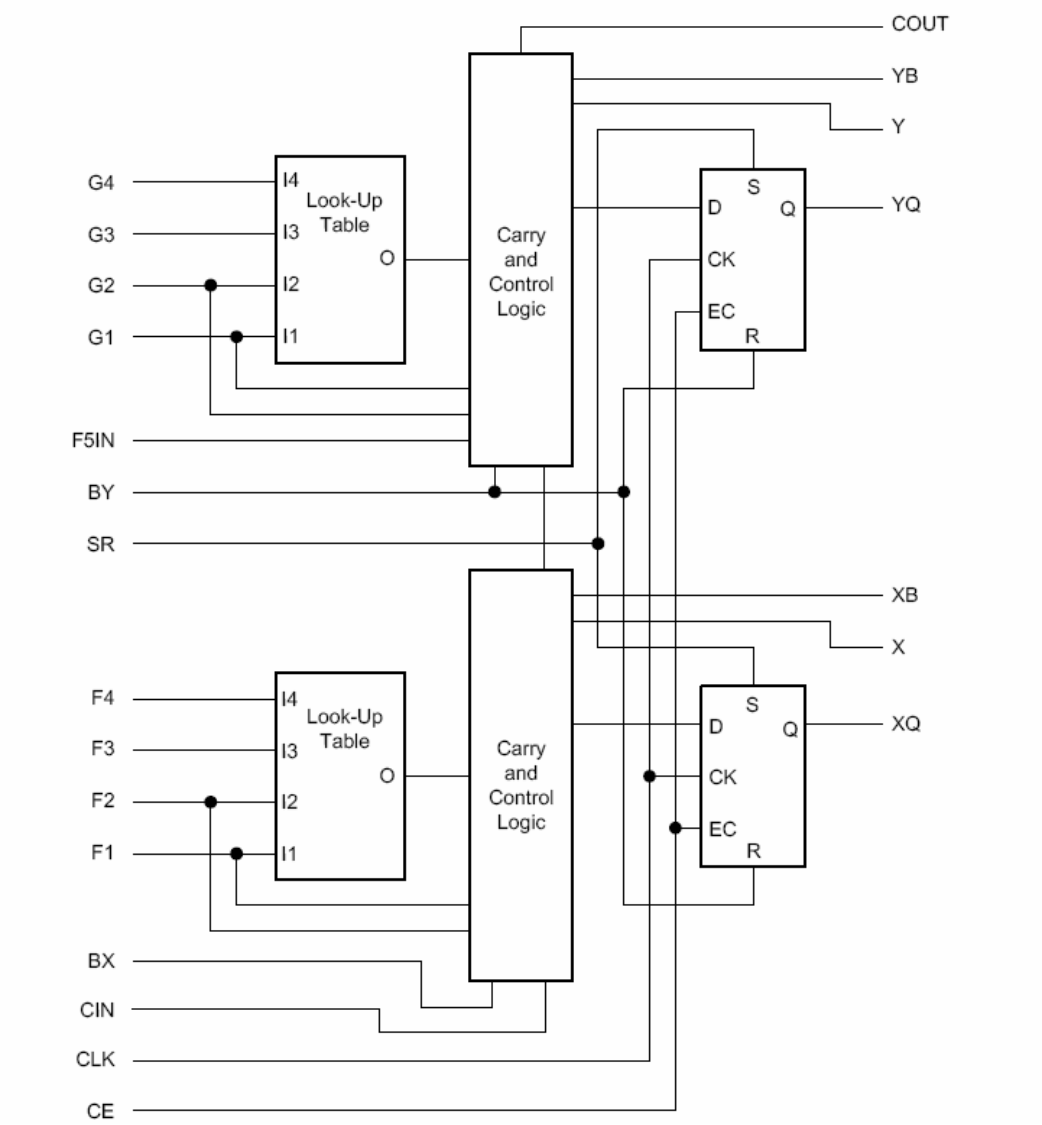
۳) پایه های I/O یا I/O block که بوسیله این بلوک ها ورود و خروج سیگنال دیجیتالی به داخل و یا از FPGA امکان پذیر شود.

هر پایه I/O به یک IOB متصل است که خود دارای اجزای مختلفی نظیر رجیستر بـرای ورود ، رجیـستر برای خروج ، مقاومت Pull-down , Pull-up بافر سه حالته برای خروجی ، کنترل Slow Rate برای کاهش نویز و یا کاهش تاخیر خروجی ، ایجاد تاخیر تصادفی برای سیگنال ورودی می باشد . میزان جریـان دهی پایه های I/O در حالت اتصال کوتاه GND و VCC حدود ۲۰۰ میلی آمپر می باشد که قابل توجـه است.

حال هر کدام از موارد بالا به طور مفصل توضیح داده خواهد شد:

۱-۲-۱ بلوک های قابل برنامه ریزی CLB

اجزای داخلی CLB در شکل امده است. تعداد CLB های FPGA بر حسب ظرفیت ان می باشد . مثلاً کوچکترین FPGA شرکت Xilinx به نام xcs05 (۵۰۰۰ گیتی) دارای ۱۰۰ عدد CLB بوده و بزرگترین آن به نام xcv3200 (چهار میلیون گیتی)دارای ۳۲۰۳۲ عدد از این CLB ها می باشد.



شکل ۳:CLB

۱-۲-۱-۱ بلوک های ترکیبی (Combinational) منطقی:

سه بلوک منطقی در CLB وجود دارد F, G , H .

دو بلوک F, G توابع ۴ به ۱ هستند که می توانند هر ترکیب منطقی ۴ به ۱ را ایجاد نمایند.

ورودی های این بلوک ها همان ورودی های CLB می باشند .بلوک H نیز می تواند هر ترکیب ۳ به ۱ را تولید کند . ورودی های H ، از خروجی F و G و یک ورودی اراد دیگر به نام HI تهیه می شود که با

استفاده از این سه بلوک می توانند هر ترکیب ۵ به ۱ یا برخی از ترکیب های ۹ به ۱ منطقی را بوجود آورد . بسته به نیاز توابع F و G می توانند به طور مستقل و جدای از هم استفاده شوند وخروجی آنها می توانند به فلیپ فلاپ رفته یا خود خروجی CLB باشند.

ساختمان داخلی این بلوک ها یک look up table می باشند که برای F و G ، ۱۶ بیت حافظه و برای H ، ۸ بیت حافظه دارد.

ورودی این بلوک ها یک خانه از حافظه را آدرس دهی می کند که مقدار آن در خروجی ظاهر می شود ، مقادیر موجود در خانه های این حافظه به وسیله برنامه بار شده در FPGA و بسته به تابع مورد نیاز برای آن بلوک تعیین می شود.

۱-۲-۱-۲ مالتی پلکسر :

این بلوک ها در شکل CLB به صورت ذوذنقه نشان داده شده اند به وسیله برنامه بار شده در FPGA برنامه ریزی می شوند. در واقع بعد از برنامه ریزی شدن FPGA ، تنها یکی از این ورودی ها به خروجی متصل می شود . این بلوک ها انتخاب مسیر حرکت سیگنال های مختلف منطقی را بر عهده دارند.

۱-۲-۱-۳ فلیپ فلاپ ها:

در شکل CLB دو عدد فلیپ فلاپ دیده می شود که امکان ایجاد توابع ترکیبی را فراهم می کنند. هر کدام از این فلیپ فلاپ ها دارای ورودی D(Data) و EC(Enable clock) هستند که بعد از برنامه ریزی شدن FPGA تنها یکی از ورودی ها reset, set سنکرون آنها فعال می شود. ورودی D فلیپ فلاپ ها می تواند از هر کدام از بلوک های F و G و H و یا دیگر ورودی های CLB تهیه شود . خروجی آن نیز

به عنوان خروجی CLB استفاده خواهد شد. لبه clock فلیپ فلاپ به صورت بالا رونده می تواند انتخاب شود.

نکته: خواص فلیپ فلاپ های سری مختلف FPGA شرکت xilinx با یکدیگر متفاوت است :

فلیپ فلاپ های سری Spartan (XCS) که پنج ولتی هستند ، تنها با لبه فعال می شوند.

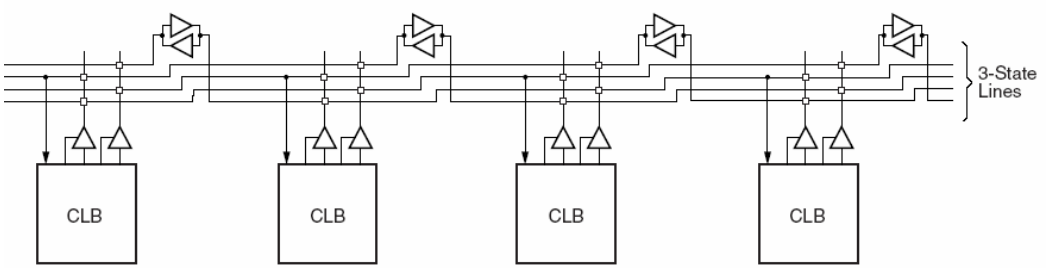
فلیپ فلاپ های سری (XCS..XL) با تغذیه ۳,۳ ولت و Spartan II (..XCS25) با تغذیه ۲,۵ ولت هم با لبه بالا و هم با سطح (Latch) می توانند فعال شوند که بوسیله نرم افزار طراحی با FPGA مشخص می شود . همچنین فلیپ فلاپ های Spartan II قابلیت Reset و Set شدن سنکرون را دارا هستند .

یک CLB می تواند به صورت RAM و ROM نیز مورد استفاده قرار گیرد که بعداً توضیح داده خواهد شد . همچنین هر CLB یک واحد به نام Carry Logic دارد که می تواند Carry مربوط به انواع توابع مختلف را به طور جداگانه تولید نماید.

مثلاً Carry مربوط به یک Adder ، هم می تواند به وسیله توابع F و G و H محاسبه شود وهم می تواند به وسیله واحد Carry فراهم شود.

۱-۲-۲ سیم ها و بلوک های ارتبا طی (Programable InterConnect) :

این قسمت برای اتصال CLB و IOB ها به یکدیگر استفاده می شود . تمامی این اتصالات از نوع سیم های فلزی متصل به سوئیچ های تکی یا سوئیچ های جعبه ای ماتریسی هستند .فراوانی این اتصالات به حدی است که تحقق هر گونه اتصال را فراهم می اورد



شکل ۴ اتصالات بین CLB

سیم کشی اتصالات اطراف CLB سه نوع است :

۱) Single-length

۲) Double-length

۳) Long-lines

همچنین بلوک های PSM(Programmable Switch Matrix) برای ایجاد اتصالات بین CLB ها بکار می رود .

تنها یکی از شش FET شکل فعال است . که برای اتصال سیم های Single و Double به کار می رود .

سری های مختلف FPGA دارای ساختمان مختلفی از لحاظ سیم های اتصال هستند.

IOB ها نیز دارای اتصال مخصوصی هستند که دور تا دور IC کشیده شده است تا بتواند هر گونه عمل اختصاص پایه های مختلف IC را به هر سیگنال داخلی امکان پذیر نماید .

سیم های کشیده شده اطراف FPGA معروف به Versaring هستند که عمل انتخاب پایه های FPGA و ساده کردن فیبر مدار چاپی را امکان پذیر میسازد .

علاوه بر اتصال بالا ، سیم ها و بافر های خاصی بـه نـام Global Nets and buffers وجـود دارد ، کـه بیشتر برای شبکه های Clock به کار می روند . هنگام طراحی مدار FPGA در هر نرم افزاری دقت شود

که حتماً برای Clock ، Set و Reset از این بافر ها و سیم های اتصال استفاده می شود . این قسمت در آینده در قسمت توضیح نرم افزار طراحی با FPGA توضیح داده خواهد شد.

۱-۲-۳ بلوک های ورودی – خروجی یا IOB :

بلوک های قابل برنامه ریزی ورودی و خروجی هستند که عمل ارتباط بین پایه های IC و مدار داخل آن را بر عهده دارند . هرIOB یک پایه IC را کنترل می کند که می تواند به صورت ورودی ، خروجی یا دو طرفه باشد.

۱-۳ ساختار IC های قابل برنامه ریزی CPLD :

در این بخش به معرفی ساختار داخلی CPLD می پردازیم

CPLD با FPGA دارای دو تفاوت عوده می باشد:

۱) حجم گیت

۲) نوع برنامه ریزی ونگهداری برنامه

۱-۳-۱ حجم گیت:

حجم FPGA ها از ۵۰۰۰ تا ۱۰ میلیون گیت است اما حجم CPLD از ۸۰۰ تا ۱۲۸۰۰ گیت می باشد. ساختار CPLD و ظرفیت گیتی آن ، این تراشه را برای کاربرد های ترکیبی مناسب ساخته است .در راقع مدارات AND ، OR ، ... که مدارات ترکیبی هستند در CPLD بیشتر از مدارات ترتیبی و حافظه دار مانند فلیپ فلاپ ، رجیستر و حافظه می باشد.

۱-۳-۲ نوع برنامه ریزی و نگهداری برنامه :

داخل CPLD ، به صورت Flash Ram بوده که توانایی برنامه ریزی شدن تا ۱۰۰۰۰ بار و نگهداری برنامه را برای مدت ۲۰ سال دارد ، که این در مقابل FPGA که بی نهایت بار می تواند برنامه ریزی شود و برنامه ان تنها تا زمانی وجود دارد که ولتاژ در مدار محفوظ می ماند، یک تفاوت عمده محسوب می شود.

ساختار داخلی CPLD شبیه به PAL است اما نسبت به آن پیچیدگی بسیار بیشتری دارد .

CPLD های Xilinx با شماره XC9500 و خانواده جدید تر از آن به نام Cool Runner که شامل دو مدل XCR3000(XPLA3) و XCR500(XPLA5) می باشند- معروف هستند.

خانواده XC9500 در رنج ۸۰۰ تا ۱۰۶۰۰ گیت ساخته می شوند که به طور مثال کوچکترین آن (XC9536) دارای ۳۶ بلوک قابل برنامه ریزی به نام Macrocell است که ۳۲ پایه I/O را در اختیار استفاده کننده قرار می دهد. در مورد خانواده Cool Runner لازم به توضیح است که این CPLD ها از نکنولوژی Fast Zero Power(FZP) استفاده می کنند که بسیار کم مصرف و سریع هستند.

۱-۴ توضیح قسمت های مختلف CPLD به طور خلاصه:

۱-۴-۱ JTAG Controller:

برای برنامه ریزی CPLD و تغییر کارکرد آن از استاندارد JTAG (IEEE1149.1) استفاده می شود .

در اینجا به وسیله چهار سیم و به صورت سریالی می توان بر حسب نیاز ، CPLD را برنامه ریزی نمود. هر CPLD تا ۱۰۰۰۰ بار قابل برنامه ریزی است و در هر بار می تواند تا ۲۰ سال برنامه خود را حفظ کند.

عمل برنامه ریزی به صورت ISP(In System Programming) صورت می گیرد و بدون نیاز به خارج نمودن IC از مدار می توان آن را برنامه ریزی کرد.

I/O ۲-۴-۱:

همان پایه های فلزی CPLD است که به چهار دسته I/O معمولی ، I/O/Clock(پایه عمومی برای Clock) ، GSR(پایه عمومی برای Set و Reset) ، GTS(پایه عمومی برای Tri state یا فعال نمودن بافر سه حالته) تقسیم می شود. البته هر کدام از پایه های عمومی در صورت نیاز می توانند مانند یک پایه I/O معمولی(ورودی یا خروجی یا هر دو)باشند.

خلاصه فصل:

ذر این بخش سعی شد تا اطلاعات جامع ولی خلاصه در مورد IC های قابل برنامه ریزی FPGA و CPLD و همچنین ساختار داخلی آنها داده شود ، گفتیم که این IC ها قابلیت گنجاندن اغلب طرح های ساده و پیچیده دیجیتال را در خود دارند. عمده تفاوت بین FPGA و CPLD در ساختار داخلی و نیز نوع ذخیره سازی طرح در داخل آنها و همچنین در ظرفیت آنها از نظر تعداد گیت می باشد.

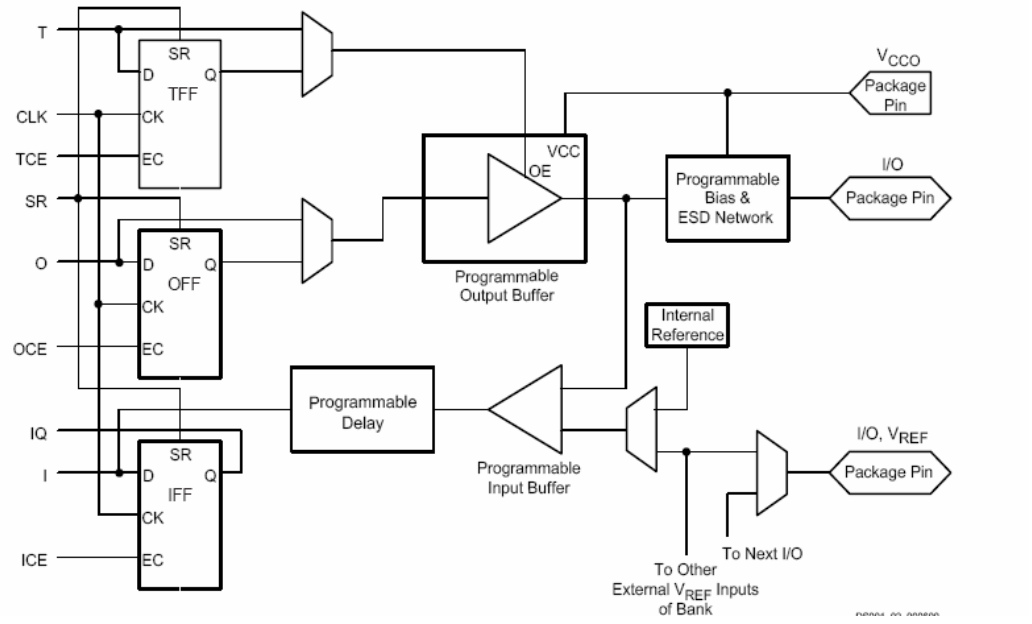
همچنین گفتیم که این IC ها انتخاب مناسبی برای تسریع در تولید محصول نهایی و نیز برای تغییر طرح پس از رسیدن محصول به دست مشتری یا به روز رسانی طرح با توجه به در خواست مشتری یا نیاز روز می باشد .

فصل ۲

۲- توصیف معماری Spartan II

۲-۱ بلوک های IO :

بلوک های $spar\tan\Pi, IO$ در شکل ۱ نشان داده شد ،خصوصیات ورودی در خروجی ها این است که سیگنالهای IO استاندارد متنوعی را پشتیبانی می کند .



شکل ۲-۱ : بلوک I/O

این ورودی های با سرعت بالا و خروجی ها توانایی پشتیبانی حافظه های جدید و پیشرفته متنوع و واسط های باس را دارا می باشند جدول ۱ لیست چندین استاندارد که با منابع مورد نیاز خروجی و ولتاژها نهایی مورد نیاز هر استاندارد آمده است . سیگنالهای کنترلی CE,CLK،برای ۳ ثبات با reset,set مشترک می باشد . برای هر ثبات ،این سیگنالها می تواند به طور مستقل پیکر بندی شوند که به صورت set سنکرون یا preset آسنگرون یا پاک کردن آسنکرون باشند .

چدول ۲-۱ : انواع I/O های استاندارد

I/O Standard	Input Reference Voltage (V_{REF})	Output Source Voltage (V_{CCO})	Board Termination Voltage (V_{TT})
LVTTL (2-24 mA)	N/A	3.3	N/A
LVCMOS2	N/A	2.5	N/A
PCI (3V/5V, 33 MHz/66 MHz)	N/A	3.3	N/A
GTL	0.8	N/A	1.2
GTL+	1.0	N/A	1.5
HSTL Class I	0.75	1.5	0.75
HSTL Class III	0.9	1.5	1.5
HSTL Class IV	0.9	1.5	1.5
SSTL3 Class I and II	1.5	3.3	1.5
SSTL2 Class I and II	1.25	2.5	1.25
CTT	1.5	3.3	1.5
AGP-2X	1.32	3.3	N/A

خصوصیات در شکل نشان داده نشده اند اما توسط نرم افزار کنترل می شوند .

با فرهای ورودی و خروجی و همه سیگنالهای کنترلی IOB به صورت مستقل کنترل می شوند .

مقاومت pull- up و pull-down اختیاری و مدار نگهدارنده ضعیف به هر قسمت وابسته می باشد (به عمل پایه و نحوه کاربرد آن)

در طول پیکره بندی خروجی ها درگیر پیکره بندی نمی باشند و به حالت امپدانس بالا می رند . مقاومت های pull- down و مدار نگه دارنده ضعیف غیر فعال می باشند و ورودی ها می توانند به صورت اختیاری pull up شوند .

فعال کردن مقاومت های pull – up در طول پیکره بندی توسط یک پایه اصلی توسط مد پایه های پیکره بندی می باشد .

اگر مقاومت های pull –up فعال نشوند همه پایه ها شناور خواهند شد در نتیجه مقاومتهای pull- up یا pull- down باید به روی پین ها فراهم شود تا در یک منطق مشخص در طول پیکر بندی باشند .

همه سطوح ارتباطی در مقابل شارژ الکترونیکی (ESD: Electronic dischrage) و در مقابل نوسان ولتاژ محافظت شده اند .

دو نوع از محافظت از سربار ولتاژ فراهم شده است ،یکی از اجازه به استفاده از ۵ ولت و دیگری این اجازه را نمی دهد . برای ولتاژ ۵ ولت ،یک ساختمان مشابه دیود زنر متصل به زمین می شود و وقتی که خروجی به حدود ۶,۵ ولت می شود روشن می شود . وقتی که ۵ ولت نیاز نمی باشد یک دیود به ولتاژ خروجی فراهم شده Vcco متصل می شود .

نوع حافظه از ولتاژ می تواند به طور مستقل برای هر سطح ارتباطی (بانک) به طور مستقل انتخاب شود .

۲-۱-۱ مسیر ورودی :

در $spar\tan\Pi$ مسیر یابی از مسیر ورودی سیگنالهای ورودی به طور مستقیم و یا از طریق فلیپ فلاپ به صورت اختیاری به منطق داخلی صورت می گیرد .

هر بافرورودی به سیگنالهای استاندارد پیتیبانی شده ولتاژ پائین اختصاص می یابد . در بعضی از این استاندارد ها بافر ورودی برای ولتاژ آستانه فراهم شده توسط کاربر اختصاص می یابد که VREF می نامند .

در هر ورودی مقاومتهای pull-up وpull- down اختیاری برای استفاده بعد از پیکره بندی می باشد .

۲-۱-۲ مسیر خروجی :

مسیر خروجی شامل بافر خروجی 3-state که سیگنال خروجی را به پایه منتقل می کند می باشد . سیگنال خروجی می تواند به بافر به طور مستقیم از طریق منطق داخلی یا از طریق IOB فلیپ فلاپ باشد.

کنترل 3-state خروجی می تواند مسیر مستقیمی از منطق داخلی یا از طریق فلیپ فلاپ که فعال یا غیر فعال کردن سنکرون را فراهم می کند باشد .

هر راه انداز خروجی می تواند به طور مستقل برای رنج وسیعی از سیگنالهای استاندارد ولتاژ پائین برنا مه ریزی شود. هر خروجی می تواند تا ۲۴mA را تولید کند .

در بعضی از سیگنالهای استاندارد خروجی ولتاژ بالا ، وابسته به فراهم کردن خارجی ولتاژ VCCO می باشد.

یک مدار نگهدارنده ضعیف به هر خروجی متصل می باشد . وقتی که انتخاب می شود مدار بر روی ولتاژ روی پایه نظارت کرده و به صورت ضعیف آن را بر روی low یا high برای تشخیص سیگنال ورودی نگه می دارد ،اگر چند مبنع سیگنال به یک پایه متصل باشند نگه دارنده ضعیف آنرا در آخرین وضعیت آن نگه میدارد .

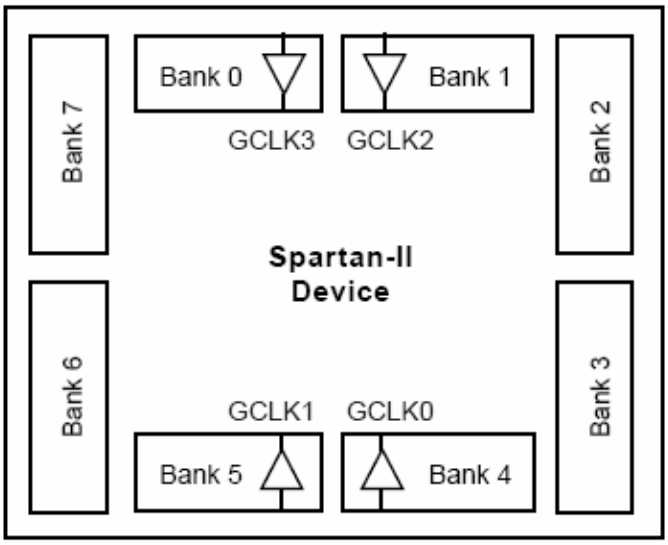
بدلیل اینکه نگهدارنده ضعیف از بافر ورودی IOB را برای نشان دادن نوع ورودی استفاده می کند،VREF باید با توجه به سیگنال استاندارد مورد استفاده فراهم شود .

مقدار VREF باید با قواعد بانک I/O مطابقت داشته باشد .

۲-۱-۳ بانک های I/O

بعضی از I/O های استاندارد که در بالا توضیح داده شد به ولتاژ VCCO و یا VREF نیاز دارد . این ولتاژ ها به صورت خارجی به پین های FPGA متصل می شود . که تغذیه یک گروه از IOB ها را بر عهده دارند که بانک نامیده می شوند .در نتیجه این محدودیت بوجود می آید که یک I / O استاندارد در یک بانک می تواند استفاده شود .

هشت بانک I/O از جداسازی هر یک از لبه های FPGA به دو بانک به وجود می آید شکل زیر



شکل ۲-۲ بانک های I/O

هر بانک دارای چندین VCCO می باشد که باید به یک ولتاژ یکسان متصل شوند . ولتاژ VCCO با توجه به استاندارد مورد استفاده مشخص می شود .

بعضی از ولتاژهای استاندارد ورودی نیاز به ولتاژ آستانه VREF دارند در این موارد پایه های I/O به صورت اتوماتیک به عنوان ورودی ولتاژ VREF می باشند .

پایه های VREF در داخل یک بانک به صورت داخلی ارتباط دارند و در نتیجه فقط یک VREF می تواند در هر بانک بکار برده شود ولی همه ولتاژ های VREF باید برای عملکرد صحیح به یک ولتاژ خارجی متصل شوند .

بافرهای ورودی که از VREFاستفاده می کند از ۵ ولت پشتیبانی نمی کنند . فقط استانداردهای PCI,LVCMOS2,LVTTL دارای ورودی ۵ ولت می باشند .

۲-۲ بلوک های منطقی قابل برنامه ریزی :

ساختار اصلی بلوک های CLB سلولهای منطق (LC) می باشد .

LC شامل چهار سیگنال ورودی ،سیگنال حامل (Carry) و عناصر ذخیره سازی باشد . خروجی سیگنال ورودی در هر LC خروجی CLB و ورودی فلیپ فلاپ D راه اندازی می کند .

هر CLB ، $spar\tan\Pi$ شامل چهار LC ،که به یک تکه مشابه سازماندهی شده است می باشد

۲-۲-۱ جدول جستجو :

سیگنالهای ورودی $spar\tan\Pi$ به عنوان چهار ورودی جدول جستجو (LUT) تکامل می یابند . به منظور عمل سیگنالهای ورودی ،هر LUT می تواند یک RAM سنکرون 1*16 بیتی را فراهم کند . از این گذشته ،دو LUT در داخل یک تکه می تواند ترکیب شوند تا یک RAM سنکرون 2*16 بیتی یا 1*32 را تولید کنند .

۲-۲-۲ عناصر ذخیره ساز :

عناصر ذخیره ساز در $spar\tan\Pi$ به یک شکل D فلیپ فلاپ حساس به لبه یا لچ حساس به سطح می باشد .

ورودی های D فلیپ فلاپ می تواند از طریق تابع مولد یا به طور مستقیم از طریق ورودی می باشد .

سیگنالهای Clock Enable , clock ،هر بخش دارای سیگنالهای RESET,SET سنکرون می باشد (SR,BY)

SR ،عنصر ذخیره ساز را مجبور به مقدار دهی حالت خاصی طول پیکره بندی می کند . BY عنصر ذخیره ساز را مجبور به حالت معکوس می کند . به طور متناوب این سیگنال ها ممکن است برای عمل کـردن بـه صورت آسنکرون طراحی شوند .

۲-۲-۳ دیگر منطق ها

مالتی پلکسر F5 در هر تکه خروجی توابع مولد را ترکیب می کند .

این ترکیب ،یک تابع مولد که می تواند هر ترکیب ۵ ورودی را تولید فراهم می کند .

به طور مشابه مالتی پلکسرF6 خروجی ها را برای همه توابع مولد در CLB با انتخـاب یکـی از خروجـی های مالتی پلکسر F5 فراهم می کند .

۲-۲-۴ منطق های ریاضی:

رقم نقلی واقع شده ،یک رقم نقلی ریاضی تولید می کند که توانایی کار با توابع ریاضی با سرعت بـالا را دار می باشد. CLB، $spar\tan\Pi$ از دو زنجیر رقم نقلی جداگانه ،در هر تکه پشتیبانی می کند .

اندازه زنجیر نقلی برای هر CLB ،۲ بیت می باشد .

منطق ریاضی شامل گیت های XOR که اجازه ایجاد تمام جمع کننده ۱ بیتی بـرای تکامـل یـک LC را می دهد .

وجود گیت های ANDبازده مالتی پلکسر را بهبود می دهند .

مسیر رقم نقلی واقع شده می تواند در توانع مولد متوالی برای تکامل پهنای توابع منطقی بکار رود .

۲-۲-۵ بافرها :

هر CLBو $spar\tan\Pi$ شامل ۲، 3-state می باشد .

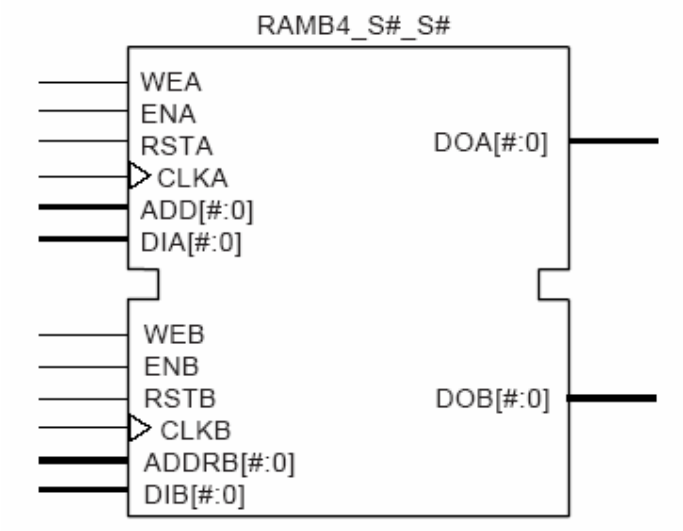
۲-۳ بلوک حافظه (RAM)

FPGA ، spartanII شامل چندین بلوک بزرگ حافظه Ram می باشد .

بلوک های حافظه RAM در ستون هایی سازماندهی شده اند . همه خانواده *spar* tan Π شامل ۲ تـا از چنین ستون هایی می باشند . یکی در امتداد لبه قائم می باشد . ایـن سـتون هـا در تمـام ارتفـاع حواشـه توسعه یافته اند . هر بلوک سلول RAM همانطور که در شکل 4 هم نشان داده شده است .

دارای پورت های دوتایی 4096 بیتی که با سیگنال کنترلی مستقل برای هر پورت می باشد .

اندازه داده دو پورت به طور مستقل می تواند مشخص شود .



شکل۲-۳ بلوک دیاگرام حافظه

بلوک های RAM $spar\tan\Pi$ شامل مسیر یابی اختصاصی برای فراهم کردن واسط مناسب با CLB ها و دیگر بلوک های RAM می باشد .

۲-۴ ماتریس مسیر یابی قابل برنامه ریزی

مسیر یابی بیشترین تاخیر مسیر را دارا می باشد که سرعت طراحی را محدود می کند .

در نتیجه جهت مسیر یابی معماری در $spar\tan\Pi$ نرم افزارهایی برای بهینه سازی طراحی شده انـد کـه این بهینه سازی تأخیر مسیرهای طولانی را کم می کند .

بهینه سازی زمان کامپایل را نیز کاهش میدهد .

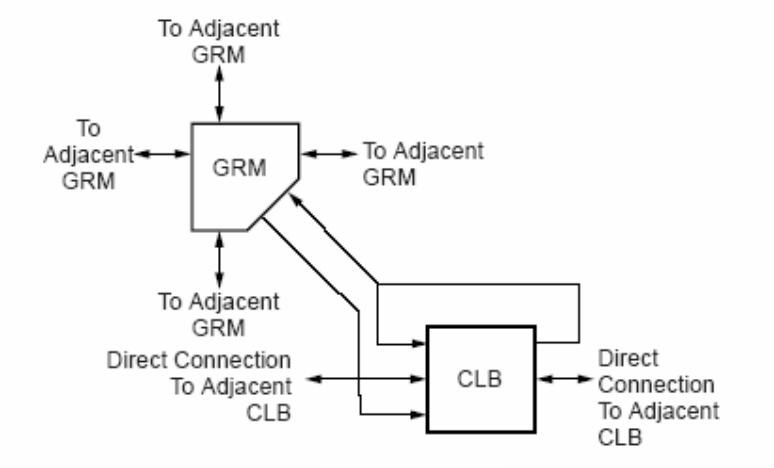
۲-۴-۱ مسیر یابی محلی :

منابع مسیر یابی محلی در شکل ۵ نشان داده شده است که سه نوع از ارتباطات زیر را برقرار می کند .

۱- ارتباط داخلی میان LUT ،فلیپ فلاپها و ماتریس مسیریابی اصلی (GRM)

۲- مسیر فیدبک داخلی CLB که ارتباط پر سرعتی را برای جابجایی دو LUT داخل یک CLB با کـم ترین أخیر ایجاد می کند .

۳- مسیر مستقیم که ارتباط پر سرعتی را میان CLB های همجوار ایجاد می کند .



شکل۲-۴

۲-۴-۲ مسیر یابی اصلی :

منابع مسیر یابی اصلی کلاک توزیع شده و دیگر سیگنالهای با فن اوت بالا می باشند .

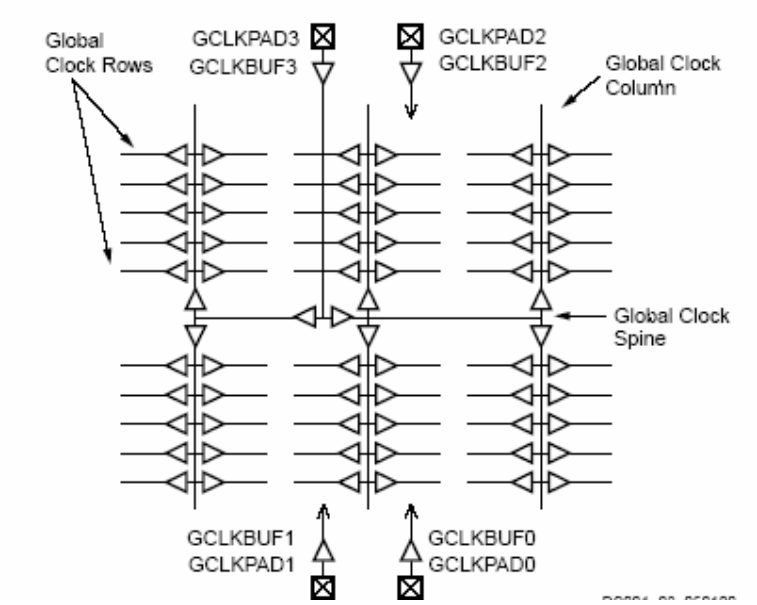
$spar\tan\Pi$ شامل ۲ ردیف از منابع مسیر یابی اصلی می باشد که به دو نوع مسیر یابی اولیه و ثانویه سیگنال کلاک با فن اوت بالا طراحی شده اند .

هر شبکه کلاک اصلی می تواند همه IOB,CLB ،پایه های کلاک و بلوک های RAM را راه اندازی کند . برای هر شبکه اصلی ،چهار بافر اصلی وجود دارد . منابع مسیر یابی ثانویه متشکل از ۲۴ خط می باشند ،۱۲ تا در عرض رویی تراشه و ۱۲ تا در سراسر عرض زیری تراشه می باشد .

۲-۵ توزیع کلاک :

خانواده $spar\tan\Pi$ دارای سرعت بالا ،کلاک مورب توزیع شده از طریق منابع مسیر یابی کلی اولیه می باشند که در زیر توضیح داده می شود .

یک نوع از کلاک توزیع شده در شکل ۶ نشان داده شده است .



شکل۲-۵ کلاک توزیع شده

چهار بافر کلی فراهم شده است که ۲ تا در بالای مرکز تراشه و ۲ تا در پائین تراشه می باشند . اینها ،چهار شبکه کلی اولیه را راه اندازی می کنند که باعث راه اندازی پایه های کلاک می شود .

چهار پایه کلاک وجود دارد ،که هر کدام نزدیک بافر کلی می باشند .

ورودی بافر اصلی از طریق هر یک از پایه ها یا از سیگنال ها در مسیر یابی همه منظوره انتخاب می شود .

حلقه تأخیر و نگهداری (DLL) ،۲ تا از شبکه های سراسری کلاک را می تواند راه اندازی کند .

DLL ناظر کلاک ورودی و کلاک توزیع شده می باشد و به طور خودکار تأخیر کلاک را تنظیم می کند .

تأخیرات اضافی دیگری نیز وجود دارند مثلاً که لبه کلاک به یک فلیپ فلاپ داخلی می رسد دقیقاً یک کلاک بعد به ورودی ها می رسند .

این سیستم حلقه بسته به طور مؤثر تأخیرات توزیع کلاک را حذف می کنـد . بـه منظـور حـذف تـاخیرات توزیع کلاک ،DLL یک کنترل پیشرفته از چندین کلاک وابـسته را فـراهم میکنـد . DLL چهـار فـاز از کلاک منبع را فراهم می کند که می تواند کلاک را ۲ برابر کند یا کلاک را بر ۱/۵ ،۲ ،۲/۵ ،۳،۴،۵،۸یـا ۱۶ تقسیم کند که دارای ۶ خروجی می باشد .

به منظور گارانتی کردن عملکرد صحیح کلاک سیستم از زمان راه انـدازی FPGA بعـد از پیکـره بنـدی ، DLL می تواند فرایند اجرای پیکره بندی فرایند ها را به تأخیر بیاندازد تا زمانی که در حالـت غیـر فعـال قرار گیرد .

۲-۶ توسعه سیستم :

متدهای پایه برای طراحی در *spar* tan Π شامل ۳ مرحله پیوسته می باشد .

۱- وارد کردن طراحی

۲- تکامل

۳- بررسی

در قسمت توضیح نرم افزار توضیح داده می شوند .

۲-۷ پیکره بندی FPGA :

پیکره بندی فرایندی است که در آن بیت هایی که توسط نرم افزار xilinx تولید شـده بـه داخـل حافظـه FPGA بار می شود .

spar tan Πاز پیکره سریال به صورت master/slave و JTAG مد پشتیبانی می کند .

۲-۷-۱ فایل پیکره بندی :

spar tan Π به وسیله بارکردن متوالی فریم های داده پیکره بندی می شود .

جدول ۲ مقدار فضای موجود در وسیله های موجود *spar* tan Π را نشان می دهد

جدول ۲-۲

Device	Configuration File Size (Bits)
XC2S15	197,696
XC2S30	336,768
XC2S50	559,200
XC2S100	781,216
XC2S150	1,040,096
XC2S200	1,335,840

۲-۷-۲ مدل های پیکره بندی :

spar tan Π چهار مد پیکره بندی را پشتیبانی می کند

- Slave serial mode
- Master serial mode
- Slave Parallel mode
- (JTAG) Boundary –scan mode

ما از این چهار مد دو مد JTAG , Slave serial mode را توضیح می دهیم .

پایه های تعیین کننده مد پیکره بندی M2,M1,M0 می باشند که ما از طریق این ۳ پایه مدی را که بـا آن FPGA برنامه ریزی می شود را مشخص میکنیم .

جدول ۳ مدل های برنامه ریزی را نشان می دهد .

جدول ۲-۳ مدل های برنامه ریزی

Configuration Mode	Preconfiguration Pull-ups	M0	M1	M2	CCLK Direction	Data Width	Serial D_{OUT}
Master Serial mode	No	0	0	0	Out	1	Yes
	Yes	0	0	1			
Slave Parallel mode	Yes	0	1	0	In	8	No
	No	0	1	1			
Boundary-Scan mode	Yes	1	0	0	N/A	1	No
	No	1	0	1			
Slave Serial mode	Yes	1	1	0	In	1	Yes
	No	1	1	1			

پیکره بندی از طریق اسکن مرزی همیشه و بطور مستقل از مد انتخابی موجود می باشد . انتخاب مد اسکن مرزی ،مدهای دیگر را غیر فعال می سازد . ۳ پایه مد (M2,M1,M0) دارای مقاومتpull-up داخلی هستند .و به صورت پیش فرض high می باشند .

۲-۷-۲-۱ سیگنال ها :

دو نوع از پایه ها که برای پیکره بندی $spar\tan\Pi$ بکار می رود وجود دارد :

پایه های اختصاصی که عمل مشخصی را برای پیکره بندی انجام می دهند و دیگر پایه ها که برای اهداف کلی I/O فقط یکبار موقعی که برنامه کاربر شروع می شود استفاده می شود .

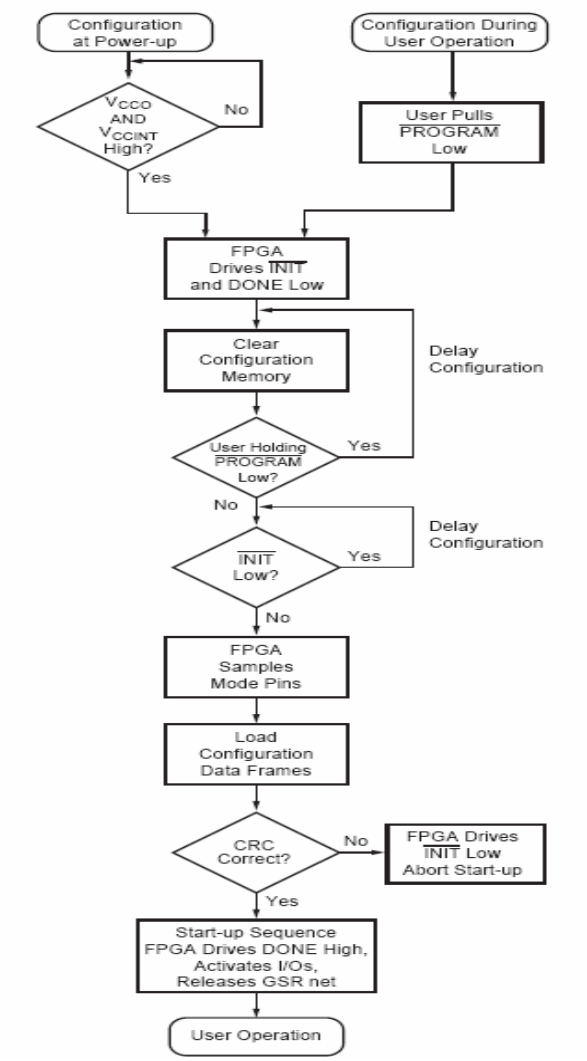
پین های اختصاصی شامل پین های مد M2,M1,M0 و پایه کلاک پیکره بندی (CCLK) ،پایه PRPGRAM ، پایه DONE و پایه های اسکن مرزی و (TDI,TDO,TMS,TCK) با توجه به مد

پیکره بندی انتخاب شده CCLKممکن است که خروجی تولید شده توسط FPGA باشد یا ممکن است که از خارج تولید شود و به عنوان ورودی FPGA به کار رود .

توجه کنید که بعضی از پایه های پیکره بندی به عنوان خروجی استفاده می شوند . برای عملکرد ،این پایه ها نیاز به VCCO, 3/3V برای راه اندازی سیگنال LVTTL دارند که پایه های VCCO بانک ۲ باید به ۳/۳ متصل شود .

۲-۷-۳ فرایند پیکره بندی :

ترتیب مراحل لازم برای پیکره بندی $spar\,\tan\Pi$ در شکل ۷ نشان داده شده است .



شکل۲-۷ ترتیب مراحل لازم برای پیکره بندی

این مراحل را در ۴ فاز می توان خلاصه کرد :

- آغاز پیکره بندی
- پیکره بندی حافظه
- بار کردن فریم های داده
- شروع

فاز پیکره بندی حافظه و شروع برای همه مدهای پیکره بندی یکسان می باشد .مرحله بار کردن فریم های داده متفاوت می باشد .

۲-۷-۳-۱ آغاز پیکره بندی :

برای آغاز پیکره بندی با قرار دادن پایه PROGRAM به صورت low ،پیکره بندی آغاز می شود . FPGA فرایند پیکره بندی رابا قرار دادن پایه DONE به صورت LOW آغاز می کند و بعد وارد فاز پیکره بندی حافظه می شود (پاک کردن حافظه) قبل از آغاز پیکره بندی VCCINT باید ۲/۵V بوده و VCCO بانک ۲ بیشتر از ۱ V باشد .

۲-۷-۳-۲ پیکره بندی حافظه :

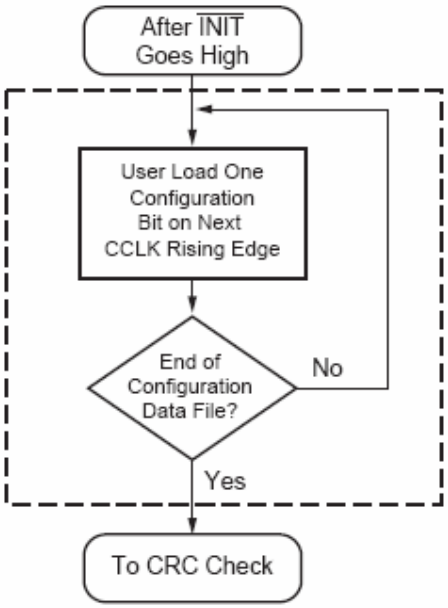
FPGA فاز پاک کردن حافظه را با قرار دادن $\overline{INIT}$ به صورت LOW نشان می دهد .

در این لحظه کاربر می تواند پیکره بندی را با نگه داشتن $\overline{PROGRAM}$ یا $\overline{INIT}$ به صورت LOW به تأخیر بیاندازد که باعث می شود FPGA در فاز پیکره بندی حافظه بماند .

توجه کنید که پایه $\overline{INIT}$ به صورت دو طرفه بوده و منطق سطح پائین در طول پیکره بندی حافظه می باشد بنابراین برای پیشگیری از تداخل از یک راه انداز $\text{open} - \text{drain}$ برای نگه داشتن پایه $\overline{INIT}$ به صورت LOW استفاده کنید .

۲-۷-۳-۳ بار کردن داده های پیکره بندی :

وقتی که پایه $\overline{INIT}$ در حالت high قرار گیرد ،کاربر می تواند بار کردن فریم ها را آغاز کند . ترتیب عملیات لازم برای بار کردن داده های پیکره بندی به روش serial mode در شکل ۸ نشان داده شده است .



شکل۲-۸ ترتیب عملیات در روش serial mode

۲-۷-۳-۳-۱ چک کردن خطاهای CRC :

در طول بار کردن داده ها ،مقدار CRC در داخل FPGA محاسبه می شود و اگر مقدار CRC درست نباشد FPGA پایه $\overline{INIT}$ را LOW کرده تا نشان دهد که در طول انتقال فریم ها خطا رخ داده و از پیکره بندی خارج می شود .

۲-۷-۳-۴ فاز شروع :

مقدار درست CRC نشان دهنده بار کردن موفق پیکره بندی داده ها می باشد . در طول شروع برنامه FPGA، چهار عمل زیر را انجام می دهد :

۱- تائید پایه DONE . اگر پایه Done ، High نشود ممکن است نشان دهنده این باشد که بار کردن غیر موفقی در طول پیکره بندی داده ها رخ داده است .

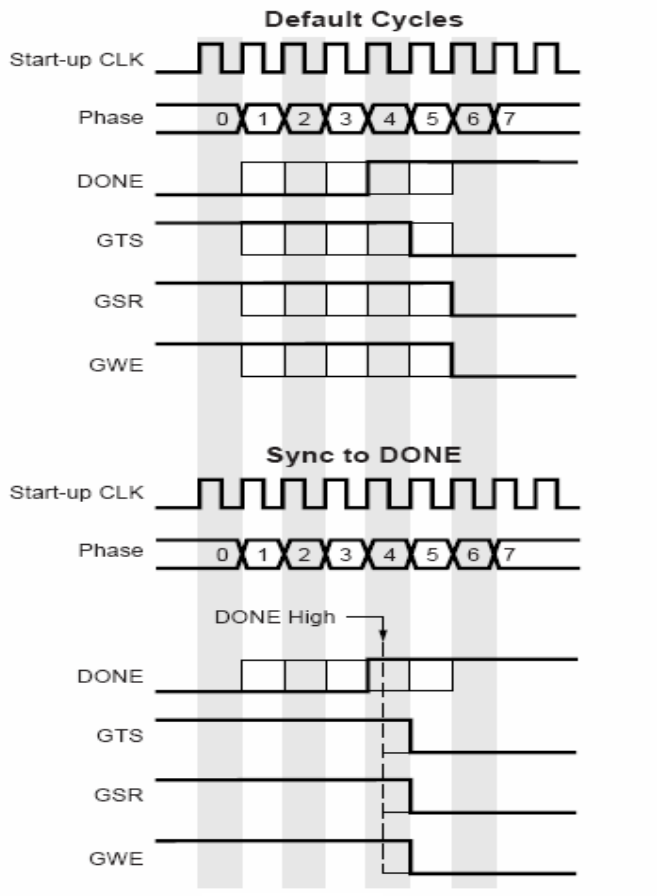
۲- آزاد سازی 3-state های سراسری (باعث می شود که همه I/O فعال شوند)

۳- خنثی کردن Reset,set سراسری (GSR) .(با این کار به همه فلیپ فلاپها اجازه می دهد که تغییر حالت دهند)

۴- فعال کردن نوشتن سراسری GWE (با این کار به RAM و فلیپ فلاپها اجازه تغییر حالت می دهد)

به صورت پیش فرض ،این عملیات سنکرون با CCLKهستند .

فاز شروع در ۸ کلاک انجام می شود که C0-C7 نامیده می شود .که در شکل ۹ نشان داده شده است .



شکل۲-۹

۲-۷-۴ مد سریال :

دو مد پیکره بندی سریال وجود دارد :

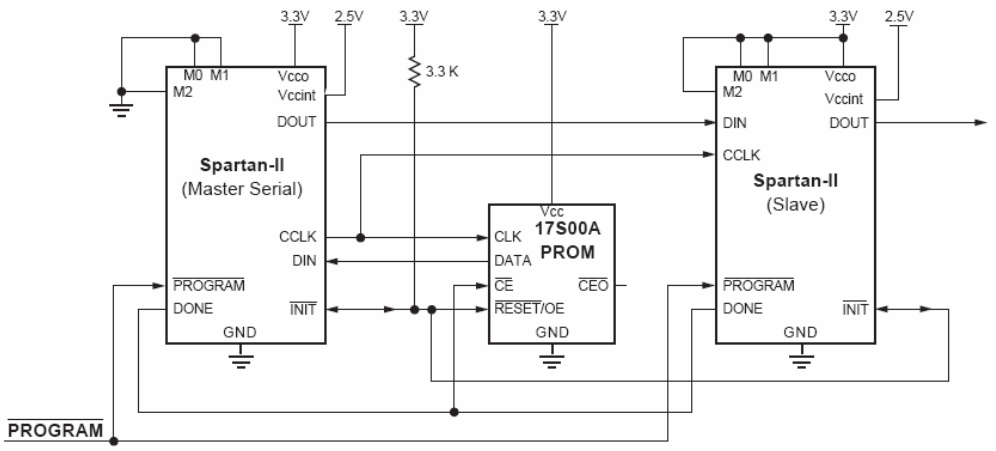
در مد Master serial , FPGA فرایند پیکره بندی را با راه اندازی CCLK به عنوان خروجی کنترل می کند .

در مد خارجی Slave serial , FPGA , CCLK را به صورت یک تابع از یک عمل خارجی دریافت می کند .که عامل خارجی فاز پیکره بندی را کنترل می کند .

در هر دو مد ، FPGA پیکره بندی را با بارکردن یک بیت در هر کلاک CCLK انجـام مـی دهـد . بیـت داده MSB برای بار شدن در پایه DIN نوشته می شود . شکل ۱۰ ترتیب بار کردن داده هـا بـه FPGA به صورت سریال نشان می دهد .

۱-۴-۷-۲ Serial slave mode :

در مد serial slave ،پایه CCLK توسط یک منبع خارجی راه اندازی می شود و به FPGA اجازه داده می شود که توسط یک وسیله منطقی دیگر پیکره بندی شود مانند ریز پردازنـده و... شـکل ۱۰ ارتبـاطی بـرای FPGA در مـد Master serial را کـه پیکـره بنـدی FPGA در مـدslave serial را از یـک PROM نشان می دهد .



شکل۲-۱۰

توضیحات لازم در قسمت فاز شروع گفته شده است .

۲-۷-۴-۲ مد اسکن مرزی : (JTAG)

در مد اسکن مرزی ،پایه غیر اختصاصی نیاز نمی باشد و پیکره بنـدی از طریـق TEST Accsess Port پیکره بندی از طریق TAP از دستورات ویژه CFG_INT استفاده می کند .

این دستورات به داده های ورودی بر روی TDI اجازه می دهد تا به بسته های داده برای پیکره بندی باس داخلی تبدیل شوند .

مراحل زیر برای پیکره بندی FPGA در مد اسکن مرزی نیاز می باشد :

۱- بار کردن دستورات CFG_IN به ثبات دستور اسکن مرزی (IR)

۲- وارد شدن به حالت شیفت داده (SDR)

۳- شیفت داده های پیکره بندی به داخل TDI

۴- قرار گرفتن در حالت Run-test-idle(RTI)

۵- بار کردن دستورات JSTART به IR

۶- بار کردن دستورات به حالت SDR

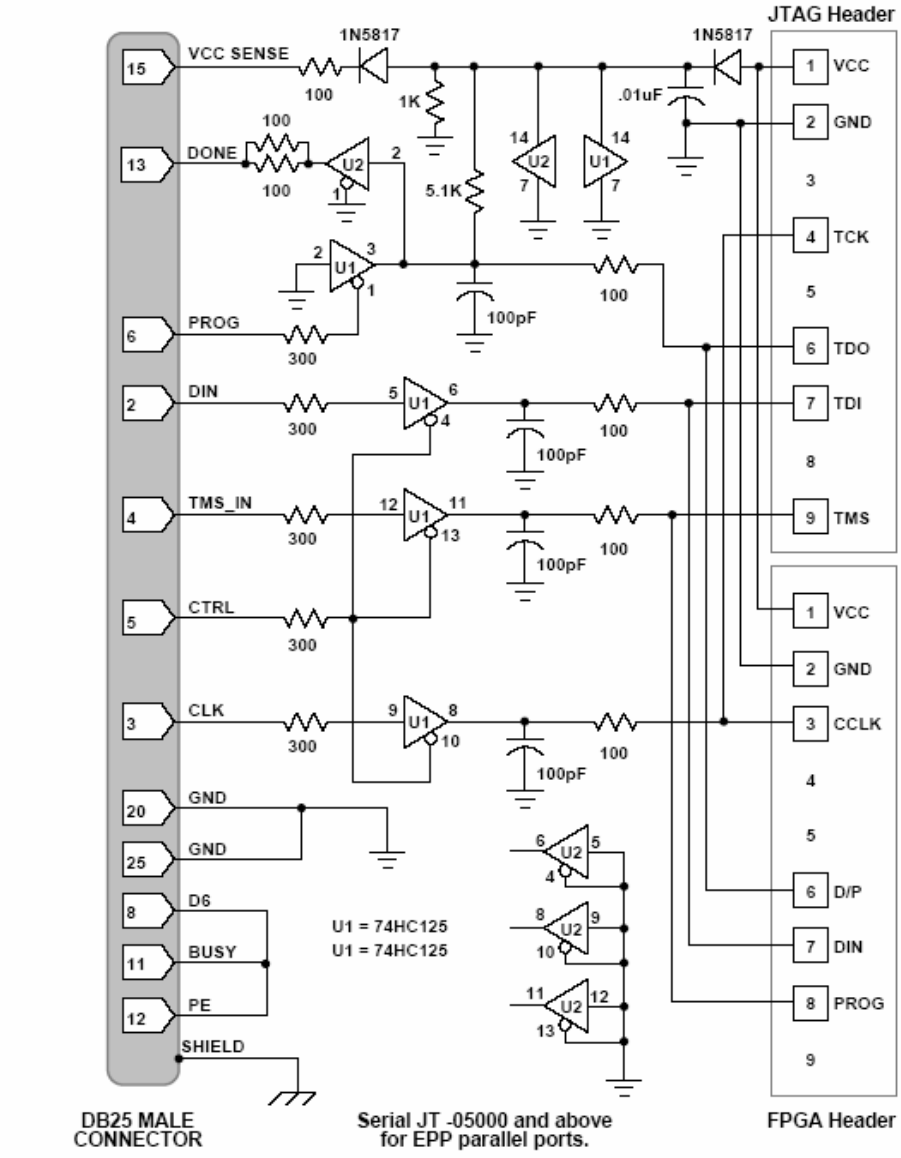
۷- کلاک TCK

۸- بازگشت به RTI

پیکره بندی و باز خوانی (read black) از طریق TAP همیشه موجود می باشد .

مد اسکن مرزی به سادگی دیگر مدهای پیکره بندی را غیر فعال می کند . مد اسکن مرزی با (lox) در مد پایه (M0,M1,M2) مشخص می شود .

شکل زیر Programmer FPGA را به روش های Serial Slave و JTAG نشان می دهد:



شکل۲-۱۱

۲-۷-۵ read black (باز خوانی) :

پیکره بندی داده ذخیره شده در $spar\tan\Pi$ و پیکره بندی حافظه برای تأیید درستی می تواند بـازخوانی شود .

در پیکره بندی داده ها امکان بازخوانی محتوی همه فلیپ فلاپها و لچ ها ، LUT RAM ها و بلوک هـای RAM وجود دارد. این توانایی برای اشکال زدایی بلا درنگ به کار می رود .

۲-۸ بکار گیری حلقه های delay –locked :(تأخیر و نگهداری)

خانواده $spar\tan\Pi$ چهار مدار تمام دیجیتال حلقه delay –locked (DLL) مـی باشـد کـه ایجـاد تأخیر انتشار صفر و حذف عدم هماهنگی میان کـلاک ورودی و کـلاک ورودی داخلـی را بـر عهـده دارد و کنترل پیشرفته کلاک های وابسته نیز بر عهده آن می باشد .

DLL های اختصاصی می توانند برای تکامل چندین مدار بکار روند تا مراحل طراحی سسیـستم را بهبـود ببخشند .

وقتی که اندازه FPGA بزرگ می شود ،توزیع کلاک در داخل تراشه بسیار مهم می باشد . تأخیر کـلاک و عدم هماهنگی کلاک باعث تداخل در کار این وسیله می شود .

$spar\tan\Pi$ این مشکل را با قرار دادن چهار مدار DLL که قبلاً توضیح داده شد بر طرف نمود .

هر DLLمی تواند ۲ شبکه مسیر یابی کلاک سراسری در داخل وسیله را راه اندازی کنـد . شـبکه توزیـع کلاک سراسری عدم توازن کلاک را به سبب تفاوتها به حداقل می رساند . با نظارت بر کلاک توسط DLL

، DLL می تواند تأخیر شبکه مسیر یابی را جبران کند و به طور مؤثر تأخیر ناشی از ورودی های خـارجی را بار کردن کلاک را حذف کند .

به منظور فراهم کردن تأخیر صفر ،DLL مـی توانـد چنـدین فـاز از کـلاک منبـع را تولیـد کنـد . DLL همچنین می تواند کلاک را دو برابر یا تقسیم کند .

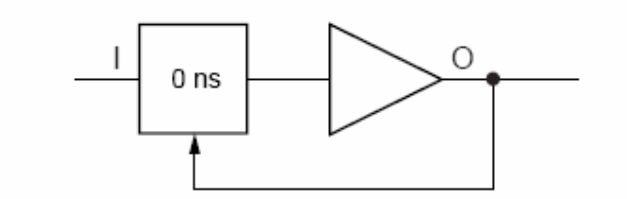
افزایش کلاک به طراح اجازه طراحی متنوعی را می دهد برای مثال کلاک منبع 50MHZ توسـط DLL می تواند FPGA را در فرکانس 100MHZ راه اندازی کند .

این تکنیک می تواند طراحی برد را ساده سازد بدلیل اینکه در داخل برد کار با چنین فرکانسی مشکل مـی باشد ولی در داخل FPGA بدلیل کوتاهی مسیر مشکل نمی باشد.

قرار دادن دو DLL به صورت سری می تواند کلاک را چهار برابر کند .

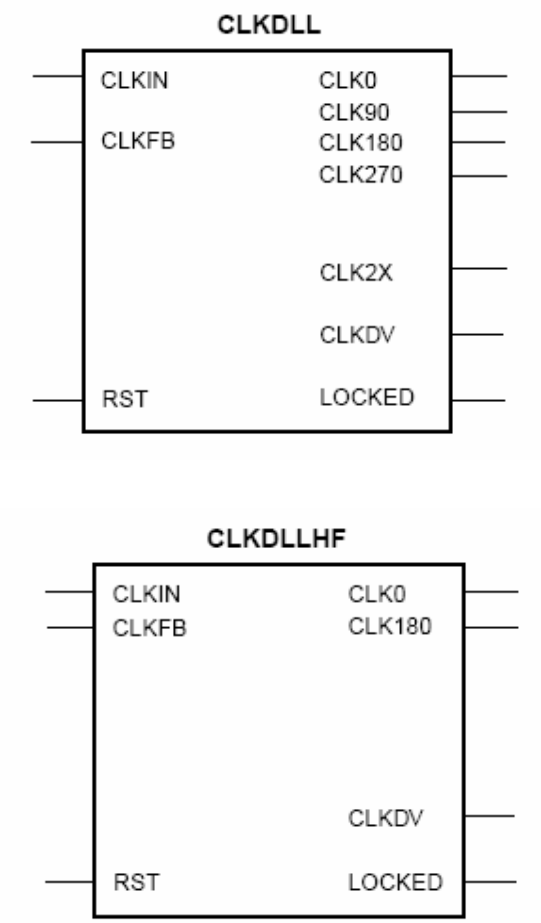
۲-۸-۱ نماد های کتاب خانه DLL :

شکل ۱۲ یک نماد کتابخوانه ای ماکرو DLL ساده را نشان می دهد (BUFGDLL) این ماکرو منجمـد به روشی مناسب و سریع برای فراهم کردن کلاک سیستم با تأخیر انتشار صفر در داخل وسیله می باشد .



شکل۲-۱۲

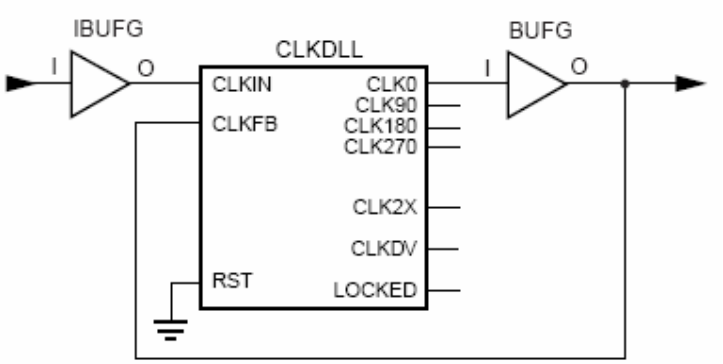
شکل ۱۳و۱۴ ، ۲ کتابخوانه DLL اولیه را نشان میدهد . این نماد ها دسترسی کامل به جزئیـات DLL را در کاربرد های پیچیده ممکن می سازد .



شکل۲-۱۳و۱۴ : ۲ کتابخوانه DLL اولیه

۲-۸-۱-۱ توصیف پایه های BUFGDLL :

بکار گیری ماکرو BUFGDLL ساده ترین روش برای فراهم کردن تأخیر صفر برای فن لوت بالا و کـلاک از ورودی خارجی می باشد . ایـن مـاکرو از BUFG,CLKDLL,TBUFG اولیـه بـرای تکامـل بیـشتر کاربرد ها بکار می رود که در شکل ۱۵ دیده می شود .



شکل ۲-۱۵

این نماد دسترسی به کنترل کلاک های وابسته یا امکان دو برابر کردن کلاک یا تقسیم کلاک را نمی دهد .

این نماد همچنین دسترسی به پایه RST یا LOCKED در DLL را نمی دهد .

برای دسترسی به این جزئیات ،طراح باید کتابخوانه اولیه DLL که در پائین توضیح داده شـده را اسـتفاده کند .

۲-۸-۱-۱-۱ منبع کلاک ورودی – I :

پایه I کلاک منبع کاربر را فراهم می کند برای ماکرو BUFGDLL فرکانس منبـع کـلاک بایـد در رنـج فرکانسی پائین که در D ata sheet مشخص شده باشد.

BUFGDLL نیاز به سیگنال کلاک منبع خارجی دارد . بنابراین ورودی خارجی فقـط مـی توانـد منبـع سیگنال برای پایه I ، BUFGDLL. باشد .

۲-۸-۱-۱-۲ کلاک خروجی – O :

کلاک خروجی پایه O نشان دهنده تأخیر جبران شده از کلاک منبع (I) می باشد این سـیگنال از مزایـای منابع مسیر یابی کلاک سراسری استفاده می کند .

کلاک خروجی دارای چرخه کار (Duty Cycle)50/50 می باشد مگر اینکه کاربر خصوصیات چرخه کـار را تغییر دهد .

۲-۸-۲ توصیف پایه های اولیه CLKDLL :

کتابخوانه CLKDLL اولیه ،دسترسی کامل به همه جزئیات مورد نیاز DLL در کاربردهـای پیچیـده بـا DLL امکان پذیر می باشد .

۲-۸-۲-۱ کلاک منبع ورودی CLKIN :

پایه CLKIN کلاک منبع کاربر را برای DLL فراهم می کند .

بافر کلاک سراسری (BUFG) از طریق CLKDLL دیگری راه انـدازی مـی شـود یـا از طریـق یکـی از بافرهای ورودی کلاک سراسری (IBUFG) باید منبع این سیگنال کلاک باشد .

۲-۸-۲-۲ فیدبک کلاک ورودی CLKDFB:

DLL نیاز به یک مرجع یا سیگنال فیدبکی برای جبران کردن تأخیر در خروجی مـی باشـد .CLKO یـا CLK2X را به عنوان فیدبک به کلاک ورودی CLKFB متصل می کنیم . با این کار فیـدبک لازم بـرای DLL فراهم می شود .

۲-۸-۲-۳ ورودی Reset :

وقتی که پایه RST فعال است سیگنال LOCKEDدر چهار منبع کلاک غیر فعال می باشد .

پایه RST (فعال بالا) باید به یک سیگنال دینامیک یا به زمین متصل باشد .

۲-۸-۲-۴ کلاک خروجی CLK2X-2X :

پایه خروجی CLK2X کلاک بافرکانس ۲ برابر تولید می کند با چرخه کاره 50/50 .

وقتی که CLKDLL , lock می شود ،خروجی CLK2Xبه 1X تغییر می یابد با چرخه کار ۲۵/۷۵ این رفتار به DLL اجازه پیدا کردن لبه درست را می دهد .

۲-۸-۲-۵ خروجی تقسیم کننده کلاک – CLKDV:

پایه خروجی تقسیم کننده کلاک CLKDV فرکانس پائین تری از کلاک منبع را فراهم می کند .

CLKDV کلاک منبع را تقسیم بر Nمی کند که N می تواند ۲،۳،۴،۵،۸/۱،۲،۵/۵ یا ۱۶ باشد .

این خصوصیات به طور اتوماتیک چرخه کار را تصحیح می کنند به طوری که پایه خروجی CLKDV چرخه کار ۵۰/۵۰ را همیشه دارا می باشد .

۲-۸-۲-۶ خروجی های کلاک CLK[0/90/180/270] :

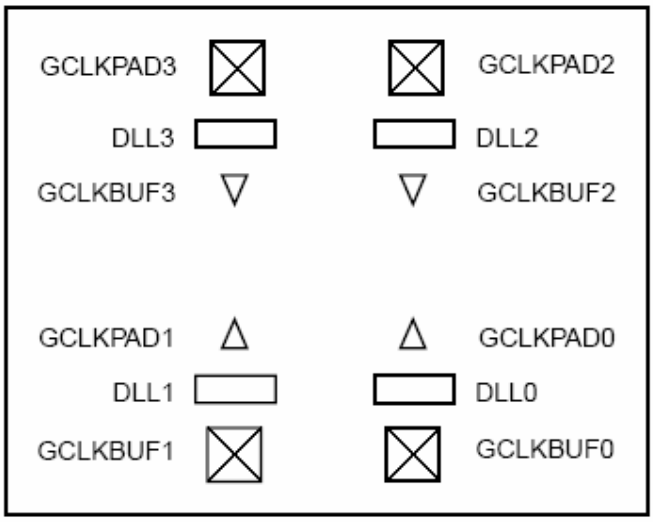
پایه خروجی کلاک CLK,1X که جبران کننده تأخیر سیگنال کلاک منبع CLKIN می باشد . CLKDLL اولیه ۳ شیفت فاز از CLKO را تولید می کند . رابطه میان شیفت فاز و دوره شیفت مطابق جدول ۴ می باشد .

جدول ۲-۴

Phase (degrees)	Period Shift (percent)
0	0%
90	25%
180	50%
270	75%

۲-۸-۳ محدودیت مکانی DLL :

در هر گوشه از FPGA یک DLL وجود دارد . محدودیت مکانی LOC ،وابسته به نماد DLL با مشخصه عددی ۰و۱و۲یا ۳ مکان DLL را کنترل می کند . موقعیت چهار DLL و منابع کلاک مطابق آن در شکل ۱۶ دیده می شود .



شکل ۲-۱۶

۲-۸-۴ کلاک ورودی :

سیگنال کلاک خروجی از DLL در اصل یک نسخه تأخیر یافته از سیگنال کـلاک ورودی مـی باشـد کـه بازتابی از بی ثباتی کلاک ورودی در خروجی شکل موج می باشـد . بـه ایـن دلیـل کیفیـت کـلاک ورودی DLL ارتباط مستقیمی با کیفیت شکل موج خروجی تولید شده توسط DLL می باشد .

در بشتر سیستم ها نوسان ساز کریستالی ،کلاک سیستم را تولید می کند . DLL می تواند با هـر نوسـان ساز کریستالی تجاری کار کند .

استفاده از خصوصیات بلوک های RAM خـانواده $spar\tan\Pi$ بلـوک هـای اختـصاصی RAM بـر روی تراشه فراهم می کند که سنکرون بوده و دارای Read/write می باشند.

هر سلول حافظه دارای ۴۰۹۶ پیت می باشد . هر پورت بلوک هـای حافظـه RAM بـه طـور مـستقل بـه صورت پورت Read/write پورت خواندن ،پورت نوشتن و می تواند برای اندازه داده دلخواه پیکـره بنـدی شود .بلوک های حافظه RAM به کاربر توانایی طراحی ساده را می دهد .

۲-۹ مدهای عملیاتی :

بلوک های حافظه RAM ،۲ مد عملیاتی را پشتیبانی می کند.

- Read through
- Write Back

۲-۹-۱ Read through :

آدرس خواندن بر روی پورت خواندن قرار می گیرد و داده ها بر روی خروجی RAM بعد از زمان دسترسی ظاهر می شوند . بعضی حافظه ها ممکن است از لچ یا نبات در خروجی استفاده کنند که این عمل وابسته به سرعت مورد نیاز می باشد .

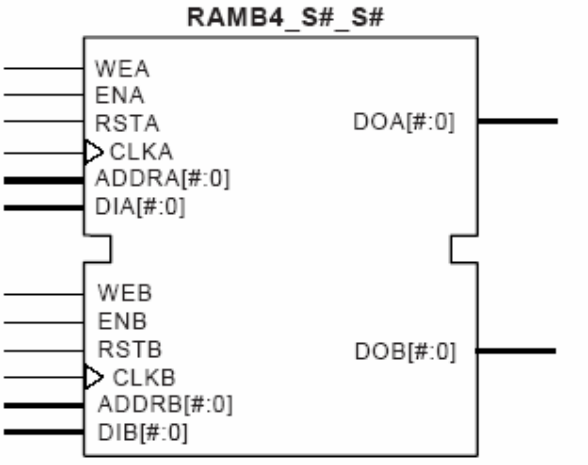
این روش به طور کلی ،رسیدگی به یک راه حل درجه دوم می باشد که مد خواندن را به تابع آسنکرون با امکان از دست دادن انتقال خط کنترل آدرس در طول تولید پالس ساعت خواندن تبدیل می کند .

۲-۹-۲ Write Back :

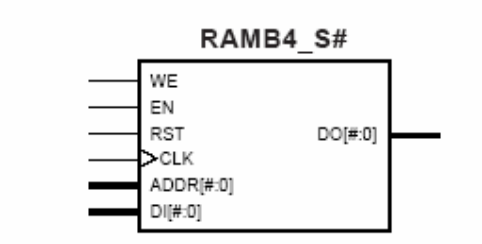
آدرس نوشتن بر روی پورت نوشن قرار می گیرد و داده های ورودی به داخل حافظه نوشته می شوند .

۲-۱۰ کتابخانه اصلی :

شکل ۱۷و۱۸ ،۲ بلوک RAM اصلی را نشان می دهد .



شکل ۲-۱۷



شکل ۲-۱۸

هر بلوک RAM عملکرد مستقلی از دیگران دارد . جدول۳ نسبت عمق و عرض بلوک های حافظه RAM را نشان می دهد .

جدول ۲-۵

Width	Depth	ADDR Bus	Data Bus
1	4096	ADDR<11:0>	DATA<0>
2	2048	ADDR<10:0>	DATA<1:0>
4	1024	ADDR<9:0>	DATA<3:0>
8	512	ADDR<8:0>	DATA<7:0>
16	256	ADDR<7:0>	DATA<15:0>

۲-۱۱ بکار گیری I/O های همه منظوره :

خانواده $spar\tan\Pi$ شامل منابع I/O با کارایی بالا که I/O های همه منظوره نامیده می شود ،مـی باشـد که رنج وسیعی از I/O های استاندارد را پشتیبانی می کند .منابع I/O همه منظوره یک خصیصه قوت می باشد که به واسطه آن می توان مقدار راه اندازی خروجی ،تـذخیر ورودی ،زمـان نگهـداری را برنامـه ریـزی کرد.همانطور که اندازه و ظرفیت FPGA بزرگ می شود سیستم های بزرگ تر و پیچیده تـری بـرای آنهـا

طراحی می شود و نیاز به افزایش تنوع I/O های استاندارد در آنها مشاهده می شود .از این گذشته سرعت کلاک سیستم همچنان درحال افزایش است ،بنابراین نیاز به I/O های با کارایی بالا بسیار مهم می باشد . وقتی که تأخیر باعث افزایشقابل توجه فشار بر سرعت سیستم می شود ، وظیفه نایل شدن به کارایی مورد نظر سیستم با افزایش I/O بیشتر وسیله های متعارف قابل پیکره بندی حل کردند هر I/O همه منظوره می تواند ۱۶تا I/O های استاندارد اجازه پشتیبانی از پهنای بزرگی از کاربرد ها را می دهد از اهداف کلی کاربرد های استاندارد تا باس های حافظه با ولتاژ پائیت و سرعت بالا بلوک های I /O همه منظوره قابلیت راه اندازی بافرهای خروجی LVTTL را به صورت اختیاری با pull up یا pull down ضعیف یا مدار نگهدارنده ضعیف ایده ال برای بکار گیری در کاربرد هایی با باس خارجی را دارند .هر IOB شامل ۳ ثبات می باشد ،یکی برای ورودی ،یکی برای خروجی و یکی برای سیگنال 3-state در داخل IOB می باشد . این ثباتهای اختیاری قبل برنامه ریزی مانند D فلیپ فلاپ یا لچ حساس به سطح می باشند .

بافر ورودی تأخیر اختیاری برای گارانتی زمان نگهداشت صفر برای سیگنال ورودی ثبات را دارا می باشد .I/O های همه منظوره منابع اختصاصی برای ولتاژ ورودی مرجع (VREF) و ولتاژ خروجی منبع (VCCO) دارند که برای تعیین مقدار ورودی و خروجی استفاده می شود جدول ۴ انواع I/O های استاندارد با VREF , VCCO مورد نیاز آمده است که در صورت استفاده از آن استاندارد در بانک مورد نظر در FPGA باید پایه VCCO , VREF به ولتاژ مورد نظر متصل شوند . در صورتی که در I/O استانداردی VREF مورد نیاز نباشد می توان از آن پایه به عنوان I/O استفاده کرد .

۲-۱۱-۱ LVTTL (TTL ولتاژ پائین):

استاندارد LVTTL یک هدف کلی برای استاندارد EIA/JESDSA برای کاربردهایی کـه بـا ولتـاژ ۳/۳ ولت کار می کنند می باشد .

این استاندارد به ولتاژ VCCO ، 3/3Vنیاز دارد و به VREF نیازی نمی باشد .

۲-۱۱-۲ LVCMOS2- CMOS ولتاژ پائین برای ۲/۵ ولت :

LVCMOS2 برای ولتاژ ۲/۵ ولت یا استانداردهای پائین تر از آن طراحی شده است . این اسـتاندارد بـه ولتاژ VCCO ، 2/5V نیاز دارد و به VREF نیاز ندارد .

۲-۱۱-۳ PCI (واسط اجزاء جانبی):

استاندارد PCI به ولتاژ VREF نیاز ندارد و به ولتاژ VCCO معادل ۳/۳ ولت نیاز دارد

برای پیکره بندی I/O های PCI با 33MHZ با 5V استاندارد کار می کند .

مابقی استانداردها با توجه به جدول ۶ مشخص می باشند .

جدول ۲-۶

I/O Standard	Input Reference Voltage (V_{REF})	Output Source Voltage (V_{CCO})	Board Termination Voltage (V_{TT})
LVTTL (2-24 mA)	N/A	3.3	N/A
LVCMOS2	N/A	2.5	N/A
PCI (3V/5V, 33 MHz/66 MHz)	N/A	3.3	N/A
GTL	0.8	N/A	1.2
GTL+	1.0	N/A	1.5
HSTL Class I	0.75	1.5	0.75
HSTL Class III	0.9	1.5	1.5
HSTL Class IV	0.9	1.5	1.5
SSTL3 Class I and II	1.5	3.3	1.5
SSTL2 Class I and II	1.25	2.5	1.25
CTT	1.5	3.3	1.5
AGP-2X	1.32	3.3	N/A

فصل ۳

۳-توضیح ساختار نرم افزار سنتز xilinx (ISE)

در این قست موارد زیر را تو ضیح می دهیم:

VHDL and Schematic design flow-1 که چگو نگی استفاده از VHDL و ابزار های الگوهای طراحی را نشان می دهد و چگونگی توصیف رفتاری و شبیه سازی زمان بندی و تکامل طراحی را نشا ن میدهد.

در این قسمت ما به توصیف و نشان دادن چگونگی استفاده از VHDL و ابزار های طراحی الگو ها و چگونگی توصیف رفتاری و شبیه سازی زمانبندی و چگونگی تکامل طراحی می پردازیم.

۳-۱ نرم افزارهای مورد نیاز:

برای انجام اموزش ما به نرم افزارهای زیر نیاز داریم:

ISE6.x -

- شبیه ساز Modelsim که شبیه سازی VHDL را پشتیبانی کند.

۲-۲ شروع افزار نرم ISE :

برای وارد شدن به محیط نرم افزار باید Project Navigator را از مسیر

Start->Programs->Xilinx ISE6.x->Project Navigator اجرا کنیم.

۲-۳ طراحی به زبان VHDL

در این قسمت یک ماژول شمارنده ۴-بیتی با طراحی سطح بالا انجام می دهیم.

طراحی شامل ماژول شما رنده HDL که توسط قالب زبان ایجاد شده می با شد.

برای شروع یک پروژه جدید ایجاد میکنیم.

۲-۴ ایجاد پروژه جدید:

یک پروژه جدید در ISE مجموعه ای فایل های لازم برای ایجاد و بار کردن طراحی در وسیله انتخاب شده

می باشد.

برای ایجاد پروژه جدید مراحل زیر را دنبال می کنیم:

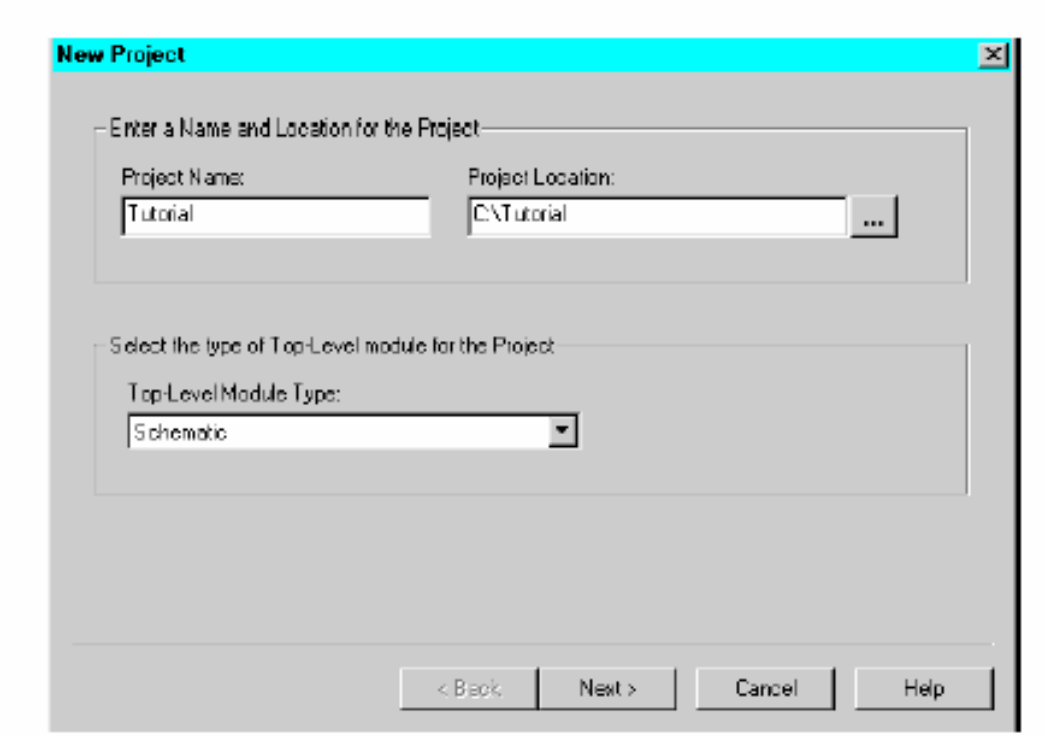
۱)File->new project را انتخاب می کنیم .

۲)در پنجره new project محل مورد نظر برای ایجاد پروژه را انتخاب می کنیم.

۳)در محل نام پروژه برای مثال نام Tutorial را می نویسیم.

۴)در قسمت Top level Module Type ‘ Schematic را انتخاب می کنیم.

مانند شکل زیر:



شکل۳-۱: پنجره ایجاد پروژه جدید

۵) بر روی next

۶) در پنجره بعدی در قسمت Device family نام خانواده ای از xilinx را که مورد نظر ماست انتخاب نیکنیم.

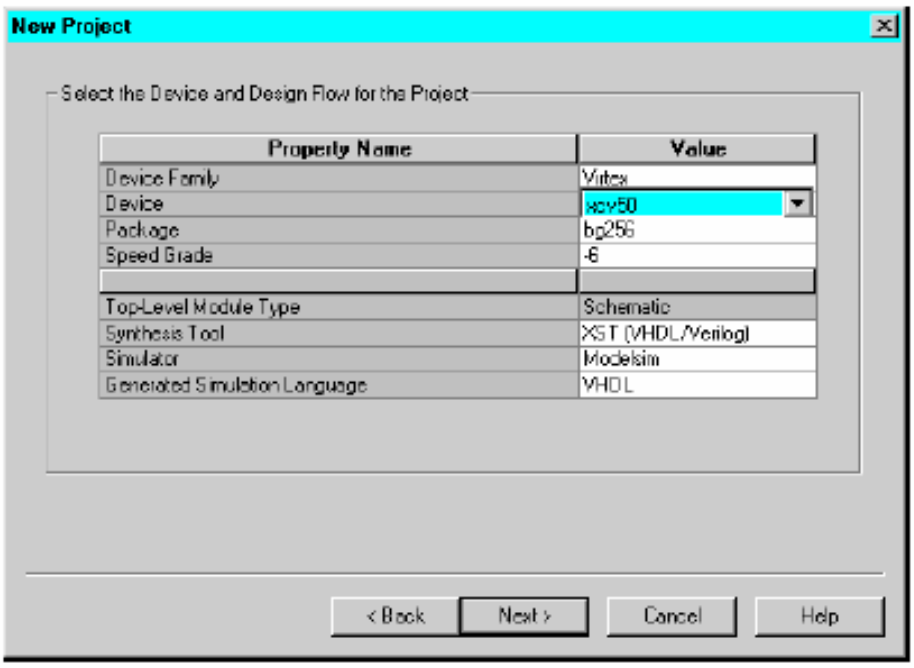
درقسمت Device نام تراشه FPGA مورد نظر را انتخاب می کنیم.

در قسمت Synthesis Tool : xst(VHdl/Verilog) را انتخاب می کنیم.

در قسمت Simulator : Modelsim را بر می گزینیم.

و در نهایت در قسمت Generated Simulation Language ' VHDL را بر می گزینیم.

مانند شکل زیر:



شکل ۳- ۲

۷) بر روی Next کلیک می کنیم.

۸) در پنجره بعد بر روی New Source کلیک کرده تا یک منبع جدید برای شمارنده به پروژه اضافه کنیم.

۹) در پنجره New Source ' VHDL Module را انتخاب می کنیم.

۱۰) در قسمت نام فایل Counter را وارد می کنیم.

۱۱) بر روی Next کلیک می کنیم.

۱۲) بر روی Finish کلیک کرده تا اضافه کردن منبع جدید به پروژه تمام شود.

با اتمام مراحل بالا یک پروژه جدید ایجاد کرده ایم .

۲-۵ تغییر ماژول شمارنده با الگوی شمارنده:

برای کامل کردن ماژول شمارنده وقرار دادن تعریف پورت ها و توصیف رفتاری برای شمارنده VHDL از ISE Language template استفاده می کنیم .

Language template شامل تعداد زیادی از الگو های VHDL و verilog وABEL برای بکار گیری در طرح می با شد.

۱) Language template را از طریق Edit->Language template یا با کلیک کردن بر روی آیکن آن می توان باز کرد.

Figure 3: **Language Templates icon**

۲) در پنجره Language Template بر روی علامت + کنار VHDL برای باز شدن لیست الگو ها کلیک می کنیم و سپس بر روی علامت + در کنار Synthesis Template کلیک کرده و counter را انتخاب می کنیم.(شکل ۴)

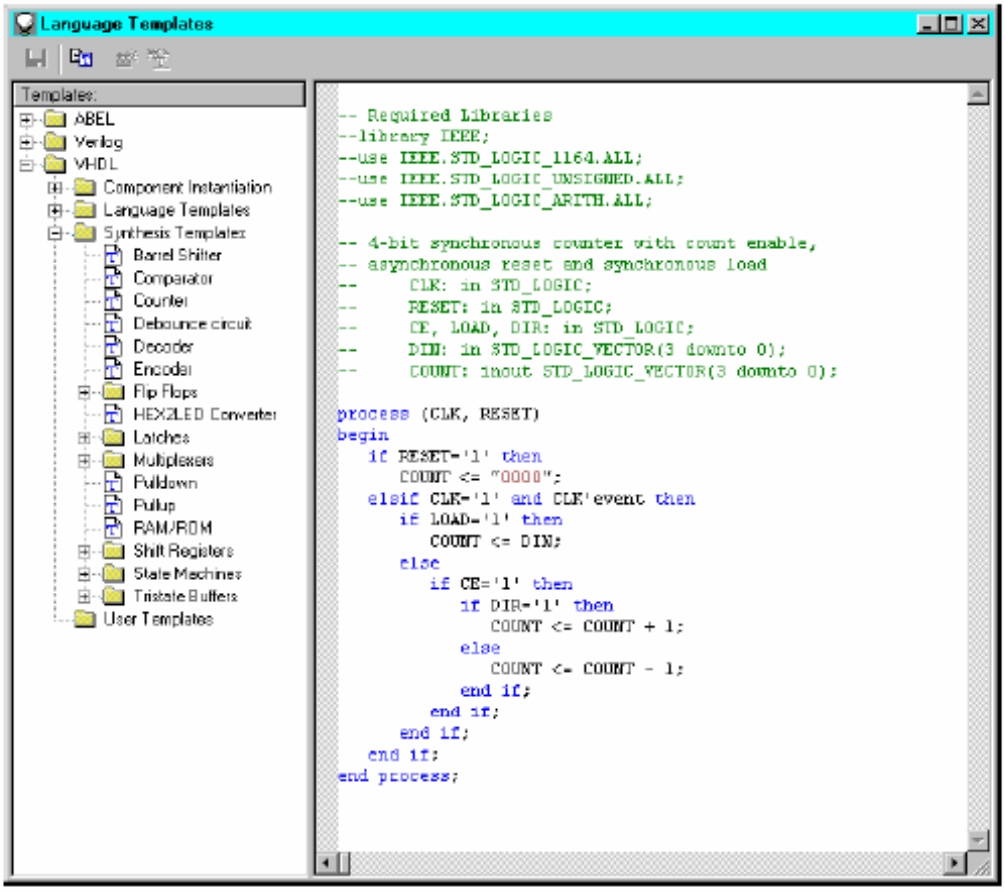
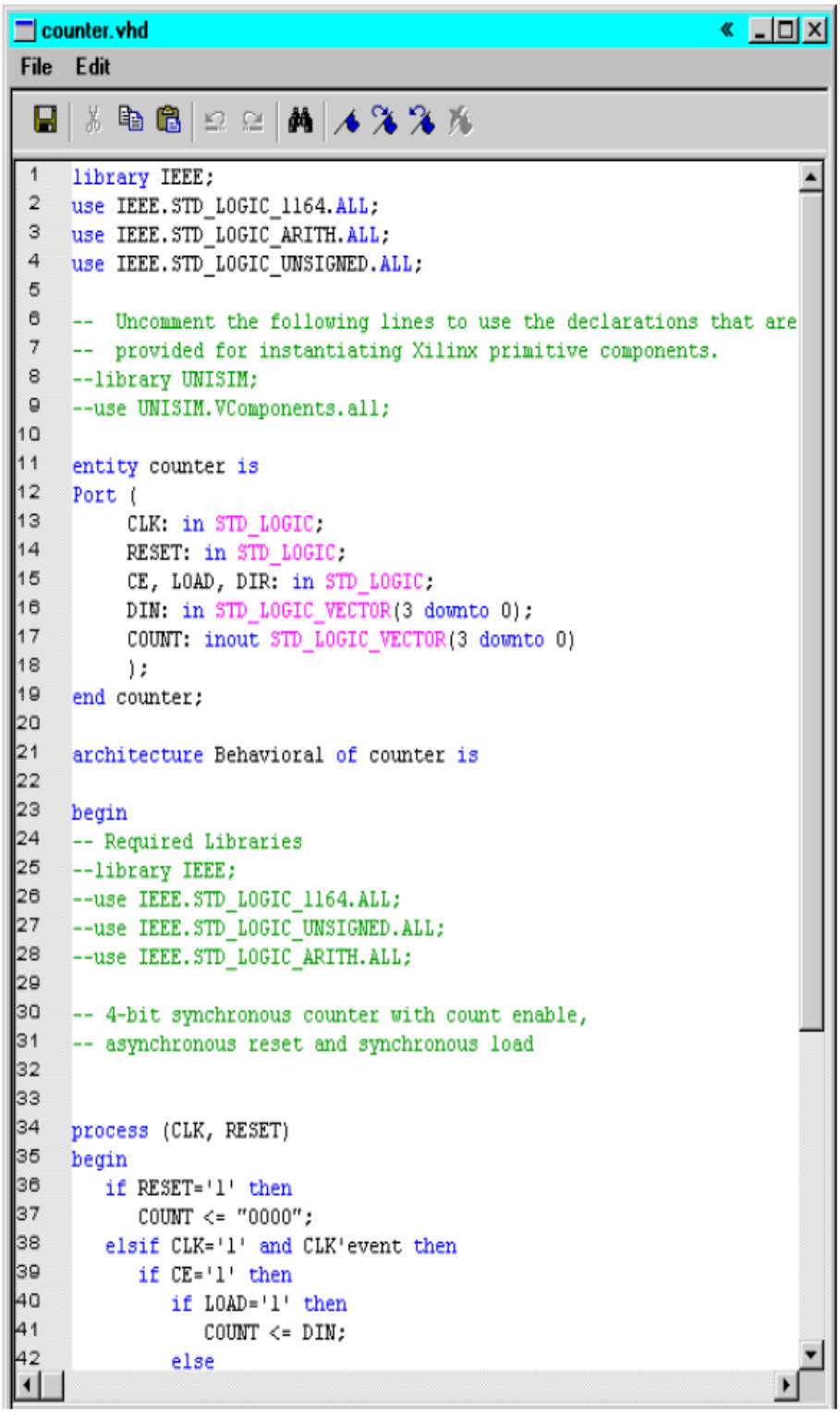


Figure 4: **Counter Language Template**

۳) محتویات الگوی شمارنده را در داخل Counter.vhd کپی می کنیم.

۴) پنجره Language template را می بندیم.

۵) بعد از فعال کردن ورودی ها و خروجی های غیر فعال در قسمت Counter انرا ذخیره می کنیم.



شکل۳-۵ : تغییر یافته ماژول شمارنده

۶-۲ ایجاد یک منبع Test bench :

برای ایجاد test bench مراحل زیر را دنبال می کنیم:

۱) در پنجره Sources in project , Counter.vhd را انتخاب می کنیم.

۲) Project->New Source را انتخاب می کنیم.

۳) بعد از مشاهده پنجره New source , Test bench waveform را انتخاب می کنیم.

۴) یک نام دلخواه برای مثال Counter_tbw را می نویسیم.

۵) بر روی Next کلیک می کنیم.

ما می توانیم پارامتر های زمانبندی را در طول شبیه سازی مشخص کنیم.

همانطور که درشکل زیر دیده میشود Test bench ایجاد شده و برای زمانبندی اماده است.

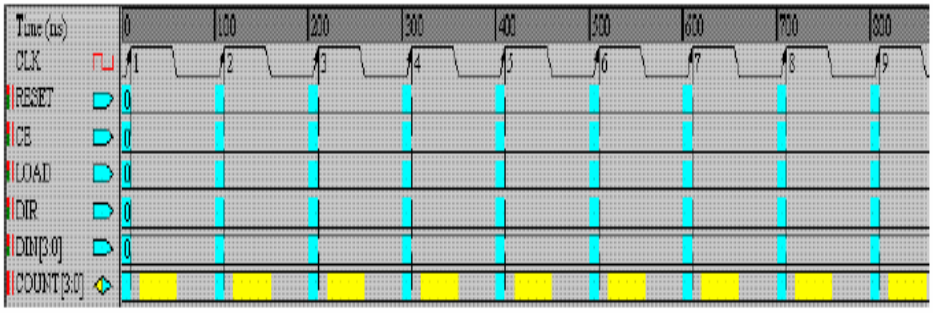


Figure 6: **Test Bench waveform in HDL Bencher**

۷-۲ مقدار دهی ورودی های شما رنده :

در این مر حله با توجه به پایه های ورودی تعریف شده می توانیم ورودی ها را در هر کلاکی که مد نظر ماست ۰ یا ۱ کرده و خروجی مورد نظر را مشاهده کنیم برای مثال ما پایه های ورودی را همانند شکل زیر تغییر می دهیم.

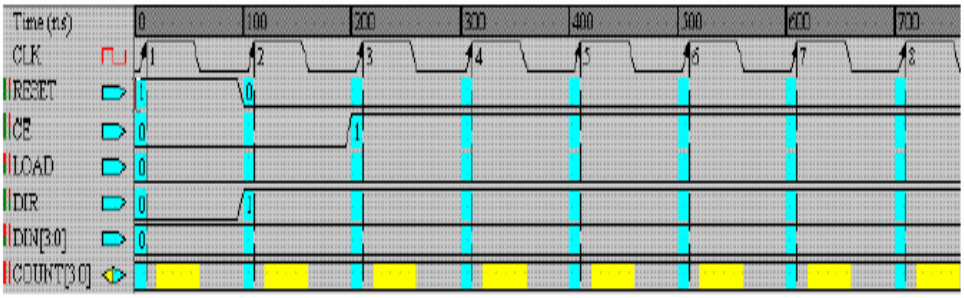


Figure 7: **HDL Bencher Stimulus and Response Entries**

۸-۲ تولید مقادیر خروجی :

ما اکنون می توانیم خروجی هایی را که انتظار داریم از ماژول شمارنده با توجه به مقدار دهی های ورودی بدست اوریم مراحل زیر را دنبال می کنیم:

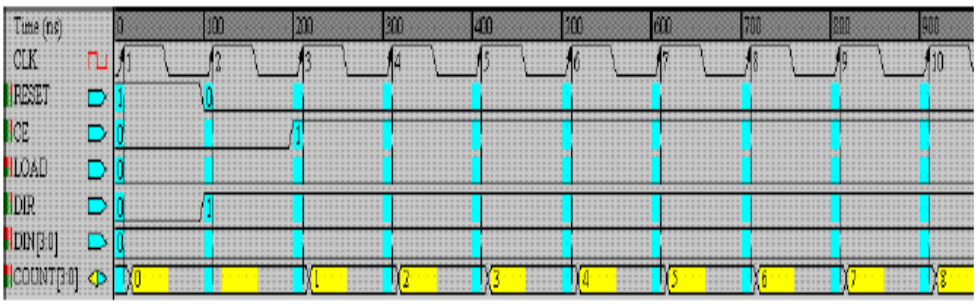
۱) Counter_tbw.tbw را در پنجره Source in project انتخاب می کنیم.

۲) در پنجره Processes for Source بر روی علامت + در مقابل Modelsim Simulator کلیک می کنیم.

۳) بر روی Generate Expected Simulation Result دو بار کلیک می کنیم .

در این حالت شبیه سازی با توجه به ورودی هایی که مشخص کردیم انجام شده و در پس زمینه انجام می شود و مقادیر خروجی ها مشخص می شوند.

خروجی شمارنده همانند شکل زیر می شود:



شکل ۸-۳ : خروجی شمارنده

۲-۹ وارد کردن طراحی :

در این بخش نشان می دهیم که چگونه یک طراحی سطح بالا که شامل ما ژول شمارنده می باشد ایجاد کنیم و نحوه سیم بندی و نام گذاری پایه ها و نشانه گذاری پایه های I/O و نشان دادن اینکه یک سیگنال ورودی یا خروجی می باشد را توضیح می دهیم.

برای ایجاد یک طراحی برای ماژول VHDL باید مراحل زیر را دنبال کنیم :

۱) در پنجره Sources in project , Counter.vhd را انتخاب می کنیم.

۲) در پنجره Processes for Source بر روی علامت + در کنار Design Entry Utilities کلیک کرده و گزینه Create Schematic symbol را انتخاب می کنیم.

با انجام این کار یک طرح از شمارنده در کتابخانه پروژه اضافه می گردد.

۲-۹-۱ ایجاد یک طراحی سطح بالا :

برای ایجاد طراحی سطح بالا مراحل زیر را دنبال می کنیم:

۱) Project->New Source را انتخاب می کنیم .

۲) نوع منبع را Schematic انتخاب می کنیم.

۳) یک نام دلخواه مثلاً top را وارد می کنیم.

۴) بر روی next کلیک کرده و سپس finish را می زنیم .

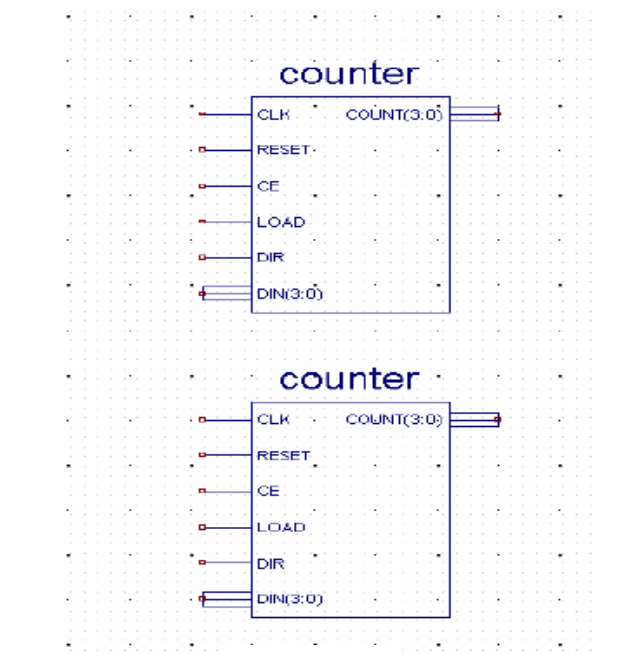
ECS اجرا شده و یک صفحه و یک صفحه خالی در داخل پنجره ECS باز می شود در ECS می توانیم دیا گرام طراحی خود را وارد کنیم.

۱) Add->Symbol راانتخاب می کنیم .

۲) Counter را انتخاب می کنیم.

۳) ۲ شمارنده در طرح خود قرار می دهیم .(با کلیک چپ موس , شمارنده را در هر جایی از صفحه که می خواهیم قرار می دهیم.

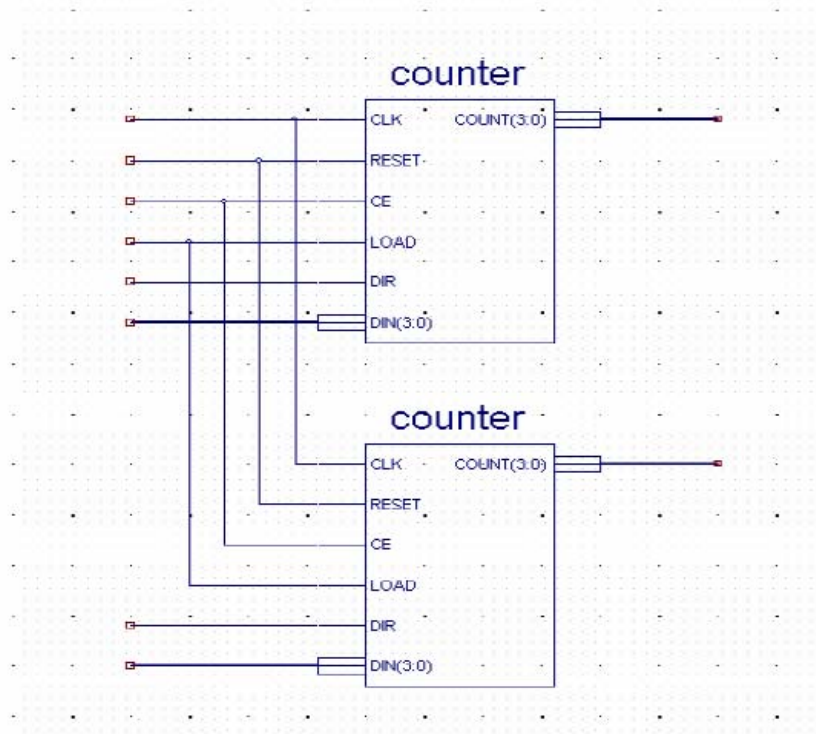
دیاگرامی که ما ایجاد کرده ایم مانند شکل زیر خواهد بود:



شکل ۳-۹

۲-۱۰ سیم بندی طرح :

برای فعال کردن حالت سیم بندی Add->wire را انتخاب می کنیم و سیم بندی را مطا بق شکل زیر انجام می دهیم:

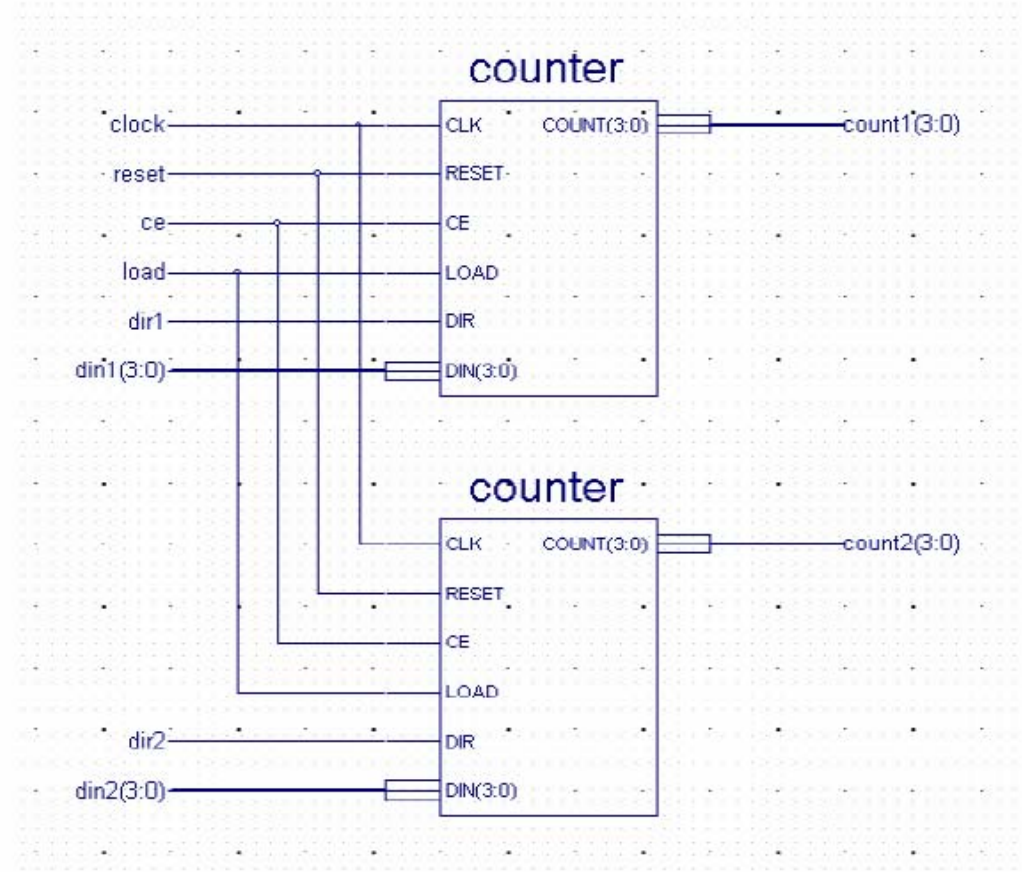


شکل ۳-۱۰

۲-۱۱ انتساب نام به ورودی ها و خروجی ها:

Add->Net name را انتخاب کرده و نام دلخواه را در ابتدا و انتهای سیم بندی انجام شده قرار می دهیم.

همانند شکل زیر:

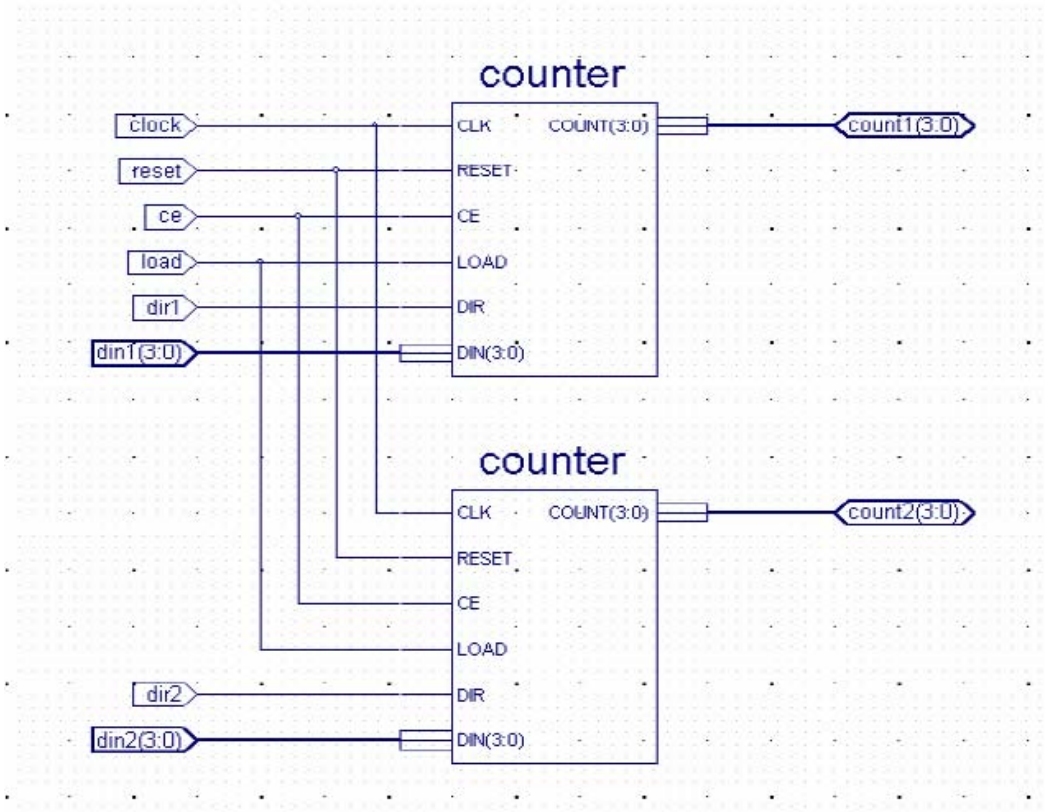


شکل ۳-۱۱

۱۲-۲ اضافه کردن نشانه های I/O :

برای اضافه کردن نشانه های I/O , add->I/O Marker را انتخاب می کنیم و به پایه های ورودی , نشانه ورودی (پایه های clock,reset,ce,load,dir1,dir2) و به پایه های din1,din2 نشانه گذرگاه را می دهیم.

مانند شکل زیر:



شکل ۳-۱۲

حال ماژول طراحی شده آماده می باشد و بعد از اتصال بورد طراحی شده به کامپیوتر می توان ان را بر روی برد برنامه ریزی کرد و نتایج را مشاهده نمود.

فصل ۴

۴- طراحی برد

۴-۱ مقدمه:

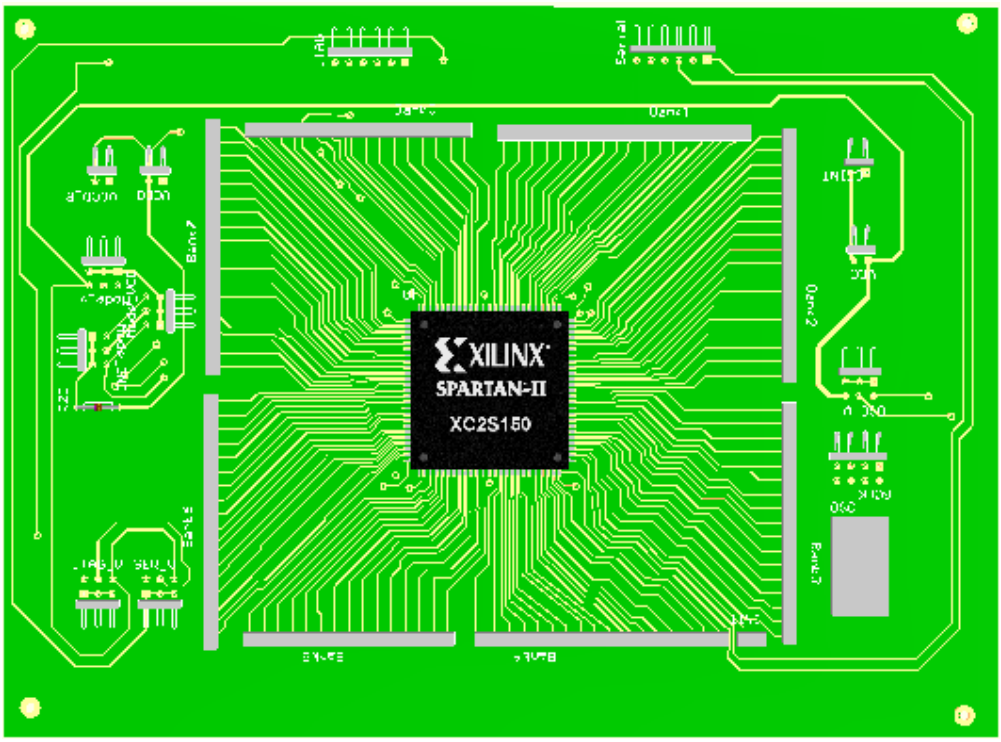
همان طور که در فصل های پیش توضیح داده شد برای ساخت بورد آموزشی FPGA ما از خانواده SPARTAN II مدل xc2s150 را انتخاب کردیم و خصوصیات آن را در فصل های پیش توضیح دادیم حال در این فصل به توضیح جزییات انجام پروژه می پردازیم.

۴-۲ فاز ۱: طراحی برد اصلی

در این پروژه پایه های I/O بر روی پایه هایی قرار گرفته اند تا هر وسیله جانبی را بتوان به آن متصل نمود . پایه هایی برای برنامه ریزی FPGA به روش های JTAG وSerial Slave تعبیه شده است . این بورد به گونه ای طراحی شده است که پایه های I/O به وسیله خاصی متصل نمی باشد و بعد از طراحی برنامه توسط کاربر چه به صورت شماتیک در Protel DXP و چه بصورت نوشتن کد VHDL ، I/O های تعریف شده را به پایه های FPGA اختصاص می دهیم که این کار را با ساخت Constraint file در Protel DXP و ISE می توانیم انجام می دهیم که در قسمت توضیح نرم افزار توضیح داده

شده است . بعد از اختصاص پایه ها وسایل جانبی مورد نظر را از طریق پایه های موجود به FPGA متصل می کنیم و عمل برنامه ریزی FPGA را توسط یکی از دو روش موجود انجام می دهیم.

شکل بورد اصلی مطابق شکل زیر می باشد:



شکل۴-۱: بورد اصلی

هما نطور که از تعریف پروژه پیداست سیستمی طراحی شده است که به واسطه آن بتوان مدار را در سطح رفتاری توصیف کرد ونتایج آن را بر روی برد مشاهده نمود بدین منظور برای تولید ورودی و خروجی های مدار وسایل جانبی تعبیه شده است که در صورت لزوم می توان از آنها استفاده نمود این برد قابلیت اتصال به LCD و Keyboard وMemory و... را دارا می باشد .

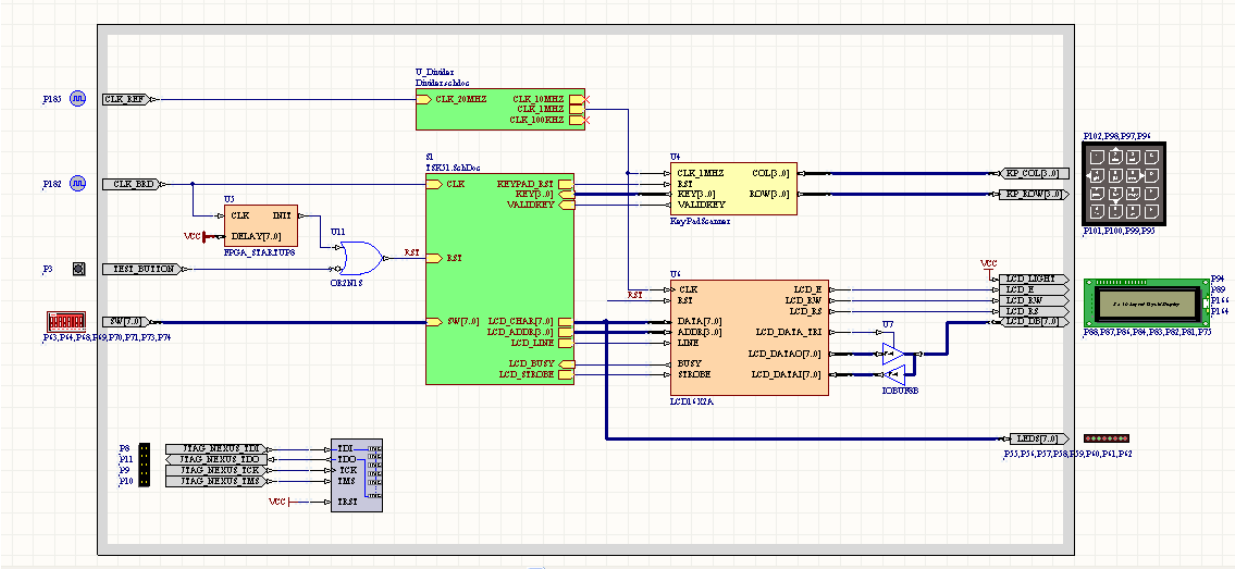
۴-۳ فاز دوم : اتصال قطعات جانبی

در این پروژه قطعات جانبی نظیر LCD و Keyboard و LED و Dip Switch و Seven segment طراحی شده بگونه ای که بعد از برنامه ریزی FPGA و تعیین پایه های ورودی و خروجی می توان از آنها استفاده نمود.

۴-۴ فاز سوم: راه اندازی قطعات جانبی

برای این بورد Dipswitch و Keyboard برای وارد کردن ورودی های مدار وLED array و LCD و Seven segment برای مشاهده نتایج تولید شده طراحی شده است .

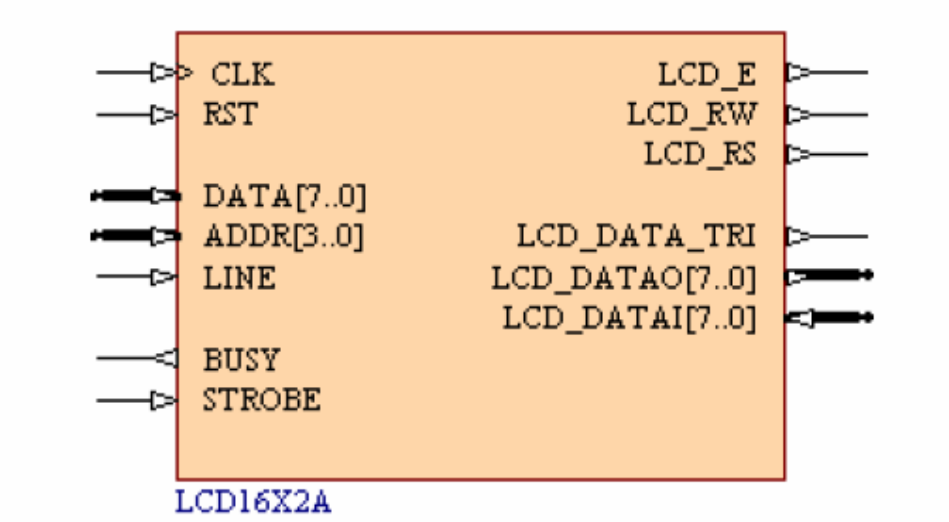
برای راه اندازی LCD و Keyboard ، ۲ راه وجود دارد راه اول استفاده از Protel DXP می باشد در Protel امکاناتی برای راه اندازی LCD و Keyboard وجود دارد که با انجام طراحی در Protel می توان آن را در FPGA بار نمود.در شکل زیر طراحی انجام شده در Protel را مشاهده می کنید



شکل ۴-۲ : راه اندازی LCD وKeyboard

حال به توضیح مراحل طراحی می پردازیم.

کنترلر LCD 16*2 یک واسط کامل و ساده میان LCD و کنترلر ایجاد می کند در شکل زیر یک کنترلر LCD نمایش داده شده است.



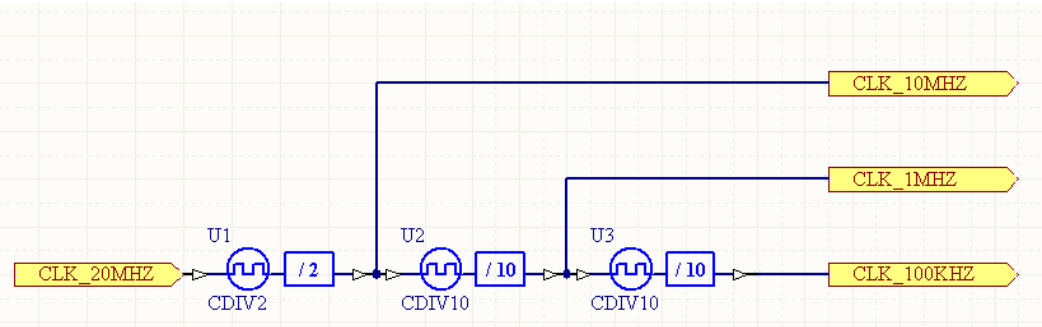
شکل ۴-۳ : کنترلر LCD

جدول توصیف پایه های کنترلر در ضمیمه آورده شده است.

راه دیگر استفاده از VHDL و نوشتن کد برای راه اندازی LCD و Keyboard می باشد در زیر کد نوشته شده برای راه اندازی LCD و Key board آمده است به واسطه این کد داده های تایپ شده روی Keyboard بر روی LCD مشاهده می شود.

برای راه اندازی Seven segment از IC ، 74LS47 استفاده شده است .

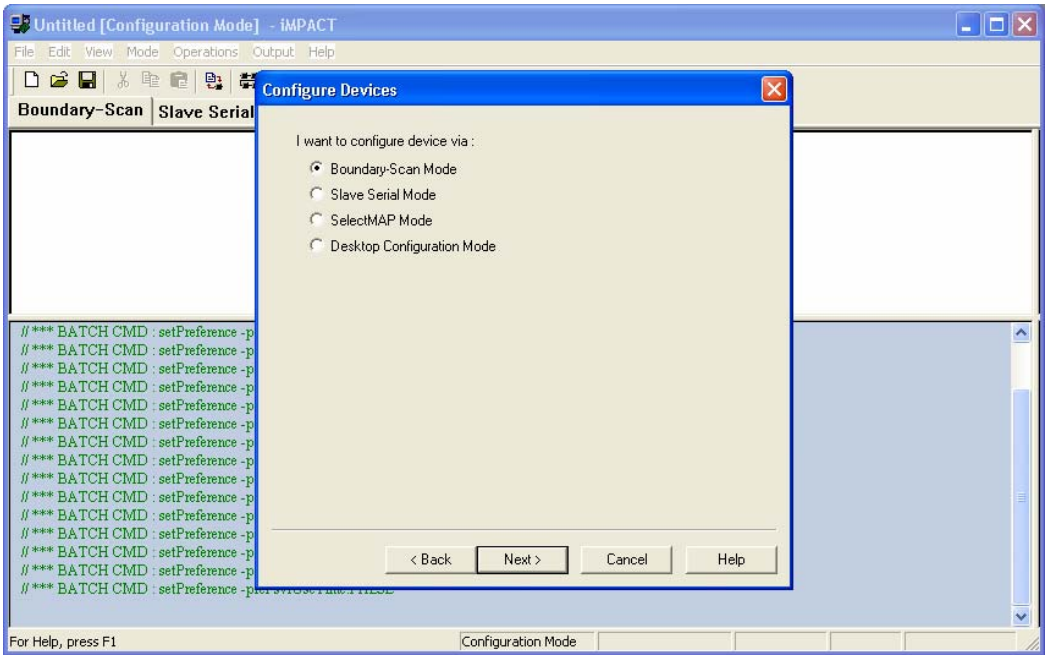
در Protel DXP امکانات متنوعی برای کار با FPGA قرار داده شده است مثلاً برای تقسیم یا ضرب فرکانس می توان از روش زیر استفاده نمود از این مدار می توان در طراحی استفاده کرده و کلاک مورد نظر را تولید نمود. فرض کنید می خواهیم شمارنده ای طراحی کنیم که هر یک ثانیه بشمارد یا به فرکاتسی کمتر یا بیشتر از فرکانس کلاک اصلی نیاز داشته باشیم برای این کار مداری مشابه مدار زیر طراحی کرده تا یک ثانیه را برای ما تولید کند.



شکل ۴-۴

۴-۵ فاز چهارم : برقراری ارتباط با نرم افزار و تست برد

بعد از اتصال programmer به کامپیوتر یکی از مد های Serial slave و یا JTAG را با تعیین M0,M1,M2 انتخاب می کنیم در نرم افزار ISE در قسمت Processes for source ، Configure Device را انتخاب می کنیم صفحه ای مشابه زیر ظاهر می شود که با توجه به مد انتخابی یکی از دو



شکل ۴-۵

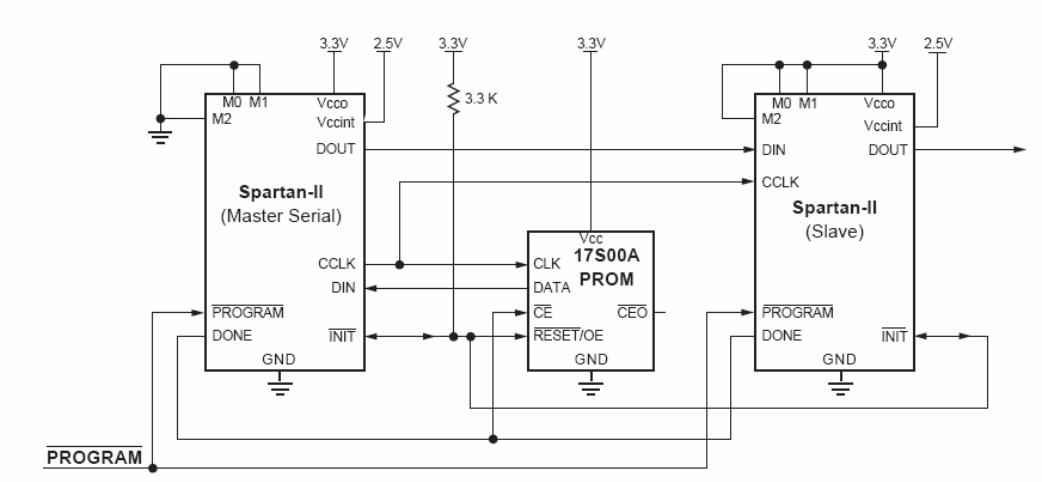
روش را انتخاب می کنیم و در صورت اتصال برد به کامپیوتر می توان برنامه مورد تظر را به درون FPGA

بار نمود.

فصل ۵

۵-پیشنهادات:

همانطور که در فصل اول گفتیم با قطع برق برنامه FPGA پاک می شود بدین منظور باید از یک PROM ، 17S00 همانطور که در شکل زیر آمده است استفاده کرد تا اگر به هر دلیلی VCC ، FPGA قطع شد بعد از برقراری VCC ، FPGA از طریق PROM بار شود.



شکل ۵-۱

از طرف دیگر گفته شد که CPLD دارای مدارات ترکیبی بیشتری نسبت به FPGA می باشد به همین منظور پیشنهاد می شود که در برد از یک CPLD استفاده شود تا با FPGA ارتباط داشته باشد.

منابع:

1- Introduction to FPGA , By Jamil Khatib , 2002

2- Digital System Design with VHDL , By Mark Zwoliski

3- VHDL Programming By Example , Douglas L.Perry

4- http://www.xilinx.com/literature/index.htm

5- http://www.xilinx.com/legal.htm

ضمیمه ۱: توصیف پایه های XC2S150

XILINX®

Spartan-II 2.5V FPGA Family: Pinout Tables

DS001-4 (v2.5) September 3, 2003 **Product Specification**

Pin Definitions

Pin Name	Dedicated Pin	Direction	Description
GCK0, GCK1, GCK2, GCK3	No	Input	Clock input pins that connect to Global Clock Buffers. These pins become user inputs when not needed for clocks.
M0, M1, M2	Yes	Input	Mode pins are used to specify the configuration mode.
CCLK	Yes	Input or Output	The configuration Clock I/O pin. It is an input for slave-parallel and slave-serial modes, and output in master-serial mode.
$\overline{\text{PROGRAM}}$	Yes	Input	Initiates a configuration sequence when asserted Low.
DONE	Yes	Bidirectional	Indicates that configuration loading is complete, and that the start-up sequence is in progress. The output may be open drain.
$\overline{\text{INIT}}$	No	Bidirectional (Open-drain)	When Low, indicates that the configuration memory is being cleared. This pin becomes a user I/O after configuration.
BUSY/DOUT	No	Output	In Slave Parallel mode, BUSY controls the rate at which configuration data is loaded. This pin becomes a user I/O after configuration unless the Slave Parallel port is retained. In serial modes, DOUT provides configuration data to downstream devices in a daisy-chain. This pin becomes a user I/O after configuration.
D0/DIN, D1, D2, D3, D4, D5, D6, D7	No	Input or Output	In Slave Parallel mode, D0-D7 are configuration data input pins. During readback, D0-D7 are output pins. These pins become user I/Os after configuration unless the Slave Parallel port is retained. In serial modes, DIN is the single data input. This pin becomes a user I/O after configuration.
$\overline{\text{WRITE}}$	No	Input	In Slave Parallel mode, the active-low Write Enable signal. This pin becomes a user I/O after configuration unless the Slave Parallel port is retained.
$\overline{\text{CS}}$	No	Input	In Slave Parallel mode, the active-low Chip Select signal. This pin becomes a user I/O after configuration unless the Slave Parallel port is retained.
TDI, TDO, TMS, TCK	Yes	Mixed	Boundary Scan Test Access Port pins (IEEE 1149.1).
V_{CCINT}	Yes	Input	Power supply pins for the internal core logic.
V_{CCO}	Yes	Input	Power supply pins for output drivers (subject to banking rules)
V_{REF}	No	Input	Input threshold voltage pins. Become user I/Os when an external threshold voltage is not needed (subject to banking rules).
GND	Yes	Input	Ground.
IRDY, TRDY	No	See PCI core documentation	These signals can only be accessed when using Xilinx PCI cores. If the cores are not used, these pins are available as user I/Os.

XC2S150 Device Pinouts

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
GND	-	P1	GND*	GND*	-
TMS	-	P2	D3	D3	-
I/O	7	P3	C2	B1	221
I/O	7	-	-	E4	224
I/O	7	-	-	C1	227
I/O	7	-	A2	F5	230
GND	-	-	GND*	GND*	-
I/O	7	P4	B1	D2	233
I/O	7	-	-	E3	236
I/O	7	-	-	F4	239
I/O	7	-	E3	G5	242
I/O	7	P5	D2	F3	245
GND	-	-	GND*	GND*	-
V_{CCO}	7	-	V_{CCO} Bank 7*	V_{CCO} Bank 7*	-
I/O, V_{REF}	7	P6	C1	E2	248
I/O	7	P7	F3	E1	251
I/O	7	-	-	G4	254
I/O	7	-	-	G3	257
I/O	7	-	E2	H5	260
I/O	7	P8	E4	F2	263
I/O	7	-	-	F1	266
I/O, V_{REF}	7	P9	D1	H4	269
I/O	7	P10	E1	G1	272
GND	-	P11	GND*	GND*	-
V_{CCO}	7	P12	V_{CCO} Bank 7*	V_{CCO} Bank 7*	-
V_{CCINT}	-	P13	V_{CCINT}*	V_{CCINT}*	-
I/O	7	P14	F2	H3	275
I/O	7	P15	G3	H2	278
I/O	7	-	-	H1	284
I/O	7	-	F1	J5	287
I/O	7	P16	F4	J2	290
I/O	7	-	-	J3	293
I/O	7	P17	F5	K5	299
I/O	7	P18	G2	K1	302
GND	-	P19	GND*	GND*	-
V_{CCO}	7	-	V_{CCO} Bank 7*	V_{CCO} Bank 7*	-
I/O, V_{REF}	7	P20	H3	K3	305
I/O	7	P21	G4	K4	308
I/O	7	-	H2	L6	311

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
I/O	7	P22	G5	L1	314
I/O	7	-	-	L5	317
I/O	7	P23	H4	L4	320
I/O, IRDY(1)	7	P24	G1	L3	323
GND	-	P25	GND*	GND*	-
V_{CCO}	7	P26	V_{CCO} Bank 7*	V_{CCO} Bank 7*	-
V_{CCO}	6	P26	V_{CCO} Bank 6*	V_{CCO} Bank 6*	-
I/O, TRDY(1)	6	P27	J2	M1	326
V_{CCINT}	-	P28	V_{CCINT}*	V_{CCINT}*	-
I/O	6	-	-	M6	332
I/O	6	P29	H1	M3	335
I/O	6	-	J4	M4	338
I/O	6	P30	J1	M5	341
I/O, V_{REF}	6	P31	J3	N2	344
V_{CCO}	6	-	V_{CCO} Bank 6*	V_{CCO} Bank 6*	-
GND	-	P32	GND*	GND*	-
I/O	6	P33	K5	N3	347
I/O	6	P34	K2	N4	350
I/O	6	-	-	N5	356
I/O	6	P35	K1	P2	359
I/O	6	-	K3	P4	362
I/O	6	-	-	R1	365
I/O	6	P36	L1	P3	371
I/O	6	P37	L2	R2	374
V_{CCINT}	-	P38	V_{CCINT}*	V_{CCINT}*	-
V_{CCO}	6	P39	V_{CCO} Bank 6*	V_{CCO} Bank 6*	-
GND	-	P40	GND*	GND*	-
I/O	6	P41	K4	T1	377
I/O, V_{REF}	6	P42	M1	R4	380
I/O	6	-	-	T2	383
I/O	6	P43	L4	U1	386
I/O	6	-	M2	R5	389
I/O	6	-	-	V1	392
I/O	6	-	-	T5	395
I/O	6	P44	L3	U2	398
I/O, V_{REF}	6	P45	N1	T3	401
V_{CCO}	6	-	V_{CCO} Bank 6*	V_{CCO} Bank 6*	-
GND	-	-	GND*	GND*	-

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
I/O	6	P46	P1	T4	404
I/O	6	-	L5	W1	407
I/O	6	-	-	V2	410
I/O	6	-	-	U4	413
I/O	6	P47	N2	Y1	416
GND	-	-	GND*	GND*	-
I/O	6	-	M4	W2	419
I/O	6	-	-	V3	422
I/O	6	-	-	V4	425
I/O	6	P48	R1	Y2	428
I/O	6	P49	M3	W3	431
M1	-	P50	P2	U5	434
GND	-	P51	GND*	GND*	-
M0	-	P52	N3	AB2	435
V_{CCO}	6	P53	V_{CCO} Bank 6*	V_{CCO} Bank 6*	-
V_{CCO}	5	P53	V_{CCO} Bank 5*	V_{CCO} Bank 5*	-
M2	-	P54	R3	Y4	436
I/O	5	-	-	W5	443
I/O	5	-	-	AB3	446
I/O	5	-	N5	V7	449
GND	-	-	GND*	GND*	-
I/O	5	P57	T2	Y6	452
I/O	5	-	-	AA4	455
I/O	5	-	-	AB4	458
I/O	5	-	P5	W6	461
I/O	5	P58	T3	Y7	464
GND	-	-	GND*	GND*	-
V_{CCO}	5	-	V_{CCO} Bank 5*	V_{CCO} Bank 5*	-
I/O, V_{REF}	5	P59	T4	AA5	467
I/O	5	P60	M6	AB5	470
I/O	5	-	-	V8	473
I/O	5	-	-	AA6	476
I/O	5	-	T5	AB6	479
I/O	5	P61	N6	AA7	482
I/O	5	-	-	W7	485
I/O, V_{REF}	5	P62	R5	W8	488
I/O	5	P63	P6	Y8	491
GND	-	P64	GND*	GND*	-

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
V_{CCO}	5	P65	V_{CCO} Bank 5*	V_{CCO} Bank 5*	-
V_{CCINT}	-	P66	V_{CCINT}*	V_{CCINT}*	-
I/O	5	P67	R6	AA8	494
I/O	5	P68	M7	V9	497
I/O	5	-	-	W9	503
I/O	5	-	-	AB9	506
I/O	5	P69	N7	Y9	509
I/O	5	-	-	V10	512
I/O	5	P70	T6	W10	518
I/O	5	P71	P7	AB10	521
GND	-	P72	GND*	GND*	-
V_{CCO}	5	-	V_{CCO} Bank 5*	V_{CCO} Bank 5*	-
I/O, V_{REF}	5	P73	P8	Y10	524
I/O	5	P74	R7	V11	527
I/O	5	-	T7	W11	530
I/O	5	P75	T8	AB11	533
I/O	5	-	-	U11	536
V_{CCINT}	-	P76	V_{CCINT}*	V_{CCINT}*	-
I, GCK1	5	P77	R8	Y11	545
V_{CCO}	5	P78	V_{CCO} Bank 5*	V_{CCO} Bank 5*	-
V_{CCO}	4	P78	V_{CCO} Bank 4*	V_{CCO} Bank 4*	-
GND	-	P79	GND*	GND*	-
I, GCK0	4	P80	N8	W12	546
I/O	4	P81	N9	U12	550
I/O	4	-	-	V12	553
I/O	4	P82	R9	Y12	556
I/O	4	-	N10	AA12	559
I/O	4	P83	T9	AB13	562
I/O, V_{REF}	4	P84	P9	AA13	565
V_{CCO}	4	-	V_{CCO} Bank 4*	V_{CCO} Bank 4*	-
GND	-	P85	GND*	GND*	-
I/O	4	P86	M10	Y13	568
I/O	4	P87	R10	V13	571
I/O	4	-	-	W14	577
I/O	4	P88	P10	AA14	580
I/O	4	-	-	V14	583
I/O	4	-	-	Y14	586
I/O	4	P89	T10	AB15	592

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
I/O	4	P90	R11	AA15	595
V_{CCINT}	-	P91	V_{CCINT}*	V_{CCINT}*	-
V_{CCO}	4	P92	V_{CCO} Bank 4*	V_{CCO} Bank 4*	-
GND	-	P93	GND*	GND*	-
I/O	4	P94	M11	Y15	598
I/O, V_{REF}	4	P95	T11	AB16	601
I/O	4	-	-	AB17	604
I/O	4	P96	N11	V15	607
I/O	4	-	R12	Y16	610
I/O	4	-	-	AA17	613
I/O	4	-	-	W16	616
I/O	4	P97	P11	AB18	619
I/O, V_{REF}	4	P98	T12	AB19	622
V_{CCO}	4	-	V_{CCO} Bank 4*	V_{CCO} Bank 4*	-
GND	-	-	GND*	GND*	-
I/O	4	P99	T13	Y17	625
I/O	4	-	N12	V16	628
I/O	4	-	-	AA18	631
I/O	4	-	-	W17	634
I/O	4	P100	R13	AB20	637
GND	-	-	GND*	GND*	-
I/O	4	-	P12	AA19	640
I/O	4	-	-	V17	643
I/O	4	-	-	Y18	646
I/O	4	P101	P13	AA20	649
I/O	4	P102	T14	W18	652
GND	-	P103	GND*	GND*	-
DONE	3	P104	R14	Y19	655
V_{CCO}	4	P105	V_{CCO} Bank 4*	V_{CCO} Bank 4*	-
V_{CCO}	3	P105	V_{CCO} Bank 3*	V_{CCO} Bank 3*	-
$\overline{PROGRAM}$	-	P106	P15	W20	658
I/O ($\overline{INIT}$)	3	P107	N15	V19	659
I/O (D7)	3	P108	N14	Y21	662
I/O	3	-	-	V20	665
I/O	3	-	-	AA22	668
I/O	3	-	T15	W21	671
GND	-	-	GND*	GND*	-
I/O	3	P109	M13	U20	674

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
I/O	3	-	-	U19	677
I/O	3	-	-	V21	680
I/O	3	-	R16	T18	683
I/O	3	P110	M14	W22	686
GND	-	-	GND*	GND*	-
V_{CCO}	3	-	V_{CCO} Bank 3*	V_{CCO} Bank 3*	-
I/O, V_{REF}	3	P111	L14	U21	689
I/O	3	P112	M15	T20	692
I/O	3	-	-	T19	695
I/O	3	-	-	V22	698
I/O	3	-	L12	T21	701
I/O	3	P113	P16	R18	704
I/O	3	-	-	U22	707
I/O, V_{REF}	3	P114	L13	R19	710
I/O (D6)	3	P115	N16	T22	713
GND	-	P116	GND*	GND*	-
V_{CCO}	3	P117	V_{CCO} Bank 3*	V_{CCO} Bank 3*	-
V_{CCINT}	-	P118	V_{CCINT}*	V_{CCINT}*	-
I/O (D5)	3	P119	M16	R21	716
I/O	3	P120	K14	P18	719
I/O	3	-	-	P19	725
I/O	3	-	L16	P20	728
I/O	3	P121	K13	P21	731
I/O	3	-	-	N19	734
I/O	3	P122	L15	N18	740
I/O	3	P123	K12	N20	743
GND	-	P124	GND*	GND*	-
V_{CCO}	3	-	V_{CCO} Bank 3*	V_{CCO} Bank 3*	-
I/O, V_{REF}	3	P125	K16	N21	746
I/O (D4)	3	P126	J16	N22	749
I/O	3	-	J14	M19	752
I/O	3	P127	K15	M20	755
I/O	3	-	-	M18	758
V_{CCINT}	-	P128	V_{CCINT}*	V_{CCINT}*	-
I/O, TRDY(1)	3	P129	J15	M22	764
V_{CCO}	3	P130	V_{CCO} Bank 3*	V_{CCO} Bank 3*	-
V_{CCO}	2	P130	V_{CCO} Bank 2*	V_{CCO} Bank 2*	-
GND	-	P131	GND*	GND*	-

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
I/O, IRDY(1)	2	P132	H16	L20	767
I/O	2	P133	H14	L17	770
I/O	2	-	-	L18	773
I/O	2	P134	H15	L21	776
I/O	2	-	J13	L22	779
I/O (D3)	2	P135	G16	K20	782
I/O, V_{REF}	2	P136	H13	K21	785
V_{CCO}	2	-	V_{CCO} Bank 2*	V_{CCO} Bank 2*	-
GND	-	P137	GND*	GND*	-
I/O	2	P138	G14	K22	788
I/O	2	P139	G15	J21	791
I/O	2	-	-	J20	797
I/O	2	P140	G12	J18	800
I/O	2	-	F16	J22	803
I/O	2	-	-	J19	806
I/O	2	P141	G13	H19	812
I/O (D2)	2	P142	F15	H20	815
V_{CCINT}	-	P143	V_{CCINT}*	V_{CCINT}*	-
V_{CCO}	2	P144	V_{CCO} Bank 2*	V_{CCO} Bank 2*	-
GND	-	P145	GND*	GND*	-
I/O (D1)	2	P146	E16	H22	818
I/O, V_{REF}	2	P147	F14	H18	821
I/O	2	-	-	G21	824
I/O	2	P148	D16	G18	827
I/O	2	-	F12	G20	830
I/O	2	-	-	G19	833
I/O	2	-	-	F22	836
I/O	2	P149	E15	F19	839
I/O, V_{REF}	2	P150	F13	F21	842
V_{CCO}	2	-	V_{CCO} Bank 2*	V_{CCO} Bank 2*	-
GND	-	-	GND*	GND*	-
I/O	2	P151	E14	F20	845
I/O	2	-	C16	F18	848
I/O	2	-	-	E22	851
I/O	2	-	-	E21	854
I/O	2	P152	E13	D22	857
GND	-	-	GND*	GND*	-
I/O	2	-	B16	E20	860
I/O	2	-	-	D21	863

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					
Function	Bank	PQ208	FG256	FG456	Bndry Scan
I/O	2	-	-	C22	866
I/O (DIN, D0)	2	P153	D14	D20	869
I/O (DOUT, BUSY)	2	P154	C15	C21	872
CCLK	2	P155	D15	B22	875
V_{CCO}	2	P156	V_{CCO} Bank 2*	V_{CCO} Bank 2*	-
V_{CCO}	1	P156	V_{CCO} Bank 1*	V_{CCO} Bank 1*	-
TDO	2	P157	B14	A21	-
GND	-	P158	GND*	GND*	-
TDI	-	P159	A15	B20	-
I/O ($\overline{CS}$)	1	P160	B13	C19	0
I/O ($\overline{WRITE}$)	1	P161	C13	A20	3
I/O	1	-	-	B19	6
I/O	1	-	-	C18	9
I/O	1	-	C12	D17	12
GND	-	-	GND*	GND*	-
I/O	1	P162	A14	A19	15
I/O	1	-	-	B18	18
I/O	1	-	-	E16	21
I/O	1	-	D12	C17	24
I/O	1	P163	B12	D16	27
GND	-	-	GND*	GND*	-
V_{CCO}	1	-	V_{CCO} Bank 1*	V_{CCO} Bank 1*	-
I/O, V_{REF}	1	P164	C11	A18	30
I/O	1	P165	A13	B17	33
I/O	1	-	-	E15	36
I/O	1	-	-	A17	39
I/O	1	-	D11	D15	42
I/O	1	P166	A12	C16	45
I/O	1	-	-	D14	48
I/O, V_{REF}	1	P167	E11	E14	51
I/O	1	P168	B11	A16	54
GND	-	P169	GND*	GND*	-
V_{CCO}	1	P170	V_{CCO} Bank 1*	V_{CCO} Bank 1*	-
V_{CCINT}	-	P171	V_{CCINT}*	V_{CCINT}*	-
I/O	1	P172	A11	C15	57
I/O	1	P173	C10	B15	60
I/O	1	-	-	A15	66
I/O	1	-	-	F12	69

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					Bndry
Function	Bank	PQ208	FG256	FG456	Scan
I/O	1	P174	B10	C14	72
I/O	1	-	-	B14	75
I/O	1	P175	D10	D13	81
I/O	1	P176	A10	C13	84
GND	-	P177	GND*	GND*	-
V_{CCO}	1	-	V_{CCO} Bank 1*	V_{CCO} Bank 1*	-
I/O, V_{REF}	1	P178	B9	B13	87
I/O	1	P179	E10	E12	90
I/O	1	-	A9	B12	93
I/O	1	P180	D9	D12	96
I/O	1	-	-	C12	99
I/O	1	P181	A8	D11	102
I, GCK2	1	P182	C9	A11	108
GND	-	P183	GND*	GND*	-
V_{CCO}	1	P184	V_{CCO} Bank 1*	V_{CCO} Bank 1*	-
V_{CCO}	0	P184	V_{CCO} Bank 0*	V_{CCO} Bank 0*	-
I, GCK3	0	P185	B8	C11	109
V_{CCINT}	-	P186	V_{CCINT}*	V_{CCINT}*	-
I/O	0	-	-	E11	116
I/O	0	P187	A7	A10	119
I/O	0	-	D8	B10	122
I/O	0	P188	A6	C10	125
I/O, V_{REF}	0	P189	B7	A9	128
V_{CCO}	0	-	V_{CCO} Bank 0*	V_{CCO} Bank 0*	-
GND	-	P190	GND*	GND*	-
I/O	0	P191	C8	B9	131
I/O	0	P192	D7	E10	134
I/O	0	-	-	D10	140
I/O	0	P193	E7	A8	143
I/O	0	-	-	D9	146
I/O	0	-	-	B8	149
I/O	0	P194	C7	E9	155
I/O	0	P195	B6	A7	158

XC2S150 Device Pinouts *(Continued)*

XC2S150 Pad Name					Bndry
Function	Bank	PQ208	FG256	FG456	Scan
V_{CCINT}	-	P196	V_{CCINT}*	V_{CCINT}*	-
V_{CCO}	0	P197	V_{CCO} Bank 0*	V_{CCO} Bank 0*	-
GND	-	P198	GND*	GND*	-
I/O	0	P199	A5	B7	161
I/O, V_{REF}	0	P200	C6	E8	164
I/O	0	-	-	D8	167
I/O	0	P201	B5	C7	170
I/O	0	-	D6	D7	173
I/O	0	-	-	B6	176
I/O	0	-	-	A5	179
I/O	0	P202	A4	D6	182
I/O, V_{REF}	0	P203	B4	C6	185
V_{CCO}	0	-	V_{CCO} Bank 0*	V_{CCO} Bank 0*	-
GND	-	-	GND*	GND*	-
I/O	0	P204	E6	B5	188
I/O	0	-	D5	E7	191
I/O	0	-	-	A4	194
I/O	0	-	-	E6	197
I/O	0	P205	A3	B4	200
GND	-	-	GND*	GND*	-
I/O	0	-	C5	A3	203
I/O	0	-	-	B3	206
I/O	0	-	-	D5	209
I/O	0	P206	B3	C5	212
TCK	-	P207	C4	C4	-
V_{CCO}	0	P208	V_{CCO} Bank 0*	V_{CCO} Bank 0*	-
V_{CCO}	7	P208	V_{CCO} Bank 7*	V_{CCO} Bank 7*	-

04/18/01

Notes:
1. IRDY and TRDY can only be accessed when using Xilinx PCI cores.
2. Pads labelled GND*, V_{CCINT}*, V_{CCO} Bank 0*, V_{CCO} Bank 1*, V_{CCO} Bank 2*, V_{CCO} Bank 3*, V_{CCO} Bank 4*, V_{CCO} Bank 5*, V_{CCO} Bank 6*, V_{CCO} Bank 7* are internally bonded to independent ground or power planes within the package.

Additional XC2S150 Package Pins

PQ208

Not Connected Pins					
P55	P56	-	-	-	-

11/02/00

FG256

V_{CCINT} Pins					
C3	C14	D4	D13	E5	E12
M5	M12	N4	N13	P3	P14
V_{CCO} Bank 0 Pins					
E8	F8	-	-	-	-
V_{CCO} Bank 1 Pins					
E9	F9	-	-	-	-
V_{CCO} Bank 2 Pins					
H11	H12	-	-	-	-
V_{CCO} Bank 3 Pins					
J11	J12	-	-	-	-
V_{CCO} Bank 4 Pins					
L9	M9	-	-	-	-
V_{CCO} Bank 5 Pins					
L8	M8	-	-	-	-
V_{CCO} Bank 6 Pins					
J5	J6	-	-	-	-
V_{CCO} Bank 7 Pins					
H5	H6	-	-	-	-
GND Pins					
A1	A16	B2	B15	F6	F7
F10	F11	G6	G7	G8	G9
G10	G11	H7	H8	H9	H10
J7	J8	J9	J10	K6	K7
K8	K9	K10	K11	L6	L7
L10	L11	R2	R15	T1	T16
Not Connected Pins					
P4	R4	-	-	-	-

11/02/00

Additional XC2S150 Package Pins *(Continued)*

FG456

V_{CCINT} Pins					
E5	E18	F6	F17	G7	G8
G9	G14	G15	G16	H7	H16
J7	J16	P7	P16	R7	R16
T7	T8	T9	T14	T15	T16
U6	U17	V5	V18	-	-
V_{CCO} BANK 0 Pins					
F7	F8	F9	F10	G10	G11
V_{CCO} Bank 1 Pins					
F13	F14	F15	F16	G12	G13
V_{CCO} Bank 2 Pins					
G17	H17	J17	K16	K17	L16
V_{CCO} Bank 3 Pins					
M16	N16	N17	P17	R17	T17
V_{CCO} Bank 4 Pins					
T12	T13	U13	U14	U15	U16
V_{CCO} Bank 5 Pins					
T10	T11	U7	U8	U9	U10
V_{CCO} Bank 6 Pins					
M7	N6	N7	P6	R6	T6
V_{CCO} Bank 7 Pins					
G6	H6	J6	K6	K7	L7
GND Pins					
A1	A22	B2	B21	C3	C20
J9	J10	J11	J12	J13	J14
K9	K10	K11	K12	K13	K14
L9	L10	L11	L12	L13	L14
M9	M10	M11	M12	M13	M14
N9	N10	N11	N12	N13	N14
P9	P10	P11	P12	P13	P14
Y3	Y20	AA2	AA21	AB1	AB22
Not Connected Pins					
A2	A6	A12	A13	A14	B11
B16	C2	C8	C9	D1	D4
D18	D19	E13	E17	E19	F11
G2	G22	H21	J1	J4	K2
K18	K19	L2	L19	M2	M17
M21	N1	P1	P5	P22	R3
R20	R22	U3	U18	V6	W4
W13	W15	W19	Y5	Y22	AA1
AA3	AA9	AA10	AA11	AA16	AB7
AB8	AB12	AB14	AB21	-	-

11/02/00

ضمیمه۲ :

Spartan-II 2.5V FPGA Family: DC and Switching Characteristics

DS001-3 (v2.7) September 3, 2003 **Product Specification**

Definition of Terms

In this document, some specifications may be designated as Advance or Preliminary. These terms are defined as follows:

Advance: Initial estimates based on simulation and/or extrapolation from other speed grades, devices, or families. Values are subject to change. Use as estimates, not for production.

Preliminary: Based on preliminary characterization. Further changes are not expected.

Unmarked: Specifications not identified as either Advance or Preliminary are to be considered Final.

Except for pin-to-pin input and output parameters, the AC parameter delay specifications included in this document are derived from measuring internal test patterns. All limits are representative of worst-case supply voltage and junction temperature conditions. Typical numbers are based on measurements taken at a nominal V_{CCINT} level of 2.5V and a junction temperature of 25°C. The parameters included are common to popular designs and typical applications. **All specifications are subject to change without notice.**

DC Specifications

Absolute Maximum Ratings(1)

Symbol	Description		Min	Max	Units
V_{CCINT}	Supply voltage relative to GND(2)		–0.5	3.0	V
V_{CCO}	Supply voltage relative to GND(2)		–0.5	4.0	V
V_{REF}	Input reference voltage		–0.5	3.6	V
V_{IN}	Input voltage relative to GND(3)	5V tolerant I/O(4)	–0.5	5.5	V
		No 5V tolerance(5)	–0.5	V_{CCO} + 0.5	V
V_{TS}	Voltage applied to 3-state output	5V tolerant I/O(4)	–0.5	5.5	V
		No 5V tolerance(5)	–0.5	V_{CCO} + 0.5	V
T_{STG}	Storage temperature (ambient)		–65	+150	°C
T_J	Junction temperature		-	+125	°C

Notes:

1. Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.
2. Power supplies may turn on in any order.
3. V_{IN} should not exceed V_{CCO} by more than 3.6V over extended periods of time (e.g., longer than a day).
4. Spartan-II I/Os are 5V Tolerant whenever the LVTTL, LVCMOS2, or PCI33_5 signal standard has been selected. With 5V Tolerant I/Os selected, the Maximum DC overshoot must be limited to either +5.5V or 10 mA, and undershoot must be limited to either –0.5V or 10 mA, whichever is easier to achieve. The Maximum AC conditions are as follows: The device pins may undershoot to –2.0V or overshoot to +7.0V, provided this over/undershoot lasts no more than 11 ns with a forcing current no greater than 100 mA.
5. Without 5V Tolerant I/Os selected, the Maximum DC overshoot must be limited to either V_{CCO} + 0.5V or 10 mA, and undershoot must be limited to –0.5V or 10 mA, whichever is easier to achieve. The Maximum AC conditions are as follows: The device pins may undershoot to –2.0V or overshoot to V_{CCO} + 2.0V, provided this over/undershoot lasts no more than 11 ns with a forcing current no greater than 100 mA.
6. For soldering guidelines, see the Packaging Information on the Xilinx web site: http://www.xilinx.com/publications/products/packaging/index.htm

Recommended Operating Conditions

Symbol	Description		Min	Max	Units
T_J	Junction temperature(1)	Commercial	0	85	°C
		Industrial	–40	100	°C
V_{CCINT}	Supply voltage relative to GND(2,5)	Commercial	2.5 – 5%	2.5 + 5%	V
		Industrial	2.5 – 5%	2.5 + 5%	V
V_{CCO}	Supply voltage relative to GND(3,5)	Commercial	1.4	3.6	V
		Industrial	1.4	3.6	V
T_{IN}	Input signal transition time(4)		-	250	ns

Notes:
1. At junction temperatures above those listed as Operating Conditions, all delay parameters increase by 0.35% per °C.
2. Functional operation is guaranteed down to a minimum V_{CCINT} of 2.25V (Nominal V_{CCINT} – 10%). For every 50 mV reduction in V_{CCINT} below 2.375V (nominal V_{CCINT} – 5%), all delay parameters increase by 3%.
3. Minimum and maximum values for V_{CCO} vary according to the I/O standard selected.
4. Input and output measurement threshold is ~50% of V_{CCO}.
5. Supply voltages may be applied in any order desired.

DC Characteristics Over Operating Conditions

Symbol	Description			Min	Typ	Max	Units
V_{DRINT}	Data Retention V_{CCINT} voltage (below which configuration data may be lost)			2.0	-	-	V
V_{DRIO}	Data Retention V_{CCO} voltage (below which configuration data may be lost)			1.2	-	-	V
I_{CCINTQ}	Quiescent V_{CCINT} supply current(1)	XC2S15	Commercial	-	10	30	mA
			Industrial	-	10	60	mA
		XC2S30	Commercial	-	10	30	mA
			Industrial	-	10	60	mA
		XC2S50	Commercial	-	12	50	mA
			Industrial	-	12	100	mA
		XC2S100	Commercial	-	12	50	mA
			Industrial	-	12	100	mA
		XC2S150	Commercial	-	15	50	mA
			Industrial	-	15	100	mA
		XC2S200	Commercial	-	15	75	mA
			Industrial	-	15	150	mA
I_{CCOQ}	Quiescent V_{CCO} supply current(1)			-	-	2	mA
I_{REF}	V_{REF} current per V_{REF} pin			-	-	20	µA
I_L	Input or output leakage current(2)			–10	-	+10	µA
C_{IN}	Input capacitance (sample tested)	VQ, CS, TQ, PQ, FG packages		-	-	8	pF
I_{RPU}	Pad pull-up (when selected) @ V_{IN} = 0V, V_{CCO} = 3.3V (sample tested)(3)			-	-	0.25	mA
I_{RPD}	Pad pull-down (when selected) @ V_{IN} = 3.6V (sample tested)(3)			-	-	0.15	mA

Notes:
1. With no output current loads, no active input pull-up resistors, all I/O pins 3-stated and floating.
2. The I/O leakage current specification applies only when the V_{CCINT} and V_{CCO} supply voltages have reached their respective minimum Recommended Operating Conditions.
3. Internal pull-up and pull-down resistors guarantee valid logic levels at unconnected input pins. These pull-up and pull-down resistors do not provide valid logic levels when input pins are connected to other circuits.

Power-On Requirements

Spartan-II FPGAs require that a minimum supply current I_{CCPO} be provided to the V_{CCINT} lines for a successful power-on. If more current is available, the FPGA can consume more than I_{CCPO} minimum, though this cannot adversely affect reliability.

A maximum limit for I_{CCPO} is not specified. Therefore the use of foldback/crowbar supplies and fuses deserves special attention. In these cases, limit the I_{CCPO} current to a level below the trip point for over-current protection in order to avoid inadvertently shutting down the supply.

Symbol	Description	Conditions		New Requirements[1] For Devices with Date Code 0321 or Later		Old Requirements[1] For Devices with Date Code before 0321		Units
		Junction Temperature[2]	Device Temperature Grade	Min	Max	Min	Max	
I_{CCPO}[3]	Total V_{CCINT} supply current required during power-on	$-40°C \le T_J < -20°C$	Industrial	1.50	-	2.00	-	A
		$-20°C \le T_J < 0°C$	Industrial	1.00	-	2.00	-	A
		$0°C \le T_J \le 85°C$	Commercial	0.25	-	0.50	-	A
		$85°C < T_J \le 100°C$	Industrial	0.50	-	0.50	-	A
T_{CCPO}[4,5]	V_{CCINT} ramp time	$-40°C \le T_J \le 100°C$	All	-	50	-	50	ms

Notes:

1. The date code is printed on the top of the device's package. See the **Device Part Marking** section in Module 1.
2. The expected T_J range for the design determines the I_{CCPO} minimum requirement. Use the applicable ranges in the junction temperature column to find the associated current values in the appropriate new or old requirements column according to the date code. Then choose the highest of these current values to serve as the minimum I_{CCPO} requirement that must be met. For example, if the junction temperature for a given design is $-25°C \le T_J \le 75°C$, then the new minimum I_{CCPO} requirement is 1.5A. If $5°C \le T_J \le 90°C$, then the new minimum I_{CCPO} requirement is 0.5A.
3. The I_{CCPO} requirement applies for a brief time (commonly only a few milliseconds) when V_{CCINT} ramps from 0 to 2.5V.
4. The ramp time is measured from GND to V_{CCINT} max on a fully loaded board.
5. During power-on, the V_{CCINT} ramp must increase steadily in voltage with no dips.
6. For more information on designing to meet the power-on specifications, refer to the application note XAPP450 "Power-On Current Requirements for the Spartan-II and Spartan-IIE Families"

DC Input and Output Levels

Values for V_{IL} and V_{IH} are recommended input voltages. Values for V_{OL} and V_{OH} are guaranteed output voltages over the recommended operating conditions. Only selected standards are tested. These are chosen to ensure that all standards meet their specifications. The selected standards are tested at minimum V_{CCO} with the respective I_{OL} and I_{OH} currents shown. Other standards are sample tested.

Input/Output Standard	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}
	V, Min	V, Max	V, Min	V, Max	V, Max	V, Min	mA	mA
LVTTL[1]	–0.5	0.8	2.0	5.5	0.4	2.4	24	–24
LVCMOS2	–0.5	0.7	1.7	5.5	0.4	1.9	12	–12
PCI, 3.3V	–0.5	44% V_{CCINT}	60% V_{CCINT}	V_{CCO} + 0.5	10% V_{CCO}	90% V_{CCO}	Note (2)	Note (2)
PCI, 5.0V	–0.5	0.8	2.0	5.5	0.55	2.4	Note (2)	Note (2)
GTL	–0.5	V_{REF} – 0.05	V_{REF} + 0.05	3.6	0.4	N/A	40	N/A
GTL+	–0.5	V_{REF} – 0.1	V_{REF} + 0.1	3.6	0.6	N/A	36	N/A
HSTL I	–0.5	V_{REF} – 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCO} – 0.4	8	–8
HSTL III	–0.5	V_{REF} – 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCO} – 0.4	24	–8
HSTL IV	–0.5	V_{REF} – 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCO} – 0.4	48	–8
SSTL3 I	–0.5	V_{REF} – 0.2	V_{REF} + 0.2	3.6	V_{REF} – 0.6	V_{REF} + 0.6	8	–8
SSTL3 II	–0.5	V_{REF} – 0.2	V_{REF} + 0.2	3.6	V_{REF} – 0.8	V_{REF} + 0.8	16	–16
SSTL2 I	–0.5	V_{REF} – 0.2	V_{REF} + 0.2	3.6	V_{REF} – 0.6	V_{REF} + 0.6	7.6	–7.6
SSTL2 II	–0.5	V_{REF} – 0.2	V_{REF} + 0.2	3.6	V_{REF} – 0.8	V_{REF} + 0.8	15.2	–15.2

Input/Output Standard	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}
	V, Min	V, Max	V, Min	V, Max	V, Max	V, Min	mA	mA
CTT	–0.5	V_{REF} – 0.2	V_{REF} + 0.2	3.6	V_{REF} – 0.4	V_{REF} + 0.4	8	–8
AGP	–0.5	V_{REF} – 0.2	V_{REF} + 0.2	3.6	10% V_{CCO}	90% V_{CCO}	Note (2)	Note (2)

Notes:
1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Tested according to the relevant specifications.

Switching Characteristics

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation netlist. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all Spartan-II devices unless otherwise noted.

Global Clock Input to Output Delay for LVTTL, *with* DLL (Pin-to-Pin)(1)

Symbol	Description	Device	Speed Grade			Units
			All	-6	-5	
			Min	Max	Max	
$T_{ICKOFDLL}$	Global clock input to output delay using output flip-flop for LVTTL, 12 mA, fast slew rate, *with* DLL.	All		2.9	3.3	ns

Notes:
1. Listed above are representative values where one global clock input drives one vertical clock line in each accessible column, and where all accessible IOB and CLB flip-flops are clocked by the global clock net.
2. Output timing is measured at 1.4V with 35 pF external capacitive load for LVTTL. The 35 pF load does not apply to the Min values. For other I/O standards and different loads, see the tables Constants for Calculating T_{IOOP} and Delay Measurement Methodology, page 10.
3. DLL output jitter is already included in the timing calculation.
4. For data *output* with different standards, adjust delays with the values shown in IOB Output Delay Adjustments for Different Standards, page 9. For a global clock input with standards other than LVTTL, adjust delays with values from the I/O Standard Global Clock Input Adjustments, page 11.

Global Clock Input to Output Delay for LVTTL, *without* DLL (Pin-to-Pin)(1)

Symbol	Description	Device	Speed Grade			Units
			All	-6	-5	
			Min	Max	Max	
T_{ICKOF}	Global clock input to output delay using output flip-flop for LVTTL, 12 mA, fast slew rate, *without* DLL.	XC2S15		4.5	5.4	ns
		XC2S30		4.5	5.4	ns
		XC2S50		4.5	5.4	ns
		XC2S100		4.6	5.5	ns
		XC2S150		4.6	5.5	ns
		XC2S200		4.7	5.6	ns

Notes:
1. Listed above are representative values where one global clock input drives one vertical clock line in each accessible column, and where all accessible IOB and CLB flip-flops are clocked by the global clock net.
2. Output timing is measured at 1.4V with 35 pF external capacitive load for LVTTL. The 35 pF load does not apply to the Min values. For other I/O standards and different loads, see the tables Constants for Calculating T_{IOOP} and Delay Measurement Methodology, page 10.
3. For data *output* with different standards, adjust delays with the values shown in IOB Output Delay Adjustments for Different Standards, page 9. For a global clock input with standards other than LVTTL, adjust delays with values from the I/O Standard Global Clock Input Adjustments, page 11.

Global Clock Setup and Hold for LVTTL Standard, *with* DLL (Pin-to-Pin)

Symbol	Description	Device	Speed Grade		Units
			-6	-5	
			Min	Min	
T_{PSDLL} / T_{PHDLL}	Input setup and hold time relative to global clock input signal for LVTTL standard, no delay, IFF,(1) with DLL	All	1.7 / 0	1.9 / 0	ns

Notes:
1. IFF = Input Flip-Flop or Latch
2. Setup time is measured relative to the Global Clock input signal with the fastest route and the lightest load. Hold time is measured relative to the Global Clock input signal with the slowest route and heaviest load.
3. DLL output jitter is already included in the timing calculation.
4. A zero hold time listing indicates no hold time or a negative hold time.
5. For data input with different standards, adjust the setup time delay by the values shown in IOB Input Delay Adjustments for Different Standards, page 7. For a global clock input with standards other than LVTTL, adjust delays with values from the I/O Standard Global Clock Input Adjustments, page 11.

Global Clock Setup and Hold for LVTTL Standard, *without* DLL (Pin-to-Pin)

Symbol	Description	Device	Speed Grade		Units
			-6	-5	
			Min	Min	
T_{PSFD} / T_{PHFD}	Input setup and hold time relative to global clock input signal for LVTTL standard, no delay, IFF,(1) without DLL	XC2S15	2.2 / 0	2.7 / 0	ns
		XC2S30	2.2 / 0	2.7 / 0	ns
		XC2S50	2.2 / 0	2.7 / 0	ns
		XC2S100	2.3 / 0	2.8 / 0	ns
		XC2S150	2.4 / 0	2.9 / 0	ns
		XC2S200	2.4 / 0	3.0 / 0	ns

Notes:
1. IFF = Input Flip-Flop or Latch
2. Setup time is measured relative to the Global Clock input signal with the fastest route and the lightest load. Hold time is measured relative to the Global Clock input signal with the slowest route and heaviest load.
3. A zero hold time listing indicates no hold time or a negative hold time.
4. For data input with different standards, adjust the setup time delay by the values shown in IOB Input Delay Adjustments for Different Standards, page 7. For a global clock input with standards other than LVTTL, adjust delays with values from the I/O Standard Global Clock Input Adjustments, page 11.

IOB Input Switching Characteristics(1)

Input delays associated with the pad are specified for LVTTL levels. For other standards, adjust the delays with the values shown in IOB Input Delay Adjustments for Different Standards, page 7.

Symbol	Description	Device	Speed Grade				Units
			-6		-5		
			Min	Max	Min	Max	
Propagation Delays							
T_{IOPI}	Pad to I output, no delay	All	-	0.8	-	1.0	ns
T_{IOPID}	Pad to I output, with delay	All	-	1.5	-	1.8	ns
T_{IOPLI}	Pad to output IQ via transparent latch, no delay	All	-	1.7	-	2.0	ns
T_{IOPLID}	Pad to output IQ via transparent latch, with delay	XC2S15	-	3.8	-	4.5	ns
		XC2S30	-	3.8	-	4.5	ns
		XC2S50	-	3.8	-	4.5	ns
		XC2S100	-	3.8	-	4.5	ns
		XC2S150	-	4.0	-	4.7	ns
		XC2S200	-	4.0	-	4.7	ns
Sequential Delays							
T_{IOCKIQ}	Clock CLK to output IQ	All	-	0.7	-	0.8	ns
Setup/Hold Times with Respect to Clock CLK(2)							
T_{IOPICK} / T_{IOICKP}	Pad, no delay	All	1.7 / 0	-	1.9 / 0	-	ns
$T_{IOPICKD}$ / $T_{IOICKPD}$	Pad, with delay(1)	XC2S15	3.8 / 0	-	4.4 / 0	-	ns
		XC2S30	3.8 / 0	-	4.4 / 0	-	ns
		XC2S50	3.8 / 0	-	4.4 / 0	-	ns
		XC2S100	3.8 / 0	-	4.4 / 0	-	ns
		XC2S150	3.9 / 0	-	4.6 / 0	-	ns
		XC2S200	3.9 / 0	-	4.6 / 0	-	ns
$T_{IOICECK}$ / $T_{IOCKICE}$	ICE input	All	0.9 / 0.01	-	0.9 / 0.01	-	ns
Set/Reset Delays							
$T_{IOSRCKI}$	SR input (IFF, synchronous)	All	-	1.1	-	1.2	ns
T_{IOSRIQ}	SR input to IQ (asynchronous)	All	-	1.5	-	1.7	ns
T_{GSRQ}	GSR to output IQ	All	-	9.9	-	11.7	ns

Notes:
1. Input timing for LVTTL is measured at 1.4V. For other I/O standards, see the table Delay Measurement Methodology, page 10.
2. A zero hold time listing indicates no hold time or a negative hold time.

IOB Output Delay Adjustments for Different Standards(1)

Output delays terminating at a pad are specified for LVTTL with 12 mA drive and fast slew rate. For other standards, adjust the delays by the values shown. A delay adjusted in this way constitutes a worst-case limit.

Symbol	Description	Standard	Speed Grade		Units
			-6	-5	
Output Delay Adjustments (Adj)					
T_{OLVTTL_S2}	Standard-specific adjustments for output delays terminating at pads (based on standard capacitive load, C_{SL})	LVTTL, Slow, 2 mA	14.2	16.9	ns
T_{OLVTTL_S4}		4 mA	7.2	8.6	ns
T_{OLVTTL_S6}		6 mA	4.7	5.5	ns
T_{OLVTTL_S8}		8 mA	2.9	3.5	ns
T_{OLVTTL_S12}		12 mA	1.9	2.2	ns
T_{OLVTTL_S16}		16 mA	1.7	2.0	ns
T_{OLVTTL_S24}		24 mA	1.3	1.5	ns
T_{OLVTTL_F2}		LVTTL, Fast, 2 mA	12.6	15.0	ns
T_{OLVTTL_F4}		4 mA	5.1	6.1	ns
T_{OLVTTL_F6}		6 mA	3.0	3.6	ns
T_{OLVTTL_F8}		8 mA	1.0	1.2	ns
T_{OLVTTL_F12}		12 mA	0	0	ns
T_{OLVTTL_F16}		16 mA	–0.1	–0.1	ns
T_{OLVTTL_F24}		24 mA	–0.1	–0.2	ns
$T_{OLVCMOS2}$		LVCMOS2	0.2	0.2	ns
T_{OPCI33_3}		PCI, 33 MHz, 3.3V	2.4	2.9	ns
T_{OPCI33_5}		PCI, 33 MHz, 5.0V	2.9	3.5	ns
T_{OPCI66_3}		PCI, 66 MHz, 3.3V	–0.3	–0.4	ns
T_{OGTL}		GTL	0.6	0.7	ns
T_{OGTLP}		GTL+	0.9	1.1	ns
T_{OHSTL_I}		HSTL I	–0.4	–0.5	ns
T_{OHSTL_III}		HSTL III	–0.8	–1.0	ns
T_{OHSTL_IV}		HSTL IV	–0.9	–1.1	ns
T_{OSSTL2_I}		SSTL2 I	–0.4	–0.5	ns
T_{OSSLT2_II}		SSTL2 II	–0.8	–1.0	ns
T_{OSSTL3_I}		SSTL3 I	–0.4	–0.5	ns
T_{OSSTL3_II}		SSTL3 II	–0.9	–1.1	ns
T_{OCTT}		CTT	–0.5	–0.6	ns
T_{OAGP}		AGP	–0.8	–1.0	ns

Notes:
1. Output timing is measured at 1.4V with 35 pF external capacitive load for LVTTL. For other I/O standards and different loads, see the tables Constants for Calculating T_{IOOP} and Delay Measurement Methodology, page 10.

CLB Switching Characteristics

Delays originating at F/G inputs vary slightly according to the input used. The values listed below are worst-case. Precise values are provided by the timing analyzer.

Symbol	Description	Speed Grade				Units
		-6		-5		
		Min	Max	Min	Max	
Combinatorial Delays						
T_{ILO}	4-input function: F/G inputs to X/Y outputs	-	0.6	-	0.7	ns
T_{IF5}	5-input function: F/G inputs to F5 output	-	0.7	-	0.9	ns
T_{IF5X}	5-input function: F/G inputs to X output	-	0.9	-	1.1	ns
T_{IF6Y}	6-input function: F/G inputs to Y output via F6 MUX	-	1.0	-	1.1	ns
T_{F5INY}	6-input function: F5IN input to Y output	-	0.4	-	0.4	ns
T_{IFNCTL}	Incremental delay routing through transparent latch to XQ/YQ outputs	-	0.7	-	0.9	ns
T_{BYYB}	BY input to YB output	-	0.6	-	0.7	ns
Sequential Delays						
T_{CKO}	FF clock CLK to XQ/YQ outputs	-	1.1	-	1.3	ns
T_{CKLO}	Latch clock CLK to XQ/YQ outputs	-	1.2	-	1.5	ns
Setup/Hold Times with Respect to Clock CLK(1)						
T_{ICK} / T_{CKI}	4-input function: F/G inputs	1.3 / 0	-	1.4 / 0	-	ns
T_{IF5CK} / T_{CKIF5}	5-input function: F/G inputs	1.6 / 0	-	1.8 / 0	-	ns
T_{F5INCK} / T_{CKF5IN}	6-input function: F5IN input	1.0 / 0	-	1.1 / 0	-	ns
T_{IF6CK} / T_{CKIF6}	6-input function: F/G inputs via F6 MUX	1.6 / 0	-	1.8 / 0	-	ns
T_{DICK} / T_{CKDI}	BX/BY inputs	0.8 / 0	-	0.8 / 0	-	ns
T_{CECK} / T_{CKCE}	CE input	0.9 / 0	-	0.9 / 0	-	ns
T_{RCK} / T_{CKR}	SR/BY inputs (synchronous)	0.8 / 0	-	0.8 / 0	-	ns
Clock CLK						
T_{CH}	Minimum pulse width, High	-	1.9	-	1.9	ns
T_{CL}	Minimum pulse width, Low	-	1.9	-	1.9	ns
Set/Reset						
T_{RPW}	Minimum pulse width, SR/BY inputs	3.1	-	3.1	-	ns
T_{RQ}	Delay from SR/BY inputs to XQ/YQ outputs (asynchronous)	-	1.1	-	1.3	ns
T_{IOGSRQ}	Delay from GSR to XQ/YQ outputs	-	9.9	-	11.7	ns
F_{TOG}	Toggle frequency (for export control)	-	263	-	263	MHz

Notes:
1. A zero hold time listing indicates no hold time or a negative hold time.

CLB Arithmetic Switching Characteristics

Setup times not listed explicitly can be approximated by decreasing the combinatorial delays by the setup time adjustment listed. Precise values are provided by the timing analyzer.

Symbol	Description	Speed Grade				Units
		-6		-5		
		Min	Max	Min	Max	
Combinatorial Delays						
T_{OPX}	F operand inputs to X via XOR	-	0.8	-	0.9	ns
T_{OPXB}	F operand input to XB output	-	1.3	-	1.5	ns
T_{OPY}	F operand input to Y via XOR	-	1.7	-	2.0	ns
T_{OPYB}	F operand input to YB output	-	1.7	-	2.0	ns
T_{OPCYF}	F operand input to COUT output	-	1.3	-	1.5	ns
T_{OPGY}	G operand inputs to Y via XOR	-	0.9	-	1.1	ns
T_{OPGYB}	G operand input to YB output	-	1.6	-	2.0	ns
T_{OPCYG}	G operand input to COUT output	-	1.2	-	1.4	ns
T_{BXCY}	BX initialization input to COUT	-	0.9	-	1.0	ns
T_{CINX}	CIN input to X output via XOR	-	0.4	-	0.5	ns
T_{CINXB}	CIN input to XB	-	0.1	-	0.1	ns
T_{CINY}	CIN input to Y via XOR	-	0.5	-	0.6	ns
T_{CINYB}	CIN input to YB	-	0.6	-	0.7	ns
T_{BYP}	CIN input to COUT output	-	0.1	-	0.1	ns
Multiplier Operation						
T_{FANDXB}	F1/2 operand inputs to XB output via AND	-	0.5	-	0.5	ns
T_{FANDYB}	F1/2 operand inputs to YB output via AND	-	0.9	-	1.1	ns
T_{FANDCY}	F1/2 operand inputs to COUT output via AND	-	0.5	-	0.6	ns
T_{GANDYB}	G1/2 operand inputs to YB output via AND	-	0.6	-	0.7	ns
T_{GANDCY}	G1/2 operand inputs to COUT output via AND	-	0.2	-	0.2	ns
Setup/Hold Times with Respect to Clock CLK(1)						
T_{CCKX} / T_{CKCX}	CIN input to FFX	1.1 / 0	-	1.2 / 0	-	ns
T_{CCKY} / T_{CKCY}	CIN input to FFY	1.2 / 0	-	1.3 / 0	-	ns

Notes:
1. A zero hold time listing indicates no hold time or a negative hold time.