بسم الله الرحمن الرحيم

دانشکده برق و رباتیک

گروه الکترونیک

پایان نامه دوره کارشناسی ارشد مهندسی برق- الکترونیک

طراحی و ساخت سیستم پردازش موازی با استفاده از FPGA

دانشجو: محمّد عرب لو

استاد راهنما:

دکتر علی سلیمانی

استاد مشاور:

دکتر علیرضا احمدی فرد

شهریور ۱۳۹۰

شماره : ۷۰۸۸۹/آ.ت.ب
تاریخ : ۹۰/۰۶/۳۰
ویرایش : ------

بسمه تعالی

مدیریت تحصیلات تکمیلی
فرم شماره (۶)

فرم صورتجلسه دفاع پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای :

محمد عربلو رشته : **برق** گرایش : **الکترونیک**

تحت عنوان : : **طراحی و ساخت سیستم پردازش موازی با استفاده از FPGA**

که در تاریخ **۹۰/۰۶/۳۰** با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح زیر است :

قبول (با درجه : خوب امتیاز ۱۷٫۵) ☑ دفاع مجدد ☐ مردود ☐

۱ـ عالی (۲۰ ـ ۱۹)
۲ـ بسیار خوب (۱۸/۹۹ ـ ۱۸)
۳ـ خوب (۱۷/۹۹ ـ ۱۶)
۴ـ قابل قبول (۱۵/۹۹ ـ ۱۴)
۵- نمره کمتر از ۱۴ غیر قابل قبول

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضـــاء
۱ـ استاد راهنما	[illegible]	استادیار	
۲ـ استاد مشاور	علیرضا احمدی فرد	استادیار	
۳ـ نماینده شورای تحصیلات تکمیلی	امیرضا معروفی	استادیار	
۴ـ استاد ممتحن	مرتضی زاهدی	استادیار	
۵ ـ استاد ممتحن	هادی گرایلو	استادیار	

رئیس دانشکده :

ای با من و پنهان چو دل، از دل سلامت می کنم

تو کعبه ای، هر جا روم قصد مقامت می کنم

هر جا که هستی حاضری، از دور در ما ناظری

شب خانه روشن می شود، چون یاد نامت می کنم

السلام علیک یا وعد الله الذی ضمنه...

تقدیم به ارواح طیبه ی شهدا و مادر فداکار و مهربانم

شکر و سپاس خدای عزوجلّ راکه همیشه برای بندگانش بهترین ها را می خواهد...

در ابتدای امر از استاد گرامی جناب آقای دکتر علی سلیمانی بدلیل راهنمایی و حمایتهایشان تشکر و قدردانی می نمایم.

همچنین از تمامی همکاران و دوستانم در دانشگاههای صنعتی شاهرود، خواجه نصیرالدّین طوسی، مؤسسه کوثر دانشگاه تهران و سازمان صداوسیما که حمایت و پشتیبانی آنها موجب دلگرمی زیاد بنده بود تشکر نموده و از خداوند منان برایشان عافیت مسئلت می نمایم.

سپاس ویژه از مادر مهربانم که دعای خیر او همیشه بدرقه راهم بوده...

دانشجو تأیید می نماید که مطالب مندرج دراین پـایان نامه (رساله) نتیجه تحقیقات خودش می باشد و در صورت استفاده از نتایج دیگران مرجع آن را ذکر نموده است.

کلیه حقوق مادی مترتب از نتایج مطالعات ، آزمایشات و نو آوری ناشی از تحقیق موضوع این پایان نامه (رساله) متعلق به دانشگاه صنعتی شاهرود می باشد .

ماه و سال: شهریور ۱۳۹۰

چکیده

پردازش سریع اطلاعات از دیر باز مورد نظر بوده و سعی شده است با ارائه سخت افزارها و الگوریتمهای مناسب و روشهای برنامه نویسی خاص از جمله پردازش موازی تا اندازه ای به این مهم نائل آیند. یکی از سخت افزارهایی که جدیداً با پیشرفت تکنولوژی استفاده عملی پیدا کرده اند FPGA ها هستند. اکنون با کمک یک یا چندFPGA به همراه قطعات جانبی سخت افزار هایی ارائه می گردد که قادرند الگوریتمهایی با بار محاسباتی بالا را اجرا کنند. کاربرد یکی از این مسائل پردازش بلادرنگ سیگنالهای تصویر و ویدئو است که شامل پیاده سازی الگوریتمهای پردازش تصویر می باشد.

پیاده سازی الگوریتم ها و روشهای بررسی شده در مقالات سالهای اخیر بر روی این تراشه ها، قابلیت پردازش موازی و استفاده زیاد این تراشه ها در سیستمهای پردازش بلادرنگ (پردازش تصویر و ویدئو، پردازش سیگنالهای رادار، روترهای اینترنت و ...) را نشان می دهند.

در این پایان نامه پس از طراحی سخت افزار، اقدام به ساخت آن نموده سپس روی نحوه پیاده سازی الگوریتم تشخیص لبه به عنوان یک موضوع مورد مطالعه و برنامه نمونه، بر روی ارزیابی عملکرد سیستم متمرکز می شویم. در واقع در این طرح به دنبال طراحی یک سخت افزار با هسته اصلی FPGA می باشیم. در کنار آن استفاده از تراشه ها و قطعات دیگر، برای ارتباط با دنیای بیرون از برد نیز مد نظر قرار گرفته شده است. از جمله اهداف پروژه طراحی و ساخت یک برد سخت افزاری با قابلیت بالا، تست سخت افزار و نشان دادن قابلیت آن با پیاده سازی یک الگوریتم لبه یابی و مقایسه میزان کارآیی سیستم با یک PC می باشد. ملاحظات طراحی سیستمهای میکروپروسسوری، طراحی شماتیک و مدار چاپی سخت افزار از جمله مطالب اصلی مطرح شده طرح می باشد.

کلمات کلیدی:

طراحی سخت افزار، پردازش سیگنال، قطعات برنامه پذیر، پردازش موازی، FPGA

فهرست مطالب

عنوان **صفحه**

۱- فصل اول: مقدمه ---------- ۱

۱-۱- مقدمه ---------- ۲

۱-۲- اهداف و انگیزش ---------- ۲

۱-۳- عناصر برنامه پذیر در حوزه کار ---------- ۴

۱-۳-۱- روند تکامل FPD ها ---------- ۵

۱-۴- FPGA (آرایه های گیتی برنامه پذیر در حوزه کار)---------- ۸

۱-۴-۱- کاربرد FPGA---------- ۱۰

۱-۴-۲- ماشین های محاسباتی مبتنی بر FPGA---------- ۱۱

۱-۴-۳- سخت افزارهای با قابلیت تغییر وظیفه در محل کار ---------- ۱۱

۱-۴-۴- فرآیند پیاده سازی روی FPGA---------- ۱۱

۱-۴-۵- فواید FPGA ها ---------- ۱۴

۱-۴-۶- معایب FPGA ها---------- ۱۶

۱-۵- تکنولوژی برنامه ریزی---------- ۱۷

۱-۶- گونه های موجود FPGA ---------- ۱۹

۱-۷- اجزای برنامه پذیر داخلی ---------- ۲۰

۱-۷-۱- بلوکهای منطقی یا Logic Blocks---------- ۲۰

۱-۷-۲- کانالهای انتقال سیگنال یا Routing Channels---------- ۲۳

۱-۷-۳- بلوکهای ورودی خروجی یا I/O BLOCKS---------- ۲۵

۱-۸- تراشه ها و بردهای جدید ---------- ۲۵

۱-۹- ساختار پایان نامه---------- ۲۹

۲- فصل دوم: پردازش موازی و FPGA ---------- ۳۱

۲-۱- مقدمه ---------- ۳۲

۲-۲- اهمیت موازی سازی ---------- ۳۲

۲-۳- پردازش موازی ---------- ۳۲
۲-۴- ابررایانه ها ---------- ۳۳
۲-۴-۱- سریعترین ابررایانه های کنونی ---------- ۳۳
۲-۵- انواع سیستم های پردازش موازی ---------- ۳۶
۲-۵-۱- دسته بندی بر اساس ساختار کنترلی ---------- ۳۶
۲-۵-۲- دسته بندی بر اساس نوع ارتباط بین پردازنده ها ---------- ۳۷
۲-۶- رایانه های با فضای آدرس مشترک ---------- ۳۷
۲-۷- رایانه های انتقال پیام ---------- ۳۸
۲-۸- خوشه های رایانه ای ---------- ۳۸
۲-۹- بررسی کارآیی موازی سازی ---------- ۳۹
۲-۹-۱- قانون آمدال ---------- ۴۰
۲-۹-۲- قانون گوستافسون ---------- ۴۱
۳- فصل سوم: مراحل طراحی و انتخاب قطعات ---------- ۴۳
۳-۱- مقدمه ---------- ۴۴
۳-۲- روند طراحی ---------- ۴۴
۳-۳- بررسی معماری ---------- ۴۹
۳-۴- پیکربندی ---------- ۵۱
۳-۵- توانایی های I/O در تراشه های خانواده اسپارتان۳ ---------- ۵۲
۳-۶- علامت گذاری تراشه ---------- ۵۲
۳-۷- بررسی اجمالی IOBs ---------- ۵۳
۳-۸- کارکرد عناصر ذخیره کننده ---------- ۵۴
۳-۹- مقاومتهای Pull up/Pull down ---------- ۵۶
۳-۱۰- مدار نگهدارنده ---------- ۵۶
۳-۱۱- توانایی Boundary-Scan ---------- ۵۷
۳-۱۲- استانداردهای قابل انتخاب برای سیگنالهای ورودی/خروجی ---------- ۵۸
۳-۱۳- امپدانس کنترل شده دیجیتالی ---------- ۶۰
۳-۱۴- سازماندهی IOBs در درون بانکها ---------- ۶۲

۳-۱۵- سازش پذیری FPGA SPARTAN---------- ۶۳
۳-۱۶- قوانین مرتبط با بانکها ---------- ۶۳
۳-۱۷- رفتار پایه های I/O در حین پیکر بندی، راه اندازی و User Mode ---------- ۶۴
۳-۱۸- بلوکهای منطقی قابل برنامه ریزی ---------- ۶۶
۳-۱۸-۱- مسیرهای اصلی منطقی ---------- ۶۷
۳-۱۸-۲- مولد توابع ---------- ۶۸
۳-۱۹- بلوک رم ---------- ۶۹
۳-۱۹-۱- آرایش بلوکهای رم در FPGA---------- ۷۰
۳-۱۹-۲- ساختار داخلی بلوک رم ---------- ۷۱
۳-۱۹-۳- معرفی سیگنال پورت بلوک رم ---------- ۷۲
۳-۱۹-۴- عملکرد گذرگاه آدرس ---------- ۷۲
۳-۱۹-۵- باس ورودی داده ---------- ۷۳
۳-۱۹-۶- باس خروجی داده ---------- ۷۳
۳-۲۰- نسبت ابعاد پورت ---------- ۷۵
۳-۲۱- عملیات داده بلوکهای رم ---------- ۷۶
۳-۲۲- ضرب کننده های اختصاص یافته ---------- ۷۷
۳-۲۳- مدیریت ساعت دیجیتال ---------- ۷۸
۳-۲۴- بررسی حلقه قفل تأخیر ---------- ۸۰
۳-۲۴-۱- ویژگیهای DLL و توابع مرتبط ---------- ۸۱
۳-۲۴-۲- ارتباطات ورودی کلاک DLL ---------- ۸۱
۳-۲۴-۳- خروجی کلاک DLL و اتصالات فیدبک ---------- ۸۲
۳-۲۴-۴- حالتهای فرکانس DLL---------- ۸۳
۳-۲۴-۵- خروجیهای شیفت فاز مربوط به مولفه DLL ---------- ۸۴
۳-۲۴-۶- خروجیهای اساسی سنتز فرکانس مربوط به اجزاء DLL---------- ۸۴
۳-۲۵- سنتزکننده فرکانس دیجیتال ---------- ۸۵
۳-۲۵-۱- حالت های فرکانس DFS ---------- ۸۷
۳-۲۵-۲- DFS با DLL یا بدون DLL ---------- ۸۷

۳-۲۵-۳- اتصالات خروجی کلاک DFS ---------- ۸۸
۳-۲۶- شیفت دهنده فاز ---------- ۸۹
۳-۲۶-۱- فعال کردن اجزای PS و انتخاب حالت ---------- ۹۰
۳-۲۶-۲- تعیین شیفت فاز خوب ---------- ۹۰
۳-۲۶-۳- حالت فاز ثابت ---------- ۹۱
۳-۲۶-۴- حالت فاز متغیر ---------- ۹۲
۳-۲۷- وضعیت اجزاء منطقی ---------- ۹۳
۳-۲۸- تثبیت کلاک های DCM قبل از User Mode ---------- ۹۳
۳-۲۹- شبکه کلاک عمومی ---------- ۹۴
۳-۳۰- اتصالات درونی ---------- ۹۷
۳-۳۱- پیکر بندی ---------- ۹۹
۳-۳۱-۱- واسط پیکربندی استاندارد ----------۱۰۱
۳-۳۱-۲- حالت های پیکربندی ----------۱۰۲
۳-۳۱-۲-۱- حالت Slave سریال ----------۱۰۲
۳-۳۱-۲-۲- حالت Master سریال ----------۱۰۳
۳-۳۱-۲-۳- حالت Slave موازی (SelectMAP) ----------۱۰۴
۳-۳۱-۲-۴- حالت Master موازی ----------۱۰۵
۳-۳۱-۲-۵- حالت اسکن مرزی (JTAG) ----------۱۰۶
۳-۳۱-۳-۱- Readback ----------۱۰۷
۳-۳۲- تغذیه Spartan-۳ FPGAs ----------۱۰۸
۳-۳۲-۱- طراحی سیستم توزیع توان (PDS) و خازن های Bypass / Decoupling --۱۰۸
۳-۳۲-۲- رفتار هنگام روشن شدن ----------۱۰۹
۳-۳۲-۳- حداقل مجاز میزان شیب VCCO در دستگاه های اولیه ----------۱۱۰
۳-۳۳- پیکربندی نگهداری داده ها ----------۱۱۰
۳-۳۴- پمپ های بدون شارژ داخلی یا نوسان سازهای آزاد ----------۱۱۱
۴- فصل چهارم: معرفی و راه اندازی سخت افزار SUT۲۰۰ ---------- ۱۱۲
۴-۱- مقدمه ----------۱۱۳

۴-۲- معرفی ----------۱۱۳
۴-۲-۱- ویژگیهای اصلی تراشه FPGA انتخابی ----------۱۱۵
۴-۲-۲- معرفی بلوکهای سخت افزار ----------۱۱۶
۴-۲-۳- سیگنالهای Handshake تراشه و نحوه ی ارتباط با آن ----------۱۲۱
۴-۲-۴- میکروکنترلر ----------۱۲۲
۴-۳- بررسی SD CARD ----------۱۲۴
۴-۳-۱- مقدمه ----------۱۲۴
۴-۳-۲- سخت افزارSD CARD ----------۱۲۵
۴-۳-۳- نحوه ارتباط SD CARD با میکروکنترلر ----------۱۲۶
۴-۳-۴- پروتکل های ارتباطی ----------۱۲۷
۴-۳-۵- راه اندازی اولیه SD CARD ----------۱۲۸
۴-۴- ملاحظات عملی در اجرای طرح ----------۱۳۰
۴-۵- مراحل طراحی ساخت یک پروژه ----------۱۳۱
۴-۶- عیب یابی مدارات ----------۱۳۵
۴-۷- مراحل اجرای پروژه ----------۱۳۷
۴-۷-۱- تعریف دقیق پروژه و اهداف مورد نظر ----------۱۳۸
۴-۷-۲- انجام مطالعات و تحقیقات لازم ----------۱۳۸
۴-۷-۳- تعیین راه حلهای ممکن و انتخاب بهترین راه ----------۱۳۹
۴-۷-۴- تهیه قطعات و طراحی سخت افزار ----------۱۳۹
۴-۷-۵- استفاده از برنامه های تست و آموزشی ----------۱۴۰
۴-۷-۶- طراحی Layout مدار چاپی ----------۱۴۰
۴-۷-۷- سفارش ساخت برد مدار چاپی ----------۱۴۱
۴-۷-۸- مونتاژ برد مدار چاپی و تست آن ----------۱۴۱
۴-۷-۹- اصلاح نقشه های شماتیک و مدار چاپی ----------۱۴۱
۴-۷-۱۰- ساخت برد نهایی ----------۱۴۲
۴-۷-۱۱- تست و ارزیابی نهایی ----------۱۴۲
۴-۷-۱۲- نوشتن برنامه های لازم روی کامپیوتر ----------۱۴۲

۴-۷-۱۳- مستند سازی و گزارش نهایی----------۱۴۳
۴-۷-۱۴- نتایج بدست آمده----------۱۴۳
۵- فصل پنجم: تشخیص لبه ----------۱۴۵
۵-۱- مقدمه ----------۱۴۶
۵-۲- معرفی ----------۱۴۶
۵-۳- تشخیص لبه ----------۱۴۷
۵-۴- گرادیان تصویر ----------۱۴۹
۵-۵- ارتباط با FPGA و پیاده سازی بر روی آن ----------۱۵۲
۵-۵-۱- نحوه پیکربندی FPGA ----------۱۵۳
۵-۶- نحوه ورود و خروج داده ها به سخت افزار ----------۱۵۶
۵-۷- پیاده سازی و سنتز طرح ----------۱۵۷
۵-۸- نتایج اعمال تصویر----------۱۶۰
۶- فصل ششم: نتیجه گیری و پیشنهادات----------۱۶۳
۶-۱- مقدمه ----------۱۶۴
۶-۲- نتیجه گیری ----------۱۶۴
۶-۳- پیشنهادات و تحقیقات آتی ----------۱۶۶
- مراجع ----------۱۶۹
- پیوست----------۱۷۱
- ضمیمه الف- شماتیک و مدار چاپی سخت افزار ----------۱۷۱
- ضمیمه ب- آشنایی با SD Card----------۱۷۸
- ضمیمه ج- آشنایی با نرم افزار Codevision----------۲۰۲
- ضمیمه د- ابر رایانه های برتر ----------۲۰۶
- ضمیمه ه- آشنایی با نرم افزار Xilinx ISE ----------۲۰۹

فهرست شکل ها

عنوان **صفحه**

شکل ۱-۱: ساختار PLA---------- ۶

شکل ۱-۲: ساختار PAL---------- ۷

شکل ۱-۳: معماری داخلی یک FPGA---------- ۸

شکل۱-۴: گام های لازم جهت پیاده سازی یک مدار روی FPGA---------- ۱۲

شکل۱-۵: مقایسه مراحل پیاده سازی یک طرح روی FPGA و MPGA---------- ۱۴

شکل۱-۶: سلول SRAM متشکل از ۵ ترانزیستور ---------- ۱۸

شکل۱-۷: نمایی از یک آنتی فیوز ---------- ۱۸

شکل۱-۸: ساختار EEPROM---------- ۱۹

شکل ۱-۹: معماری های مختلف FPGA ---------- ۲۰

شکل ۱-۱۰: ساختار Slice ها در اسپارتان۳ ---------- ۲۲

شکل۱-۱۱: معماری کلی مسیر نمایی---------- ۲۴

شکل۱-۱۲: نمایی از اتصالات داخلی یک FPGA مبتنی بر آنتی فیوز---------- ۲۴

شکل۱-۱۳: ساختار ورودی/خروجی نوعیFPGA ---------- ۲۵

شکل۲-۱- ابررایانه Roadrunner ساخت IBM ---------- ۳۴

شکل۲-۲- قانون آمدال---------- ۴۱

شکل۳-۱- معماری خانواده اسپارتان۳ ---------- ۵۱

شکل ۳-۲- علامت گذاری تراشه از نوع بسته بندی QFP ---------- ۵۳

شکل۳-۳- نمودار ساده شده یک ورودی/خروجی ---------- ۵۵

شکل۳-۴- اتصال مقاومتهای مرجع ---------- ۶۱
شکل ۳-۵- بانکهای I/O اسپارتان۳ ---------- ۶۲
شکل ۳-۶- طرح Slice ها در درون CLB---------- ۶۶
شکل ۳-۷- مسیر داده بلوک رم ---------- ۷۲
شکل ۳-۸-پورت دوگانه---------- ۷۲
شکل ۳-۹ پورت یگانه ---------- ۷۲
شکل ۳-۱۰- شکل موج عملیات داده بلوک رم با انتخاب WRITE-FIRST---------- ۷۶
شکل۳-۱۱- شکل موج عملیات داده بلوک رم با انتخاب READ-FIRST ---------- ۷۷
شکل ۳-۱۲- شکل موج عملیات داده بلوک رم با انتخاب NO-CHANGE ---------- ۷۷
شکل ۳-۱۳-ضرب کننده ۱۸ بیتی آسنکرون ---------- ۷۸
شکل۳-۱۴- ضرب کننده ۱۸بیتی با رجیستر ---------- ۷۸
شکل ۳-۱۵- بلوک کارکرد DCM و سیگنالهای مرتبط آن ---------- ۸۰
شکل ۳-۱۶: نمودار ساده شده عملکرد DLL ---------- ۸۰
شکل ۳-۱۶-۱ الی۴: ارتباطات ورودی کلاک DLL ---------- ۸۲
شکل ۳-۱۷: خروجیهای کلاک DLL ---------- ۸۵
شکل ۳-۱۸: شکل موجهای شیفت دهنده فاز---------- ۹۲
شکل ۳-۱۹: شبکه کلاک اسپارتان۳ ---------- ۹۶
شکل ۳-۲۰-۱-الی ۴، اتصالات درونی ---------- ۹۹
شکل ۳-۲۱- اتصالات مربوط به مستر سریال و اسلیو سریال ---------- ۱۰۳
شکل ۳-۲۲: اتصالات حالت Slave موازی ---------- ۱۰۵

شکل ۳-۲۳- نمودار اتصالات برای حالت Master موازی---------- ۱۰۶

شکل ۳-۲۴: توالی راه اندازی پیش فرض ---------- ۱۰۸

شکل ۴-۱- نمای کلی سخت افزار SUT۲۰۰---------- ۱۱۴

شکل ۴-۲- ساختار داخلی SD CARD ---------- ۱۲۵

شکل ۴-۳- : پایه هایSD CARD در مدهای SPI و SD ---------- ۱۲۶

شکل ۴-۴- نحوه ارتباط SD CARD با میکروکنترلر ---------- ۱۲۷

شکل ۴-۵- پروتکل باس باقابلیت ارتباط باچندکارت حافظه باسرعت ۱۰۰ mbps ---------- ۱۲۸

شکل ۴-۶- روندنمای راه اندازی ---------- ۱۳۰

شکل ۵-۱: نقابهای گوناگون برای محاسبه گرادیان تصویر---------- ۱۵۲

شکل ۵-۲: تنظیمات ورودی iMPACT---------- ۱۶۹

شکل ۵-۳: تنظیمات پروگرامر---------- ۱۵۴

شکل ۵-۴: پیکربندی تراشه---------- ۱۵۵

شکل ۵-۵: پنجره برنامه ریزی تراشه---------- ۱۵۵

شکل ۵-۶: روند کلی ورود و خروج اطلاعات بین کامپیوتر و برد---------- ۱۵۶

شکل۵-۷: روند نمای ارتباط بین میکروکنترلر و FPGA---------- ۱۵۹

شکل ۵-۸- نتایج حاصل از اعمال الگوریتم تشخیص لبه بر روی تصویر ۱۵*۱۵---------- ۱۶۰

فهرست جداول

عنوان **صفحه**

جدول ۱-۶: مقایسه هزینه کل تولید MPGA و FPGA ها نسبت به حجم تولید ---------- ۱۷

جدول ۱-۲: مقایسه ویژگیهای تراشه های برتر شرکت Xilinx ---------- ۲۶

جدول۲-۱- ابررایانه های برتر دنیا---------- ۳۴

جدول۳-۱- جدول ورودی/خروجی تراشه خانواده اسپارتان۳ ---------- ۵۲

جدول ۳-۲- مقدار مطلق حداکثر ولتاژ قابل تحمل تراشه ---------- ۵۷

جدول ۳-۳- استانداردهای ورودی/خروجیهای تک پایانه (بر حسب ولت) ---------- ۵۹

جدول ۳-۴- استانداردهای ورودی/خروجی دیفرانسیلی ---------- ۶۰

جدول ۳-۵: تعداد بلوکهای رم ---------- ۷۱

جدول ۳-۶- سیگنالهای DFS---------- ۸۶

جدول ۳-۷- ویژگیهای DFS---------- ۸۶

جدول ۳-۸- مکانیزم انتخاب BUFGMUX---------- ۹۵

جدول ۳-۹- تنظیمات پینهای پیکربندی ---------- ۱۰۰

جدول ۳-۱۰- داده های پیکربندی اسپارتان۳ ---------- ۱۰۰

جدول ۳-۱۱- نیازمندیهای شیب زمان ولتاژ تغذیه---------- ۱۱۱

جدول ۳-۱۲- سطوح ولتاژ مورد نیاز برای حفظ کردن محتوای RAM ----------۱۱۱

جدول ۴-۱- تراشه های خانواده اسپارتان۳---------- ۱۱۵

جدول ۴-۲- مقایسه دو تراشه USB---------- ۱۱۹

جدول ۴-۳- ولتاژ و جریان مصرفی کارت حافظه ---------- ۱۲۶

جدول ۵-۱- مقایسه سرعت نرم افزار و سخت افزار در برخی از الگوریتمهای پردازش تصویر -- ۱۶۰

جدول ۵-۲- خلاصه سنتز طرح و ماژول های اصلی آن ---------- ۱۶۱

فهرست اصطلاحات اختصاری

Term of acronym	Explanation
ASIC	Application Specific Integrated Circuit
BGA	Ball grid array
CAD	Computer-aided design
CLB	Configurable Logic Blocks
CPLD	Complex programmable logic Device
CRC	Cycle Redundancy Code
CSD	Card Specific Data
DCM	Digital Clock Manager
DCI	Digitally Controlled Impedance
DDR	Double Data rate
DFS	Digital Frequency Synthesizer
DLL	Delay Lock Loop
ESD	Electro-Static Discharge
FDDR	Double Data rate D-FF
FLOPS	Floating Point per Seconds
FPD	Field Programmable Device
FPGA	Field- Programmable Gate Array
FPS	Fine Phase Shift
GCLKs	Global Clocks
GP I/Os	General-purpose User I/Os
GPR	General Purpose Register
GSR	Global Set/Reset

HDL	Hardware description language
IBUF	Input Buffer
IOB	Input Output Block
ISP	In System Programming
JTAG	Joint Test Action Group
LUT	Look-up Table
LVCMOS	Low Voltage CMOS
LVDS	Low Voltage Differential Signal
MIMD	Multiple Instruction Multiple Data
MPI	Message Passing Interface
MPGA	Mask-Programmable Gate Array
MPP	Massive Parallel Processing
NRE	Non Recurring Eng
OCR	Operating Conditions Register
PDS	Power Distribution Systems
PLA	programmable logic array
PVM	Parallel virtual machine
RTL	Register Transfer Level
TAP	Test Access Port
TCK	Test Clock
TDI	Test Data Input
TDO	Test Data Output
TMS	Test Mode Select
UMA	Uniform Memory Access
VCP	Virtual Com Port
XST	Xilinx Synthesis Technology

فصل اول

مقدمه

۱-۱- مقدمه

در ابتدای این فصل اهداف و انگیزه های انجام این پروژه و مروری بر فعالیتهای صورت پذیرفته آورده شده است. در ادامه به معرفی FPGA و روند تکاملی آنها پرداخته، مشکلات طراحی مدار، کاربردها و ضرورت انجام کار اشاره شده است و در نهایت ساختار کلی پروژه طرح می شود.

۱-۲- اهداف و انگیزش

استفاده از FPGA ها با قابلیتهای فراوانی که در پیاده سازی الگوریتمهای پیچیده خصوصاً پردازش سیگنالهای تصویر دارند روز به روز در حال گسترش و توسعه است. بعنوان نمونه استفاده از تراشه Pro Virtex II در کنار سایر بلوکها نظیر کانالهای مستقل DDR-SDRAM، PCI BUS و لینک مدار واسط دوربین، سرعت پردازش را تا 2^{27} تصویر(در ۲ ثانیه) افزایش می دهد. پیاده سازی الگوریتم ها و روشهای بررسی شده در مقالات سالهای اخیر بر روی این پردازنده ها، قابلیت پردازش موازی و استفاده زیاد این تراشه ها در سیستمهای پردازش بلادرنگ (پردازش تصویر و ویدئو، پردازش سیگنالهای رادار، روترهای اینترنت و ...) همگی بر قدرت و قابلیت این تراشه ها اشاره دارند [۱,۲,۳].

با توجه به پیشرفت و توسعه پردازنده ها، استفاده و بهره از آنها نیازمند بررسی جامع این نوع پردازنده ها و مدارات واسط مرتبط می باشد. شناخت بیشتر این سیستم ها و میزان کاربرد و کارآیی آنها در این موضوع و موضوعات مشابه می تواند مفید باشد. پردازش سریع اطلاعات از دیر باز مورد نظر بوده و سعی شده است با ارائه سخت افزار ها و الگوریتمهای مناسب و روشهای برنامه نویسی خاص از جمله پردازش موازی تا اندازه ای به این مهم نائل آیند و با توجه به نیاز هر روز آنرا توسعه دهند. یکی از سخت افزارهایی که اخیراً با پیشرفت تکنولوژی استفاده عملی پیدا کرده اند FPGA ها هستند. اکنون با کمک یک یا چندFPGA به همراه قطعات جانبی سخت افزار هایی ارائه می گردد که قادرند الگوریتمهایی با بار محاسباتی بالا را اجرا کنند. برای پیاده سازی این الگوریتم ها، بایستی آنها را بصورت موازی تبدیل و با پیکر بندی ساختار درونی FPGA چندین پردازنده ساده طراحی و مسئله را بصورت موازی حل کنند. لذا

آشنایی با این نوع پردازنده از اهمیت ویژه ای برخوردار است. بررسی اولیه در تعدد استفاده از این نوع پردازنده در پروژه های صنعتی، تجاری، نظامی و... گویای مطلب فوق است. بعنوان نمونه در اکثر روترهای اینترنت، دستگاه های پخش ویدئو حرفه ای، انواع کنترلرهای رادار و... از FPGA استفاده شده است.

در این پروژه به دنبال طراحی یک سخت افزار با هسته اصلی FPGA می باشیم. در کنار آن استفاده از تراشه ها و قطعات دیگر، برای ارتباط با دنیای بیرون از برد نیز مد نظر قرار گرفته شده است. در ابتدا می بایست تعیین کنیم که ساختار سخت افزار چگونه باشد از چه اجزایی تشکیل شده باشد و قطعات مصرفی چگونه انتخاب گردند و ارتباط سخت افزار با دنیای بیرون چگونه باشد. لازم بذکر است که کاربرد سخت افزار طراحی شده در پیاده سازی انواع الگوریتم ها و برنامه های پردازشی- کنترلی می باشد. بعنوان مثال در سیستم های حفاظتی، سیستم های پردازش موازی، بکار گیری سخت افزار در رباتهای هوشمند و پروژه های مرتبط از انواع استفاده های سخت افزار طراحی شده می باشد.

همانگونه که اشاره شد، طراحی یک سخت افزار پردازشگر قوی و ارتباطات آن از عمده اهداف این طرح است. لذا با توجه به این موضوع و مسائل مرتبط با آن، مشکلات اصلی مواجهه شده در انجام طرح عبارتند از:

- ملاحظات شماتیک مدار از جهت تغذیه های متفاوت تراشه ها
- ارتباطات تراشه های گوناگون با یکدیگر
- کارکرد مدار در حالت فرکانس های بالاتر از ۲۰ مگاهرتز
- توزیع توان در کل سیستم و ملاحظات زمین مدار برای توزیع یکسان ولتاژ ورودی بدون تغییر در آن[۱].

۱ -Power Distribution Systems

- با توجه به اینکه مدارچاپی سخت افزار مورد نظر دولایه طراحی شده است، ارتباط بین لایه ها و اتصالات آنها می بایستی به دقت مورد توجه قرار گرفته شود تا اثر نامطلوبی در سیگنال مورد نظر نگذارد.
- ملاحظات شیارهای ارتباطی در سخت افزار، از جمله قسمت های مهم در طراحی مدارچاپی می باشد. با توجه به اینکه فاصله های پایه های تراشه های FPGA و برخی دیگر از قطعات برد بسیار نزدیک به یکدیگر (۹ میل[۱]) است، این مورد دست طراح را برای ارتباط آسان پایه ها و تراشه ها با یکدیگر مشکل ساز می نماید و بدلیل این موضوع جانشانی قطعات در برد، با جابجایی بسیار قطعات همراه است تا اینکه حالت بهینه طرح بدست آید.
- ملاحظات طراحی سخت افزار از لحاظ جایگذاری قطعات در برد، مکان آنها و فاصله آنها با یکدیگر که یکی از مهمترین بخش های طراحی مدارچاپی می باشد.
- لحیم کاری قطعات روی برد و مونتاژ آن

۱-۳- عناصر برنامه پذیر در حوزه کار

توسعه قطعات پیچیده برنامه ریزی در حوزه کار (FPD[۲]) به میزان قابل توجهی طراحی سخت افزار های دیجیتال را دگرگون نموده است. بر خلاف نسل های گذشته تکنولوژی سخت افزار، که در آن طراحی های در سطح برد شامل تعداد زیادی از تراشه های حاوی گیت های پایه بود، در حال حاضر هر طرح دیجیتال تولید شده شامل قطعات با ظرفیت بالاست. این موضوع نه تنها برای قطعات الکترونیکی سفارشی محض نظیر پردازنده ها و حافظه ها صادق است بلکه در مورد مدارهای منطقی نظیر شمارنده ها، ثبات ها و رمز گشاها نیز اعتبار دارد. هنگامیکه طراحی سیستم های حجیم مد نظر باشد، طراحان این سیستم ها را به صورت مجتمع درون آرایه های گیتی با گنجایش بالا پیاده می نمایند، اما هزینه مهندسی غیر قابل برگشت[۳] بالا و زمان طولانی، سبب شده تا آرایه های متشکل از گیت برای اهداف تهیه

---

۱ - ۰.۲۲۸٦mm (۱mil=۰.۰۲٥٤mm)
۲ -Field Programmable Device
۳ - Non Recurring Eng

نسخه اولیه و سایر کاربردهای با تولید محدود مناسب نباشد. بهمین دلیل بسیاری از نسخه های اولیه مدار و طرح های تولیدی از FPD ها استفاده می نمایند. از جمله فواید چشمگیر FPD ها می توان به هزینه پایین طراحی، ریسک اقتصادی کم و بدلیل برنامه ریزی شدن قطعه توسط کاربر نهایی، مدت زمان کم تولید محصول و سادگی اعمال تغییرات بعدی روی طرح اشاره نمود. برای انتخاب از بین انواع متعدد FPD طراح وظیفه دارد تا یک بررسی روی بهترین قطعه قابل استفاده و ابزار CAD[۱] مختص به فروشنده انجام دهد.

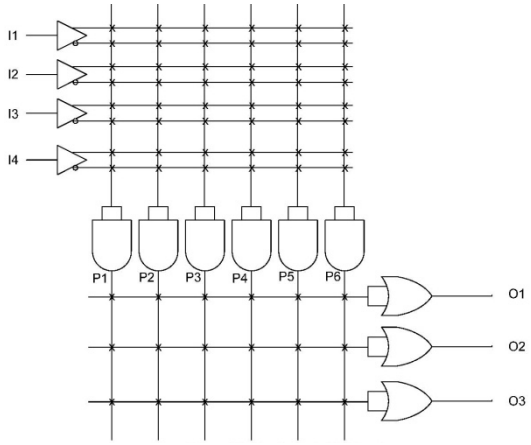
۱-۳-۱- روند تکامل FPD ها

اولین تراشه قابل برنامه ریزی توسط کاربر که برای پیاده سازی مدارات منطقی نیز مورد استفاده واقع شده است، حافظه فقط خواندنی برنامه پذیر (PROM) است که خطوط آدرس آن بعنوان ورودی و خطوط داده آن بعنوان خروجی مدار منطقی عمل می نمایند. توابع منطقی غالباً نیاز به تعداد محدودی از p-term[۲] ها دارند، در حالیکه PROM دارای یک رمزگشای کامل برای ورودیهای آدرس خود می باشد.

از اینرو PROM ها برای تحقق مدارهای منطقی معمولی مناسب نبوده و اکثر قسمت های آن ممکن است بدون استفاده باقی بماند. بهمین دلیل طراحان کمتر از آنها برای این منظور استفاده می کنند. اولین قطعه ای که بطور اختصاصی برای پیاده سازی مدارهای منطقی طراحی گردید، آرایه منطقی قابل برنامه ریزی یا PLA[۳] (شکل۱-۱) است. یک PLA دارای دو صفحه منطقی[۴] است. یک صفحه قابل برنامه ریزی با استفاده از تکنیک AND سیم بندی شده قرار گرفته است. ساختار PLA این امکان را فراهم می سازد که هر کدام از ورودیهای مربوط به صفحه AND یا مکمل آنها با یکدیگر OR شده و به این ترتیب هر خروجی این صفحه با یک p-term از ورودیها متناظر خواهد بود.

---

۱ - Computer-aided design
۲ -Predicate Term
۳ -programmable logic array
٤ -Logical Plane



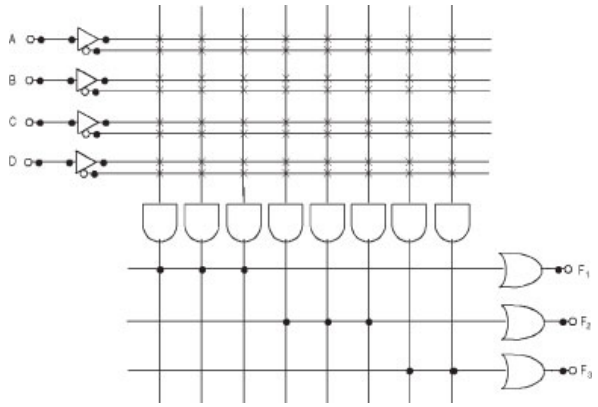
شکل ۱-۱: ساختار PLA

بهمین صورت کاربر قادر است هر خروجی صفحه OR را بنحوی تنظیم نماید که مجموع تعدادی از خروجی های صفحه AND را شامل باشد. به این ترتیب PLA ها برای پیاده سازی مدارها به شکل مجموع حاصلضرب ها[۱] مناسبند. صفحات AND و OR می توانند تعداد زیادی ورودی داشته باشند.

این ساختار دارای معایب هزینه بالای ساخت و تا حدی بازدهی سرعتی کم است. این عیب ها ناشی از برنامه پذیر بودن دو سطح موجود در PLA است، چرا که هم ساخت دو صفحه قابل برنامه ریزی مشکل است و هم وجود کلیدهای قابل برنامه ریزی در هر دو صفحه تأخیر انتشار قابل توجهی را تحمیل می سازد. برای برطرف نمودن این ضعف، شرکت Monolithic Memories Inc (که بعدها توسط شرکت AMD خریداری گردید) قطعات PAL را معرفی نمود. همانطور که در شکل۱-۲ مشاهده می شود، PAL ها تنها یک سطح قابل برنامه ریزی را شامل هستند. برای جبران کمبود انعطاف پذیری ساختار که ناشی از وجود صفحه OR ثابت است، PAL ها انواع متنوعی با تعداد ورودی، خروجی و گیت های OR با اندازه های متنوع عرضه شده اند. پایین آمدن تأخیر کلی در PAL ها این امکان را پدید می آورد که برای پیاده سازی مدارهای ترتیبی در خروجی گیت های OR فلیپ فلاپ هایی تعبیه شوند.

۱ -SOP

PAL ها همچنین بعنوان ساختار پایه قطعات پیچیده تر دیگری مطرح شده اند. معمولاً تمام FPD های ساده را که شامل PLA، PAL و قطعات مشابه PAL می باشند در مقوله ای به نام SPLD قرار می دهند که مشخصه اصلی آنها قیمت کم و بازدهی سرعتی بسیار بالاست.



شکل ۱-۲: ساختار PAL

پیشرفت تکنولوژی، عرضه قطعاتی را که گنجایش بالاتری از SPLD دارند ممکن ساخته است. مشکل افزایش گنجایش SPLD آن است که با افزایش تعداد ورودیهای صفحه منطقی قابل برنامه ریزی، وسعت ساختمان آن به سرعت رو به رشد می گذارد. تنها راه حل تهیه نمودن قطعات پر گنجایش مبتنی بر SPLD، اتصال قابل برنامه ریزی چند SPLD روی یک تراشه منفرد به یکدیگر است. چنین ساختاری CPLD نامیده می شود. این ایده اولین بار توسط شرکت Altera در اولین تراشه های Classic EPLD استفاده شد. بخاطر بازار رو به رشد FPD های بزرگ، شرکتهای دیگری نیز افزاره های CPLD را معرفی نمودند.

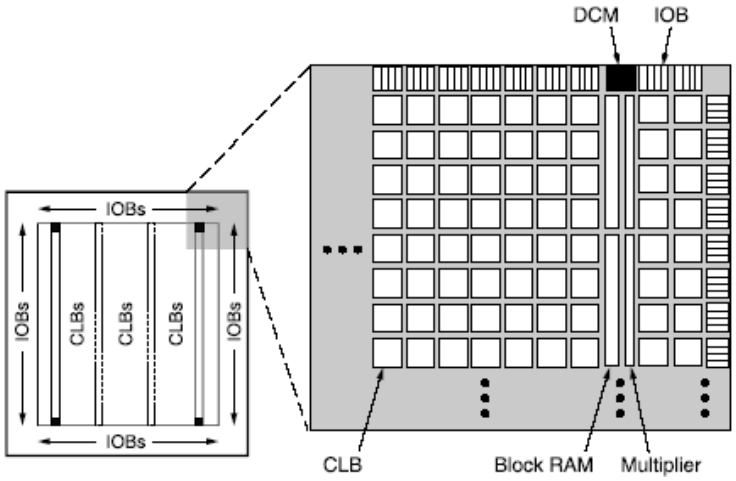
CPLD ها گنجایش منطقی در حدود ۵۰ برابر یک SPLD نوعی ارائه می نمایند، لیکن افزایش گنجایش این معماری ها به دلیل پایه قرار دادن تحقق توابع منطقی دو سطحی مشکل است.

پر ظرفیت ترین تراشه های منطقی همه منظوره که اکنون در دسترس هستند، آرایه های متشکل از گیت ها هستند که عموماً بنام MPGA[۱] نامیده می شوند. یک MPGA حاوی آرایه ای از ترانزیستورهای

۱ -Mask-Programmable Gate Array

از پیش ساخته شده است که توسط ساخت اختصاصی لایه های اتصالات سیمی در کارخانه، مدار مورد نظر کاربر را تحقق می بخشند. به دلیل اینکه در طول فرآیند ساخت تراشه، در قسمت کارخانه ساخت سیلیکون عمل سفارشی سازی انجام می پذیرد، زمان ساخت طولانی بوده و هزینه تطبیق طراحی با خواست کاربر نسبتاً بالاست.

همانند MPGA ها، FPGA ها متشکل از آرایه ای از عناصر مداری غیر ویژه (بلوکهای منطقی) و تجهیزات اتصال این عناصر به یکدیگر هستند. لیکن استفاده کننده نهایی این قطعه را از طریق برنامه ریزی قالب بندی می کند. شکل ۱-۳ نشان دهنده معماری داخلی یک FPGA نوعی است.



شکل ۱-۳: معماری داخلی یک FPGA

۱-۴- FPGA (آرایه های گیتی برنامه پذیر در حوزه کار)

ادوات FPGA مدارات مجتمع الکترونیک قابل برنامه ریزی هستند که دارای ساختار دو بعدی بوده و قادر به پیاده سازی مدارات چند سطحی می باشند و بهمین دلیل انعطاف پذیری آنها بسیار بیشتر از PLA ها است که تنها قادر به پیاده سازی مدارات دو سطحی هستند.

FPGA در سال ۱۹۸۵ توسط شرکت Xilinx معرفی گردید. از آن زمان تا کنون FPGA های متنوعی توسط شرکت های Actel، Altera، Plessey، AMD، Quick Logic، Algotronix، Concurrent

Logic و Crosspoint Solution طراحی گردیده اند. گنجایش منطقی FPGA ها به چند صد هزار گیت معادل می رسد.

FPGA ها شامل آرایه های دو بعدی از بلوکهای منطقی هستند که توسط تجهیزات اتصال به یکدیگر ارتباط یافته اند. این اتصالات قابل برنامه ریزی بوده و به کاربر این امکان را می دهند که بلوکهای منطقی را به یکدیگر متصل سازد. اتصالات داخلی مشتمل بر قطعاتی از سیم هستند که می توانند دارای طول های متفاوت باشند. در سرتاسر اتصالات بینابینی کلیدهای قابل برنامه ریزی موجودند که وظیفه اتصال قطعات سیم ها به بلوک های منطقی و به یکدیگر را بر عهده دارند. اندازه، ساختار و تعداد بلوکهای ساختمانی و مقدار اتصالات و نحوه اتصال در بین FPGA های مختلف فرق می کند.

تفاوت بین معماری های مختلف FPGA ها از تکنولوژی مختلف برنامه ریزی و هدف نهایی کاربرد ناشی می شود. FPGA ها به سه دسته مبتنی بر SRAM، مبتنی بر آنتی فیوز و مبتنی بر EEPROM تقسیم می شوند. به منظور تسهیل در پیاده سازی دامنه وسیعی از مدارها، وجود حداکثر انعطاف پذیری در FPGA ها از اهمیت ویژه ای برخوردار است. این موضوع بدین معنی است که طراحی بلوک منطقی و مدارات اتصال بایستی بگونه ای باشد که بتواند محدوده وسیعی از مدارهای دیجیتال را تحقق بخشد. روشهای بسیاری در راستای طراحی ساختار خود FPGA ها وجود دارند که ذاتاً موازنه ی بین پیچیدگی و انعطاف پذیری بلوکهای منطقی و تجهیزات اتصال را بهمراه دارند.

ساختار و محتوای یک بلوک منطقی در حکم معماری آن تلقی می شود. معماری بلوک منطقی به روشهای متفاوتی می تواند طراحی گردد. بعضی از بلوکهای منطقی FPGA ها به طور ساده از دو گیت NAND دو ورودی تشکیل شده اند. بلوکهای جدید در حال حاضر از ساختارهای پیچیده تری نظیر مالتی پلکسر، جدول جستجو و ... تشکیل یافته اند.

در برخی از پژوهشها، معماری هایبریدی برای بلوک منطقی FPGA پیشنهاد شده است. بلوکهای منطقی هایبریدی شامل یک PAL و یک جدول بازبینی هستند که تعداد ورودی ها، خروجی ها و p-

term ها در PAL، تعداد ورودی ها در جدول بازبینی بر اساس داده های آماری حاصل از پیاده سازی مدارات نمونه به شکل بهینه ای تعیین می شوند.

ساختمان و محتوای تجهیزات مسیر نمایی در یک FPGA به معماری مسیرنمایی موسوم است. معماری مسیر نمایی حاوی قطعات سیم و کلیدهای قابل برنامه ریزی است. کلیدهای قابل برنامه ریزی به طرق مختلفی ساخته می شوند. ازجمله توسط Pass Transistor کنترل شونده با SRAM، آنتی فیوز و ترانزیستورهای EEPROM. امروزه تحقیقات در زمینه بلوک منطقی FPGA، بیشتر بر روی ارائه معماری های آسنکرون و FPGA های ممیز شناور متمرکز شده است.

۱-۴-۱- کاربرد FPGA

تقریباً همه کاربردهایی که در حال حاضر از MPGA ها، PLD ها و مدارهای منطقی مقیاس کوچک SSI استفاده می نمایند، بر روی FPGA ها قابل پیاده سازی هستند. در ذیل برخی از زمینه های مرتبط با این کاربردها بیان شده است:

- مدارهای ASIC

یک FPGA وسیله پیاده سازی مدارهای منطقی دیجیتال است. این تراشه ها بالاخص جهت پیاده سازی مدارهای ASIC مناسبند. بعضی از نمونه های چنین کاربردهایی عبارتند از: کنترل کننده FIFO چند مگا بیتی، کنترلر DRAM با تصحیح خطا، کنترلر چاپگر، مدار تشخیص کاراکتر نوری و ...

امروزه مدارهای آنالوگ FPGA هم با روند رو به رشدی در بازار قابل دسترسی هستند. در حال حاضر مدارهای موجود در بازار اغلب حاوی تعدادی از قطعات SSI هستند. در بسیاری از شرایط می توان این تراشه ها را با FPGA جایگزین نمود که در چنین حالتی فضای اشغالی روی سطح بردی که این تراشه ها روی آن نصب شده اند، به مراتب کاهش یافته و این در حالیست که مصرف توان کاهش و سرعت پردازش افزایش خواهد یافت.

FPGA ها برای تهیه نسخه اولیه یک مدار منطقی ایده آل هستند. هزینه پایین پیاده سازی و زمان کوتاه تحقق فیزیکی یک طرح مفروض، فواید عمده ای را نسبت به روش های متداول ساخت نسخه اولیه فراهم نموده است. علاوه بر این تغییرات بعدی بر روی طرح اولیه بسیار کم هزینه است.

۱-۴-۲- ماشین های محاسباتی مبتنی بر FPGA

امکان برنامه ریزی مجدد FPGA بصورت درون سیستمی رده جدیدی از کامپیوتر ها را معرفی نموده است. این ماشین ها در بردارنده بردی هستند که حاوی چنین FPGA هایی است. ایده نهفته در ورای این طرح آن است که یک برنامه نرم افزاری به مجموعه ای از قطعات سخت افزاری کامپایل گردد. این سخت افزار با برنامه ریزی مناسب FPGA تحقق می یابد. یا بعبارت دیگر هر الگوریتم نرم افزاری قابلیت پیاده سازی مستقیم توسط یک سخت افزار اختصاصی را دارد که این سخت افزار می تواند توسط یک FPGA تحقق یابد.

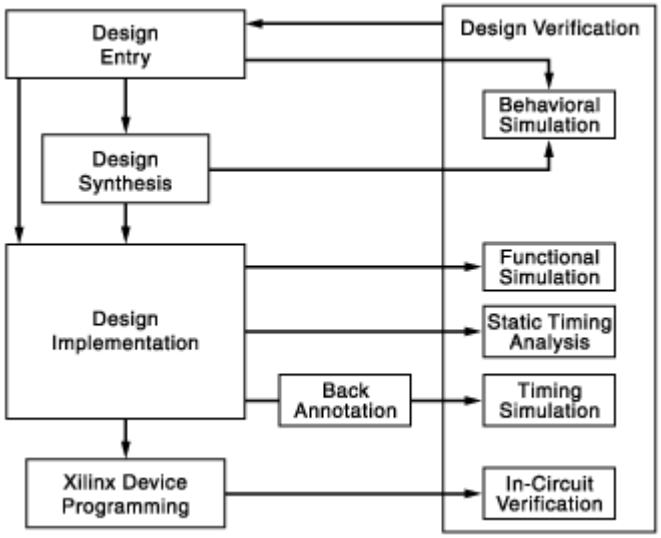
۱-۴-۳- سخت افزارهای با قابلیت تغییر وظیفه در محل کار

چنانچه لازم باشد آرایش عملیاتی یا عملکرد لحظه ای ماشینی که در حال انجام وظیفه است تغییر یابد، جذابیت بیشتر FPGA ها نمایان می گردد. یک مثال از چنین ماشینی تجهیزات کامپیوتری است که در محل دوری قرار گرفته و لازم است ترکیب آن در محل عملیاتی خود بمنظور تصحیح خطای عملکرد و یا خطای طراحی تغییر یابد. در چنین حالتی بدون جداسازی برد اصلی این ماشین جهت اعمال تغییرات، می توان این عملیات را در همان مکان تعبیه ماشین انجام داد. متذکر می شود که مناسبترین نوع FPGA برای استفاده در این حالت، نوعی است که حاوی کلیدهای قابل برنامه ریزی مجدد باشد.

۱-۴-۴- فرآیند پیاده سازی روی FPGA

چنانچه طراح بخواهد به نحو احسن از FPGA استفاده نماید، بایستی به ابزار کاری طراحی به کمک کامپیوتر دسترسی داشته باشد. شکل۱-۴ نمایانگر گام هایی است که جهت پیاده سازی یک مدار روی FPGA در یک سیستم CAD نوعی طی می گردد[۴]. نقطه شروع فرآیند طراحی، وارد کردن مدار مطلوب برای پیاده سازی است. این گام معمولا بصورت کشیدن شماتیک یا نوشتن کد توصیف سخت افزاری مثلاً به زبان VHDL یا بصورت بیان عبارات بولی طی می شود. معمولاً بدون توجه به اینکه مدار چگونه توصیف شده است، این توصیف به فرم یک استاندارد داخلی نظیر عبارات بولی ترجمه می گردد. سپس عبارات بولی توسط ابزار بهینه سازی منطقی پردازش می گردد که در این فرآیند عبارات دستخوش تغییر می شوند.

هدف از تصحیح این عبارات، بهینه سازی مساحتی یا سرعتی مدار نهایی است. البته ترکیبی از دو هدف نیز می تواند دنبال شود. بهینه سازی مذکور عمدتاً با استفاده از کمینه سازی جبری صورت گرفته و برای پیاده سازی مدار منطقی به هر صورتی و نه فقط برای FPGA مناسب است. بعد از این عمل عبارت بولی بهینه سازی شده به صورت مداری متشکل از بلوک های منطقی FPGA تبدیل می گردند. این عمل با استفاده از نرم افزار نگاشت تکنولوژی انجام می پذیرد.



شکل۱-۴: گام های لازم جهت پیاده سازی یک مدار روی FPGA

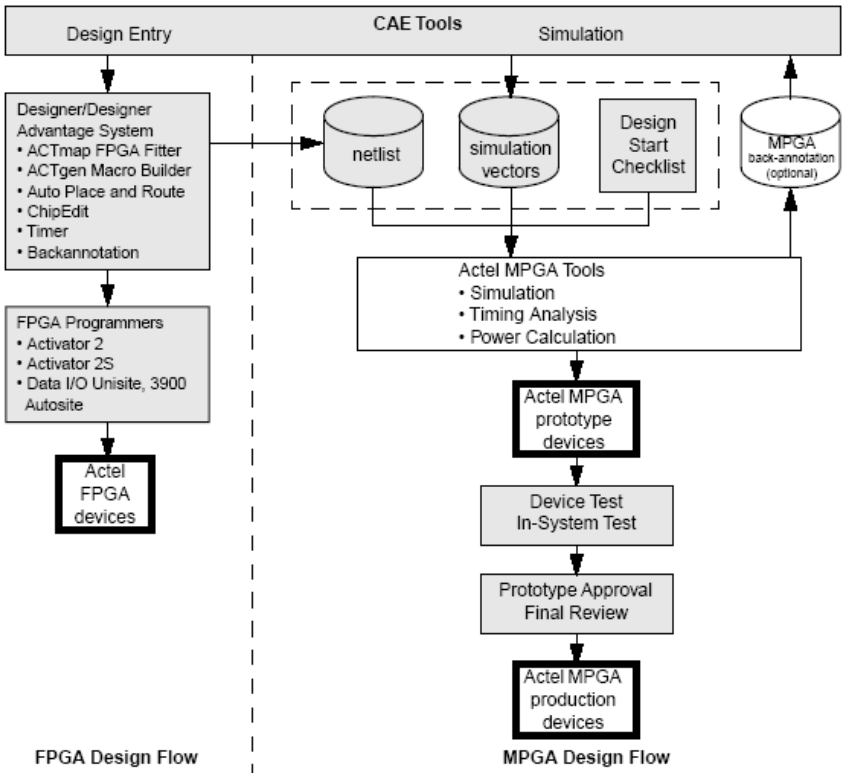
هر مدار ترکیبی یا ترتیبی را می توان بصورت یک گراف جهت دار بدون حلقه نمایش داد. این گراف چند سطحی است و بدلیل ساختار آرایه ای بلوکهای منطقی FPGA، لزومی به تبدیل گراف به فرم دو سطحی برای تحقق نیست. از این جهت بهینه سازی به دو صورت امکان پذیر است. اولاً بهینه ساز تک تک گره های گراف را که خود دو سطحی هستند، تحت بهینه ساز ی دو سطحی قرار داده و سپس شیوه های منتظم بهینه سازی چند سطحی را روی کل گراف اعمال می نماید. سپس نرم افزار نگاشتگر تکنولوژی کل گراف را به تکه های کوچکتری تقسیم می نماید که هر یک توسط یک بلوک منطقی قابل تحقق هستند. سپس یکی از قابلیت های FPGA برای تحقق مدارهای چند سطحی، همانگونه که اشاره شد، در وجود وسعت عمل برای بهینه ساز است. نگاشتگر باید تلاش کند تا از حداقل بلوکهای منطقی موجود در FPGA استفاده کند که این موضوع به بهینه سازی مساحتی در این جا تعبیر می شود. از دیدگاهی متفاوت، ممکن است تلاش نگاشتگر به حداقل رساندن طبقات بلوکهای منطقی جهت بهینه سازی تأخیری مدار بوده و در این راستا کاهش تعداد بلوکها در درجه دوم اهمیت می باشد.

بعد از تصویر شدن مدار به درون بلوکهای منطقی FPGA، لازم است که در مورد محل قرار گرفتن هریک از بلوکها در داخل آرایه FPGA تصمیم گیری شود. نرم افزار جانمایی به چنین مسئله ای می پردازد. سعی یک الگوریتم فرضی جانمایی آن است که عمل جانمایی به نحوی صورت گیرد که مجموع طول اتصالات مورد استفاده برای همبندی بلوکها کمینه گردد. بایستی متذکر شد که فرآیند جانمایی در مورد FPGA ها مشابه آن چیزی است که در ارتباط با مدارهای VLSI پیاده سازی شده با سلولهای استاندارد انجام می پذیرد. گام نهایی در این سیستم CAD که توسط نرم افزار مسیر نمایی انجام می پذیرد، تخصیص قطعات سیم ها و انتخاب کلیدهای برنامه پذیر جهت برقراری ارتباط بین بلوکهای منطقی است. نرم افزار مسیر نمایی باید اطمینان حاصل کند که صد درصد اتصالات لازم قابل شکل گیری اند. یعنی تجهیزات اتصال تعبیه شده در سطح FPGA برای برقراری ارتباطات لازم کفایت می کند، که در غیر اینصورت مدار مورد نظر روی تنها یک FPGA قابل پیاده سازی نخواهد بود. به علاوه، گاهاً لازم می شود مسیر نمایی به نحوی صورت پذیرد که تأخیر روی مسیر کمینه گردد. در این مورد نیز

فرآیند مسیر نمایی مشابه فرآیند متناظر در محیط سلول های استاندارد است، با این تفاوت که در مورد FPGA ها تجهیزات مسیرنمایی یعنی قطعات سیم و کلیدها ثابتند. به محض تکمیل موفقیت آمیز مراحل جانمایی و مسیر نمایی، توسط واحد برنامه ریزی خروجی سیستم CAD به داخل FPGA سرازیر می گردد. کل فرآیند پیاده سازی یک مدار روی FPGA ممکن است از چند ثانیه تا حدود چند دقیقه به طول انجامد.

۱-۴-۵- فواید FPGA ها

شکل ۱-۵ مراحل پیاده سازی یک طرح روی FPGA و MPGA را مقایسه می کند[۵]. مراحل ورود و بازبینی طرح برای هر دو تکنولوژی مشابه است ولی تفاوت قابل توجهی در مراحل بعدی وجود دارد. با واگذاری عمل پیاده سازی نهایی طرح به کاربر از طریق تعبیه کلیدهای قابل برنامه ریزی، بخشی از مراحل طراحی برای FPGA حذف گردیده است.



شکل۱-۵: مقایسه مراحل پیاده سازی یک طرح روی FPGA و MPGA

۱. هزینه های پایین ابزار مورد استفاده

چنانچه لازم باشد طرحی روی MPGA پیاده سازی شود، لازم است که ماسک یا ماسک هایی جهت تعیین الگوهای سیم بندی ساخته شود. هر ماسک حدود چند هزار دلار قیمت دارد و هزینه ماسک طبیعتاً روی تعداد کل محصولات تولید شده سر شکن می گردد. از آنجاییکه موارد محدودی از پروژه ها نیاز به بیش از ده هزار واحد از محصول تولید شده دارند، لذا هزینه تمام شده ماسک به ازای هر قطعه نسبتاً بالاست. چنین مرحله ای برای FPGA وجود ندارد و از این بابت در تولیدات با تعداد کم FPGA مقرون به صرفه است.

۲. عرضه سریع محصول

ساخت یک طرح بر مبنای MPGA چندین هفته به طول می انجامد که عمده این زمان مربوط به ویژه سازی قطعه بر طبق خواسته کاربر است. این درحالیست که ویژه سازی FPGA در حدود چند دقیقه طول می کشد.

۳. ریسک کم

در مورد FPGA، تکرار طراحی ناشی از وجود خطا در یکی از مراحل هزینه زیادی نداشته و مدت زمان زیادی طول نمی کشد. این موضوع بدلیل هزینه کم NRE[۱] است. NRE معرف گام هایی از طراحی است که برای ارتقاء کیفیت و یا اصلاح یک طرح بایستی دستخوش تغییر و تحول گردند.

۴. بازبینی مؤثر طراحی

بدلیل هزینه های بالای NRE و زمان طولانی ساخت طرح مبتنی بر MPGA، کاربران MPGA بایستی طرح خود را قبل از ساخت به طور وسیع به وسیله شبیه سازی بازبینی نماید. در یک طرح MPGA که به وسیله شبیه سازی بازبینی شده است امکان بروز خطا وجود دارد چرا که استفاده از شبیه

۱ -Non Recurring Engineering

ساز سریع از دقت آن کاسته و مدل استفاده شده نیز برای تسریع کار شبیه ساز شده است. FPGA به مشکلات فوق مبتلا نیست و به جای استفاده از شبیه سازهای کند، کاربر قادر است FPGA را روی برد واقعی و در شرایط واقعی قرار داده و آنرا از نظر عملیاتی بازبینی کند.

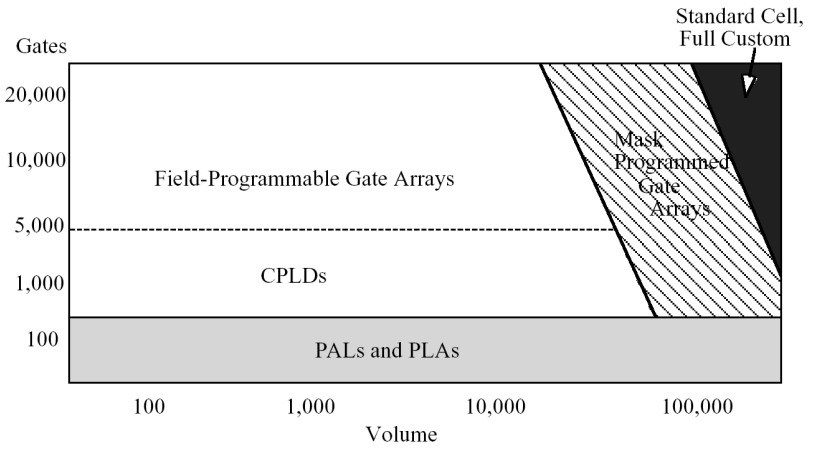
۵. هزینه پایین آزمون نهایی

تمام مدارهای مجتمع بایستی از نقطه نظر ساخت و بسته بندی مناسب مورد آزمایش قرار گیرند. طرح هایی که روی MPGA پیاده شده اند، شامل هزینه های اضافی مربوط به آزمون هستند. نظیر تولید برنامه آزمون و بکارگیری تجهیزات آزمون فیزیکی، وقتی قطعه بطور کامل ساخته شود. FPGA این هزینه ها را مرتفع نموده است. تهیه برنامه های خوب آزمون بسیار مشکل است و برای MPGA که طرح های متنوعی توسط سازنده روی آنها پیاده می گردد، بایستی هر بار این برنامه نوشته شود. لیکن چون FPGA ها در کارخانه و توسط سازنده ویژه سازی نمی گردند، بیش از یک برنامه برای آزمون تمام آنها لازم نیست و از این بابت معقول است که تلاش عمده ای روی تهیه این برنامه آزمون گذاشته شود.

۱-۴-۶- معایب FPGA ها

FPGA ها دارای هزینه سربار ناشی از وجود عناصر برنامه ریزی هستند. وجود این اجزاء علاوه بر تحمیل تأخیر انتشار سیگنال، از فضایی که به بخش فعال و عملیاتی اختصاص می یابد می کاهد. یک مقایسه سرعتی بین MPGA و FPGA بیانگر این نکته است که سرعت یک مدار نوعی پیاده شده روی FPGA حدود سه مرتبه کندتر از مدار مشابه پیاده شده روی MPGA است. همچنین گنجایش منطقی یک FPGA نوعی، یعنی میزان ناحیه فعّال و مفید نسبت به ناحیه غیرفعّال حدود ۸ تا ۱۲ برابر در تکنولوژی مشابه نسبت به MPGA کمتر است. علاوه بر این چنانچه حجم تولید بسیار بالا باشد هزینه کل نسبت به استفاده از MPGA بالاتر است. در جدول ۱-۶ هزینه کل تولید MPGA ها و FPGA ها نسبت به حجم تولید مقایسه شده است. مشاهده می شود که با گذر از FPGA به MPGA و عبور از تعدادی معین، تولید مدار با استفاده از MPGA توجیه پذیر می نماید[۶] .

جدول ۱-۶: مقایسه هزینه کل تولید MPGA و FPGA ها نسبت به حجم تولید



۱-۵- تکنولوژی برنامه ریزی

از نقطه نظر تکنولوژی برنامه ریزی، FPGA ها به سه گروه تقسیم می گردند:

۱. FPGA های مبتنی بر SRAM

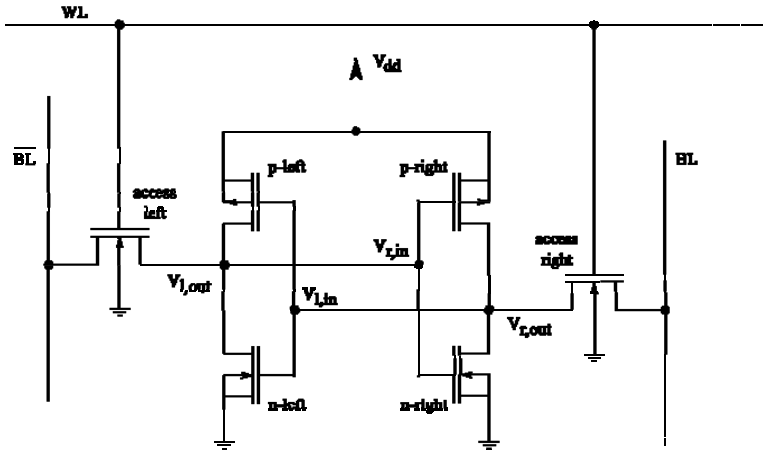
۲. FPGA های مبتنی بر آنتی فیوز

۳. FPGA های مبتنی بر EPROM یا EEPROM

یادآور می شود که دسته سوم عمدتاً در گروه CPLD ها قرار می گیرند و به آنها FPLD گفته می شود. لذا در تقسیم بندی اشاره شده، عنوان FPGA برای گروه سوم انتخاب نمی شود.

- تکنولوژی برنامه ریزی SRAM

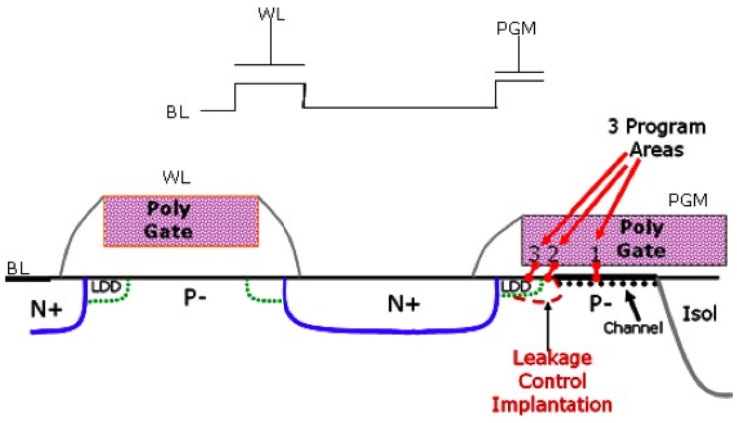
در این تکنولوژی سلول هایی که قابلیت برنامه ریزی و تغییر وضعیت را دارند از SRAM ساخته می شوند. واضح است که این تکنولوژی امکان برنامه ریزی مکرّر را به کاربر می دهد. یک نمونه از سلول SRAM در شکل۱-۶ دیده می شود. این شکل سلول SRAM متشکل از ۶ ترانزیستور اثر میدانی را به نمایش می گذارد. یک SRAM می تواند وظیفه کنترل یک سوئیچ را بر عهده داشته باشد. که از این امکان در برنامه ریزی بخش های مختلف FPGA استفاده می شود.



شکل۱-۶: سلول SRAM متشکل از ۵ ترانزیستور

- تکنولوژی برنامه ریزی آنتی فیوز

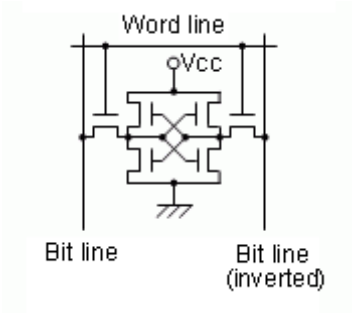
در این تکنیک سلول های قابل برنامه ریزی از عناصری بنام آنتی فیوز ساخته شده اند. آنتی فیوزها قطعاتی هستند که بر عکس فیوزها، در حالت عادی عایق جریان بوده و بعد از برنامه ریزی جریان را از خود عبور می دهند. شکل۱-۷ نمایی از یک آنتی فیوز را نشان می دهد. با اعمال جریان نسبتاً زیاد بین لایه POLY و DIFFUSION لایه میانی (لایه عایق) ذوب شده و به حالت عادی در می آید. لایه عایق ترکیبی شیمیایی بنام Oxide-Nitrogen-Oxide(ONO) است که ضخامتی حدود ۱۰ نانو متر دارد. ولتاژ مورد نیاز برای تغییر دادن وضعیت آنتی فیوز در حدود ۱۶ ولت است. آنتی فیوز در حالت عادی مقاومتی حدود ۱۰۰ گیگا اهم داشته و در حالت هدایت حدود چند ده اهم مقاومت از خود نشان می دهد.



شکل۱-۷: نمایی از یک آنتی فیوز

- تکنولوژی برنامه ریزی (E)EPROM

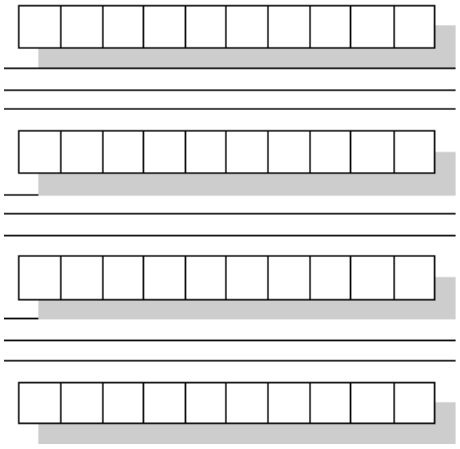
در این تکنولوژی از ترانزیستور ماسفت به عنوان عنصر نگهدارنده مقدار صفر یا یک استفاده می گردد. اما این ترانزیستور با ترانزیستور ماسفت معمولی اندکی تفاوت دارد. این ترانزیستور علاوه بر گیت عادی، دارای یک گیت شناور نیز است. برای برنامه ریزی نمودن سلول حافظه ساخته شده از این ترانزیستور ولتاژی به گیت شناور اعمال می شود که سبب می شود بار الکتریکی بین گیت شناور و گیت عادی به دام افتد. این بار ذخیره شده از این پس سبب می شود که ترانزیستور همیشه روشن باقی بماند. برای زدودن بار الکتریکی کافی است با اشعه ماوراء بنفش(uv) به قسمتی که بار را در خود ذخیره کرده است، تاباند و یا از طریق اعمال جریان الکتریکی این عمل را انجام داد. شکل۱-۸ ساختار EEPROM را نشان میدهد.



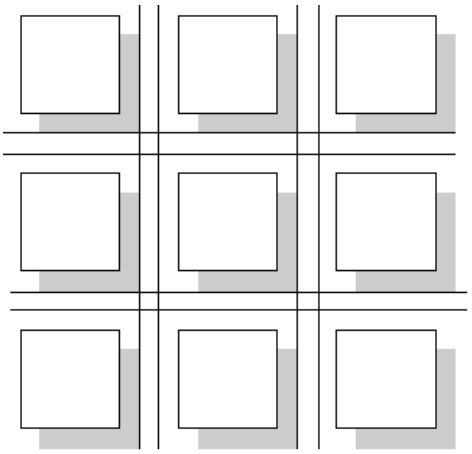
شکل۱-۸: ساختار EEPROM

۱-۶- گونه های موجود FPGA

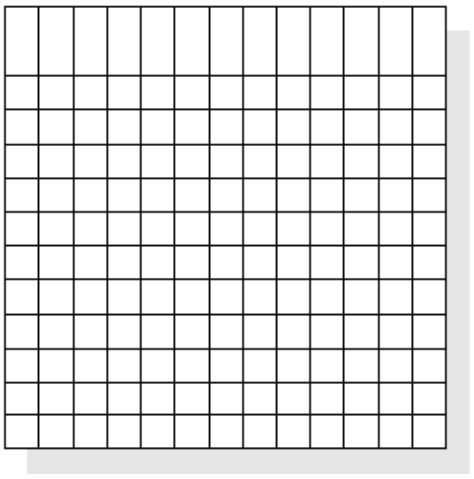
در سالهای گذشته شرکت های متعددی انواع مختلفی از FPGA ها را معرفی نموده اند. با وجود این که هر یک از این FPGA ها دارای ویژگی های منحصربفردی هستند، می توان از دیدگاه معماری آنها را به چهار دسته تقسیم کرد: آرایه متقارن، آرایه های ردیفی، PLD سلسله مراتبی و دریای گیت ها[۴]و[۵]. شکل ۱-۹ نمایشگر کلیات معماری این چهار دسته است.



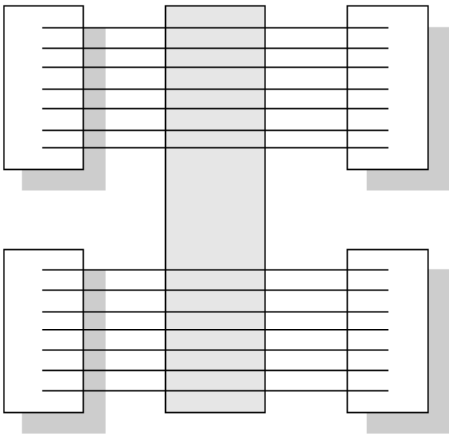
آرایه ردیفی



آرایه متقارن



دریای گیت ها



PLD سلسله مراتبی

شکل ۱-۹: معماری های مختلف FPGA

۱-۷- اجزای برنامه پذیر داخلی

۱-۷-۱- بلوکهای منطقی یا Logic Blocks

همانگونه که اشاره شد، FPGA ها عمدتاً بر دو دسته مبتنی بر SRAM و آنتی فیوز تقسیم می شوند. در دسته اول وظیفه برقراری اتصالات و شکل دهی CLB بر عهده بیت های SRAM نهاده شده است. لذا بطور ساده خود CLB متشکل از ساختاری است که از سلول های SRAM ساخته شده و می تواند تابع

منطقی را تحقق بخشد. این ساختار به جدول جستجو[1] موسوم است. یک جدول جستجو یک حافظه متشکل از 2^n سلول حافظه است که در اینجا n تعداد ورودیهای این LUT است. هر LUT با n ورودی قادر است توابعی با n متغیر و کمتر را تحقق بخشد.

واحد اصلی تشکیل دهنده بلوک منطقی در FPGAهای مبتنی بر SRAM، معمولاً LUT است. منظور از بکاربردن واژه واحد اصلی آن است که واحدهای فرعی دیگری نیز داخل CLB وجود دارند. از آن جمله می توان به فلیپ فلاپها، مالتی پلکسرها و گیت های ساده اشاره کرد. هدف از بکار بردن فلیپ فلاپها در داخل بلوک منطقی آن است که امکان تحقق مدارات ترتیبی بدون آنکه از خود CLB ها جهت تحقق عناصر فیدبک دار استفاده شود، وجود داشته باشد.

از مالتی پلکسرها نیز برای تولید خروجی های متنوع استفاده شده است. بعنوان مثال چنانچه لازم باشد، خروجی LUT بدون آنکه از میان فلیپ فلاپها بگذرد مستقیماً به خروجی CLB متصل می شود و یا حتی به یکی از ورودیهای خود LUT برگردانده می شود. استفاده از گیت های ساده نیز عمدتاً بخاطر سهولت بخشیدن به تهیه مکمل سیگنالهای میانی در داخل بلوک منطقی است.

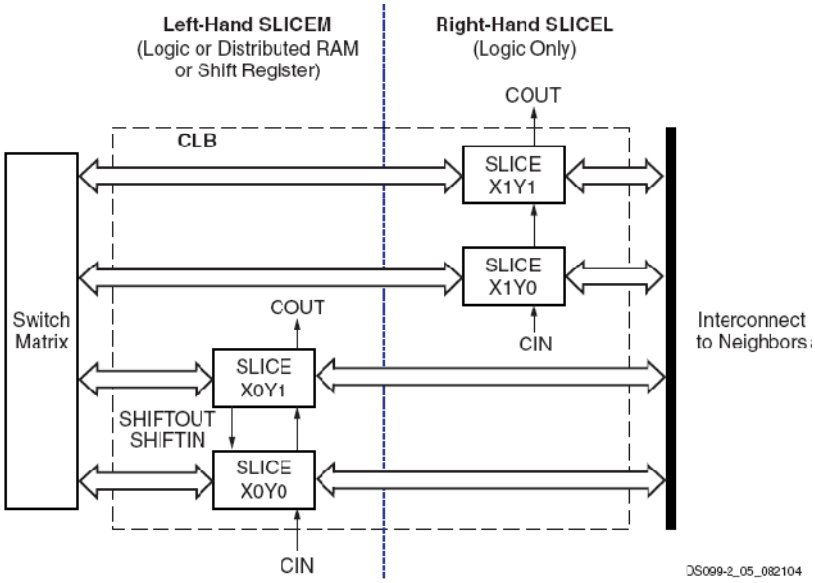
۱-۷-۱-۱- بلوک منطقی در FPGA های اسپارتان ۳ شرکت Xilinx

معماری اسپارتان۳ همانطور که در شکل ۱-۳ دیده می شود از نوع آرایه متقارن است[۷] . در FPGA های این سری، هر CLB از چهار بخش که هر کدام یک Slice نامیده می شود تشکیل شده است. هر Slice دارای ۲ عدد LUT چهار ورودی و ۲ عدد المان حافظه می باشد. المان های حافظه می توانند به شکل فلیپ فلاپ یا Latch بکار روند.

دو Slice نیمه چپ هر CLB، SLICEM نامیده شده و می تواند بعنوان تابع منطقی، حافظه ی RAM یا شیفت رجیستر بکار رود. دو Slice نیمه راست هر CLB، SLICEL نامیده شده و تنها بعنوان تابع منطقی بکار می رود. شکل ۱-۱۰ ساختار این دو Slice را نشان می دهد.

---

[1] -Look up Table



شکل ۱-۱۰: ساختار Slice ها در اسپارتان۳

قابلیت های دیگر CLB های این سری عبارتند از:

۱. منابع مشترک Slicem و Slicel

- دو عدد LUT چهار ورودی
- دو عدد المان حافظه
- دو عدد wide-function Mux
- Carry & arithmetic logic

لذا هر دوی این Slice ها می توانند در کاربردهای زیر بکار روند:

- پیاده سازی توابع منطقی
- عملکرد حسابی

۲. خروجی LUT ها می توانند به شکل عادی یا رجیستر شده از CLB خارج شوند.

۳. تأخیر انتشار یک LUT مستقل از تابع پیاده شده توسط آن می باشد.

۴. LUT های موجود در Slicem های هر CLB می توانند بعنوان شیفت رجیسترهای ۱۶ بیتی بکار روند.

۵. توابع با بیش از ۴ ورودی می توانند با سری کردن LUT ها یا با استفاده از مالتی پلکسرهای با ورودی زیاد[۱] ساخته شوند.

۶. المان حافظه قابلیت برنامه ریزی در حالت های D-FF و LATCH را دارند. این المان ها همچنین دارای سیگنالهای SET و RESET سنکرون و Clock Enable هستند.

۷. قطبش سیگنالهای کنترلی قابل انتخاب و مستقل از هم می باشند.

۱-۷-۲- کانالهای انتقال سیگنال[۲]

همانگونه که در شکل ۱-۳ مشاهده شد، قسمتی از فضای داخلی تراشه به شیارهای ارتباطی جهت انتقال سیگنال اختصاص یافته است. مسیر حرکت و نحوه اختصاص این کانالها در گونه های مبتنی بر SRAM و آنتی فیوز متفاوت است.

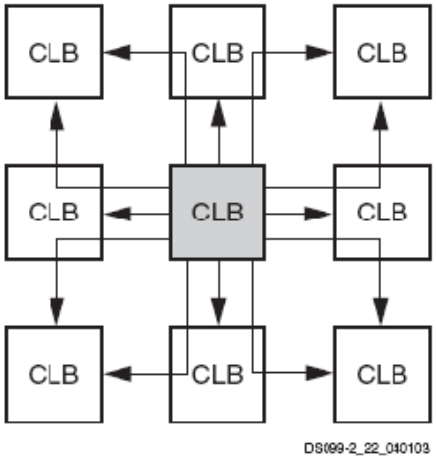
در FPGA های مبتنی بر SRAM در محل تقاطع کانالهای افقی و عمودی یک جعبه حاوی ماتریسی از سوئیچ ها[۳] قرار دارد که امکان برقراری اتصال بین شیارهای افقی و عمودی را فراهم می سازد. در اطراف بلوکهای منطقی نیز جعبه دیگری با عنوان جعبه اتصال[۴] تعبیه شده که توسط سوئیچ های داخل آن، شیارهای انتقال سیگنال به ورودی ها و خروجی های بلوک منطقی متصل می گردند. از آن جهت که وجود سوئیچ های قابل برنامه ریزی به تأخیر انتشار سیگنال اضافه می کند و گذشته از آن نیازی وجود ندارد که تمام خطوط انتقال سیگنال به طور کامل بیکدیگر متصل گردند. تعداد این سوئیچ ها در هر دو نوع از جعبه ها محدود است. شکل۱-۱۱ خواننده را جهت درک بهتر وضعیت و ساختار معماری مسیر نمایی یاری خواهد کرد.

در مورد FPGA هایی که در آنها از آنتی فیوز استفاده شده، جعبه های مذکور بدلیل ساختار خاص این گونه از تراشه ها وجود ندارند. شیارهای عمودی و افقی سیگنال در سراسر تراشه گسترده شده اند.

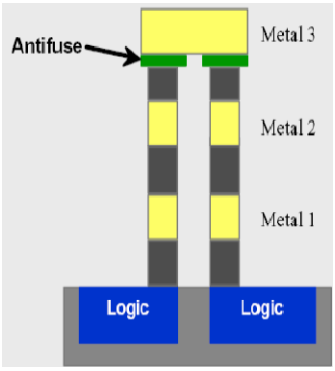
---

۱ - Wide multiplexer
۲ - Routing Channels
۳ - Switch Box
٤ - Connection Box

شیارهای افقی انتقال سیگنال بین طبقات مختلف بلوکهای منطقی قرار گرفته اند. شیارهای عمودی با شیارهای افقی مذکور تلاقی داشته و گذشته از آن روی ردیفهای بلوکهای منطقی با کانالهای افقی انتقال سیگنال را فراهم می سازد. در شکل ۱-۱۲ نمایی از اتصالات داخلی یک FPGA مبتنی بر آنتی فیوز دیده می شود.



شکل۱-۱۱: معماری کلی مسیر نمایی



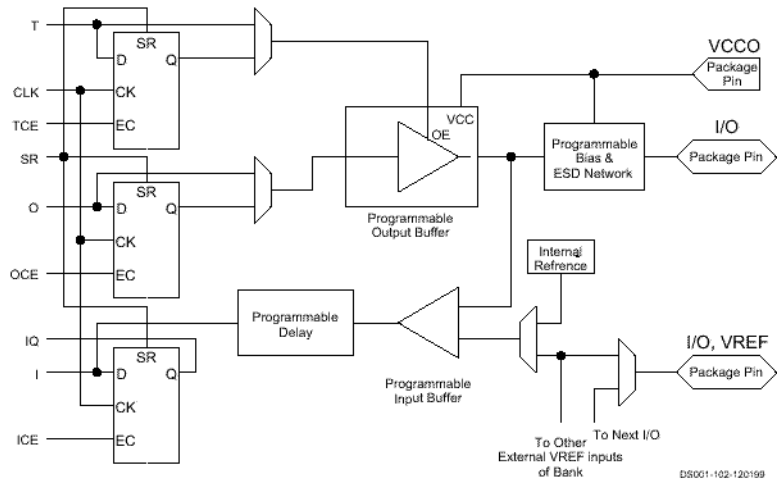
شکل۱-۱۲: نمایی از اتصالات داخلی یک FPGA مبتنی بر آنتی فیوز

وجود سوئیچ های قابل برنامه ریزی در سر راه انتقال سیگنال به تأخیر انتشار سیگنال اضافه میکند و از آنجایی که در مدارهای منطقی احتمالاً نیاز به انتقال سیگنالهای فراگیر نظیر پالس ساعت و سیگنالهای کنترلی وجود دارد، لذا خطوط ویژه ای در سراسر FPGA قرار گرفته اند که دارای کمترین سوئیچ در طول مسیر خود هستند. از طرفی ارتباط بین بلوکهای مجاور ضروری است (سیگنالهای محلی). بنابراین

خطوط انتقال سیگنال دیگری نیز در سراسر FPGA گسترده شده اند که دارای تعداد زیادی از سوئیچ های قابل برنامه ریزی در سر راه خود هستند. بنابراین چنانچه طول خطوط انتقال سیگنال را به فاصله بین دو سوئیچ قرار گرفته در مسیر آنکه هیچ سوئیچ دیگری بین آن دو وجود ندارند تعبیر کنیم، می توان گفت که در داخل هر FPGA خطوط انتقال سیگنال با طول های متفاوت وجود دارد.

۱-۷-۳- بلوکهای ورودی خروجی یا I/O BLOCKS

بمنظور آنکه امکان برقراری ارتباط دو طرفه با جهان خارج از FPGA و داخل آن وجود داشته باشد، بلوکهای I/O با استانداردهای سیگنال مختلفی در تمام FPGA ها وجود دارند. شکل۱-۱۳ یک ساختار ورودی/خروجی نوعی را نشان می دهد[۷].



شکل۱-۱۳: ساختار ورودی/خروجی نوعی FPGA

وجود فلیپ فلاپها در بلوک های I/O سبب بهبود بازده سرعتی FPGA می شود. چون در مرحله زمانبندی مجدد در فرآیند بهینه سازی سعی بر آن است که عناصر تأخیر به حواشی مدار منتقل گردند.

۱-۸- تراشه ها و بردهای جدید

تراشه های جدید FPGA که تا کنون توسط شرکت Xilinx ارائه شده است به شرح زیر می باشد:

تراشه های این شرکت در چهار دسته کلی خانواده های SPARTAN، VIRTEX و اخیراً تراشه های ARTIX و KINTEX تقسیم می شوند. تفاوت این خانواده ها در کارآیی، مصرف توان، مقدار LOGIC CELL، بلوک RAM، تواناییهای مدارات واسط و... است[۷]. در جدول ۱-۲ مقایسه ای بین تراشه های برتر این شرکت صورت پذیرفته است.

جدول ۱-۲: مقایسه ویژگیهای تراشه های برتر شرکت Xilinx

ویژگیها	Artix-۷	Kintex-۷	Virtex-۷	Spartan-٦	Virtex-٦
Logic Cells	۳٥۲,۰۰۰	٤۸۰,۰۰۰	۲,۰۰۰,۰۰۰	۱٥۰,۰۰۰	۷٦۰,۰۰۰
Block RAM	۱۹Mb	۳٤Mb	۸٥Mb	٤.۸Mb	۳۸Mb
DSP Slices	۱,۰٤۰	۱,۹۲۰	٥,۲۸۰	۱۸۰	۲,۰۱٦
DSP Performance (symmetric FIR)	۱,۱۲۹GMACS[۱]	۲,٤٥۰GMACS	٦,۷۳۷GMACS	۱٤۰GMACS	۲,٤۱۹GMACS
Transceiver Count	۱٦	۳۲	۹٦	۸	۷۲
Transceiver Speed	٦.٦Gb/s	۱۲.٥Gb/s	۲۸.۰٥Gb/s	۳.۲Gb/s	۱۱.۱۸Gb/s
Total Transceiver Bandwidth (full duplex)	۲۱۱Gb/s	۸۰۰Gb/s	۲,۷۸٤Gb/s	٥۰Gb/s	٥۳٦Gb/s

۱- Giga multiply-accumulates per second

ویژگیها	Artix-۷	Kintex-۷	Virtex-۷	Spartan-٦	Virtex-٦
Memory Interface (DDR۳)	۱,۰٦٦Mb/s	۱,۸٦٦Mb/s	۱,۸٦٦Mb/s	۸۰۰Mb/s	۱,۰٦٦Mb/s
PCI Express®	Gen۲x٤	Gen۲x۸	Gen۳x۸	Gen۱x۱	Gen۲x۸
Agile Mixed Signal (AMS)/XADC	Yes	Yes	Yes		Yes
Configuration AES	Yes	Yes	Yes	Yes	Yes
I/O Pins	٦۰۰	۵۰۰	۱,۲۰۰	۵۷٦	۱,۲۰۰
I/O Voltage	۱.۲V, ۱.۳۵V, ۱.۵V, ۱.۸V, ۲.۵V, ۳.۳V	۱.۲V, ۱.۳۵V, ۱.۵V, ۱.۸V, ۲.۵V, ۳.۳V	۱.۲V, ۱.۳۵V, ۱.۵V, ۱.۸V, ۲.۵V, ۳.۳V	۱.۲V, ۱.۵V, ۱.۸V, ۲.۵V, ۳.۳V	۱.۲V, ۱.۵V, ۱.۸V, ۲.۵V

بردهای ارائه شده توسط شرکت بسته به کاربردها و استفاده از آن متفاوت می باشد. بردها در رده های کلی زیر ارائه شده است:

- Broadcast
- Automotive
- Consumer
- Industrial
- Wired Specific Boards

بعنوان نمونه یکی از جدیدترین بردهای ارائه شده توسط شرکت در ذیل معرفی شده است.

Virtex-٦ FPGA ML٦٠٥ Evaluation Kit

برد ارزیابی ML٦٠٥ دارای تمام اجزای اصلی سخت افزار های مورد نیاز برای طراحی سیستمی با عملکرد و کارآیی بالا، اتصالات سریال و واسط حافظه پیشرفته می باشد[٧]. برخی از ویژگیهای این برد در زیر قابل مشاهده است:

FPGA: XC٦VLX٢٤٠T-١FFG١١٥٦

Configuration

- Onboard configuration circuitry (USB to JTAG)
- ١٦MB Platform Flash XL
- ٣٢MB Parallel (BPI) Flash
- System ACE CF with ٢GB Compact FLASH (CF) Card

Communications and Networking

- ١٠/١٠٠/١٠٠٠ Tri-Speed Ethernet (GMII, RGMII, SGMII, MII)
- SFP transceiver connector
- GTX port (TX, RX) with four SMA connectors
- USB To UART Bridge
- USB Host Port and USB Peripheral Port
- PCI Express x٨ Edge Connector (card supports up to x٤ Gen٢ with Virtex-٦ LX٢٤٠T-١ silicon)

Memory

- DDR٣ SO-DIMM (٥١٢ MB)
- BPI Linear Flash (٣٢ MB) (Also available for configuration)
- IIC EEPROM (٨ Kb)

Clocking

- ٢٠٠ MHz Oscillator (Differential)
- ٦٦ MHz Socketed Oscillator (Single-Ended)
- SMA Connectors for external clock (Differential)
- GTX Reference Clock port with ٢ SMA connectors

Input/Output and Expansion Ports

- ١٦x٢ LCD character display

- DVI Output
- System Monitor
- User Push buttons (٥), DIP switches (٣), LEDs (١٣)
- User GPIO with two SMA connectors
- Two FMC Expansion Ports
- High Pin Count (HPC)
- Eight GTX Transceivers
- ١٦٠ SelectIOs
- Low Pin Count (LPC)
- One GTX Transceiver
- ٦٨ SelectIOs

  Power
- ١٢V wall adapter or ATX
- Voltage and Current measurement capability of ٢.٥V, ١.٥V, and ١.٢V, ١.٠V supplies

در بردهای جدید علاوه بر تراشه های FPGA از پردازنده های DSP، INTEL ATOM و برخی از پردازنده های دیگر در کنار FPGA برای بهبود و افزایش قدرت پردازش سیستم استفاده شده است[٧].

١-٩- ساختار پایان نامه

این پایان نامه شامل ۶ فصل است. معرفی و بررسی روند تکامل تراشه های FPGA، کاربردها، اهداف و موضوعات اصلی طرح در فصل اول مورد بررسی قرار گرفته است. در فصل دوم به بررسی پردازش موازی، ابررایانه ها و استفاده از FPGA در پردازش موازی پرداخته شده است. بررسی موازی سازی، انواع سیستم های پردازش موازی و استفاده از آنها از جمله موضوعاتی است که در فصل ۲ به آنها اشاره شده است.

در فصل سوم به روند طراحی و انتخاب قطعات، بررسی جزئی تراشه FPGA اسپارتان۳، دلایل انتخاب و قابلیتهای این پردازنده خاص پرداخته شده است. بررسی کامل قسمتهای مختلف FPGA نظیر بلوکهای

منطقی برنامه پذیر، بلوکهای رم، مدیریت کلاک دیجیتال، حلقه قفل تأخیر، بلوکهای ورودی/خروجی و بلوکهای ضرب کننده در این بخش صورت پذیرفته است.

فصل چهارم به معرفی، راه اندازی و پیاده سازی طرح بر روی سخت افزار اختصاص دارد. در این فصل بطور خاص به ارائه بلوک دیاگرام کلی مدار، معرفی و بررسی بلوکهای مختلف طراحی شده و ارتباطات آنها با یکدیگر، مراحل بکارگیری سخت افزار و تست برنامه های نمونه آزمایشی و ارتباطی در سخت افزار پرداخته شده است.

در فصل پنجم به بررسی الگوریتم تشخیص لبه تصاویر، بعنوان برنامه نمونه پیاده شده بر روی FPGA و نحوه برقراری ارتباط بین سخت افزار و دنیای بیرون از آن پرداخته شده است. ارتباط بین سخت افزار و کامپیوتر و ارسال و دریافت اطلاعات، آشنایی با نحوه برنامه ریزی FPGA از جمله بخش های این فصل می باشد.

در فصل ششم نیز به مرور کلی، نتیجه گیری بر این طراحی و ارائه پیشنهادات برای کارهای آتی و پیش رو خواهیم پرداخت.

فصل دوم

پردازش موازی و FPGA

۲-۱- مقدمه

در این فصل به بررسی مبحث پردازش موازی و مسائل مرتبط با آن خواهیم پرداخت. در ابتدا به بررسی ابررایانه ها، سیستم های پردازش موازی، دسته بندی آنها و در ادامه به کاربردهای FPGA در پردازش موازی اشاره کرده و در انتهای این فصل کارآیی موازی سازی در سیستم ها را مورد بررسی قرار می دهیم.

۲-۲- اهمیت موازی سازی

امروزه بشر برای محاسبه و حل مسائلی از جمله مسائل فیزیک کوانتوم، هواشناسی، بررسی تغییرات آب و هوا (از جمله تحقیق در مورد گرم شدن زمین) مدل سازی مولکولی (مطالعه ساختارها و محتویات ترکیبات شیمیایی، بلورها و ...)، شبیه سازی فیزیکی (شبیه سازی هواپیما در تونل هوا، شبیه سازی انفجار سلاح های هسته ای و تحقیق در مورد هم جوشی هسته ای)، رمزنگاری و ... در دانشگاههای بزرگ، مراکز نظامی و آزمایشگاههای تحقیقاتی نیازمند سامانه های پردازش موازی[۱] و ابررایانه ها[۲] است که برای پاسخگویی به نیاز سرعت بوجود آمده اند. موضوع مورد اشاره از جمله مواردی است که بیش از پیش مورد توجه نهاد های علمی و تحقیقاتی قرار گرفته است[۸]. در این پایان نامه نیز مختصراً به این موضوع و اهمیت آن پرداخته شده است.

۲-۳- پردازش موازی

پردازش موازی اجرای یک فرآیند بطور همزمان روی چندین پردازنده، برای کوتاه کردن زمان رسیدن به جواب است. ایده ی این روش بر این مبنا می باشد که هر مسئله بطور معمولی قابل قسمت به چندین مسئله به اندازه ی کوچکتر است که این مسئله های کوچکتر می توانند بصورت همزمان حل شده و در نهایت با هم ترکیب شوند تا نتیجه ی نهایی سریع تر بدست آید. استفاده از چندین پردازنده و بکاربستن

---

۱ -Parallel Processing
۲ - Super Computer

آنها بصورت همزمان و در واقع تقسیم برنامه اصلی بین پردازنده ها در یک طرح کلی یا استفاده از پردازنده هایی که قابلیت اجرای موازی طرح ها را دارند (FPGA) برای این امر راهگشا است[۸]. با توجه به این موضوع که بسیاری از الگوریتم های سطح پایین در پردازش سیگنال، دارای عملیات های تکراری هستند و ذاتاً برای موازی سازی مناسب هستند، می توان از موازی سازی سخت افزاری عملیات پردازش سیگنال، برای بهبود زمان محاسباتی بهره جست. بعنوان مثال برای اعمال موازی سازی در پردازش های تصویر سطح پایین تقسیم بندی یک تصویر به قسمتهای جداگانه و توزیع آنها روی شبکه ای از پردازنده ها است که این عمل به موازی سازی داده معروف است و یک روش عمومی و مؤثر در پردازش های سطح پایین برای موازی سازی می باشد [۱۹].

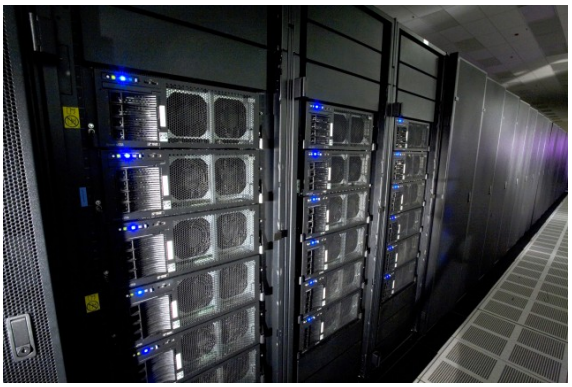
۲-۴- ابررایانه ها

ابررایانه به رایانه ای گفته می شود که در زمان معرفی آن، مقام نخست میزان ظرفیت محاسبه در واحد زمان را در دنیا داشته باشد. عبارت "ابررایانه" برای اولین بار توسط مجله ((نیویورک ورلد))[۱] در سال ۱۹۲۹، برای اشاره به جدول سازهای آی بی ام[۲] در دانشگاه کلمبیا بکار رفت. ابررایانه های امروز، رایانه های شخصی فردا هستند. برخی از ابررایانه های گذشته مانند Cray-۱ که اوایل دهه ۸۰ میلادی در شبیه سازهای با مقیاس بزرگ به کار رفته است، رایانه های موازی نبودند. اما امروزه همه ابررایانه ها، رایانه های موازی هستند. برخی از آنها با بکارگیری پردازنده ها و شبکه های ویژه ساخته می شوند. ولی بیشتر آنها بر اساس سخت افزارهای معمول و سیستم های عامل و نرم افزارهای متن باز[۳] ساخته می شوند. سرعت ابررایانه ها بر اساس FLOPS[۴] محاسبه می شود که مخفف عملیات ممیز شناور در هر لحظه است و معمولاً هم یک پسوند SI مثل ترا یا پتا با آن همراه است[۸].

۲-۴-۱- سریعترین ابررایانه های کنونی

۱ - New York World Newspaper
۲ -IBM
۳ -Open Source
٤ -Floating Point Per Seconds

ابررایانه K computer, SPARC٦٤ VIIIfx ٢.٠GHz ساخت ژاپن در سال ٢٠١١ سریعترین ابررایانه ارائه شده با سرعت Gflops ٨٧٧٣٦٣٢ می باشد[٩] . ابررایانه Roadrunner ساخت IBM در ژوئن ٢٠٠٨ سریع ترین ابررایانه ای بود که با قدرت پردازش PFLOPS ١٠٢٦ به بازار ارائه شد. در شکل ٢-١ نمایی از این ابررایانه قابل مشاهده است.



شکل٢-١- ابررایانه Roadrunner ساخت IBM

BLUE GENE/L ساخت IBM، با قدرت پردازشی TFLOPS ۴٧٨ یکی از برترین ابررایانه است که در سال ٢٠٠٧ بعنوان سریعترین ابررایانه دنیا شناخته شده بود.

لیستی از ١٠ ابررایانه برتر دنیا در جدول ٢-١ نشان داده شده است. اکثر این رایانه ها از نوع MPP[١] که از انواع پیشرفته تر خوشه های رایانه ای است، بشمار می آیند.

جدول٢-١- ابررایانه های برتر دنیا

Rank	Site	Manufacturer	Computer	Country	Year	Processor
١	RIKEN Advanced Institute for Computational Science (AICS)	Fujitsu	K computer, SPARC۶۴ VIIIfx ٢.٠GHz, Tofu	Japan	٢٠١١	SPARC٦٤ VIIIfx ٢٠٠٠ MHz

١ -Massive Parallel Processing

			interconnect			
۲	National Supercomputing Center in Tianjin	NUDT	NUDT TH MPP, X۵۶۷۰ ۲.۹۳Ghz ۶C, NVIDIA GPU, FT-۱۰۰۰ ۸C	China	۲۰۱۰	Intel EM۶۴T
۳	DOE/SC/Oak Ridge National Laboratory	.Cray Inc	Cray XT۵-HE Opteron ۶-core ۲.۶ GHz	United States	۲۰۰۹	AMD x۸۶_۶۴
۴	National Supercomputing Centre in Shenzhen ((NSCS	Dawning	Dawning TC۳۶۰۰ Blade, Intel X۵۶۵۰, NVidia Tesla C۲۰۵۰ GPU	China	۲۰۱۰	Intel EM۶۴T
۵	GSIC Center, Tokyo Institute of Technology	NEC/HP	HP ProLiant SL۳۹۰s G۷ Xeon ۶C X۵۶۷۰, Nvidia GPU, Linux/Windows	Japan	۲۰۱۰	ntel EM۶۴T
۶	DOE/SC/LBNL/NERSC	Cray Inc.	Cray XE۶ ۱۲-core ۲.۱ GHz	United States	۲۰۱۰	AMD x۸۶_۶۴
۷	Commissariat a l'Energie Atomique ((CEA	Bull SA	Bull bullx super-node S۶۰۱۰/S۶۰۳۰	France	۲۰۱۰	Intel EM۶۴T
۸	DOE/NNSA/LANL	IBM	BladeCenter QS۲۲/LS۲۱ Cluster, PowerXCell ۸i ۳.۲ Ghz / Opteron DC	United States	۲۰۰۹	Power
۹	National Institute for Computational Sciences/University of Tennessee	Cray Inc.	Cray XT۵-HE Opteron ۶-core ۲.۶ GHz	United States	۲۰۰۹	AMD x۸۶_۶۴
۱۰	Forschungszentrum (Juelich (FZJ	IBM	Blue Gene/P Solution	Germany	۲۰۰۹	Power

۲-۵- انواع سیستم های پردازش موازی

۲-۵-۱- دسته بندی بر اساس ساختار کنترلی

یک دسته بندی مورد قبول برای رایانه های موازی توسط مایکل فلین[۱] در اواسط دهه ۶۰ میلادی ارائه شد. این دسته بندی بر اساس مدل برنامه نویسی، آن ها را به (تک دستور/چند داده) SIMP، (چند دستور/چند داده) MIMD، (تک دستور/ تک داده) SISD، (چند دستور/تک داده) MISD تقسیم بندی کرد. در یک رایانه SIMD مانند CM-۲ یا رایانه های دهه ۸۰ کمپانی NCUBE، همه پردازنده ها در هر کلاک رایانه عملیات مشابه را انجام می دهند که توسط Control Unit کنترل می شود. در این مدل که با عنوان برنامه با داده موازی شناخته می شود زبانهای سطح بالا مانند C استفاده می شوند و محاسبات و انتقال اطلاعات بین پردازشها در هر کلاک همزمان روی داده های گوناگون اجرا می شوند[۸][۱۰].

در یک سیستم MIMD[۲] هر یک از واحدهای پردازشی به شکل موازی عملیات خود را مستقل از هم اجرا می کنند و همزمانی در تناوبهای زمانی مشخص با بکارگیری رد و بدل پیام انجام می شود. در پردازنده های FPGA نیز از چنین ساختاری استفاده شده است. بدین صورت که برنامه هایی که در بدنه Process یک طرح کلی می آیند، بصورت همزمان و با هم اجرا می شوند که هر برنامه می تواند چندین بدنه Process داشته و بدین صورت، عملیات پردازش موازی و موازی سازی برنامه ها را عملی نمود. همچنین هم پخش موازی داده ها و هم ردوبدل پیام و همزمانی تحت کنترل کاربر است. برای نمونه از سیستم های MIMD می توان به Intel Gamma و Touch Stone Delta که دارای پردازنده های کمتر امّا نیرومندتری هستند، Gray-۹۰ و همچنین اولین نسل IBM SP۲ اشاره کرد. با اینکه معمولاً بخاطر یکنواختی بین پردازشها، طراحی کامپایلر و برنامه ها برای ابر رایانه ها ی SIMD ساده تر است، بیشتر این سیستم ها دارای کارآیی محاسباتی بسیار پایینی هستند. دلیل این امر انعطاف پذیری پایین آنها در اجرای بخش هایی از برنامه است که بخش های مشابهی برای موازی سازی نداشته باشد. چنین موردی

---

۱ -Michael Flynn
۲ -Multiple Instruction Multiple Data

در سیستم های پردازشگر مبتنی بر FPGA کمتر مشاهده می شود. بدلیل اینکه FPGA بخش های مشابهی برای موازی سازی را داراست.

رایانه های MIMD می توانند هم به شکل حافظه مشترک ساخته شوند مانند SGI Origin ۲۰۰۰ و هم بصورت حافظه پراکنده[۱] مانند سیستم IBMSP، حافظه اشتراکی به این معنی است که دسترسی به یک فضای آدرس توسط همه پردازنده ها با بکار گیری یک رویه[۲] همزمان کننده میسر است. در سیستم های با حافظه غیر اشتراکی، رویه های جداگانه برای انتقال اطلاعات مورد نیاز است. یکی از روشهای متداول امروزی برای برنامه نویسی این است که حافظه فیزیکی پخش شده باشد ولی حافظه آدرس مشترک باشد. این روش انعطاف پذیری بالا را بهمراه برنامه نویسی ساده تر فراهم می آورد.

۲-۵-۲- دسته بندی بر اساس نوع ارتباط بین پردازنده ها

علاوه بر دسته بندی بالا که بر اساس ساختار کنترلی استوار است، از دیدگاهی دیگر رایانه های موازی را می توان بر اساس نوع ارتباط بین پردازنده ها دسته بندی کرد. دو روش اصلی انتقال داده بین پردازنده ها عبارتند از:

انتقال داده با نوشتن و خواندن در یک فضای آدرس مشترک (shared address space) و انتقال داده با بکار گیری رد و بدل پیام.

۲-۶- رایانه های با فضای آدرس مشترک

در این دیدگاه یک فضای آدرس که بین همه پردازنده ها مشترک است وجود دارد که همه پردازنده ها با ایجاد تغییر در داده های موجود در این فضا با یکدیگر کنش دارند. به رایانه هایی که فضای آدرس مشترک دارند و برنامه نویسی تک برنامه/چند داده (SIMD) را پشتیبانی می کنند چند پردازنده ای (Single program/Multiple data) نیز می گویند. اگر مدت زمان لازم برای دسترسی به هر کلمه در

۱ -Distributed Memory
۲ -Procedure

سیستم (محلی یا عمومی) برابر با بقیه باشد به آن رایانه با دسترسی یکنواخت به حافظه (UMA[1]) می گویند و در غیر اینصورت به آن رایانه با دسترسی غیر یکنواخت (NUMA[2]) گویند.

در اینجا لازم است تفاوت بین رایانه های با حافظه مشترک و رایانه های با فضای آدرس مشترک که در بالا توضیح داده شد مشخص شود. رایانه های با فضای حافظه مشترک به رایانه هایی گفته می شود که در آنها از جهت فیزیکی یک بخش حافظه وجود داشته باشد که همه پردازنده ها به آن دسترسی داشته باشند. ولی در رایانه با فضای آدرس مشترک ممکن است حافظه از جهت فیزیکی پخش شده باشد و هر پردازنده بخش حافظه مربوط به خود را داشته باشد، اما کماکان حافظه ها در یک فضای آدرس دهی مشترک برای پردازنده ها قرار داشته باشند.

۲-۷- رایانه های انتقال پیام[3]

در این دیدگاه کل سیستم بصورت مجموعه ای از گره های محاسباتی دیده می شود که هریک دارای فضای آدرس جداگانه هستند. هر گره می تواند یک تک پردازنده و یا یک سیستم چند پردازنده ای داشته باشد. انتقال اطلاعات بین پردازنده ها از طریق یک سری پیام صورت می پذیرد. پایه رد و دل داده ها دو عمل ارسال و دریافت پیام ها است و باقی عملیات با بکار گیری این دو عمل اصلی انجام می گیرد. همچنین از آنجایی که در یک ارسال پیام، مقصد باید مشخص باشد، هر پردازنده باید یک شماره مشخصه داشته باشد. علاوه بر این هر پردازنده باید اطلاعاتی مانند تعداد پردازشهای در حال اجرا نیز داشته باشد. واسط های برنامه نویسی متعددی برای برنامه نویسی با انتقال پیام طراحی شده که از بین آنها می توان MPI[4] و PVM[5] را نام برد.

۲-۸- خوشه های رایانه ای

---

۱ -Uniform Memory Access
۲ -No Uniform Memory Access
۳ -Message Passing
٤ -Message Passing Interface
٥ -Parallel virtual machine

خوشه های رایانه ای دسته ای از سامانه های موازی اند که از اتصال شماری از رایانه های شخصی تشکیل می شوند. بدلیل تولید انبوه رایانه های شخصی، قیمت آنها و در نتیجه خوشه های رایانه ای بسیار کمتر از دیگر سامانه های موازی با ساختارهای ویژه است. در مقایسه، افزایش قیمت پردازنده ها (یا سامانه های ویژه) نسبت به توان پردازشی آنها بصورت نمایی است. در حالیکه افزایش قیمت خوشه ها، ضریبی خطی از شمار رایانه های بکار رفته در آنها (و در نتیجه ضریب خطی از توان پردازشی شان) است. این قیمت کمتر، سبب روی آوردن هرچه بیشتر خوشه های رایانه ای شده است.

لیستی از تعدادی از ابر رایانه های برتر دنیا در بخش پیوست ضمیمه د ارائه شده است[۹].

۲-۹- بررسی کارآیی موازی سازی

اخیراً مباحثی در پردازش سریع داده‌ها مطرح شده است که ایجاب می کند با یک سرمایه گذاری اصولی راه کارهای مطرح شده برای آنها را بررسی و مورد تفحص قرار داد. این مباحث نه تنها در علم کامپیوتر، بلکه در تمام علوم که احتیاج به ابزارهایی برای کاهش زمان پردازش دارند می تواند وجود داشته باشد. در زیر چند نمونه از کارهایی که می توان با استفاده از پردازش موازی انجام داد ذکر شده است:

۱- محاسبات با سرعت زیاد

۲- سرویس دهنده های حجیم اطلاعات و شبکه

۳- سرویس دهنده های وب با کارایی بالا

۴- شبیه سازی های حجیم مثل شبیه سازی پرواز، شبیه سازی پیش بینی آب و هوای سه بعدی

۵- سرویس دهنده های دانشگاه های مجازی و ...

یکی از معیارهای بررسی کارآیی موازی سازی، افزایش سرعت است (Speed up). چیزی که در پردازنده های FPGA بدلیل ساختار آنها با پردازش موازی و اجرای همزمان برنامه ها بدنبال رسیدن به

این مهم هستیم. اندازه افزایش سرعت برابر است با زمان اجرای سریعترین الگوریتم پیاپی شناخته شده در سخت ترین حالت بر زمان اجرای الگوریتم موازی در سخت ترین حالت.

$$S = \frac{T(I)}{T(N)} \qquad (۱-۲)$$

با جایگذاری T(N) در معادله ۲-۱:

$$S = \frac{T(I)}{T_{ser} + T_{par} + T_{oh}} \qquad (۲-۲)$$

که در آن:

T (i): زمان اجرای یک الگوریتم بوسیله i پردازنده

T_{ser}: زمان اجرای بخش های پیاپی[۱] برنامه های موازی

T_{par}: زمان اجرای بخش های موازی برنامه موازی

T_{oh}: زمان سربار در برنامه موازی است.

کارآیی سامانه های موازی را می توان با قوانین آمدال و گوستافسون[۲] بررسی کرد.

۲-۹-۱- قانون آمدال: این قانون برای محاسبه بیشینه افزایش سرعت به ازای افزایش شمار پردازنده ها بکار می رود. در اینصورت این قانون به شکل زیر نمایش داده می شود:

$$S = \frac{۱}{F + \frac{(۱ - F)}{N}} \qquad (۳-۲)$$

۱ -Sequential
۲ - Amdahl & Gustafson's law

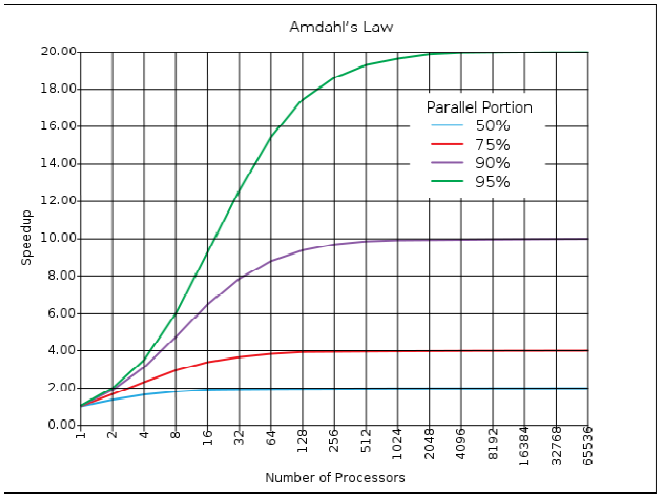
که در آن:

S: بیشینه افزایش سرعت

F: کسری از برنامه که پیاپی است و نمی تواند موازی شود.

N: شمار پردازنده ها

هنگامیکه N به سوی بینهایت برود، A بسوی $\frac{۱}{F}$ میل خواهد کرد. به این معنی که بیشینه افزایش کارآیی $\frac{۱}{F}$ خواهد بود. قانون آمدال که در سال ۱۹۶۷ توسط جین آمدال، یکی از طراحان آی.بی.ام بیان شد، یکی از انگیزه هایی است که سبب شد پردازش موازی دیرتر بکارگرفته شود.



شکل۲-۲- قانون آمدال (افزایش سرعت در موازی سازی وابسته به اندازه بخش های پیاپی برنامه است)

۲-۹-۲- قانون گوستافسون: این قانون بیان می کند که هر مسئله بزرگ می تواند به خوبی موازی شود. این قانون بسیار نزدیک به قانون آمدال است. این قانون را می توان به شکل زیر نشان داد:

$$S = \frac{۱}{F + s = p - \propto (p - ۱) = \propto + p(۱ - \propto)} \qquad (۴-۲)$$

که در آن:

S: افزایش سرعت

P: شمار پردازنده ها

∝: بخشی از برنامه که نمی تواند موازی شود

می توان تعریف کرد که:

$$a(n) + b(n) = ۱ \qquad (۵-۲)$$

که n اندازه مسئله و b بخشی از مسئله است که می تواند موازی شود. اگر با افزایش اندازه مسئله، $a(n)$ کم شود، آنگاه اگر n به سمت بی نهایت میل کند، S به سوی P میل خواهد کرد.

تفاوت قانون گوستافسون و آمدال در این است که قانون آمدال، اندازه بخش پیاپی را نسبت به کل ثابت می داند که با افزایش اندازه مسئله، بخش پیاپی نیز افزایش می یابد. اما قانون گوستافسون اندازه بخش پیاپی را ثابت می داند و افزایش اندازه مسئله را بر آن بی تاثیر می شمارد. در سال ۱۹۹۸، گوستافسون و گروهی از پژوهشگران آزمایشگاه ساندیا[۱] توانستند با بکارگیری رایانه ncuber/۱۰ با ۱۰۲۴ پردازنده، به بهره سرعت در حدود ۱۰۰۰ برسند با اینکه a میان ۰.۰۰۴ و ۰.۰۰۸ بوده است. با آزمودن الگوریتم های دیگر، درستی قانون گوستافسون به اثبات رسید و این نتیجه بدست آمد که بهره گیری اقتصادی از سیستم های موازی ممکن است[۸][۱۰]. در پردازنده های FPGA نیز با اندازه گیری سرعت پردازش در یک پروسس مجزا و مقایسه سرعت اجرای آن الگوریتم مورد نظر با سیستم های دیگر می توان به نتیجه مطلوب برای کار مورد نظر رسید[۱۷].

۱ -Sandia

فصل سوم

مراحل طراحی و انتخاب قطعات

۳-۱- مقدمه

در فصول قبلی معرفی و بررسی FPGA ها و پردازش موازی مورد بررسی قرار گرفت. در این فصل ابتدا به بررسی روند طراحی سخت افزار (شماتیک و مدار چاپی)، انتخاب قطعات و تراشه های مورد نظر، ارتباطات بین تراشه ها و همچنین ارتباط برد با دنیای بیرون از برد، تست و مونتاژ برد و ملاحظات مد نظر در طراحی سخت افزار پرداخته شده است. در ادامه به بررسی جزئی برخی قطعات مهم و پرکاربرد و ویژگی های این تراشه ها پرداخته و با تمام قابلیت های آنان آشنا خواهیم شد. آشنایی با قطعات، نحوه کارکرد و چگونگی ارتباط با آنها از جمله اهداف اصلی این فصل می باشد.

۳-۲- روند طراحی

همانطور که در فصول قبلی نیز به آن اشاره شد، در این پروژه به دنبال طراحی یک سخت افزار با هسته پردازنده اصلی FPGA می باشیم. در کنار آن استفاده از تراشه ها و قطعات دیگر، برای ارتباط با دنیای بیرون از برد نیز مد نظر قرار گرفته شده است. در ابتدا می بایست تعیین کنیم که ساختار سخت افزار چگونه باشد، از چه اجزایی تشکیل شده باشد و قطعات مصرفی چگونه انتخاب گردند و ارتباط سخت افزار با دنیای بیرون چگونه باشد. تهیه یک برد پردازشگر قوی برای کاربردهای متنوع از جمله اهداف اصلی طرح می باشد. لذا بدین منظور از تراشه های متنوعی در برد استفاده شده است.

بمنظور طراحی سخت افزار، ابتدا بررسی جامعی از تراشه های پردازشگر موجود در بازار صورت پذیرفت تا با توجه به کارکرد هر کدام و قابلیت های آنان، انتخاب قطعات صورت پذیرد.

پردازنده FPGA انتخاب شده برای این طرح XC۳S۴۰۰ از خانواده Spartan۳ شرکت Xilinx است. انتخاب این FPGA بدلیل کارآیی بالای آن و استفاده زیاد از این نوع تراشه در ادوات و وسایل حرفه ای ویدئو، نظیر انواع میکسرهای صدا و تصویر، انواع پخش کننده ها سیگنالهای ویدئویی و همچنین کاربردهای متنوع نظامی است. سریعترین و بروزترین نوع در زمان انتخاب نیز از جمله موارد مؤثر در انتخاب قطعه بوده است. بطور کلی در دنیای طراحی FPGA، اندازه یا همان مقدار گیت ها، برخلاف

تصور اولیه حرف اول را نمی زند. در واقع باید FPGA را بر اساس نیازهای طرح مانند تعداد پایه های I/O، منابع منطقی موجود، بلوکهای کارکردی ویژه ی در دسترس و ... انتخاب نمود. برای انتخاب قطعه مناسب می بایستی موارد زیر را بعنوان پارامترهای اصلی مد نظر داشت[۱۱]:

- فناوری FPGA
- منابع اصلی و بسته بندی
- واسطه های I/O همه منظوره
- RAM، ضرب کننده ها و دیگر المان های تعبیه شده
- قابلیت های I/O گیگابیتی
- دسترس پذیری IP
- کلاس های سرعت

با توجه به اینکه این تراشه دارای ٥٦Kb حافظه رم توزیع شده و ۲۸۸Kb بلوک رم می باشد، برای کارهایی که نیاز به حافظه بیشتر می باشد، یک حافظه سریع ۸Mb در کنار FPGA تعبیه شده است. این رم دارای ۴ بلوک از نوع استاتیک رم است که مستقیماً به تراشه متصل شده است.

برای افزایش کارآیی و قابلیتهای سخت افزار، تراشه های دیگری نیز به برد اضافه شده است. طرح اولیه سخت افزار با توجه به بررسی بردهای مشابه و در دسترس صورت پذیرفته است. بدین منظور بررسی جامعی بر روی بردهای موجود در بازار و همچنین کیت های آماده شرکت Xilinx صورت پذیرفت. در طراحی، این موضوع دارای اهمیت است که سخت افزار طراحی شده، قابلیت ها، امکانات و کارآیی سیستم های مشابه را دارا بوده و حتی در برخی زمینه ها کامل تر از آنها نیز باشد. بدین منظور برد FPGA SPARTAN۳A/AN Starter Kit Board را بعنوان مرجع طراحی در نظر گذاشته و مورد بررسی قرار گرفت.

قطعات و ارتباطاتی که برای سخت افزار SUT۲۰۰ طراحی شده اند، عبارتند از:

- Spartan۳ FPGA Xilinx ٤۰۰K-Gate XC۳S٤۰۰PQ۲۰۸
- ۲Mbit Flash In-System Programmable Configuration PROMs
- ۸Mbit Synchronous SRAM
- FPGA Configuration Storage
- X٦ Or X۳۰ Data Interface
- ۸bit MCU AVR, ATMEGA۱۲۸
- ۱۲bit,۱۰us ADC
- ۸bit DAC
- Serial Interface (RS۲۳۲,RS٤۸٥)
- USB Interface
- SD CARD Interface
- Character LCD Display
- Expansion Connector
- Push-Button Key
- Discrete LEDs

به منظور ارتباط با سیگنالهای آنالوگ نیز بعنوان ورودی مدار، یک مبدل آنالوگ به دیجیتال ۱۲ بیتی با نرخ نمونه برداری ۱۰۰Ksps در مدار در نظر گرفته شده است. این مبدل در سه حالت کاری راه اندازی می شود که در سخت افزار طراحی شده در حالت Stand-Alone راه اندازی شده است. این مبدل ورودیهای از محدوده ٥- ولت الی ٥+ ولت را می پذیرد. خروجی بیت های مبدل، هم به میکروکنترلر متصل هستند و هم به FPGA. این کار از طریق اتصال جامپرها در کانکتور مربوطه صورت می پذیرد.

همچنین یک مبدل دیجیتال به آنالوگ ۸ بیتی نیز در نظر گرفته شده است تا در برخی از برنامه های کاربردی و آزمایشی، این امکان برای سخت افزار نیز وجود داشته باشد. ارتباطات بین برد و دنیای بیرون از آن از طریق پورتهای سریال RS۲۳۲/٤٨٥ و ارتباط USB امکان پذیر است. یک کارت حافظه SD نیز برای ذخیره سازی اطلاعات در مدار تعبیه شده است. برای ارتباط RS۲۳۲ از تراشه MAX۲۳۲ استفاده شده است. این تراشه مستقیماً به پورت USART میکروکنترلر متصل شده است. برای مسافتهای طولانی تر نیز تراشه ADM٤٨٥ به یکی دیگر از پورتهای USART میکروکنترلر ارتباط داده شده است.

ارتباط USB یکی از پرکاربردترین ارتباطات در بردهای امروزی بشمار می آید. سهولت در استفاده کاربران، سریع بودن و ارتباط ساده با تراشه های جانبی از جمله مزایای این ارتباط است. برای این امر از تراشه FT۲۴۵ شرکت FTDI CHIP استفاده شده است. این تراشه یک مبدل پارالل (۸بیت) به USB است که تا نرخ ۸ Mbit ps اطلاعات را جابجا می نماید. برای کسب اطلاعات جامع در ارتباط با مدارات واسط USB و چگونگی استفاده از آنها به مرجع [۱۵] مراجعه نمایید.

بنابراین آشنایی با قطعات، کاربرد و کارکرد آنها گام اول در روند طراحی سخت افزار می باشد. پس از انتخاب قطعات با توجه به نیازهای طرح مورد نظر، گام بعدی طراحی شماتیک اولیه مدار و مدار چاپی آن می باشد. در شماتیک اولیه، طرح کلی مدار، ارتباطات بین قطعات با یکدیگر و با بیرون از برد، ملاحظات کلی راه اندازی تراشه ها از قبیل تغذیه و فوت پرینت (طرح قطعات در مدارچاپی) قطعات با توجه به سایز دقیق تمامی تراشه ها و قطعات استفاده شده در برد در نظر گرفته می شود. بعد از طراحی شماتیک و بررسی دقیق آن، نوبت به طراحی مدار چاپی می رسد. جایگذاری قطعات در مکان مناسب، بررسی دقیق فوت پرینت قطعات انتخاب شده، ملاحظات طراحی مدار چاپی نظیر کشیدن شیارهای ارتباطی برد، محدودیتهای فرکانسی و ارتباطات بین لایه ای در برد از مهمترین بخش های طراحی مدار چاپی می باشد.

در این پروژه، یک شماتیک و مدار چاپی اولیه طراحی گردیده و در آن از قطعات اصلی استفاده شده است. گام بعدی تکمیل شماتیک اولیه و افزودن تمامی قطعات مورد نظر به طرح و طراحی نهایی مدار چاپی می باشد. که در این گام می بایست تمامی قطعات در شماتیک قرار داشته باشند و تمامی ملاحظات طراحی سخت افزار در نظر گرفته شوند. در این مرحله ما فقط با برقراری ارتباطات بهینه فی ما بین قطعات بدنبال تکمیل سخت افزار هستیم. استفاده از مارکاژ (توضیحات و شماره های قطعات) در این مرحله از اهمیت خاصی برخوردار است تا در مرحله بعدی که مونتاژ برد است، کار سختی را پیش رو نداشته باشیم.

گام بعدی طراحی، مونتاژ برد است. در این مرحله قطعات بر روی برد لحیم کاری شده و سخت افزار مورد نظر تکمیل می گردد. تست اهمی و اتصالی از اهمیت ویژه ای برخوردار است. بدلیل حرارت و گرمای زیاد لحیم کاری، ممکن است که شیارها با یکدیگر اتصال کوتاه شوند و یا حتی یک قطعی در مدار صورت پذیرد. بدین منظور، صحت و سلامت تراشه ها و ارتباطات بعد از عملیات لحیم کاری نیز می بایست مورد توجه قرار گرفته و تست شوند.

عملیات طراحی مدار چاپی حائز اهمیت می باشد. توزیع سیگنال زمین در سرتاسر برد[۱] در مدار چاپی یکی از مسائل مهم و قابل توجه است. مسایلی که در طراحی مدار چاپی می بایستی مورد توجه قرار گیرد بدین شرح می باشد:

- جاینشانی بهینه و صحیح قطعات در برد با توجه به ارتباطات آن در طرح شماتیک مدار، بگونه ای که کمترین شیار برای برقراری ارتباط قطعات لازم باشد.
- ملاحظات کشیدن شیارها از لحاظ سایز با توجه به نوع سیگنال حامل آن و زاویه ی شیارها در سرتاسر برد
- ارتباطات بین لایه ای در برد
- سایز پدهای ارتباطی بین لایه ها با توجه به نوع تراشه و سیگنال ارتباطی
- استفاده از مارکاژ (توضیحات و شماره های قطعات)
- در دسترس بودن تغذیه مدار در سرتاسر برد
- ملاحظات مربوط به خازنهای دی کوپلینگ تراشه ها
- ملاحظات خطوط انتقال سیگنال

طراحی سیستم توزیع توان خوب (PDS) برای همه طرح های FPGA ، به ویژه برای برنامه های کاربردی با عملکرد بالا[۲] مهم است. نتایج طراحی مناسب باعث عملکرد کلی بهتر، کاهش جیتر کلاک و

۱ - Grounding
۲ - high-performance

DCM و به طور کلی تقویت سیستم می شود. قبل از طراحی برد مدار چاپی (PCB) برای طراحی FPGA، طراحی سیستم های توزیع قدرت (PDS) با استفاده از خازن های Bypass/Decoupling بررسی می شود.

تغذیه ورودی مدار ۱۲+ ولت (DC) است. تراشه FPGA نیازمند سه ولتاژ ۱.۲,۲.۵،۳.۳ بترتیب برای I/O ها، هسته اصلی و منابع منطقی می باشد. بدین منظور ولتاژ I/O میکروکنترلر نیز ۳.۳ ولت انتخاب شده تا ارتباط فی مابین آنها به راحتی و بدون نیاز به تغییری در سطوح ولتاژ، امکان پذیر باشد. تراشه های دیگر نیازمند تغذیه های ۵+ و ۱۲- نیز می باشند که در مجموع شش سطح ولتاژ برای مدار در نظر گرفته شده است. تراشه اصلی تغذیه مدار، آی سی LM۳۱۷ می باشد.

بعد از مراحل فوق، تست قطعات و اجرای برنامه های نمونه مرحله بعدی می باشد. از جمله مسائل و مشکلاتی که در این مرحله با آن مواجه هستیم، بحث برنامه ریزی و پیکربندی FPGA می باشد. این تراشه دارای پنج مُد پیکربندی است که بدلیل ویژگیها و قابلیت ها، در این طرح دو نوع پیکربندی اسکن مرزی (JTAG) و پیکربندی از طریق حافظه فلش سریال، مد نظر قرار گرفته است. برای پیکربندی تراشه، نیاز به یک مدار واسط دیگری نیز می باشد که دو نوع ارتباط از طریق پورت موازی کامپیوتر و یا از طریق پورت USB در این مورد دسترس است. نحوه کار با این پورت و اطلاعات مورد نیاز جهت پیکربندی تراشه، در قسمت مراجع [۱۵] قابل مشاهده می باشد. در این مرحله با ارتباط برقرار کردن با FPGA و تراشه های دیگر که مورد نیاز است، با نرم افزارهای مربوطه، برنامه های نمونه ای جهت تست سخت افزار بر روی آن پیاده سازی می نماییم. در این طرح، برنامه تست LED، شیفت رجیستر و برخی فیلترهای دیجیتال، بعنوان برنامه های آزمایشی در برد پیاده سازی شده است.

در ادامه این فصل بمنظور آشنایی با تراشه FPGA انتخاب شده و بررسی جزئی آن، به مطالب اصلی این پردازنده ها می پردازیم:

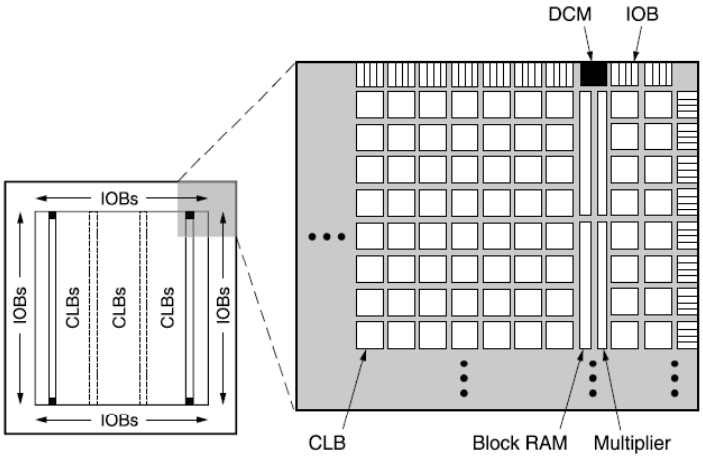
۳-۳- بررسی معماری

معماری خانواده اسپارتان ۳ شامل پنج عنصر اصلی برنامه پذیر می باشد:

- بلوکهای منطقی قابل پیکر بندی (CLBs): این بلوکها، شامل جداول LUT مبتنی بر رم هستند. این بلوکها برای پیاده سازی عناصر ذخیره پذیر و منطقی که بصورت فلیپ فلاپ یا LATCH قابل استفاده شدن هستند می باشند. CLBها توانایی برنامه ریزی شدن برای اجرای گستره وسیعی از توابع منطقی را بعنوان ذخیره کننده اطلاعات دارند.
- بلوکهای ورودی/خروجی (IOB): این بلوکها جریان اطلاعات بین پینهای I/O و منطق دورنی قطعه را کنترل می کنند. هر I/O پشتیبان کننده دیفرانسیلی اطلاعات بهمراه کارآیی سه حالته می باشد. ۲۶ استاندارد مختلف سیگنالها که شامل ۸ استاندارد با کارآیی بصورت دیفرانسیلی می باشد در جدول ۳-۱ نشان داده شده است. این اطلاعات شامل نرخ دوبرابر دیتا (DDR) نیز می شود. همچنین ویژگی کنترل کننده امپدانس دیجیتالی (DCI)، فراهم کننده تطبیق پایانه های طرح مدار است تا اینکه بصورت ON-CHIP عمل تطبیق امپدانس صورت پذیرد. این امر باعث ساده سازی طراحی مدار می شود.
- بلوکهای رم: فراهم کننده ذخیره اطلاعات در فرم ۱۸Kbit، در فرمهای دوگانه است.
- بلوکهای ضرب کننده (Multiplier Blocks) پذیرنده دو عدد ۱۸ بیتی باینری بعنوان ورودی است و نتیجه آنرا محاسبه می کند.
- بلوکهای مدیریت کلاک دیجیتال (DCM): فراهم کننده سیگنالهای کلاک برای درجه بندی سرخود است. در واقع یک راه حل دیجیتال کردن کامل کلاک، برای توزیع، تاخیر، ضرب، تقسیم و شیفت فاز است. این عناصر در شکل ۳-۱ نشان داده شده است.

یک حلقه ای از IOB، آرایه ای از CLB ها را احاطه کرده است. تراشه XC۳S۵۰ تنها یک ستون از بلوک رم جاسازی شده در آرایه دارد. تمام تراشه ها در رنج XC۳S۲۰۰ الی XC۳S۲۰۰۰ در بردارنده دو ستون از بلوکهای رم هستند. تراشه های سری ۴۰۰۰ و ۵۰۰۰ دارای چهار ستون بلوک رم هستند. هر ستون از چندین بلوکهای رم ۱۸Kbit تشکیل شده و هر ستون با یک ضرب کننده اختصاص یافته در

ارتباط است. واحدهای DCMs در قسمت انتهایی بلوکهای رم قرار گرفته اند. خانواده اسپارتان ۳ در بردارنده یک شبکه قوی از سوئیچ ها و مسیرهایی است که تمام عناصر تابعی پنج گانه را بیکدیگر وصل کرده و منتقل کننده سیگنالها از بین آنها است. هر یک از عناصر با یک سوئیچ ماتریسی در ارتباط است که جواز چندین ارتباط برای مسیردهی را می دهد.



شکل۳-۱- معماری خانواده اسپارتان۳

۳-۴-پیکربندی[۱]

FPGA خانواده اسپارتان۳ با بارگذاری اطلاعات پیکربندی، به درون حافظه LATCH های استاتیک CMOS و قابل برنامه ریزی مجدد، برنامه ریزی می شوند. قبل از راه اندازی FPGA اطلاعات پیکربندی در یک حافظه خارجی PROM یا حافظه های غیرفرار دیگر بصورت درون بردی یا بیرون بردی ذخیره می گردند. بعد از راه اندازی الکتریکی FPGA اطلاعات پیکربندی بر اساس یکی از پنج روش زیر به درون FPGA نوشته می شوند:

مستر پارالل، اسلیو پارالل، مستر سریال، اسلیو سریال و JTAG

۱ -Configuration

حافظه ای که برای برنامه ریزی تراشه نیاز است از خانواده حافظه های فلش PROM شرکت Xilinx می باشد که شامل تراشه XCF۰۰S PROM برای پیکربندی سریال و XCF۰۰P PROM برای پیکربندی موازی است.

۳-۵-توانایی های I/O در تراشه های خانواده اسپارتان۳

ویژگیهای انتخاب I/O تراشه های اسپارتان۳، ۱۸ استاندارد بصورت تک پایانه ای و ۸ استاندارد بصورت دیفرانسیلی را پشتیبانی می نماید. تعدادی از استانداردها، ویژگی DCI را پشتیبانی می کند که این ویژگی، پایانه ای بصورت مدار مجتمع را در خود تراشه مهیا می نماید تا انعکاسهای سیگنالهای ناخواسته را حذف کند.

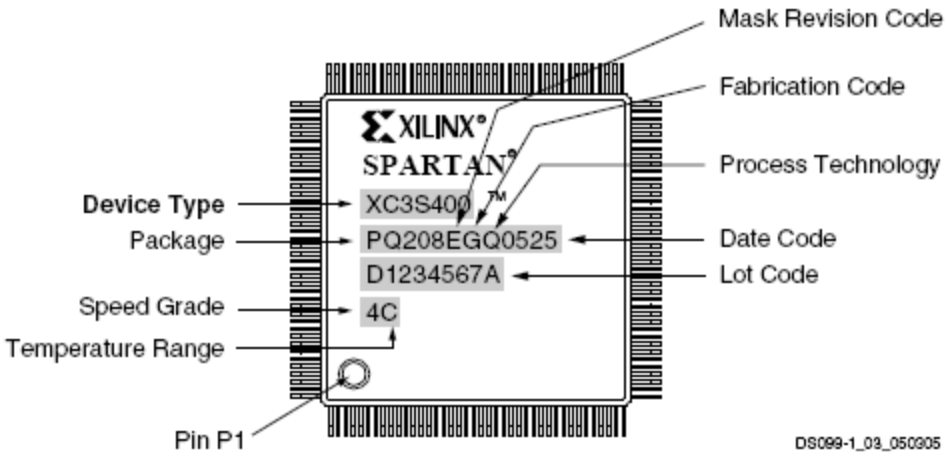
جدول ۳-۱ نشان دهنده تعداد I/O ها برای هر بسته از خانواده اسپارتان ۳ بصورت کاربری (User) یا دیفرانسیلی می باشد.

جدول۳-۱- جدول ورودی/خروجی تراشه خانواده اسپارتان۳

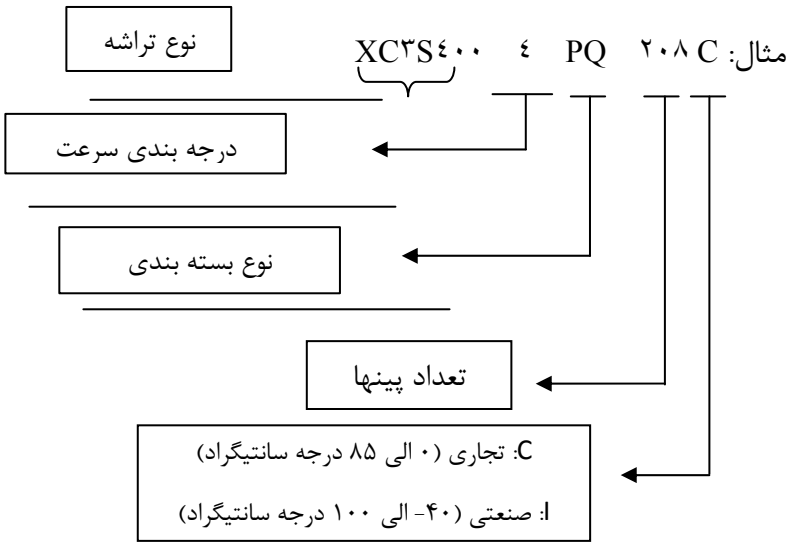
	Available User I/Os and Differential (Diff) I/O Pairs by Package Type																			
Package	VQ100 VQG100		CP132(1) CPG132		TQ144 TQG144		PQ208 PQG208		FT256 FTG256		FG320 FGG320		FG456 FGG456		FG676 FGG676		FG900 FGG900		FG1156(1) FGG1156	
Footprint (mm)	16 x 16		8 x 8		22 x 22		30.6 x 30.6		17 x 17		19 x 19		23 x 23		27 x 27		31 x 31		35 x 35	
	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff
XC3S50	63	29	89(1)	44(1)	97	46	124	56	-	-	-	-	-	-	-	-	-	-	-	-
XC3S200	63	29	-	-	97	46	141	62	173	76	-	-	-	-	-	-	-	-	-	-
XC3S400	-	-	-	-	97	46	141	62	173	76	221	100	264	116	-	-	-	-	-	-
XC3S1000	-	-	-	-	-	-	-	-	173	76	221	100	333	149	391	175	-	-	-	-
XC3S1500	-	-	-	-	-	-	-	-	-	-	221	100	333	149	487	221	-	-	-	-
XC3S2000	-	-	-	-	-	-	-	-	-	-	-	-	333	149	489	221	565	270	-	-
XC3S4000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	489	221	633	300	712(1)	312(1)
XC3S5000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	489	221	633	300	784(1)	344(1)

۳-۶- علامت گذاری تراشه

شکل ۳-۲ علامت گذاری تراشه را برای بسته بندی QFP نشان می دهد.



شکل ۳-۲- علامت گذاری تراشه از نوع بسته بندی QFP



۳-۷- بررسی اجمالی IOBs

بلوکهای ورودی خروجی[۱] یک واسط دو جهته قابل برنامه ریزی را بین پینهای I/O و منطق درونی FPGA فراهم می کنند. نموداری از ساختار درونی IOB در شکل ۳-۳ قابل مشاهده می باشد.

سه مسیر اصلی در ارتباط با IOB وجود دارد:

۱ - Input Output Block

مسیر خروجی، مسیر ورودی و مسیر سه حالته[1]. هر مسیر دارای جفتهای عناصر ذخیره کننده خود است که بعنوان Latch یا رجیستر عمل می کنند.

۳-۸-کارکرد عناصر ذخیره کننده[2]

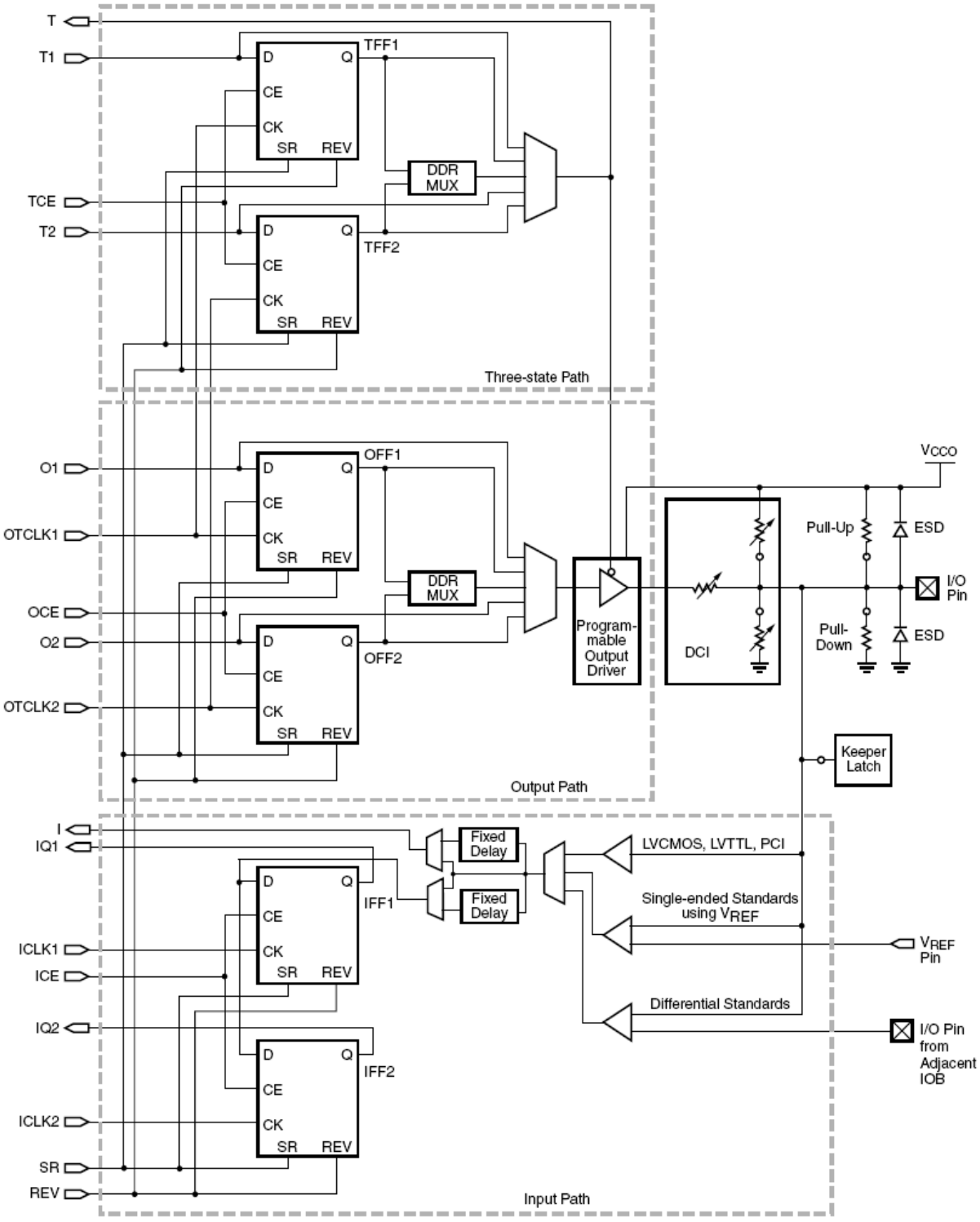
سه جفت از عناصر ذخیره کننده در هر IOB وجود دارد. یک جفت برای هر کدام از سه مسیر ذکر شده فوق. قابلیت این کار وجود دارد که هر کدام از این عناصر ذخیره کننده را مانند یک D-Type flip-flop یا یک latch حساس به سطح[3] پیکربندی کرد.

جفت عنصر ذخیره کننده واقع بر روی مسیر خروجی یا مسیر ۳state را می توان با یک مالتی پلکسر مخصوصی با هم بکار برد تا اینکه انتقال DDR[4] را تولید کرد. این عمل با سنکرون کردن اطلاعات با لبه بالا رونده سیگنال کلاک و تبدیل کردن آنها به بیتهای سنکرون با هر دو لبه های بالارونده و پایین رونده، تکمیل می شود.

ترکیب دو رجیستر و یک مالتی پلکسر، بوجودآورنده یک FDDR (Double Data rate D-FF) است.

---

۱ -۳STATE
۲ -Storage Element
۳ - level sensitive latch
٤- Double Data rate



شکل۳-۳- نمودار ساده شده یک ورودی/خروجی

بر طبق شکل ۳-۳، خط کلاک OTCLK۱، ورودی CLK مربوط به رجیستر بالایی را روی خروجی و مسیر ۳ حالته متصل می کند. بصورت مشابه خط کلاک OTCLK۲ ورودی CK را برای رجیستر پایینی

برروی مسیرهای خروجی و سه حالته متصل می کند. رجیسترهای بالایی و پایینی در مسیر ورودی خطوط مستقل کلاک دارند.(lclk۱, lclk۲)

خط فعال ساز OCE، ورودی CE مربوط به رجیسترها را روی مسیر خروجی متصل می کند. بصورت مشابه TCE، ورودی CE برای جفت رجیسترها بر روی مسیر سه حالته را متصل کرده و CE برای جفت رجیسترها بر روی مسیر ورودی یکسان است. خط SET/RESET که به IOB وارد شده است، برای هر شش رجیستر مشترک است. هر عنصر ذخیره ساز علاوه بر کنترل کردن پلاریته سیگنالها، ویژگیهای متعددی را پشتیبانی می کند.

۳-۹- مقاومتهای Pull up/Pull down

مقاومتهای انتخابی Pull up/Pull down برای راه اندازی سطح ولتاژهای بالا و پایین بترتیب در ورودی-خروجیهای بدون استفاده بکار برده می شوند. مقاومتهای Pull up، بطور دلخواه هر پایه IOB را به VCCO متصل کرده و مقاومت Pull down هر پایه را به زمین بطور دلخواه وصل می کند. آنها را در کد برنامه نویسی HDL می توان بعنوان محدودیت، کامپوننت یا بعنوان شاخص معرفی کرد. این مقاومتها را می توان برای تمام IO های بدون کارکرد با استفاده از گزینه مولد جریان بیت[۱] انتخاب کرد.

یک سطح ولتاژ پایین بر روی پد HSWAP-EN، مقاومتهای Pull-up را بر روی تمام I/O، در مدت پیکر بندی فعال می کند. مقاومتهای Pull up/Pull down در خانواده اسپارتان۳، بطور مشخصی از مقاومتهای Pull up/Pull down خانواده های پایین تر Xilinx، قوی تر است.

۳-۱۰- مدار نگهدارنده[۲]

۱- Bit Stream Generator
۲ - Keeper Circuit

هر I/O دارنده یک مدار نگهدارنده انتخابی است که نگهدارنده آخرین سطح ولتاژ بر روی خط می باشد. بعد از اینکه تمامی راه اندازها از مدار خارج شدند، این گزینه برای جلوگیری از شناور شدن (Floating) خط باس در زمانیکه تمامی درایوهای متصل شده در حالت امپدانس بالا[1] هستند، می باشد.

حفاظت از تخلیه الکتریکی(ESD[2])

دیودهای برش، تمامی پدهای قطعه را از خسارتهای ناشی از تخلیه الکتریکی به خوبی محافظت می کنند. هر I/O دو دیود برش دارد. یک دیود با اتصال P-N از پد به VCCO رفته و دیگری با اتصال N-P از پد به طرف زمین کشیده شده است. در مدت کارکرد، بطور معمول در وضعیت خاموش بایاس می شوند. این دیودهای برش، بدون توجه به استانداردهای سیگنال همیشه به پد وصل می شوند. حضور این دیود ها توانایی اسپارتان۳ برای تحمل کردن سیگنالهای ولتاژ بالا را ارتقاء می بخشد. مقدار ولتاژ ورودی که هر I/O می تواند داشته باشد در جدول ۳-۲ قابل مشاهده است.

جدول ۳-۲- مقدار مطلق حداکثر ولتاژ قابل تحمل تراشه

Symbol	Description	Conditions		Min	Max	Units
V_{CCINT}	Internal supply voltage relative to GND			–0.5	1.32	V
V_{CCAUX}	Auxiliary supply voltage relative to GND			–0.5	3.00	V
V_{CCO}	Output driver supply voltage relative to GND			–0.5	3.75	V
V_{REF}	Input reference voltage relative to GND			–0.5	V_{CCO} + 0.5	V
V_{IN}	Voltage applied to all User I/O pins and Dual-Purpose pins relative to GND(2, 4)	Driver in a high-impedance state	Commercial	–0.95	4.4	V
			Industrial	–0.85	4.3	
	Voltage applied to all Dedicated pins relative to GND(3)		All temp. ranges	–0.5	V_{CCAUX} + 0.5	V

۳-۱۱- توانایی Boundary-Scan

تمامی I/O های خانواده اسپارتان۳، تست Boundary-Scan را که با IEEE۱۱٤۹.۱ سازگار است، پشتیبانی می کنند. در طول عملیات Boundary-Scan، مانند EXTEST یا HIGHZ مقاومتهای Pull down ورودی خروجی، فعال هستند.

۱- high-impedance state
۲- Electro-Static Discharge

۳-۱۲- استانداردهای قابل انتخاب برای سیگنالهای ورودی/خروجی

IOB، ۱۸ استاندارد متفاوت تک پایانه[۱] را پشتیبانی می کند. این استاندارد ها در جدول۳-۳ نشان داده شده است. همانطور که در جدول ۳-۴ نیز نشان داده شده است، اکثر IOB ها که در جفت های مشخصی بکار می روند، پشتیبان ۸ استاندارد سیگنال بصورت دیفرانسیلی هستند[۷].

دو سطح ولتاژ بکار برده شده (VCCO,VREF) تعیین کننده استاندارد سیگنال مطلوب است. خطوط VCCO، فراهم کننده جریان برای درایور خروجی است. ولتاژ در این خطوط، تعیین کننده سوییینگ ولتاژ خروجی برای تمامی استاندارد ها است بجز GTLP, GTL. تمام استاندارد های تک پایانه (Single Ended) بجز LVCMOS,LVTTL,PCI، نیازمند یک ولتاژ مرجع (Vref) برای بایاس سطح جداکننده سوئیچ کننده ورودی است. در یک مرتبه که اطلاعات پیکربندی به درون FPGA بارگذاری می شوند که پین ورودی/خروجی مربوط به بانکی از FPGA را برای کارکرد استانداردی مشخص فرا می خواند، تعداد کمی از پینهای رزرو I/O بر روی همان بانک، بطور اتوماتیک به ورودیهای Vref تبدیل می شوند. زمانیکه یکی از استانداردهای LVCMOS بکار برده می شود، این پینها ورودی/خروجی باقی می مانند. این امر بعلت این است که ولتاژ VCCO آستانه سوئیچ ورودی را بایاس کرده است. در اینصورت به ولتاژ مرجع نیازی نیست.

سطوح VCCO,VREF برای بدست آوردن استاندارد تک پایانه مطلوب با توجه به جدول ۳-۳ انتخاب می شوند.

۱- single-ended

جدول ۳-۳- استانداردهای ورودی/خروجیهای تک پایانه (بر حسب ولت)

Signal Standard (IOSTANDARD)	V_{CCO}		V_{REF} for Inputs(1)	Board Termination Voltage (V_{TT})
	For Outputs	For Inputs		
GTL	Note 2	Note 2	0.8	1.2
GTLP	Note 2	Note 2	1	1.5
HSTL_I	1.5	-	0.75	0.75
HSTL_III	1.5	-	0.9	1.5
HSTL_I_18	1.8	-	0.9	0.9
HSTL_II_18	1.8	-	0.9	0.9
HSTL_III_18	1.8	-	1.1	1.8
LVCMOS12	1.2	1.2	-	-
LVCMOS15	1.5	1.5	-	-
LVCMOS18	1.8	1.8	-	-
LVCMOS25	2.5	2.5	-	-
LVCMOS33	3.3	3.3	-	-
LVTTL	3.3	3.3	-	-
PCI33_3	3.0	3.0	-	-
SSTL18_I	1.8	-	0.9	0.9
SSTL18_II	1.8	-	0.9	0.9
SSTL2_I	2.5	-	1.25	1.25
SSTL2_II	2.5	-	1.25	1.25

استانداردهای دیفرانسیلی، یک جفتی از سیگنالها را بکار می گیرند که پلاریته یکی برعکس دیگری می باشد. توانایی حذف نویز در این استاندارد، اجازه دهنده انتقال اطلاعات با سرعتهای بسیار بالا می باشد. یک L-Number منحصر بفردی (بخشی از نام پین قطعه) معرف جفت خطوط مرتبط با هر بانک است. برای هر جفت، حروف N,P بترتیب معرف خط اصلی و خط وارونه می باشد. بعنوان مثال پین IO-L۴۳P-۷ و IO-L۴۳N-۷ معرف خطوط اصلی و وارونه هستند که دربرگیرنده جفت خط L۴۳ در بانک ۷ می باشند.

خطوط VCCO، فراهم کننده جریان برای خروجی است. خطوط VccAux تأمین کننده توان برای ورودیهای دیفرانسیلی است که باعث مستقل شدن آنها از ولتاژ VCCO برای یک بانک I/O می شود. انتخاب سطح VCCO برای رسیدن به استاندارد دیفرانسیلی مطلوب بر طبق جدول ۳-۴ صورت می پذیرد.

جدول ۳-۴- استانداردهای ورودی/خروجی دیفرانسیلی

Signal Standard (IOSTANDARD)	V_{CCO} (Volts)		V_{REF} for Inputs (Volts)
	For Outputs	For Inputs	
LDT_25 (ULVDS_25)	2.5	-	-
LVDS_25	2.5	-	-
BLVDS_25	2.5	-	-
LVDSEXT_25	2.5	-	-
LVPECL_25	2.5	-	-
RSDS_25	2.5	-	-
DIFF_HSTL_II_18	1.8	-	-
DIFF_SSTL2_II	2.5	-	-

البته باید توجه داشت که نیاز فراهم کردن VCCO,VREF تحمیل کننده محدودیتهایی است. مانند اینکه کدام استاندارد توانایی کارکرد در یک بانک را دارد.

۳-۱۳- امپدانس کنترل شده دیجیتالی[۱]

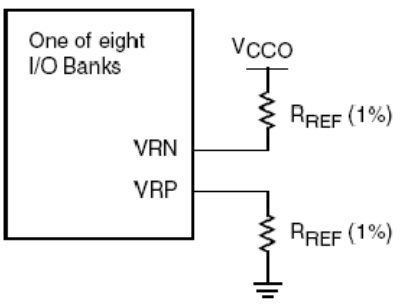
زمانیکه مسیر چرخشی تأخیر یک سیگنال خروجی (از خروجی به ورودی و برعکس) از زمان خیزش[۲] و فرود[۳] گذر می کند، این دلیل مناسبی است برای اینکه ما نیازمند به مقاومتهایی در انتهای مسیر حمل سیگنال می باشیم. این مقاومتها بطور مؤثری امپدانس I/O را با امپدانس خط انتقال مطابقت و یکسان می کنند. این امر مانع انعکاسهایی می شود که بطور مخالف یکپارچگی سیگنال را تحت تأثیر قرار می دهند. به هر حال با تعداد زیاد ورودی/خروجیها در قطعات جدیدتر، افزودن مقاومتها نیازمند تعداد بیشتری قطعه و ناحیه بزرگتری از برد می باشد. بنابراین در تعدادی از بسته بندیها، بطور مثال Ball grid array (BGA) همیشه این امکان وجود ندارد که مقاومتها نزدیک پینها قرار بگیرند.

DCI با فراهم کردن دو نوع پایان دهی درون تراشه ای[۴]، به این مشکل پاسخ داده است: پایان دهی موازی با شبکه ای از مقاومتهای مجتمع و پایان دهی سری که با کنترل کردن امپدانس درایوهای

۱ -Digitally Controlled Impedance
۲- Rise time
۳- Fall time
٤- On Chip

خروجی بدست می آید. DCI هر دو نوع پایان دهی را انجام می دهد تا بطور دقیق تطبیق امپدانس خطوط را انجام دهد. این پردازش تنظیمات، جبران کننده تفاوتهای امپدانس I/O است که در نتیجه تغییرات دمای محیط، ولتاژ تغذیه و مراحل ساخت قطعه بوجود آمده است. در زمانیکه درایورهای خروجی از مدار خارج می شوند، پایان دهی سری، یک امپدانس بالایی را منجر می شود. بر خلاف آن مقاومتهای پایان دهی موازی، در مقدار تعیین شده باقی می مانند. DCI، فقط برای استانداردهای I/O مخصوصی، در دسترس است (اطلاعات بیشتر در برگاه اطلاعات تراشه، جدول۹). همچنین پنج روش پایه ای برای برنامه ریزی پایانه ها وجود دارد.

ویژگیهای DCI، برای هر کدام از هشت بانک قطعه، بطور مستقل عمل می کنند. هر بانک دارای یک پین مرجع N (VRN) و پین مرجع P (VRP) برای کالیبره کردن درایور و مقاومت پایانه می باشد. فقط زمانیکه یک استاندارد DCI در یک بانک استفاده می شود، این دو پین همانند VRN و VRP استفاده می شوند و زمانیکه استاندارد DCI استفاده نمی شود، این دو پین همانند I/O قابل استفاده هستند. همانطور که در شکل ۳-۴ مشاهده می شود، یک مقاومت خارجی به VRN بصورت Pull-up و یکی دیگر به VRP بعنوان Pull down بایستی افزوده گردد. هر دوی مقاومتها مقدار یکسانی دارند (بطور معمول ۵۰ اهم با یک درصد خطا). استانداردهایی که دربردارنده حرف "DV۲" هستند، یک مقدار مرجع بکار می برند که دو برابر امپدانس خط است. DCI، تثبیت کننده امپدانس درایور خروجی برای همسان کردن مقدار مقاومت مرجع طبق استاندارد می باشد و همیشه مقاومتهای پایان دهی درون تراشه ای را برای همسان کردن مقدار مقاومت مرجع تنظیم می کند.



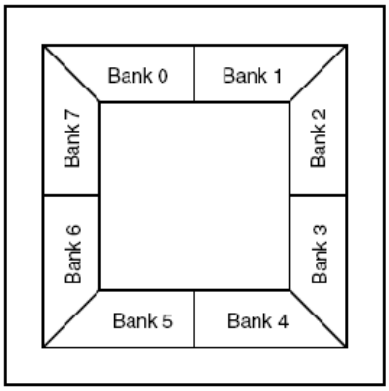
شکل۳-۴- اتصال مقاومتهای مرجع

قوانینی که راهنمای استفاده از DCI می باشند طبق زیر است:

- به ازای هر بانک بیش از یک استاندارد ورودی/خروجی DCI با یک پایان دهی یکتا، امکان پذیر نیست.
- برای هر بانک بیش از یک استاندارد ورودی/خروجی DCI، با یک پایان دهی مجزا و دوبخشی امکان پذیر نیست.
- پایان دهی یکتا، پایان دهی دو بخشی، درایور امپدانس کنترل شده و درایور امپدانس کنترل شده با نصف امپدانس می تواند در یک بانک وجود داشته باشد.

۳-۱۴- سازماندهی IOBs در درون بانکها

IOB ها در طول ۸ بانک قرار گرفته اند. هر طرف تراشه دارای دو بانک است. همانند شکل ۳-۵ برای تمامی بسته ها، هر بانک دارای خط Vref مجزا می باشد. بعنوان مثال خط Vref مربوط به بانک ۳ از تمامی خطوط Vref دیگر بانکها مجزا و مستقل می باشد. برای بسته بندیهای FG FT،PQ،VQ هر بانک دارای خطوط VCCO مختص می باشد. خط VCCO بانک ۷ از تمامی خطوط VCCO مختص می باشد. بنابراین قطعات اسپارتان ۳ در این بسته بندیها، پشتیبان ۸ تغذیه VCCO مستقل هستند.



شکل ۳-۵- بانکهای I/O اسپارتان۳

بر خلاف سایرین، در بسته بندیهای TQ۱٤٤ و CP۱۳۲، VCCO ها از داخل به یکدیگر متصلند. در نتیجه خانواده اسپارتان۳ در این بسته بندیها، پشتیبان چهار تغذیه VCCO مستقل هستند.

۳-۱۵- سازش پذیری[۱] FPGA SPARTAN

در خانواده اسپارتان۳، تمامی قطعات با بسته بندی مورد نظر سازش پذیر هستند. هنگامیکه نیاز به منابع منطقی آینده از ظرفیت تراشه اسپارتان در حال استفاده بیشتر می گردد، یک تراشه بزرگتر در بسته بندی یکسان می تواند جایگزین آن شود. قطعات بزرگتر ممکن است دارای خطوط Vref و Vcco بیشتری برای پشتیبانی از تعداد بیشتری از ورودی/خروجیها باشند. در قطعات بزرگتر، پینهای بیشتری می توانند از I/O به خطوط Vref تبدیل شوند. بنابراین برای ارتقاء سیستم مهم است که در زمان طراحی برد با در نظر گرفتن پین های اضافی در طرح سیستم برنامه ریزی لازم انجام شود. این خانواده از لحاظ پینها با هیچ کدام از خانواده های قبلی سازش پذیر نمی باشد.

۳-۱۶- قوانین مرتبط با بانکها

زمانیکه ورودی/خروجیها به بانکها تخصیص داده می شود، مهم است که قوانین Vcco زیر رعایت شود:

- هیچ کدام از پینهای Vcco در FPGA بدون اتصال نباشد.
- تمامی خطوط Vcco مرتبط با هر بانک، با سطح ولتاژ یکسانی تنظیم گردد.
- سطوح Vcco که با تمام استانداردها بکار برده می شود و به ورودی/خروجی بانکها تخصیص یافته است، می بایستی تطبیق داده شوند. نرم افزار Xilinx این مورد را بررسی می کند. لازم بذکر است که فقط یکی از استاندارد های زیر بازای هربانک در خروجی می تواند باشد: LVDS،LDT،LVDS-EXT و یا RSDS. این محدودیت برای هر ۸ بانک می باشد حتی اگر سطوح Vcco بین تمامی بانکها مشترک باشد[۷].

---

۱ -Compatibility

- اگر هیچ کدام از استانداردهای تخصیص یافته به ورودی/خروجی، Vcco را بکار نبرند، تمامی خطوط Vcco را به ولتاژ ۲.۵ ولت وصل نمایید.
- بطور کلی به Vcco بانک۴، سطح ولتاژ ۲.۵ ولت فراهم می شود. برای مدت زمان راه اندازی تا انتهای پیکربندی ولتاژ یکسانی را برای Vcco های بانک۵، درمدت پیکربندی یا عملیات موازی یا Readback فراهم کنید.

اگر هر کدام از استاندارد های اختصاص یافته به ورودی بانکها، Vref را بکار برند، لازم است که قوانین زیر نیز رعایت شود:

- تمام پینهای Vref درون بانک را به ولتاژ یکسانی متصل کنید.
- سطح Vref که با تمامی استانداردها بکار برده می شود و به ورودی بانک تخصیص یافته می بایستی تطبیق داده شوند.

۳-۱۶-۱- استثناهایی برای استانداردهای I/O بانکها

بانک ۵ در هر خانواده اسپارتان۳ در بسته بندیهای CP۱۳۲،VQ۱۰۰ یا TQ۱٤٤ استاندارد سیگنال DCI را پشتیبانی نمی کند. در این مورد بانک ۵، پینهای VRN و VRP را ندارد. بهمین ترتیب بانک ۴ و ۵ در هر خانواده اسپارتان۳ در بسته بندی VQ۱۰۰، استانداردهای سیگنال بکار برده شده با Vref را پشتیبانی نمی کند. در این مورد، دو بانک، هیچ پین Vref را ندارند.

۳-۱۶-۲- ولتاژ تغذیه :IOBs

سه منبع متفاوت، تغذیه ورودی/خروجیها را تأمین می کنند:

تغذیه Vcco، Vcc int و Vcc aux

۳-۱۷- رفتار پایه های I/O در حین پیکر بندی، راه اندازی و User Mode

زمانیکه تغذیه تراشه وصل نشده است، تمام پایه های I/O در حالت امپدانس بالا[۱] هستند. قبل از اتمام مرحله راه اندازی تراشه[۲] مقادیر ولتاژهای Vcco بانک۴، Vccint و Vccaux می بایستی به حداقل سطح پیشنهاد شده برسد. در این حالت تمامی I/O ها نیز همچنین در حالت امپدانس بالا خواهند بود. Vcco بانک۴، Vccint و Vccaux در خدمت مدار ریست بهنگام راه اندازی[۳] بعنوان ورودی هستند.

یک سطح ولتاژ پایین در ورودی پایه HSWAP-EN، باعث فعال شدن مقاومتهای بالابر در I/O در سراسر مدت پیکربندی FPGA از زمان راه اندازی خواهد شد. سطح ولتاژ بالا باعث غیر فعال شدن مقاومتهای Pullup و شناور بودن ورودی/خروجیها در این مدت خواهد شد. اگر پین HSWAP-EN شناور باشد، یک مقاومت داخلی Pullup، HSWAP-EN را در حالت high قرار خواهد داد. به محض اینکه تغذیه فراهم گردد، FPGA شروع به مقدار دهی حافظه پیکر بندی خود می کند. در همین زمان FPGA، بصورت داخلی مدار ست-ریست عمومی را فعال نموده (GSR) که بصورت ناهمزمان تمامی عناصر ذخیره کننده IOB را در حالت LOW تنظیم می کند.

بعد از اتمام مقدار دهی اولیه[۴]، پایه INIT-B، سطح ولتاژ بالا را گرفته و از ورودیهای M۰، M۱ و M۲ نمونه برداری کرده تا مُد پیکر بندی را مشخص کند. در این زمان، اطلاعات پیکربندی به داخل FPGA بارگذاری می شود. راه اندازی I/O در زمان پیکربندی در حالت امپدانس بالا باقی خواهد ماند (با یا بدون مقاومتهای Pullup که توسط پین HSWAP-EN مشخص شده است).

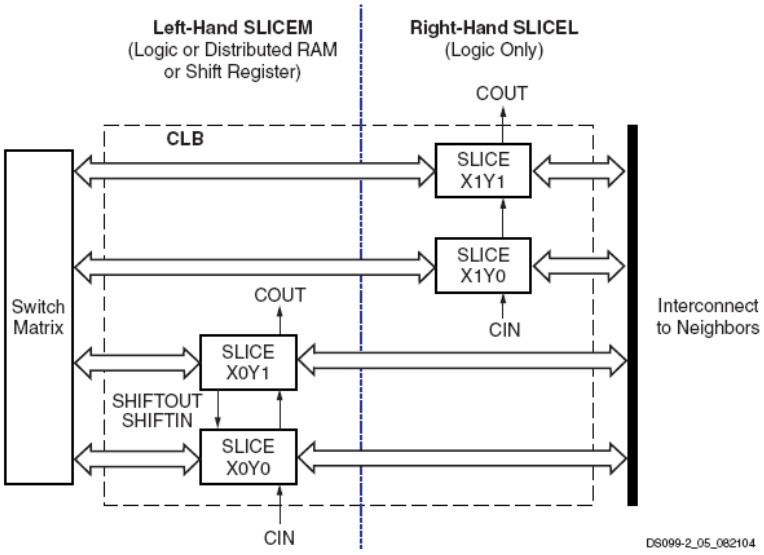
شبکه GTS[۵] در مدت راه اندازی[۶] رها شده و این شبکه پایان پیکربندی و آغاز عملیات طراحی در User Mode را ثبت می کند. در این زمان، I/O هایی که به سیگنال اختصاص یافته است، فعال شده، در حالتیکه تمامی I/O های غیر قابل استفاده در حالت امپدانس بالا باقی خواهد ماند. در User Mode، تمامی مقاومتهای Pullup داخلی در ورودی/خروجیها، غیر فعال می شوند وHSWAP-E بعنوان ورودی

۱ - high impedance
۲ - power on
۳ - power on reset
٤ - Initialization
٥ - Global Three State
٦ - Start Up

بی تأثیر[۱] عمل خواهد کرد. گزینه انتخاب تولید کننده رشته بیت[۲] در نرم افزار توسعه Xilinx، بعنوان یک پین غیر کاربرد، مشخص کننده این است که در مجموع I/O ها، مقاومتهای بالابر، پایین بر (یا بدون مقاومت در User Mode) را دارند یا خیر.

۳-۱۸- بلوکهای منطقی قابل برنامه ریزی (CLB[۳])

بلوکهای منطقی قابل برنامه ریزی (CLB)، تشکیل دهنده اصلی ترین منبع منطقی برای پیاده سازی همزمان مدارات ترکیبی می باشد. همانطور که در شکل ۳-۶ دیده می شود؛ هر CLB دربردارنده چهار قطعه[۴] متصل بهم است.



شکل ۳-۶- طرح Sliceها در درون CLB

این Slice ها در جفت هایی گروه بندی شده اند. هر جفت بصورت یک ستون با یک خط انتقال زنجیری سازماندهی شده است. شکل فوق نشان دهنده CLB است که در قسمت پایینی سمت چپ ردیف قرار گرفته است.

---

۱ - Don’t Care
۲ - Bitstream Gen
۳ - Configurable Logic Blocks
٤ -Slice

۳-۱۸-۱- مسیرهای اصلی منطقی

کارکرد هر قسمت از CLB، دو مسیر اطلاعات نزدیک به هم یکسان هستند که با ترم های top و bottom متمایز شده است. توضیح اینکه اسامی بکار برده شده در ارتباط با مسیر bottom است (اسامی مسیر top، در پرانتزها نشان داده شده است). مسیر اصلی از یک سوئیچ ماتریسی (از درون متصل بهم) در بیرون CLB آغاز شده است. چهار مسیر F۱ الی F٤(یا G۱ الی G٤ در مسیر بالایی) به قسمت معین وارد شده و مستقیم به LUT وصل می شوند.

در داخل هر قسمت، مسیر ۴ بیتی پایینی از میان یک Function Generator که انجام دهنده عملیاتهای منطقی است عبور می کند (F یا G). اطلاعات خروجی یک Function Generator، (D) ۵ مسیر ممکن را پیشنهاد می کند:

۱. خارج شدن از Slice از طریق خط X (یا Y) و بازگشت به اتصال داخلی
۲. در داخل Slice، X (یا Y) بعنوان ورودی برای DXMUX یا (DYMUX) در نظر گرفته شده که تغذیه کننده اطلاعات ورودی است. D، اطلاعات خروجی F یا G، FFX (یا FFY) عنصر ذخیره کننده، Q بعنوان خروجی عنصر ذخیره کننده راه انداز XQ(یا YQ) است که از قسمت مورد نظر خارج می شوند.
۳. کنترل کننده مالتی پلکسر CYMUXF (یا CYMUXG) بر روی خط انتقال زنجیره ای
۴. با در نظر گرفتن انتقال زنجیره ای، برای دروازه ی XOR (XORF یا XORG) که اجرا کننده عملیات ریاضی است بعنوان ورودی در نظر گرفته شود و در انتها نتیجه X یا Y را تولید کند.
۵. راه اندازی کردن مالتی پلکسر F٥MUX برای پیاده سازی توابع منطقی عریض تر از چهار بیت. لازم بذکر است که خروجی D هر دوی F-LUT و G-LUT بعنوان اطلاعات ورودی برای این مالتی پلکسر در نظر گرفته می شوند.

علاوه بر توضیحاتی که برای مسیر اصلی منطقی (MLP) در بالا داده شد، دو مسیر بای پس که به قسمت مورد نظر وارد می شود وجود دارد (BX و BY). در داخل FPGA، BX در نیمه پایینی Slice (یا BY در نیمه بالایی) می تواند به هر یک از بخش های زیر دسترسی داشته باشد:

- بای پس کردن هر دوی LUT و عنصر ذخیره کننده، سپس خارج شدن از Slice همانند BXOUT (یا BYOUT) و در انتها بازگشت به اتصال داخلی
- بای پس کردن LUT، سپس عبور کردن از عنصر ذخیره کننده از طریق ورودی D قبل از خارج شدن همانند سیگنال XQ یا YQ.
- کنترل کردن مالتی پلکسر تابعی عریض F٥MUX یا F٦MUX.
- از طریق مالتی پلکسرها، با در نظر گرفتن بعنوان ورودی برای انتقال زنجیره ای
- راه اندازی کردن ورودی D١، مربوط به LUT
- قابل ذکر است BY می تواند کنترل ورودی REV مربوط به هردوی عناصر ذخیره کننده FFY و FFX را انجام دهد.
- و در پایان مالتی پلکسر DIG-MUX می تواند سیگنال BY را به خط DIG سوئیچ کند که در اینصورت از Slice خارج می شود.

۳-۱۸-۲- مولد توابع[۱]

هر دو LUTs (F,G) در یک اسلایس دارای ۴ ورودی منطقی (A١-A٤) و یک خروجی یکتای D هستند. این مورد، امکان اجازه برنامه ریزی عملیات منطقی هر ۴ متغیر از نوع بولین (Boolean) در داخل آنها را می دهد. بنابراین مالتی پلکسرهای توابع عریض، برای ترکیب LUT با CLB یکسان یا از میان CLB های متفاوت می تواند بطور مؤثری بکار برده شود. این امر باعث می شود که توابع منطقی با امکان داشتن ورودیهای بیشتر ساخته شوند. LUT های موجود در سمت راست و در سمت چپ جفت

١ - Function Generator

Slice ها نه تنها توابع منطقی را که در بالا به آن اشاره شد پشتیبانی می کنند، بلکه می توانند همانند ROM عمل کنند. این ROM با اطلاعات پیکربندی تراشه[۱] مقداردهی اولیه می شود.

LUT های موجود در سمت چپ جفت Slice ها مربوط به هر CLB، دو تابع اضافی تر را پشتیبانی می کند (همانند ستون X۰ در شکل ۳-۶). لازم بذکر است جفت Slice های سمت راست، این کار را انجام نمی دهند. اول اینکه این امکان نیز وجود دارد تا LUT سمت چپ را همانند یک RAM توزیع شده برنامه ریزی کرد. این نوع تهیه کردن حافظه، مقدار بافرینگ اطلاعات را هر جایی در طول مسیر اطلاعات متعادل می کند. LUT سمت چپ، ۱۶ بیت را ذخیره می کند. چندین LUT سمت چپ می توانند برای ذخیره کردن مقدار بیشتری از اطلاعات به روشهای گوناگون ترکیب شوند.

با انتخاب پورت دوگانه، دو LUT را ترکیب کرده، بنابراین دسترسی به حافظه از دو خط اطلاعات مستقل امکان پذیر می شود. گزینه ROM توزیع شده، امکان بارگذاری مجدد اطلاعات در حافظه FPGA در مدت پیکربندی آن را میسر می سازد.

دوم اینکه این امکان نیز وجود دارد که هر کدام از LUT های سمت چپ را همانند یک شیفت رجیستر ۱۶ بیتی برنامه ریزی کرد. با بکار بستن این روش، هر LUT می تواند تأخیری به مدت یک الی شانزده سیکل ساعت در اطلاعات سریال بوجود آورد. چهار LUT سمت چپ مربوط به یک CLB، توانایی ترکیب شدن برای تولید تأخیری تا ۶۴ سیکل ساعت را دارا هستند. خطوط SHIFTIN و SHIFTOUT، LUT را برای تولید شیفت رجیستری بزرگتر تقسیم می کند. همچنین این امکان وجود دارد که شیفت رجیسترها از میان تعدادی از CLB ها ترکیب گردند. تأخیر برنامه ریزی شده بوجود آمده حاصل از ترکیب، می تواند برای زمانبندی خط لوله اطلاعات[۲] بکار برده شود.

۳-۱۹- بلوک رم

۱ - Configuration
۲ - Timing Data Pipeline

تمام تراشه های خانواده اسپارتان۳، بلوکهای رم را پشتیبانی می کنند. این رم ها در بلوکهای ۱۸Kbit و بصورت سنکرون با قابلیت برنامه ریزی مجدد سازماندهی شده اند. بلوکهای رم نسبتاً مقدار زیادی از اطلاعات را خیلی مؤثرتر از بخش رم های توزیع شده[۱] که در بخش قبل توضیح داده شد، ذخیره می کنند. عرض و عمق هر رم قابل برنامه ریزی می باشد. بنابراین چندین بلوک می توانند ترکیب شوند تا اینکه حافظه های عریض تر و عمیق تری بدست آیند. یکی از انتخابهای اولیه این است که بلوک رم بعنوان حافظه پورت یکتا (Single) یا پورت دوگانه (Dual) بکار برده شود. بعنوان مثال نام RAM۱۶-S[Wa]-S[Wb] تعیین کننده حافظه پورت دوگانه است. در حالیکه اعداد صحیح Wa و Wb بترتیب تعیین کننده عرض کلی مسیر اطلاعات در پورتهای Wa و Wb می باشد. بنابراین RAM۱۶-S۹-S۱۸ یک رم با پورت دوگانه، با پورت عریض ۹ بیتی a و یک پورت عریض ۱۸ بیتی b. نامی در فرم RAMB۱۶-S[w] مشخص کننده یک رم یکتا است که پارامتر صحیح w، تعیین کننده عرض کلی مسیر اطلاعات یک پورت منفرد است. دیگر توابع حافظه FIFO، تبدیل عرض مسیر اطلاعات، ROM و غیره، با بکار بردن نرم افزار Core Generator در دسترس خواهند بود.

۳-۱۹-۱- آرایش بلوکهای رم در FPGA

تراشه XC۳S۵۰ دارای یک ستون از بلوک رم می باشد. تمام تراشه ها در رنج XC۳S۲۰۰ الی XC۳S۲۰۰۰ در بردارنده دو ستون از بلوکهای رم هستند. تراشه های سری ۴۰۰۰ و ۵۰۰۰ دارای چهار ستون بلوک رم هستند. مکان قرار گیری ستونهای بلوک رم در شکل ۳-۱ قابل مشاهده می باشد. در یک تراشه مورد بررسی تعداد کلی بلوکهای رم در دسترس، بطور برابر در طول ستونها توزیع شده است. جدول ۳-۵ نشان دهنده ی تعداد بلوکهای رم، ظرفیت ذخیره سازی اطلاعات و تعداد ستونها برای هر تراشه را نشان می دهد.

---

۱ - Distributed RAM

جدول ۳-۵: تعداد بلوکهای رم

Device	Total Number of RAM Blocks	Total Addressable Locations (bits)	Number of Columns
XC3S50	4	73,728	1
XC3S200	12	221,184	2
XC3S400	16	294,912	2
XC3S1000	24	442,368	2
XC3S1500	32	589,824	2
XC3S2000	40	737,280	2
XC3S4000	96	1,769,472	4
XC3S5000	104	1,916,928	4

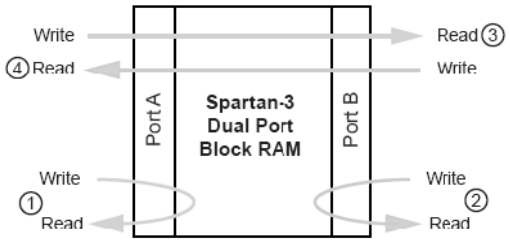
بلوک رم و ضرب کننده ها به یکدیگر متصل هستند تا اجرای عملیات های همزمان را امکان پذیر سازند. به هر حال با توجه به اینکه ضرب کننده ها ورودیهایش را با بیتهای اطلاعات بالایی مربوط به بلوک رم به اشتراک می گذارد، حداکثر عرض مسیر داده بلوک رم در این مورد ۱۸ بیت است.

۳-۱۹-۲- ساختار داخلی بلوک رم

بلوک رم یک ساختار پورت دوگانه دارد. دو پورت داده یکسان که A وB نامیده می شوند، اجازه دسترسی به بلوک رم را می دهند. حداکثر ظرفیت آنها ۱۸۴۳۲ بیت یا ۱۶۳۸۴بیت (در زمانیکه هیچ خط یکسانی بکار برده نشود) می باشد. هر پورت بصورت مجزا داده های تنظیم کردن تخصیص یافته ای دارد از جمله خطوط کنترل و خط کلاک برای عملیات همزمان خواندن و نوشتن.

همانطور که در شکل ۳-۷ نشان داده شده است، چهار مسیر اصلی داده وجود دارد:

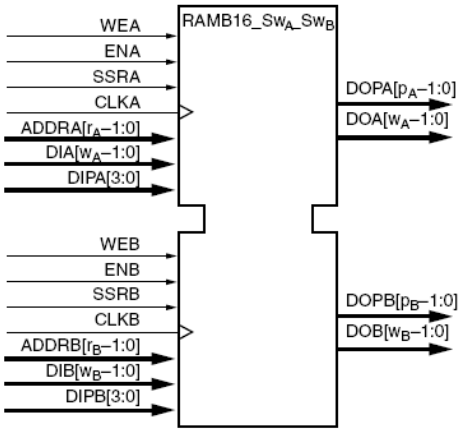
۱. نوشتن در پورت A و خواندن از پورت A
۲. نوشتن در پورت B و خواندن از پورت B
۳. انتقال داده ها از پورت A به پورت B
۴. انتقال داده ها از پورت B به پورت A



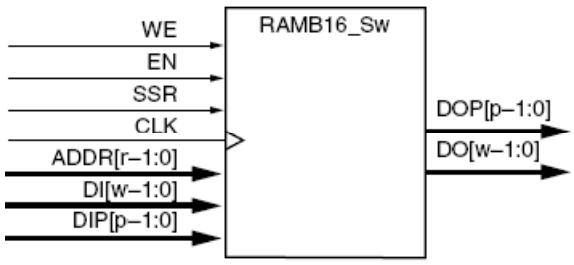
شکل ۳-۷- مسیر داده بلوک رم

۳-۱۹-۳- معرفی سیگنال پورت بلوک رم

پورت دوگانه RAMB۱۶-S[Wa]-S[Wb] و پورت یگانه RAMB۱۶-S[w] با سیگنالهای مربوط به آنها در شکلهای ۳-۸ و ۳-۹ بصورت جداگانه نشان داده شده است. سیگنالها در جدول ۱۲ برگه اطلاعات تراشه معرفی شده اند.



شکل ۳-۸-پورت دوگانه



شکل ۳-۹ پورت یگانه

۳-۱۹-۴- عملکرد address bus

باس آدرس، محل حافظه را برای عملیات نوشتن و خواندن انتخاب می کند. عرض پورت(w) مرتبط با مسیر داده، تعیین کننده تعداد خطوط آدرس در دسترس است. در زمانیکه پورت فعال شد(ENB=High یا ENA) ترانزیستورهای آدرس می بایستی زمان راه اندازی و نگهداری در ارتباط با کلاک را (CLKA یا CLKB) بگذرانند (همانطور که در برگه اطلاعات تراشه آمده است)[۷].

۳-۱۹-۵- باس ورودی داده

داده ها در ورودی باس DI در محل آدرس حافظه با لبه های فعال کلاک نوشته می شوند. این امکان وجود دارد که عرض مسیر داده پورت را بصورت ۱،۲،۴،۹،۱۸ یا ۳۶ بیتی برنامه ریزی نمود. این انتخاب برای هر دو مسیر DI وDO یک پورت داده شده بکار برده می شود. لازم بذکر است که هر پورت بصورت مستقل عمل می کند. برای پورتی با عرض(w)، تعداد محل های قابل آدرس دهی برابر است با $\frac{۱۶.۳۸۴}{W-P}$ که p، تعداد بیتهای همسان (parity) می باشد. هر محل حافظه دارای عرضی به اندازه w است.(دربرگیرنده بیت های مشابه parity)

ورودیهای parity، نشان دهنده بیت های اضافی که در مسیر داده هستند می باشد. این امر باعث تشخیص خطا می گردد. تعداد بیت های پریتی p که در باس DI هستند (بطور مشابه در DO) بستگی به عرض کلی مسیر داده دارد. این موضوع در جدول ۳-۶ آمده است.

۳-۱۹-۶- باس خروجی داده

دسترسی اساسی به داده زمانیکه WE غیر فعال است، رخ خواهد داد. خروجیهای DO، داده های ذخیره شده در محل آدرس دهی شده حافظه را منعکس می کنند.

اگر یکی از دو مشخصه زیر انتخاب شود، دسترسی به داده با فعال بودن WE نیز امکان پذیر است:

Write-First و Read-First:

Write-First نشان دهنده داده ورودی جدید در خروجی پورت DO است و داده را در محل آدرس RAM می نویسد. Read-First نشان دهنده داده قبلی ذخیره شده در رم بر روی خروجی پورت DO در زمانیکه داده جدید در رم نوشته می شود است. مشخصه سوم، No Change، خروجی DO را با فعال کردن WE، در حافظه بارگذاری می کند. این امکان وجود دارد که عرض مسیر داده کلی را در نرخهای ۱،۲،۴،۹،۱۸ یا ۳۶ بیت برنامه ریزی نمود. این برنامه ریزی برای دو مسیر DI و DO بکاربرده می شود.

خروجیهای داده Parity: همانند ورودیهای داده Parity می باشد.

فعال کننده عملیات نوشتن (Write Enable): در زمانیکه با سیگنال EN همراه شود، این ورودی نوشتن داده ها را در رم فعال می کند. هنگامیکه WE غیر فعال است و EN فعال، عملیات خواندن کماکان امکان پذیر است. در این حالت، یک حافظه معین، داده ها را از محل حافظه آدرس دهی شده به خروجی های DO عبور می دهد.

فعال ساز کلاک ساعت (Clock Enable): این ورودی، سیگنال کلاک را برای همزمان کردن بلوکهای رم فعال می کند. برای نوشتن داده ها در ورودیهای DI، برای بروز کردن داده ها در خروجیهای DO با ست/ریست کردن حافظه خروجی DO، زمانیکه EN سیگنال کلاک بکار برده نشود، عملیات فوق، غیر فعال می گردند.

Set/Reset: در زمانیکه بکار برده می شود، این پین خروجی DO را با مقداری که مشخصه SRVAL تنظیم شده است، یکسان می کند. عملیات ست/ریست در یک پورت بر عملیات و کارکرد پورتهای دیگر اثری ندارد و محتوای حافظه داده را دستکاری نمی نماید. این سیگنال با سیگنال کلاک همزمان می باشد.

کلاک: ورودی سیگنال کلاک برای عملیات خواندن و نوشتن است و اینکه این عملیاتها همزمان شوند. تمام ورودیهای پورت نیاز به زمان راه اندازی[۱] با درنظر گرفتن لبه های فعال سیگنال کلاک دارند. این موضوع برای باس خروجی داده نیز مطرح است.

۳-۲۰- نسبت ابعاد پورت[۲]:

در یک پورت در دسترس، امکان انتخاب تعداد متفاوتی از عرضها (w-p) برای باسهای DI/DO وجود دارد. در جدول ۳-۶ می توان این موضوع را مشاهده کرد. این دو باس، همیشه دارای عرض یکسانی هستند و انتخاب عرض باس داده برای هر پورت مستقل می باشد. اگر عرض باس داده پورت A از پورت B متفاوت باشد، بلوک رم بصورت اتوماتیک، عملیات همگون کردن باس ها را انجام می دهد. زمانیکه داده ها در یک پورت با عرض باس کم نوشته شده، سپس از یک پورت با باس عریض خوانده می شود، پورت دومی بصورت مؤثری باس های با عرض کم را ترکیب کرده تا باس عریضی را فرم دهد. بطور مشابه زمانیکه داده در یک پورت با باس عریض نوشته شده، سپس از یک پورت با عرض کم خوانده می شود، پورت دومی بخشهای عریض را به باسهای کم عرض تقسیم خواهد کرد. هنگامیکه باس داده ۸ بیت یا بیشتر است، بیت پریتی اضافی در دسترس می شود. عرض کلی مسیر داده (W) ار مجموع عرض باس DI/DO و هر بیت پریتی(P) بدست می آید.

جدول ۳-۶ : نسبت ابعاد پورتهای A وB

DI/DO Bus Width (w – p bits)	DIP/DOP Bus Width (p bits)	Total Data Path Width (w bits)	ADDR Bus Width (r bits)	No. of Addressable Locations (n)	Block RAM Capacity (bits)
1	0	1	14	16,384	16,384
2	0	2	13	8,192	16,384
4	0	4	12	4,096	16,384
8	1	9	11	2,048	18,432
16	2	18	10	1,024	18,432
32	4	36	9	512	18,432

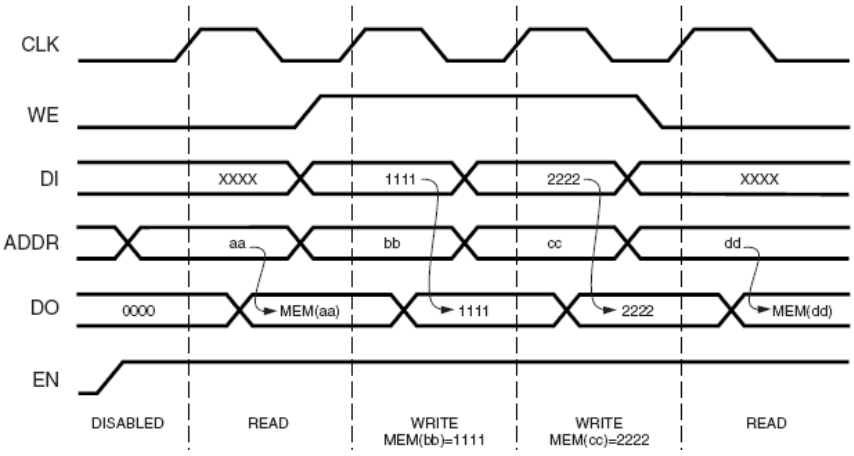
۱ - Setup Time
۲ - Aspect Ratio

۳-۲۱- عملیات داده بلوکهای رم:

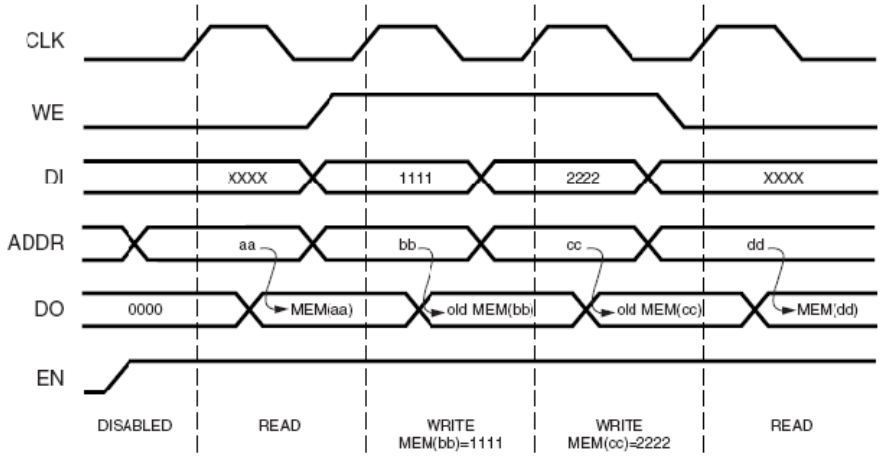
نوشتن داده و دسترسی به داده در بلوک رم، بصورت عملیاتهای همزمان هستند که بصورت مستقل در هر کدام از دو پورت رخ می دهد. هنگامیکه سیگنالهای WE و EN، لبه مؤثر کلاک را فعال می کنند، داده در باس ورودی DI در محل آدرس دهی شده بلوک رم توسط خطوط ADDR، نوشته می شوند.

تعداد شرایط متفاوتی وجود دارد برای اینکه کدام داده می تواند در خروجی DO در دسترس باشد. دسترسی پایه ای به داده همیشه تا زمانیکه ورودی WE غیر فعال است، اتفاق می افتد. تحت این شرایط، داده ذخیره شده در محل حافظه آدرس دهی شده، توسط خطوط ADDR از یک حافظه خروجی به خروجی های DO عبور می کند. زمانبندی پایه ای دسترسی به داده در بخشی از تصاویر ۳-۱۰ ،۳-۱۱ و ۳-۱۲ در مدت LOW بودن WE نشان داده شده است. همچنین هنگامیکه از ورودی WE استفاده می کنیم، داده می تواند در خروجیهای DO در دسترس قرار گیرد. این مورد با دو مشخصه زیر به سرانجام خواهد رسید:

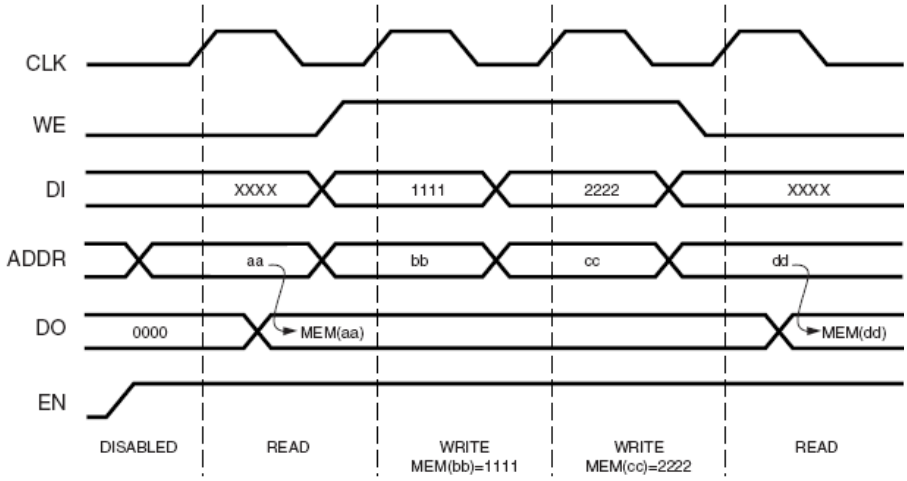
با انتخاب کردن مشخصه WRITE-FIRST، داده در محل آدرس دهی شده حافظه نوشته شده و همچنین به خروجی DO انتقال داده می شود. زمانبندی WRITE-FIRST، در بخشی از شکل ۳-۱۰ در زمانیکه WE، در حالت High است، نشان داده شده است.



شکل ۳-۱۰- شکل موج عملیات داده بلوک رم با انتخاب WRITE-FIRST



شکل۳-۱۱- شکل موج عملیات داده بلوک رم با انتخاب READ-FIRST



شکل ۳-۱۲- شکل موج عملیات داده بلوک رم با انتخاب NO-CHANGE

۳-۲۲- ضرب کننده های اختصاصی[۱]

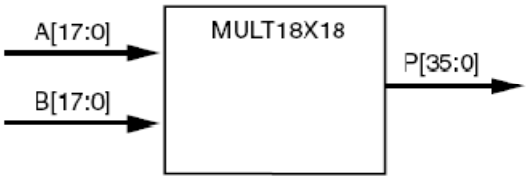
تمامی تراشه های اسپارتان۳، دارای ضرب کننده های تعبیه شده ای هستند که دو کلمه ۱۵ بیتی را بعنوان ورودی دریافت کرده و یک خروجی ۳۶ بیتی را تولید می کند. باس ورودی ضرب کننده، داده ها را در دو فرم مکمل دریافت می کند (۱۸ بیت علامتدار یا ۱۷ بیت بی علامت). در ابتدا ضرب کننده با هر بلوک رم همسان می شود. نزدیکی فیزیکی ایندو، داده گردانی مؤثر را تضمین می کند. با آبشاری[۲] بستن

۱ -Dedicated Multiplier
۲ -Cascading

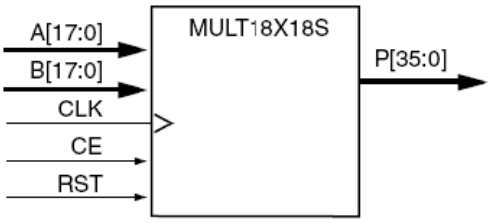
ضرب کننده ها، اجازه محاسبه ضرب بیش از سه عدد را می دهد. بعلاوه اینکه عریض تر از ۱۸ بیت نیز می توانند باشند.

ضرب کننده ها در یک طراحی با دو روش زیر بکار برده می شوند:

با یک مدل آسنکرون که MULT۱۸*۱۸ نامیده می شود (شکل ۳-۱۳) و مدلی با یک رجیستر (شکل ۳-۱۴) سیستم تولید کننده Core، ضرب کننده ها را بر اساس این فرضیات اولیه تولید کرده که می تواند پیکربندی شود تا رنج وسیعی از نیازها را برآورده نماید.



شکل ۳-۱۳-ضرب کننده ۱۸ بیتی آسنکرون



شکل۳-۱۴- ضرب کننده ۱۸بیتی با رجیستر

۳-۲۳- مدیریت کلاک دیجیتال[۱](DCM)

تراشه های اسپارتان۳، فراهم کننده کنترل کامل از طریق فرکانس کلاک، شیفت فاز و ... بوسیله واحد DCM هستند. DCM از یک حلقه قفل تأخیر (Delay Locked Loop)، (در واقع یک سیستم کنترل دیجیتال کامل) برای حفظ کردن کاراکترهای سیگنال کلاک با دقت بالا بکار می بندد، حتی با وجود اینکه تغییرات نرمالی را در ولتاژ و درجه حرارت داشته باشیم. هر خانواده از اسپارتان۳، دارای ۴ واحد DCM

۱ -Digital Clock Manager

است بجز اولین عضو XC۳S۵۰ که دو واحد DCM دارد. DCM ها در قسمت انتهایی ستونهای بلوکهای رم قرار گرفته اند (شکل۳-۱).

قابل توجه است که DCM سه کار مهم را پشتیبانی می کند:

- رفع انحراف سیگنال کلاک:

انحراف کلاک مشخص کننده مقداری است که ممکن است سیگنال کلاک، پایین تر از آن شرایط مطلوب قرار گیرد. این مورد زمانی اتفاق می افتد که تفاوتهای کمی در مسیرهای تأخیر باعث می شود که سیگنال کلاک در زمانهای متفاوتی به مکانهای مختلفی برسد. این انحراف کلاک باعث افزایش زمان Setup و Hold time شده که این مورد در فرکانسهای بالا نامطلوب می باشد. مخصوصاً زمانیکه مبحث تایمینگ مد نظر است. DCM با پس خور گرفتن از کلاک، باعث حذف انحراف سیگنال کلاک و خطاهای احتمالی می گردد. در نتیجه دو سیگنال کلاک، بوجود آورنده ارتباط فاز صفر هستند (Zero phase). DCM بطور مؤثری از بین برنده تأخیر توزیع کلاک خروجی می باشد. این سیگنال ممکن است در مسیر سیگنالی که از کلاک خروجی DCM به ورودی پس خور آن است، واقع شده باشد.

- سنتز فرکانس:

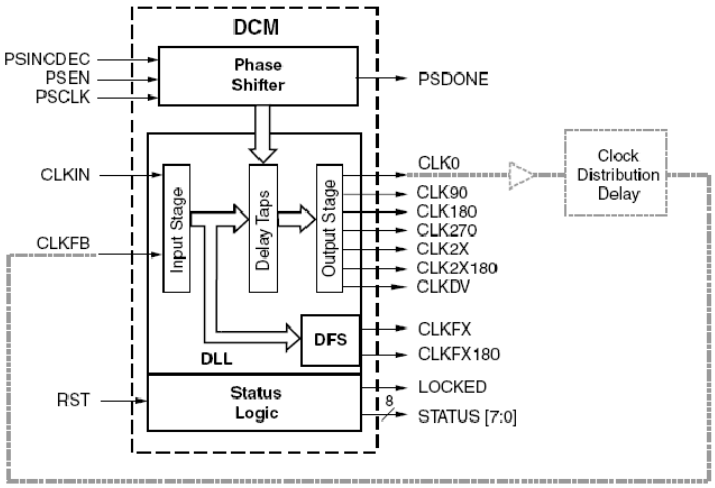
سنتز فرکانس با یک سیگنال ورودی کلاک فراهم شده است. DCM می تواند طیف وسیعی از فرکانسهای خروجی مختلف کلاک را تولید کند. این عمل با ضرب یا تقسیم فرکانس سیگنال ورودی ساعت توسط هریک از عوامل گوناگون انجام می شود.

- شیفت فاز:

DCM توانایی شیفت فاز از تمامی خروجیهای سیگنالهای کلاک با توجه به سیگنال ساعت ورودی را دارد.

همانطور که در فوق اشاره شد، DCM دارای چهار مؤلفه عملکردی است:

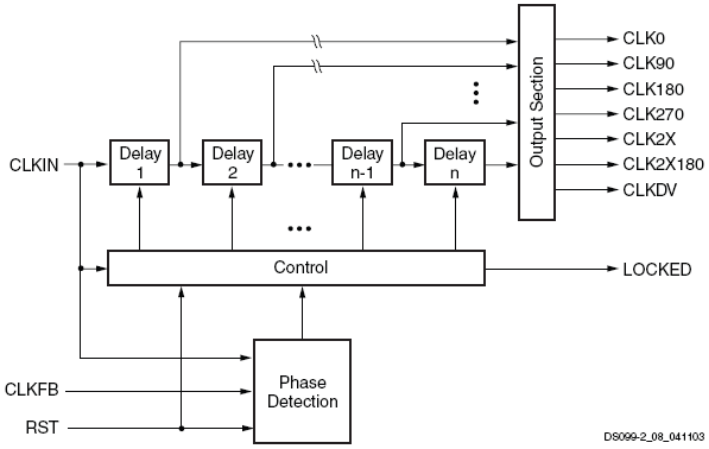
DLL، سنتز کننده دیجیتال فرکانس، شیفت دهنده فاز و Status Logic. همانطور که در شکل ۳-۱۵ نشان داده شده است، هر مؤلفه سیگنالهای مرتبط با آن را دارد.



شکل ۳-۱۵- بلوک کارکرد DCM و سیگنالهای مرتبط آن

۳-۲۴- بررسی حلقه قفل تأخیر

عملکرد اصلی DLL، حذف انحراف کلاک است. مسیر اصلی سیگنال DLL در برگیرنده یک مرحله ورودی است که توسط یک سری از عناصر گسسته تأخیر دنبال می شود که به نوبه خود به مرحله خروجی منجر می شود. این مسیر با منطق برای تشخیص فاز و کنترل به شکل یک سیستم کامل با پس خور در شکل ۳-۱۶ نشان داده شده است.



شکل ۳-۱۶: نمودار ساده شده عملکرد DLL

مؤلفه های DLL، دارای دو ورودی کلاک هستند. CLKIN و CLKFB با هفت خروجی.

مجموعه ای از خروجی ها در حالت فرکانس پایین در دسترس هستند. سیگنالهای کلاکی که به ورودی CLKIN داده شده بعنوان شکل موج مرجع در نظر گرفته می شود. DLL عملیات جستجو را انجام داده تا اینکه سیگنال فیدبک را در ورودی CLKFB تعیین کند.

در زمان حذف انحراف سیگنال، رویکرد معمول برای استفاده از DLL بشرح زیر است:

سیگنال CLK۰، از طریق شبکه توزیع کلاک به تمامی رجیسترهای آن گذرانده می شود. این رجیستر ها هم بصورت داخلی و هم خارجی در FPGA هستند. بعد از گذشتن از شبکه توزیع کلاک، سیگنال کلاک به DLL از طریق خط فیدبک که CLKFB نامیده می شود، بر می گردد. بلوک کنترل در درون DLL، خطای فاز را بین CLKIN و CLKFB اندازه گیری می کند. این خطای فاز، میزانی از انحراف کلاک است که شبکه توزیع کلاک تولید کرده است. بلوک کنترل تعداد مناسبی عناصر تأخیر را برای لغو انحراف کلاک فعال می کند. در زمانیکه DLL، سیگنال CLK۰ را هم فاز با CLKIN دید، باعث می شود که خروجی قفل شود. این عمل نشان دهنده قفل شدن در سیگنال CLKIN می باشد.

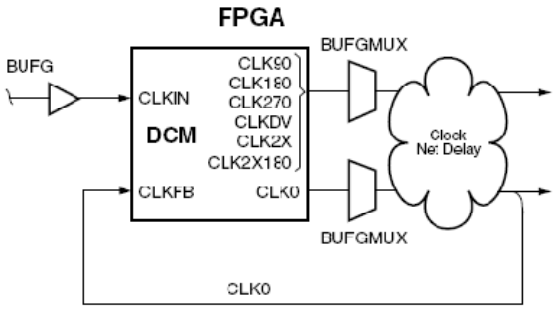
۳-۲۴-۱- ویژگیهای DLL و توابع مرتبط

تعدادی از گزینه های مختلف عملکردی، می تواند برای مولفه های DLL تنظیم شود. این عمل از طریق استفاده از ویژگیهای شرح داده شده در برگه اطلاعات تراشه قابل انجام است.

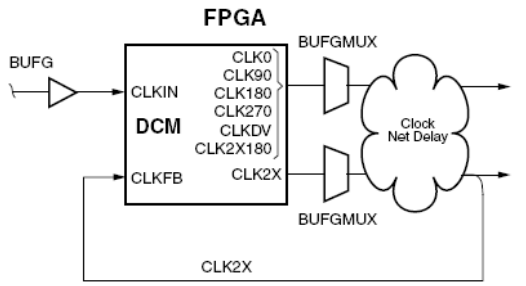
۳-۲۴-۲- ارتباطات ورودی کلاک DLL

یک منبع کلاک خارجی به FPGA از طریق بافر ورودی کلاک (IBUF) وارد می شود. این منبع بطور مستقیم به شبکه عمومی کلاک یا یک بافر ورودی دسترسی دارد (IBUF). سیگنالهای کلاک درون FPGA راه انداز یک شبکه عمومی کلاک هستند. این امر با استفاده از بافر مالتی پلکسر کلاک انجام می

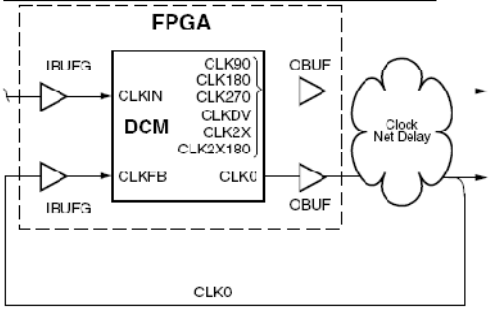
شود (BUFGMUX). شبکه عمومی کلاک، مستقیماً به ورودی CLKIN متصل می شود. ارتباطات داخلی و خارجی در شکلهای ۳-۱۶-۱ الی ۳-۱۶-۴ نشان داده شده است.



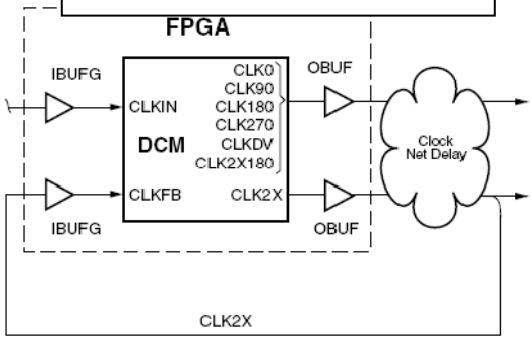
شکل ۱۶-۱- on chip با پس خور CLK۰



شکل ۱۶-۲- on chip با پس خور CLK۲X



شکل ۱۶-۳- off chip با پس خور CLK۰



شکل۱۶-۴- off chip با پس خور CLK۲x

۳-۲۴-۳- خروجی کلاک DLL و اتصالات فیدبک

چهار خروجی از نه خروجی کلاک DCM توانایی راه اندازی چهار بافر (BUFGMUX) را در هر دو جهت بالا و پایین دارند. تمام خروجیهای کلاک DCM، می تواند منابع مسیریابی عمومی را راه اندازی کند (شامل اتصالات داخلی بافرهای OBUF). حلقه فیدبک برای عملیات DLL ضروری است و با راه اندازی ورودی CLKFB بهمراه سیگنالهای CLK۰ یا CLK۲X تأسیس می شود. بنابراین هر تأخیر توزیع ناخواسته کلاک در حلقه فیدبک وجودخواهد داشت. همچنین این امکان وجود دارد که این دو سیگنال را برای همزمانی هر کدام از هفت خروجیهای DLL بکار برد. CLK۰، CLK۹۰، CLK۱۸۰، CLK۲۷۰، CLKDV، CLK۲X یا CLK۲*۱۸۰ مقداری که به CLK_FEEDBACK اختصاص می یابد، می

بایستی با اتصالات فیزیکی فیدبک تطابق داشته باشد. بعنوان مثال مقدار ۱X برای مورد CLK۰، ۲X برای مورد CLK۲X. اگر که DCM در یک کاربردی استفاده شود که نیازی به DLL نداشته باشد (فقط DFS بکار برده شود)، بنابراین حلقه فیدبکی وجود نخواهد داشت و CLK_FEEDBACK تنظیم نمی شود.

فیدبک CLK۲X فقط در سریهای 'E' و بالاتر پشتیبانی می شود. دو گزینه اساسی برای تعیین نحوه اتصال خروجی کلاک DLL و اتصالات فیدبک وجود دارد؛ همزمان سازی دورن تراشه ای (on chip) و همزمان سازی off chip. این موضوع در شکلهای ۳-۱۶-۱ الی ۳-۱۶-۴ نشان داده شده است.

در مورد همزمان سازی on-chip، این امکان وجود دارد که هر کدام از هفت خروجی سیگنال کلاک را از طریق منابع مسیر دهی عمومی به رجیسترهای درونی FPGA متصل کرد. همچنین یک بافر عمومی کلاک (BUFG) یا BUFGMUX موجب دسترسی به شبکه عمومی کلاک می شود. حلقه پس خور با مسیر دهی CLK۰(یا CLK۲X) به شبکه عمومی کلاک درست شده است که به نوبه خود، ورودی CLKFB را راه اندازی می کند.

در مورد همزمان سازی Off-chip (شکلهای ۳-۱۶-۱ و ۳-۱۶-۴)، CLK۰ (یا CLK۲X) بعلاوه هر کدام از سیگنالهای کلاک خروجی دیگر DLL که از FPGA خارج شده اند، بافرهای خروجی را (OBUF) برای راه اندازی یک شبکه کلاک خارجی، بعلاوه رجیسترهای در درون برد، بکار می بندند. همانطور که در شکل ۳-۱۶-۳ نشان داده شده است، حلقه فیدبک توسط CLK۰ با بکاربردن یک IBUF به درون FPGA تشکیل شده که بطور مستقیم به شبکه کلاک عمومی دسترسی دارد.

۳-۲۴-۴- مدهای فرکانس DLL

DLL دو مُد مجزای راه اندازی را پشتیبانی می کند. این مُد ها عبارتند از: فرکانس بالا و فرکانس پایین که هر کدام در یک رنج فرکانسی مختلف، مشخص شده است. زمانیکه این مد در حالت low باشد، هرکدام از هفت خروجیهای کلاک DLL در حالت فرکانسهای متوسط به پایین راه اندازی می شوند.

همچنین در حالت High این قضیه برای CLK۰، CLK۱۸۰ و CLKDV وجود دارد. مابقی خروجیهای کلاک DLL، برای بکار برده شدن در رنج فرکانس بالا در دسترس نیستند.

- انطباق ورودیهای فرکانس بالا

اگر که فرکانسهای سیگنال CLKIN بالا بود (حالتی که بیشترین مقدار ورودی مجاز را دارد بود) آنرا توسط یک CLK_DIVIDE_BY_۲ به یک مقدار قابل قبول تقسیم می کند. زمانیکه این مورد به مقدار True تنظیم شده باشد، فرکانس CLKIN توسط یک عامل از دو عاملی که به DCM وارد می شود تقسیم می شود.

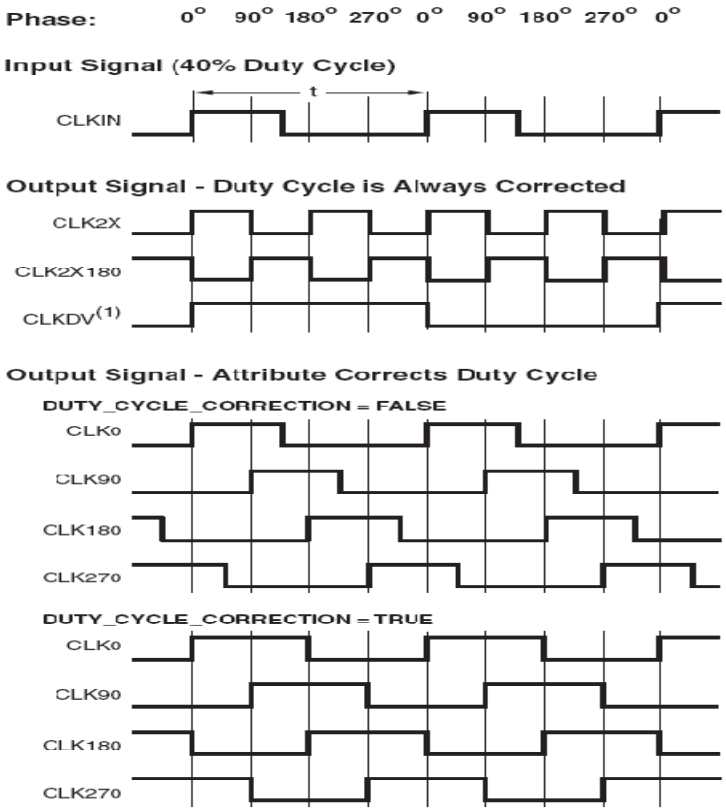
۳-۲۴-۵- خروجیهای شیفت فاز مربوط به مولفه DLL

علاوه بر CLK۰، برای همترازی شیفت فاز سیگنال CLKIN، DLL همچنین خروجیهای CLK۹۰، CLK۱۸۰ و CLK۲۷۰ را برای سیگنالهای شیفت فاز شده به اندازه ۹۰°، ۱۸۰° و ۲۷۰° فراهم می کند. زمانبندی در مد فرکانس پایین در شکل ۳-۱۷ نشان داده شده است. قابل توجه است که سیگنالهای خروجی CLK۹۰، CLK۱۸۰ و CLK۲۷۰ در مد فرکانس بالا در دسترس نیستند. برای کنترل بهتر از افزایش ۹۰° به مبحث شیفت فاز رجوع شود.

۳-۲۴-۶- خروجیهای اساسی سنتز فرکانس مربوط به اجزاء DLL

اجزاء DLL، گزینه های اساسی را برای ضرب و تقسیم فرکانسی برای رسیدن به انعطاف پذیری بیشتر نسبت به اجزاء DFS را دارا هستند. این عملیات در سیگنال کلاک خروجی، با فرکانس هایی که کسری یا مضربی از فرکانس کلاک ورودی باشند، نتیجه می شود. سیگنال CLK۲X یک سیگنال هم فاز را که دو برابر فرکانس CLKIN است تولید می کند. خروجی CLK۲X۱۸۰ همچنین فرکانس ورودی را دو برابر کرده ولی با این تفاوت که ۱۸۰ درجه اختلاف فاز نسبت به آن را داراست. خروجی CLKDIV فرکانس کلاکی را تولید می کند که کسر از پیش تعیین شده فرکانس CLKIN است. سیگنال

CLKDV_DEVIDE، تعیین کننده عاملی است که فرکانس CLKIN را تقسیم می کند. زمانبندی مرتبط در مد فرکانس پایین در شکل ۳-۱۷ نشان داده شده است. قابل توجه است که خروجیهای CLK۲X و CLK۲X۱۸۰ در حالت فرکانس بالا در دسترس نیستند.



شکل ۳-۱۷: خروجیهای کلاک DLL

۳-۲۵- سنتزکننده فرکانس دیجیتال[۱] (DFS)

اجزاء DFS، سیگنال های فرکانس زمان سنجی که محصول فرکانس زمان سنجی در ورودی CLKIN و نسبت دو عدد صحیح تعیین شده از سوی کاربر می باشد را تولید می کند. با توجه به طیف گسترده ای از فرکانس خروجی ممکن از جمله نسبت مجوزها، ویژگیهای DFS هنوز هم انعطاف پذیری بیشتر نسبت گزینه های ترکیب اساسی DLL همانطور که در بخش قبل شرح داده شده است، را ارائه می کند. دو جزء

۱ -Digital Frequency Synthesizer

DFS به خروجی های CLKFX و CLKFX۱۸۰، همانطور که در جدول ۳-۶ تعریف شده است، تخصیص می یابند.

جدول ۳-۶- سیگنالهای DFS

Attribute	Description	Values
DFS_FREQUENCY_MODE	Chooses between High Frequency and Low Frequency modes	Low, High
CLKFX_MULTIPLY	Frequency multiplier constant	Integer from 2 to 32
CLKFX_DIVIDE	Frequency divisor constant	Integer from 1 to 32

سیگنال در خروجی CLKFX۱۸۰ اساساً برگردان از سیگنال CLKFX می باشد. این دو خروجی همیشه در چرخه وظیفه ۵۰ ٪ نشان داده می شوند. همچنین این درست است که سیگنال CLKIN نشان داده نمی شود. خروجی های کلاک DFS هم زمان با خروجی های هفتمین کلاک DLL مشتق می شوند.

شمارشگر نسبت، مقدار عدد صحیح اختصاص یافته به ویژگی CLKFX_MULTIPLY و مخرج، مقدار عدد صحیح اختصاص یافته به به ویژگی CLKFX_DIVIDE می باشد. این ویژگی ها در جدول ۳-۷ شرح داده شده اند.

جدول ۳-۷- ویژگیهای DFS

Attribute	Description	Values
DFS_FREQUENCY_MODE	Chooses between High Frequency and Low Frequency modes	Low, High
CLKFX_MULTIPLY	Frequency multiplier constant	Integer from 2 to 32
CLKFX_DIVIDE	Frequency divisor constant	Integer from 1 to 32

فرکانس خروجی (fCLKFX) می تواند به عنوان تابعی از فرکانس کلاک ورودی (fCLKIN) به شرح زیر توضیح داده شود :

$$fCLKFX = fCLKIN*(CLKFX_MULTIPLY / CLKFX_DIVIDE) \qquad (۱-۳)$$

با توجه به این دو ویژگی، این امکان وجود دارد تا هر ترکیبی از مقادیر عدد صحیح، به شرطی که دو شرط را داشته باشند، به یکدیگر اختصاص یابند :

۱. دو عدد در درون محدوده خود قرار می گیرند.
۲. فرکانس fCLKFX از عبارت بالا مطابق با مشخصات فرکانس عامل DCM محاسبه می شود.

برای مثال، اگر CLKFX_MULTIPLY = ۵ و CLKFX_DIVIDE = ۳ ، فرکانس سیگنال کلاک خروجی از سیگنال کلاک ورودی، ۵/۳ خواهد بود.

۳-۲۵-۱- حالت های فرکانس DFS

DFS دو حالت عملکردی را پشتیبانی می کند، فرکانس بالا و پایین فرکانس، که هر یک با بیش از یک محدوده مشخص و فرکانس کلاک مختلف مشخص میشود. ویژگی DFS_FREQUENCY_MODE بین دو حالت را انتخاب می کند. هنگامی که ویژگی در LOW تنظیم می شود، حالت فرکانس پایین اجازه می دهد تا دو خروجی DFS در محدوده فرکانس پایین تا متوسط عمل کنند. هنگامی که ویژگی در حالت HIGH تنظیم شود، حالت فرکانس بالا اجازه می دهد تا این دو خروجی در بالاترین فرکانس ممکن عمل کنند.

۳-۲۵-۲- DFS با DLL یا بدون DLL

جزء DFS می تواند با DLL یا بدون جزء DLL استفاده شود: بدون DLL، جزء DFS فرکانس سیگنال CLKIN را با توجه به CLKFX_MULTIPLY مربوطه و مقدارهای CLKFX_DIVIDE تکثیر و یا تقسیم می کند و یک کلاک را با یک فرکانس هدف جدید در CLKFX و خروجی های CLKFX۱۸۰ تولید می کند. اگرچه آنها متعلق به به جزء DLL طبقه بندی می شوند، ورودی CLKIN

با جزء DFS به اشتراک گذاشته می شوند. این مورد نمی تواند از حلقه بازخورد بهره بگیرند؛ بنابراین نمی تواند برای تأخیر توزیع کلاک[1] درست باشند.

با DLL، DFS همانطور که در قبل شرح داده شده، تنها با بهره مندی بیشتر از حذف تأخیر پخش کلاک عمل می کند. در این مورد، یک حلقه بازخورد از خروجی CLK۰ به ورودی CLKFB باید وجود داشته باشد. اجزای DLL و DFS با هم کار می کنند تا به این مرحله تصحیح برسند که به شرح زیر است: با توجه به مقدارهای داده شده برای CLKFX_MULTIPLY و ویژگیهای CLKFX_DIVIDE، DLL عنصر تأخیر را که خروجی لبه کلاک همزمان با لبه کلاک ورودی باشد را هر وقت که از نظر علم ریاضی امکان پذیر باشد، انتخاب می کند. برای مثال وقتی CLKFX_MULTIPLY = ٥ و CLKFX_DIVIDE = ٣ ، لبه های کلاک ورودی و خروجی همزمان با هر سه دوره ورودی که معادل با زمان دوره های خروجی پنج می باشد، خواهد شد.

مقدارهای کوچکتر CLKFX_MULTIPLY و CLKFX_DIVIDE سریعتر به زمان قفل می رسند. بدون داشتن عوامل مشترک در این دو ویژگی، هم ترازی یک بار با هر تعداد سیکل برابر با مقدار CLKFX_DIVIDE، رخ می دهد. بنابراین، توصیه می شود که کاربر این مقادیر را توسط عامل مشترک در هر جا که امکان پذیر بود، کاهش دهد. به عنوان مثال، با توجه به CLKFX_MULTIPLY = ٩ و CLKFX_DIVIDE = ٦، یک عامل از سه بازده CLKFX_MULTIPLY =٣و CLKFX_DIVIDE=٢ حذف می شود. در حالی که هر دو مقدار که در ضرب فرکانس کلاک با ٣/٢ نتیجه خواهد داد، دومین مقدار، DLL را به قفل سریع تر منجر می سازد.

۳-۲۵-۳- اتصالات خروجی کلاک DFS

دو مورد اساسی وجود دارد که چگونگی اتصال خروجی های کلاک DFS را تعیین می کند: بر روی تراشه و خارج از تراشه که هر کدام به ترتیب در شکل ۳-۱۶-۱ و ۳-۱۶-۳ نشان داده شده اند. این شبیه

---

۱ - clock distribution delay

به آنچه که قبلاً برای جزء DLL شرح داده شده است، می باشد. در مورد On Chip، این امکان وجود دارد تا اتصال دو سیگنالهای کلاک خروجی DFS از طریق منابع عمومی به ثبات داخلی FPGA انجام شود. در هر صورت بافر کلاک عمومی (BUFG) و یا BUFGMUX توانایی دسترسی به شبکه کلاک عمومی را دارا می باشد. حلقه بازخورد اختیاری از این طریق شکل گرفته است و CLK۰ را به شبکه عمومی کلاک مسیر دهی می نماید که از ورودی CLKFB گرفته می شود.

در مورد Off Chip، دو سیگنال کلاک خروجی DFS، بعلاوه CLK۰ برای حلقه بازخورد اختیاری، می تواند در FPGA با استفاده از بافر خروجی (OBUF) خارج شود تا شبکه کلاک را بعلاوه رجیسترهای درون برد راه اندازی کند. حلقه بازخورد با استفاده از سیگنال برگشتی CLK۰ به FPGA و با کمک IBUFG شکل می گیرد که به طور مستقیم به شبکه کلاک عمومی و یا IBUF دسترسی دارد. سپس، شبکه کلاک عمومی مستقیماً به ورودی CLKFB متصل می شود.

۳-۲۶- شیفت دهنده فاز[۱] (PS)

DCM دو روش برای کنترل فاز سیگنال خروجی کلاک DCM، نسبت به سیگنال CLKIN ارائه می کند: اول، نه خروجی کلاک وجود دارد که از DLL برای رسیدن به رابطه فاز مورد نظر استفاده می کند: CLK۰ ، CLK۹۰، CLK۱۸۰، CLK۲۷۰، CLK۲X، CLK۲X۱۸۰، CLKDV CLKFX ، و CLKFX۱۸۰. این خروجی ها کنترل فاز "درشت یا ضخیم " را موجب می شوند.

روش دوم از جزء PS که در این بخش شرح داده شده است، استفاده می کند تا درجه کنترل بهتری را ارائه دهد. جزء PS تنها زمانی در دسترس است که DLL در حالت فرکانس پایین عمل کند. فاز جزء PS کلاک های خروجی DCM را با بکارگیری تغییر فاز خوب[۲] (FPS) بین سیگنالهای CLKFB و CLKIN درون جزء DLL، شیفت فاز می دهد. کاربر می تواند این فاز خوب را به تفکیک پذیری پایین ۲۵۶ /۱ از چرخه CLKIN یا یک تب تأخیر (DCM_TAP)،(هر کدام بزرگتر است) جابجا کند. در هنگام استفاده،

۱ - Phase Shifter
۲ - Fine Phase Shift

جزء PS فاز همه نه سیگنال خروجی کلاک DCM را با هم جابجا می کند. اگر جزء PS همراه با خروجی کلاک DCM مانند CLK۹۰، CLK۱۸۰، CLK۲۷۰ ، CLK۲X۱۸۰ و CLKFX۱۸۰ استفاده شود، سپس تغییر فاز خوب آنچه که به فاز درشت از قبل اضافه شده را شیفت فاز می دهد.

۳-۲۶-۱- فعال کردن اجزای PS و انتخاب حالت

ویژگی CLKOUT_PHASE_SHIFT جزء PS را برای استفاده و انتخاب بین دو حالت عملکرد فعال می سازد. این ویژگی سه مقدار ممکن را دارد:

هیچ کدام (None)، ثابت (Fixed) و متغیر (Variable)

وقتی CLKOUT_PHASE_SHIFT در حالت None تنظیم می شود، جزء PS غیر فعال است و ورودی های آن، PSEN، PSCLK و PSINCDEC باید به GND متصل شوند. مجموعه ای از شکل موجها در شکل ۳-۱۸ موارد غیر فعال را نشان می دهد که در این شکل، وقتی DLL هم ترازی فاز صفر از سیگنال های CLKFB و CLKIN که بر اساس آن جزء PS که دیگر اثری ندارد، حفظ می کند. جزء PS با تنظیم ویژگی به یکی از دو مقدار ثابت یا متغیر فعال می شود و به ترتیب حالت فاز ثابت و فاز حالت متغیر را انتخاب می کند. این دو حالت در بخش های زیر شرح داده می شوند:

۳-۲۶-۲- تعیین شیفت فاز خوب

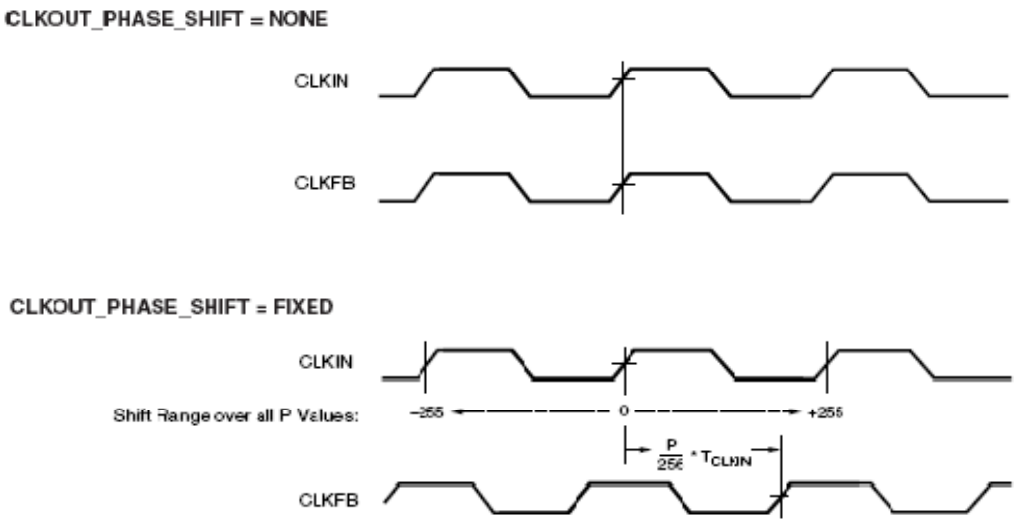
کاربر شیفت فاز از CLKFB نسبت به CLKIN را توسط تنظیم یا پذیرفتن مقدار ویژگی PHASE_SHIFT کنترل می کند. این مقدار باید یک عدد صحیح با دامنه ۲۵۵- تا ۲۵۵+ باشد. جزء PS از این مقدار برای محاسبه مورد نظر تغییر فاز خوب (Tps) به عنوان کسری از دوره تناوب CLKIN (TCLKIN) استفاده می کند. با توجه به مقدارهای داده شده برای PHASE_SHIFT و TCLKIN، ممکن است محاسبه TPS به شرح زیر باشد :

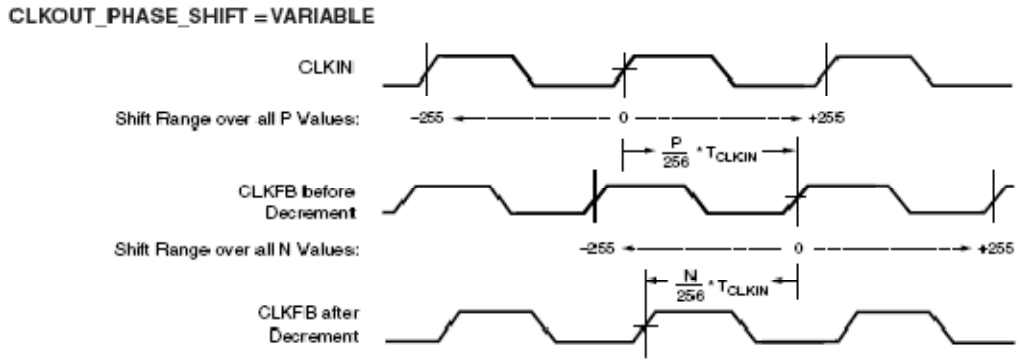
TPS =(PHASE_SHIFT/۲۵٦) * TCLKIN (۲-۳)

هر دو حالت های فاز ثابت و فاز متغیر از این محاسبات استفاده میکنند. اگر مقدار PHASE_SHIFT صفر باشد، همان زمانی که جزء PS غیر فعال باشد، CLKFB و CLKIN در فاز خواهد ماند. هنگامی که مقدار PHASE_SHIFT مثبت باشد، سیگنال CLKFB بعد از آن با توجه به CLKIN در زمان شیفت پیدا خواهد کرد. اگر مقدار ویژگی منفی باشد، سیگنال CLKFB پیش از آن با توجه به زمان CLKIN منتقل خواهد شد.

۳-۲۶-۳- حالت فاز ثابت

این حالت شیفت فاز مطلوب را بعنوان کسری از TCLKIN را با توجه به معادله تعیین شده (۳-۱) و مقدار کاربر P انتخاب شده PHASE_SHIFT ثابت می کند. مجموعه ای از شکل موجها در شکل ۳-۱۸ رابطه بین CLKFB و CLKIN در حالت فاز ثابت را نشان می دهد. در حالت فاز ثابت، PSEN ، PSCLK و ورودی های PSINCDEC استفاده نمی شوند و باید به GND متصل شوند. تغییر فاز ثابت نیاز به نسخه ۱۰.۱.۰۳ یا بالاتر نرم افزار ISE دارد.





شکل ۳-۱۸: شکل موجهای شیفت دهنده فاز

۳-۲۶-۴- حالت فاز متغیر

حالت " فاز متغیر " تغییر فاز خوب در زمان را بصورت پویا با استفاده از سه ورودی جزء PS، یعنی PSEN، PSCLK و PSINCDEC، تنظیم می کند. پس از پیکربندی دستگاه، جز PS در ابتدا TPS را توسط معادله ارزیابی (۳-۱) برای مقدار اختصاص یافته به ویژگی PHASE_SHIFT تعیین می کند. سپس به صورت پویا تنظیم می شود که تغییر فاز، از سه ورودی PS برای افزایش یا کاهش تغییر فاز خوب استفاده می کند.

PSINCDEC با سیگنال کلاک PSCLK، که خود توسط PSEN فعال شده است، همزمان شده است. ممکن است ورودی PSCLK را با سیگنال CLKIN و یا هر سیگنال کلاک دیگری مدیریت کرد. درخواست برای تنظیم فاز به شرح زیر آمده است:

برای هر چرخه PSCLK که PSINCDEC آن High می باشد، جزء PS ۱/۲۵۶ از چرخه CLKIN را به TPS اضافه می کند. به طور مشابه، برای هر چرخه PSCLK فعال شده که PSINCDEC در حالت Low است، جزء PS، ۱/۲۵۶ از چرخه CLKIN را از TPS کاهش می دهد. تنظیم فاز ممکن است نیاز به ۱۰۰ چرخه CLKIN به علاوه سه چرخه PSCLK داشته باشد تا تأثیر گذار باشد که در آن نقطه خروجی PSDONE برای یک چرخه PSCLK در حالت High قرار می گیرد. این پالس نشان می دهد که

جزء PS تنظیم حاضر را به پایان رسانده است و در حال حاضر آماده درخواست بعدی می باشد. در زمان راه اندازی مجدد (RST)، TPS را به زمان شیفت اصلی یعنی زمانیکه مقدار ویژگی PHASE_SHIFT تعیین شده باشد، باز می گرداند. مجموعه ای از شکل موجها در شکل ۳-۱۸ رابطه بین CLKFB و CLKIN را در حالت فاز متغیر نشان می دهد.

۳-۲۷- وضعیت اجزاء منطقی[۱]

وضعیت جزء منطق نه تنها حالت DCM را گزارش می کند بلکه با استفاده از مجموعه ای از راه اندازیهای مجدد[۲]، DCM را به حالت اولیه شناخته شده باز می گرداند. به عنوان یک قانون، ورودی ریست (RST) تنها بعنوان پیکربندی دستگاه یا تغییر فرکانس CLKIN استفاده می شود. ریست DCM تأثیری بر ویژگی ارزش ها نمی گذارد، (بعنوان مثال CLKFX_MULTIPLY و CLKFX_DIVIDE) و اگر استفاده نشود، RST باید به GND متصل شود.

۳-۲۸- تثبیت کلاک های DCM قبل از User Mode

ممکن است که اتمام پیکربندی دستگاه حتی تا بعد از اینکه DLL به شرایط قفل با استفاده از ویژگی STARTUP_WAIT، با تأخیر همراه باشد. این گزینه اطمینان می دهد که FPGA وارد حالت User Mode نشده است. (بعنوان مثال، آغاز عملیات تابعی[۳]) تا زمانی که همه کلاکهای سیستم که توسط DCM تولید شده است، پایدار شده باشند. به منظور دستیابی به تأخیر، لازم است ویژگی را به True و همچنین گزینه BitGen از LCK_cycle را به یکی از شش چرخه ساخت تا راه اندازی فاز پیکربندی تنظیم نمود. چرخه انتخاب شده، نقطه ای را که پیکربندی متوقف خواهد شد (تا زمانیکه خروجی قفل شده High خواهد شد) را تعیین می کند.

---

۱ - The Status Logic Component
۲ - means of resetting
۳ - functional operation

۳-۲۹- شبکه کلاک عمومی[۱]

تراشه های اسپارتان۳ هشت ورودی کلاک عمومی GCLK۰ - GCLK۷ را دارا هستند. این ورودی ها دسترسی به شبکه های با خازن کم[۲] و کم اریب[۳] که برای حمل سیگنال های فرکانس بالا مناسب هستند را فراهم می کند. شبکه کلاک اسپارتان۳ در شکل ۳-۱۹ نشان داده شده است. GCLK۰ الی GCLK۳ در مرکز لبه پایین واقع شده اند و GCLK٤ الی GCLK۷ در مرکز لبه بالا واقع شده است.

هشت کلاک عمومی Multiplexers (همچنین عناصر BUFGMUX نیز نامیده می شوند) ارائه می شوند تا سیگنال ها را از ورودی های کلاک عمومی بپذیرند و آنها را به شبکه کلاک داخلی و همچنین DCMs مسیردهی کنند. چهار عنصر BUFGMUX، در مرکز از پایین لبه درست بالای ورودی های GCLK۰ - GCLK۳ واقع شده اند. چهار عنصر باقی مانده BUFGMUX ، در مرکز از بالای لبه، درست بالای ورودی های GCLK۷ - GCLK٤ واقع شده اند. جفت عناصر BUFGMUX با ورودی عمومی، همانطور که در شکل ۳-۱۹ نشان داده شده است به اشتراک گذاشته می شوند. به عنوان مثال، ورودی های GCLK٤ و GCLK٥ هر دو به طور بالقّوه با BUFGMUX٤ و BUFGMUX٥ واقع در مرکز فوقانی راست در تماس هستند. ورودی کلاک تفاضلی از یک جفت ورودی GCLK برای اتصال به یک عنصر BUFGMUX تنها استفاده می کند. هر عنصر BUFGMUX ، که در شکل ۳-۱۹ نشان داده شده است، یک مالتی پلکسر ۲ به ۱ است که قابلیت دریافت سیگنال از هر چهار منبع زیر را دارا می باشد:

۱. یکی از چهار ورودی کلاک عمومی - در بالا یا پایین - زمانیکه عنصر BUFGMUX استفاده می شود.
۲. هر یک از چهار ورودی در نزدیکی دو خط افقی.

---

۱ - Global Clock Network
۲- low-capacitance
۳ - low-skew

۳. هر یک از چهار خروجی از DCM در ربع سمت راست، زمانیکه عنصر BUFGMUX استفاده می شود.

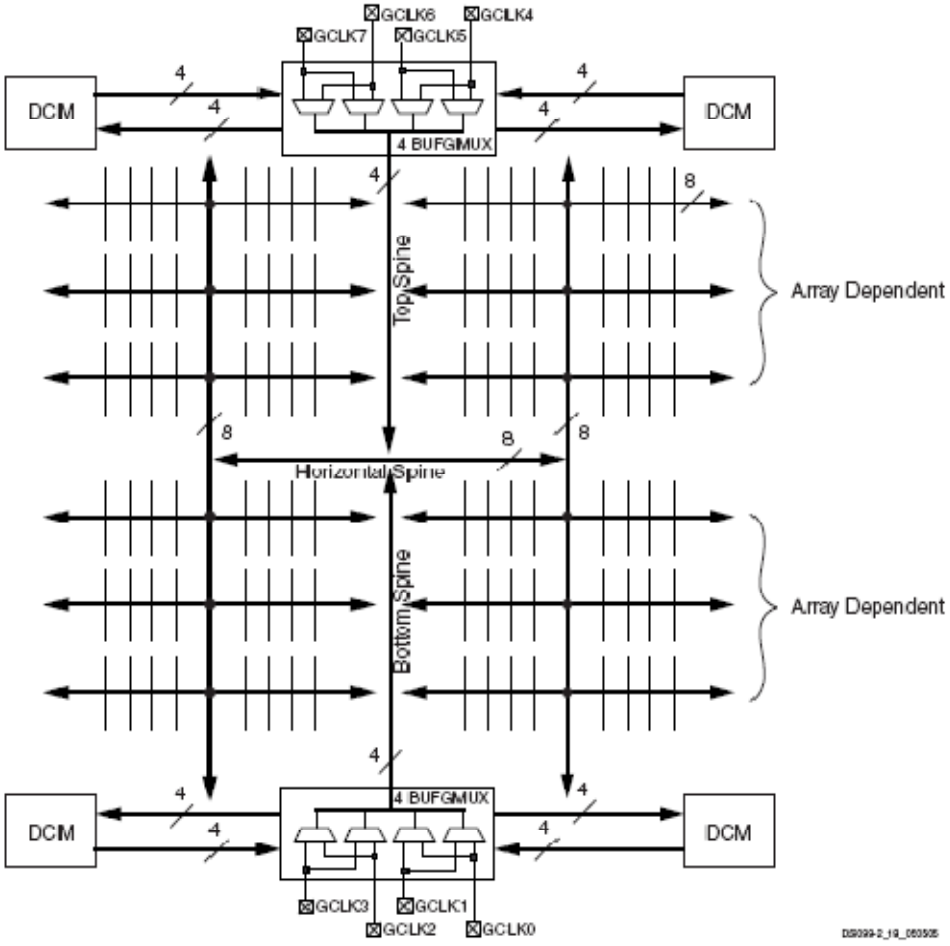
۴. هر یک از چهار خروجی از DCM در ربع سمت چپ، زمانیکه عنصر BUFGMUX استفاده می شود.

خط انتخاب مالتی پلکسر(S) انتخاب می کند که کدام یک از این دو ورودی I_0 یا I_1 ، از سیگنال خروجی BUFGMUX (O) که در جدول ۳-۸ شرح داده شده است، مشتق شود.

سوئیچ کردن از یک کلاک به کلاکهای دیگر باعث glitchless می شود (یک اشکال جزئی، یک اشکال کوچک که خصوصاً "به وسیله موج ولتاژ یا اختلال الکتریکی که باعث خطایی در ارسال داده می شود" ایجاد می گردد) و به گونه ای انجام می شود که زمان خروجی High و Low هیچ گاه کمتر از زمان High وLow هرکلاک ورودی نمی شود.

جدول ۳-۸- مکانیزم انتخاب BUFGMUX

S Input	O Output
0	I0 Input
1	I1 Input



شکل ۳-۱۹: شبکه کلاک اسپارتان۳

دو ورودی کلاک می تواند بدون توجه به یکدیگر ناهمزمان باشند، و ورودی S می تواند در هر زمان تغییر کند به جز زمان راه اندازی کوتاه قبل از لبه بالارونده از کلاک انتخابی حاضر (I۰ یا I۱). نیازمندیهای این تنظیم زمان می تواند منجر به خروجی پالس تعریف نشده بشود. بافر اولیه کلاک BUFG از سیگنال کلاک بر روی شبکه کلاک گرفته می شود و در اصل همان عنصر BUFGMUX است، فقط بدون مکانیسم انتخاب کلاک می باشد. بطور مشابه، BUFGCE اولیه، بافر کلاک فعالی را با استفاده از مکانیزم انتخاب BUFGMUX، ایجاد می کند. هرکدام از BUFGMUX سیگنالهای ورودی کلاک را به دو مقصد ممکن بافر می کنند:

۱. ستون عمودی متعلق به همان طرف- بالا یا پایین– هنگامی که عنصر BUFGMUX استفاده شود. دو ستون- بالا و پایین – هر کدام شامل چهار کلاک خطی عمودی، که هر یک از طریق یکی از عناصر BUFGMUX اجرا می شوند. سیگنال های کلاک به ستون افقی هشت خطی می رسند، که در دهانه قرار می گیرد. به نوبه خود، ستون افقی به کلاک تابع متصل که به CLBs دسترسی دارند، انشعاب می یابد.

۲. ورودی کلاک هر کدام از DCM در همان سمت قرار می گیرند- بالا یا پایین – هنگامی که عنصر BUFGMUX استفاده می شود.

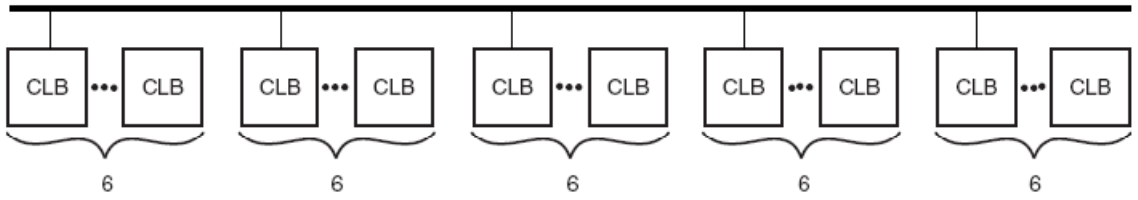
استفاده از هر دو عنصر BUFGMUX یا BUFG (بافر کلاک عمومی) یک ورودی عمومی در طراحی را قرار می دهد. به منظور به حداقل رساندن اتلاف قدرت دینامیکی شبکه کلاک، نرم افزار توسعه Xilinx به صورت خودکار همه بخش های خط کلاک که طراحی می شوند اما استفاده نمی شوند را غیر فعال می کند. خط کلاک عمومی بطور ایده آل ورودی کلاک را بر روی عناصر مختلف کلاک در داخل FPGA، راه اندازی می کند، از جمله فلیپ فلاپ های IOB یا CLB یا اینکه بلوکهای RAMs. خط کلاک عمومی نیز به صورت اختیاری از ورودی های ترکیبی گرفته می شود. با این حال، این کار بارگذاری اضافی در خط کلاک که همچنین ممکن است جیتر[۱] کلاک را تحت تأثیر قرار دهد. در حالت ایده آل، راه اندازی ورودی های ترکیبی از سیگنال استفاده می کنند که از ورودی BUFGMUX یا عنصر BUFG مشتق می شود. برای اطلاعات بیشتر ، به "استفاده از منابع عمومی در کلاک" در فصل UG۳۳۱ برگه اطلاعات تراشه مراجعه نمائید.

۳-۳۰- اتصالات درونی[۲]

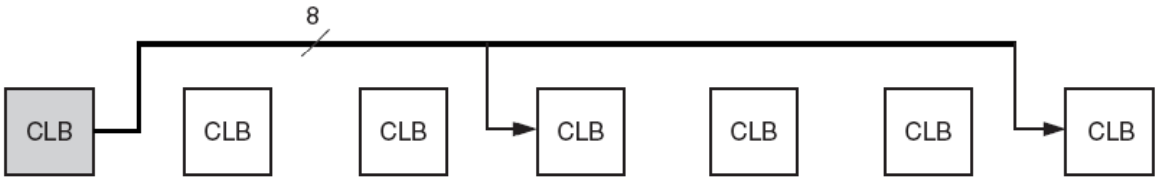
اتصالات درونی (یا مسیر دهی[۱]) سیگنالها را از میان عناصر مختلف توابعی از تراشه اسپارتان۳ عبور می دهد. لازم بذکر است که چهار نوع اتصال وجود دارد: خطوط طویل، خطوط Hex (مبنای شانزده)، خطوط

۱ -Jitter
۲ - Interconnect

دوگانه یا جفت، و خطوط مستقیم. خطوط طولانی به یکی از هر شش CLBs متصل می شوند (به شکل ۳-۲۰-۱ نگاه کنید). از آنجا که آنها ظرفیت خازنی پایینی دارند، این خطوط مناسب برای حمل سیگنال های فرکانس بالا و حداقل اثرات با بارگذاری می باشند (بعنوان مثال، سیگنال مورب). اگر تمام هشت ورودی کلاک عمومی از قبل با هم متصل شوند و سیگنال های کلاک اضافی باقی بمانند، برای این منظور خطوط طولانی جایگزین خوبی می باشند. خطوط Hex (مبنای شانزده) به هر یک از سه CLBs متصل می شود (به شکل ۳-۲۰ نگاه کنید). این خطوط میان خطوط طولانی و خطوط دوگانه (Double) با توجه به ظرفیت شان قرار می گیرند، رویکرد خطوط مبنای شانزده ویژگی های فرکانس بالا از خطوط طولانی را در همان زمان دارد و در نتیجه ارتباط بیشتر را فراهم می آورد. خطوط دوگانه به هر CLB متصل می شوند (به شکل ۳-۲۰ نگاه کنید). در مقایسه با انواع خطوط قبلی مورد بحث، خطوط دوگانه درجه بالاتری از انعطاف پذیری را هنگام اتصال ارائه می کنند. خطوط مستقیم هرگونه دسترسی مستقیم CLB به همسایه خود CLBsها را امکان پذیر می سازند (به شکل ۳-۲۰ نگاه کنید). این خطوط اغلب برای هدایت سیگنال از یک CLB "منبع" به خطوط دوگانه، مبنای شانزده، یا خطوط طویل و سپس از طریق اتصالات طولانی به یک خط مستقیم دسترسی به CLB "مقصد" استفاده می شوند. برای اطلاعات بیشتر، به "نحوه استفاده از اتصالات" در فصل UG۳۳۱ برگه اطلاعات تراشه مراجعه نمایید.

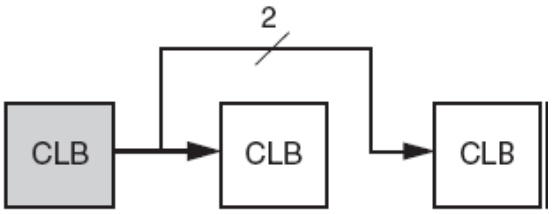


شکل ۳-۲۰-۱- خطوط طویل

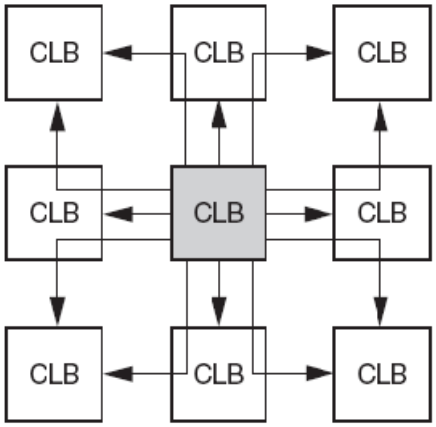


۱ -Routing

شکل ۳-۲۰-۲-خطوط Hex



شکل ۳-۲۰-۳- خطوط دوگانه



شکل ۳-۲۰-۴-خطوط مستقیم

۳-۳۱- پیکر بندی[۱]

تراشه های اسپارتان۳ توسط بارگذاری داده های خاصی در حافظه داخلی پیکربندی می شوند. تنظیمات با استفاده از زیر مجموعه ای از پینهای تراشه صورت می پذیرد که برخی از آنها "اختصاصی" فقط با یک عملکرد می باشند و در حالی که دیگران، با " هدفی دوگانه " صورت می گیرد و می توانند مورد استفاده دوباره به عنوان کارکرد عمومی I/O[۲] عامل پیکربندی قرار بگیرند. بسته به نوع طراحی سیستم، چندین حالت پیکربندی پشتیبانی می شود. پین های M۰، M۱، M۲ پینهای اختصاصی پیکربندی هستند. تنظیمات حالت پینها در جدول ۳-۹ نشان داده شده است.

۱ -Configuration
۲ - general-purpose User I/Os

جدول ۳-۹- تنظیمات پینهای پیکربندی اسپارتان۳

Configuration Mode(1)	M0	M1	M2	Synchronizing Clock	Data Width	Serial DOUT(2)
Master Serial	0	0	0	CCLK Output	1	Yes
Slave Serial	1	1	1	CCLK Input	1	Yes
Master Parallel	1	1	0	CCLK Output	8	No
Slave Parallel	0	1	1	CCLK Input	8	No
JTAG	1	0	1	TCK Input	1	No

پین ورودی HSWAP_EN چگونگی حالت پین های I/O را که در هنگام پیکربندی استفاده نمی شوند، تعریف می کند. به طور پیش فرض، HSWAP_EN در حالت High قرار دارد. که مقاومتهای Pull-up را در I/O در مدت پیکربندی غیر فعال می کند. زمانیکه HSWAP_EN در حالت Low باشد، I/O مقاومتهای Pull-up را در مدت پیکربندی دارند. پینهای پیکربندی اختصاصی (CCLK DONE، PROG_B، M۲، M۱، M۰، HSWAP_EN) و پین های JTAG (TDI ، TMS ، TCK و TDO) همیشه از مقاومت های Pull-up متصل به VCCAUX در طول پیکربندی استفاده میکند، صرف نظر از مقدار پین HSWAP_EN. به طور مشابه، پین دو کاره INIT_B یک مقاومت داخلی Pull-up متصل به VCCO_٤ یا VCCO_BOTTOM را بسته به بسته بندی تراشه دارد. بسته به نوع انتخاب حالت پیکربندی، FPGA خروجی کلاک CCLK ایجاد می کند، یا CCLK یک ورودی است که کلاک خروجی را پذیرا می باشد.

یک گزینه ای نیز در دسترس است که می تواند پین ها را برای حفظ عملکرد پیکربندی خود حتی پس از اینکه پیکر بندی دستگاه کامل باشد، مورد استفاده قرار دهد. اگر این گزینه انتخاب نشده باقی بماند، سپس پین های پیکربندی به استثنای CCLK، PROG_B ، DONE می توانند به عنوان I/O با عملکردی معمول مورد استفاده قرار بگیرند. این ویژگی در برنامه های کاربردی ارزشمند می باشد که داده های پیکربندی را در حالت User I/O خوانده می شود. جدول ۳-۱۰ تعداد کل بیت مورد نیاز برای پیکربندی هر یک از FPGA و همچنین PROMs مناسب برای ذخیره سازی بیت را نشان می دهد.

جدول ۳-۱۰- داده های پیکربندی اسپارتان۳

Device	File Sizes	Xilinx Platform Flash PROM	
		Serial Configuration	Parallel Configuration
XC3S50	439,264	XCF01S	XCF08P
XC3S200	1,047,616	XCF01S	XCF08P
XC3S400	1,699,136	XCF02S	XCF08P
XC3S1000	3,223,488	XCF04S	XCF08P
XC3S1500	5,214,784	XCF08P	XCF08P
XC3S2000	7,673,024	XCF08P	XCF08P
XC3S4000	11,316,864	XCF16P	XCF16P
XC3S5000	13,271,936	XCF16P	XCF16P

حداکثر طول bitstream که اسپارتان۳ پشتیبانی می کند، در زنجیره ۴۲۹۴۹۶۷۲۶۴ بیت (٤Gbits) می باشد، تقریباً معادل زنجیره با ۳۲۳ تراشه XC۳S۵۰۰۰. این تنها محدودیت برای زنجیره سریال است جایی که داده های پیکربندی FPGA از طریق پینDOUT عبور می کند. چنین محدودیتی برای زنجیره هایJTAG وجود ندارد.

۳-۳۱-۱- واسط پیکربندی استاندارد

سیگنالهای پیکربندی متعلق به یکی از این دو مقوله مختلف می باشند: اختصاصی و یا کارکرد دوگانه. که این دسته بندی مشخص می کند که کدام خطوط FPGA تغذیه سیگنال را فراهم می کند. پین های پیکربندی اختصاصی شامل PROG_B، HSWAP_EN ، TDI ، TMS، TCK، TDO، DONE، CCLK، و M۰ - M۲ می باشند. این پین ها توسط VCCAUX تغذیه می شوند. ۵ سیگنال در حالت های پیکربندی سریال استفاده می شوند که از VCCO_٤ تغذیه می کنند. هر دو سیگنال اختصاصی در بالا توضیح داده شده است و سیگنال های کارکرد دوگانه رابط پیکربندی را تشکیل می دهند. پین های اختصاصی، توسط VCCAUX ۲.۵V تغذیه می شوند، و همیشه از استاندارد I/O LVCMOS۲۵ استفاده می کنند. با این حال سیگنال های کارکرد دوگانه، توسط VCCO_٤ و همچنین VCCO_۵ در حالت پیکربندی موازی تغذیه می شوند. ساده ترین رابط پیکربندی از ۲.۵ ولت

برای ٤_VCCO و ٥_VCCO، در صورت لزوم استفاده می کند. با این حال، ٤_VCCO و در صورت نیاز، ٥_VCCO می توانند ولتاژ ۲.۵ ولت باشند اما پس از آن رابط پیکربندی ولتاژ دو سطح را خواهد داشت: ۲.۵ ولت برای VCCAUX و عرضه جداگانه VCCO. سیگنال های کارکرد دوگانه به طور پیش فرض سطوح ورودی و خروجی LVCMOS برای تامین ولتاژ VCCO می باشند.

۳-۳۱-۲- حالت های پیکربندی

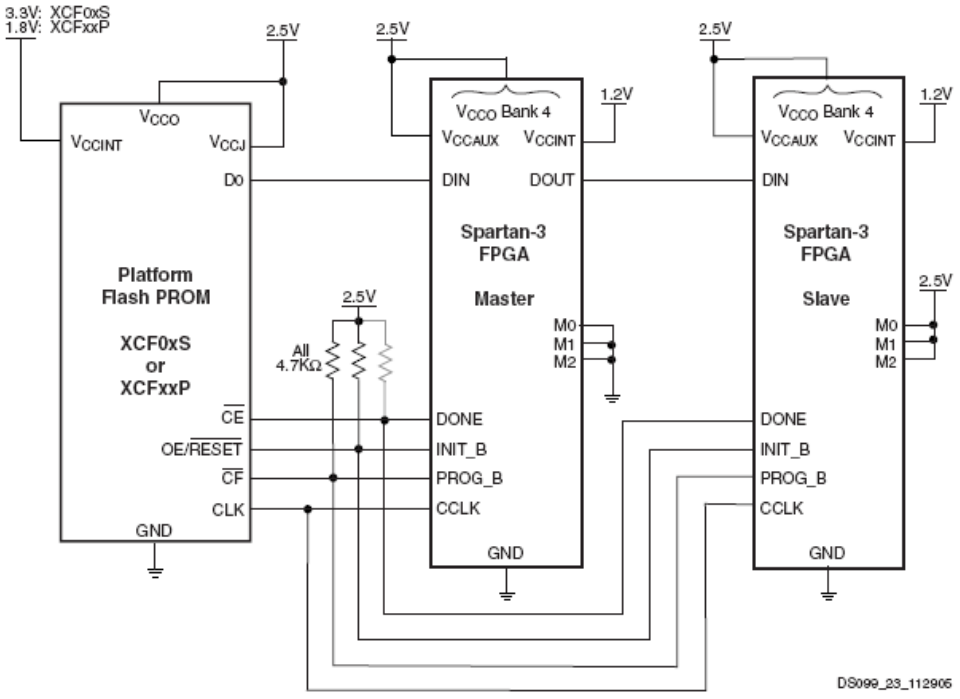
اسپارتان۳ از پنج حالت پیکربندی زیر پشتیبانی می کند:

- حالت Slave سریال
- حالت Master سریال
- حالت Slave موازی (SelectMAP)
- حالت Master موازی (SelectMAP)
- حالت اسکن مرزی (JTAG) (IEEE۱۵۳۲, IEEE۱۱٤۹.۱)

۳-۳۱-۲-۱- حالت Slave سریال

در حالت Slave سریال، FPGA اطلاعات پیکربندی در شکل بیت سریال از یک PROM سریال و یا سریال های دیگر منبع داده های پیکربندی، دریافت می کند. FPGA در سمت راست شکل ۳-۲۱ برای حالت Slave سریال تنظیم می شود. پین CCLK در FPGA ورودی در این حالت می باشد. Bitstream سریال باید در پین ورودی DIN مدت کوتاهی قبل از هر لبه بالارونده یک کلاک خارجی CCLK تنظیم شود. چند FPGAمی تواند، زنجیر اصلی برای پیکربندی از یک منبع جداگانه باشد. پس از اینکه FPGA خاص پیکربندی شد، داده ها برای دستگاه بعدی و پین به DOUT داخلی وارد می شوند. اطلاعات بر

روی پین های DOUT موجب بروز تغییرات در لبه CCLK می شود. حالت Slave سریال با استفاده از حالت پین <۱۱۱> (M۰، M۱، M۲) انتخاب می شود.



شکل ۳-۲۱- اتصالات مربوط به مستر سریال و اسلیو سریال

۳-۳۱-۲-۲- حالت Master سریال

در حالت Master سریال، FPGA پین های CCLK را راه اندازی می نماید، که به عنوان پین دو طرفه I/O رفتار می کند. FPGA در مرکز شکل ۳-۲۱ برای حالت سریال Master تنظیم می شود و به پیکربندی PROM سریال متصل می شود و و ورودی های CCLK از هر زنجیره اصلیFPGAs پیکربندی می شوند. FPGA از کلاک پیکربندی در پین CCLK به سریال Xilinx PROM، که در پاسخ، ارائه سریال داده های به ورودی FPGA DIN، ایجاد می شوند. FPGA این اطلاعات را در لبه بالارونده CCLK می پذیرد. پس از اینکه پیکربندی FPGA به اتمام رسید، آن داده را بر روی پین DOUT به دستگاه FPGA بعدی عبور می دهد. اطلاعات DOUT پس از لبه پایین رونده کلاک CCLK ظاهر می شوند.

رابط حالت Master سریال به Slave سریال به جز نوسان ساز داخلی که کلاک پیکربندی (CCLK) را ایجاد می کند، یکسان است. طیف گسترده ای از فرکانس را می توان همیشه برای CCLK انتخاب کرد، که در فرکانس پیش فرض از ۶ مگاهرتز شروع می شود. بیت های پیکربندی سپس CCLK را به فرکانسی بالاتر برای بقیه پیکربندی تبدیل می کنند.

۳-۳۱-۲-۳- حالت **Slave موازی** (SelectMAP)

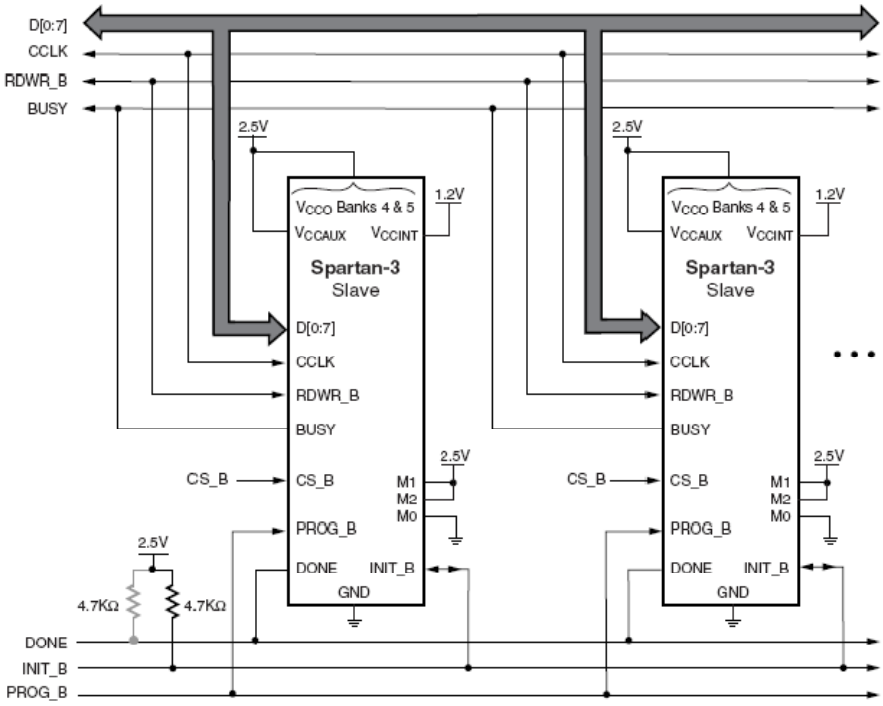
حالت های موازی یا SelectMAP از پیکربندی سریع پشتیبانی می کند. داده گسترده بایت[۱] بر روی FPGA نوشته می شود. با پرچم BUSY جریان داده ها را می توان کنترل نمود. منبع خارجی، داده گسترده ۸ بیتی را ارائه می کند: CCLK، سیگنال انتخاب تراشه active-Low (CS_B) و سیگنال نوشتن active-Low (RDWR_B).

اگر BUSY به FPGA اضافه شود (High)، سپس داده ها باید نگه داشته شوند تا زمانی که BUSY به حالت Low رود. داده ها همچنین می تواند با استفاده از حالت Slave موازی خوانده شوند. اگر RDWR_B اضافه شود، داده های پیکربندی FPGA را می توان بعنوان بخشی از عملیات خواندن مجدد خواند. پس از پیکربندی، ممکن است هر یک از پین های چند منظوره (DIN/D۰-D۷، DOUT، INIT_B، CS_B و RDWR_B) را بعنوان I/O مورد استفاده قرار داد. برای این کار، به سادگی می توان گزینه اصرار BitGen را بر روی NO تنظیم کرد. سیگنال های دلخواه برای پیکربندی پین های چند منظوره با استفاده از نرم افزار Xilinx اختصاص می یابند. روش دیگر، ممکن است که استفاده از پورت پیکربندی باشد (به عنوان مثال تمام پین های پیکربندی گرفته شده با هم) هنگامی که درحالت کاربر عمل می کند، ادامه یابد. این مورد با تنظیم کردن گزینه اصرار بر روی Yes کامل می شود. چند FPGA می توانند با استفاده از حالت Slave موازی پیکربندی شوند و نیز می توانند به به طور همزمان راه اندازی شوند. شکل ۳-۲۲ اتصالات دستگاه را نشان می دهد. برای پیکربندی چندین تراشه متعدد در این راه

---

۱ - Byte-wide data

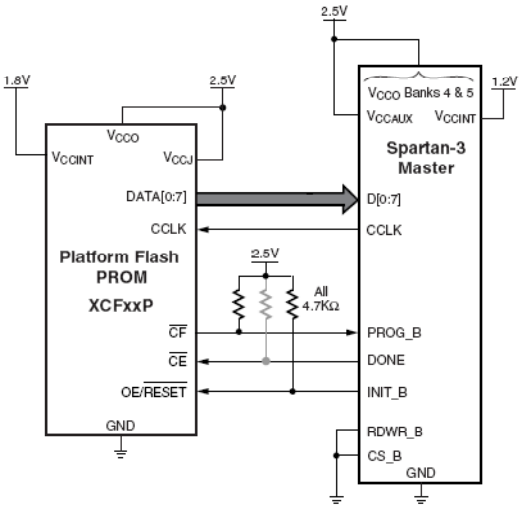
همه پینهای CCLK، داده ها، RDWR_B و BUSY در همه دستگاهها در حالت موازی استفاده می شود. دستگاه های شخصی به طور جداگانه توسط پین CS_B از هر دستگاه به نوبه خود بارگذاری می شود و داده های مناسب را می نویسند.



شکل ۳-۲۲: اتصالات حالت Slave موازی

۳-۳۱-۲-۴- حالت Master موازی

در این حالت، FPGA از اطلاعات گسترده بایت پیکربندی می شود و FPGA کلاک پیکربندی CCLK را فراهم می کند. در حالت پیکربندی Master ، CCLK به عنوان پین دو طرفه I/O رفتار می کند. زمانبندی شبیه به حالت موازی Slave است با این تفاوت که CCLK توسط FPGA فراهم می شود. اتصالات تراشه در شکل ۳-۲۳ نشان داده شده است.



شکل ۳-۲۳- نمودار اتصالات برای حالت Master موازی

۳-۳۱-۲-۵- حالت اسکن مرزی[۱] (JTAG)

در حالت اسکن مرزی پین های اختصاص داده شده برای پیکربندی FPGA استفاده می شوند. پیکربندی به طور کامل از طریق IEEE ۱۱۴۹.۱ (TAP[۲]) انجام می شوند. پیکربندی FPGA با استفاده از حالت اسکن مرزی سازگار با استاندارد IEEE ۱۱۴۹.۱-۱۹۹۳ و استاندارد IEEE ۱۵۳۲ برای تراشه های قابل برنامه ریزی مجدد درون سیستمی انجام می شود. تنظیمات از طریق پورت اسکن مرزی صرف نظر از حالت پیکربندی انتخاب شده، همیشه در دسترس است. با این حال، در برخی موارد حالت تنظیم پین ممکن است بر برنامه ریزی مناسب دستگاه به علت فعل و انفعالات مختلف تأثیر بگذارد. برای مثال، اگر پین های حالت در Master سریال یا حالت Master موازی و PROM مرتبط قبلاً برنامه ریزی شده باشد، بنابراین پس از آن رابط پیکربندی بالقوه ای برای برنامه ریزی از طریق JTAG و داده های PROM وجود دارد. حالت اسکن مرزی انتخاب شده حالت های دیگر را غیر فعال می کند و قابل اعتماد ترین حالت در زمان برنامه نویسی از طریق JTAG می باشد.

۱ - Boundary-Scan
۲ - Test Access Port

فرآیند پیکربندی همچنین می تواند با پین PROG_B آغاز شود. پایان پاکسازی حافظه توسط سیگنال INIT_B که High می شود، انجام می شود. در این مرحله، اطلاعات پیکربندی بر روی FPGA نوشته می شود. FPGA سیگنالهای تنظیم/بازیابی عمومی (GSR[۱]) را در پایان پیکربندی راه اندازی می کند و همه فلیپ فلاپ ها مجدداً راه اندازی می شود (Reset). اتمام تمامی مراحل بالا توسط پین DONE انجام می شود. به طور پیش فرض راه اندازی با یک چرخه CCLK بعد از اینکه DONE در حالت High می رود انجام می شود و سیگنال سه حالته عمومی (GTS[۲]) منتشر می شود. این امر اجازه می دهد که خروجی های دستگاه، فعال شوند. چرخه بعدی CCLK، سیگنال نوشتن عمومی (GWE[۳]) که منتشر می شود را فعال می کند. این اجازه می دهد که عناصر ذخیره کننده داخلی، آغاز تغییر حالت را در پاسخ به طراحی منطقی و کلاک کاربر امکان پذیر سازند.

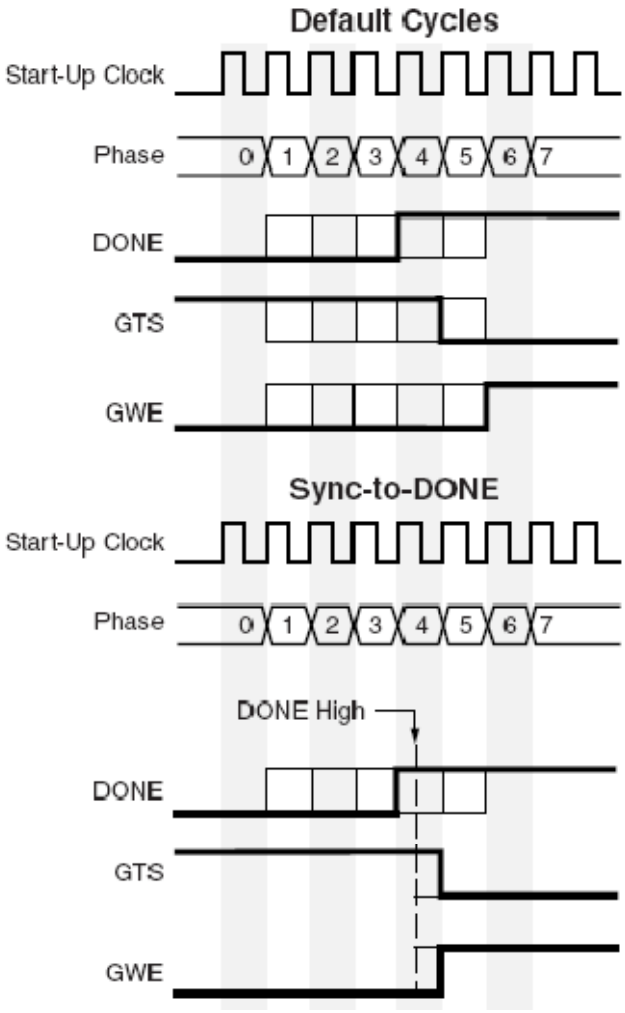
زمان نسبی رویدادهای پیکربندی می تواند از طریق گزینه های موجود در BitGen و نرم افزار Xilinx، تغییر کند. علاوه براین، رویدادهای GWE و GTS می تواند به صورت وابسته بر روی پین های DONE از چندین دستگاه ایجاد شود و دستگاه ها را مجبور به شروع همزمان می کند. همچنین دنباله می تواند در هر مرحله متوقف شود تا زمانی که قفل در هر DCM بدست بیاید.

Readback -۱-۳-۳۱-۳

با استفاده از حالت Slave موازی، اطلاعات پیکربندی از FPGA قابل خواندن می شود. Readback فقط در حالت های Slave موازی و اسکن مرزی پشتیبانی می شود. همراه با پیکربندی داده ها، ممکن است خواندن دوباره تمامی محتواها از همه ثبات ها، حافظه توزیع شده، و منابع حافظه سیستم مسدود شده نیز رخ دهد. این قابلیت برای اشکال زدایی بهنگام استفاده می شود.

---

۱ - Global Set/Reset
۲ - Global Three-State signal
۳ - Global Write Enable



شکل ۳-۲۴: توالی راه اندازی پیش فرض

۳-۳۲-تغذیه Spartan-۳ FPGAs

تولیدکنندگان گوناگون منابع تغذیه راه حل های کامل تغذیه را برای Xilinx FPGAs ارائه کرده اند. از جمله برخی از آنها با رگولاتورهای مجتمع چند ریلی[۱]و به طور خاص طراحی شده برای FPGAs های اسپارتان۳ ارائه کرده اند.

۳-۳۲-۱- طراحی سیستم توزیع توان (PDS[۲]) و خازن های Bypass / Decoupling

۱ - integrated multi-rail regulators
۲ - Power Distribution System

طراحی سیستم توزیع قدرت خوب (PDS) برای همه طرح های FPGA ، به ویژه برای برنامه های کاربردی با عملکرد بالا[1] مهم است. نتایج طراحی مناسب باعث عملکرد کلی بهتر، کاهش جیتر کلاک و DCM و به طور کلی تقویت سیستم می شود. قبل از طراحی برد مدار چاپی (PCB) برای طراحی FPGA، طراحی سیستم های توزیع قدرت (PDS) با استفاده از خازن های Bypass/Decoupling بررسی می شود (اطلاعات بیشتر در برگه اطلاعات تراشه XAPP۶۲۳).

۳-۳۲-۲- رفتار هنگام روشن شدن[2]

FPGAs های اسپارتان۳ دارای مدار ریست هنگام راه اندازی هستند، (POR) که بر سه تغذیه مورد نیاز برای موفقیت پیکربندی FPGA نظارت می کند. در هنگام روشن شدن، مدار POR، FPGA را در حالت ریست نگه می دارد تا VCCO VCCAUXو VCCINT مربوط به BANK٤ به سطوح آستانه ورودی مربوطه برسد. پس از رسیدن به هر سه آستانه مربوطه، ریست POR آزاد شده و فرآیند پیکربندی FPGA آغاز می شود. از آنجا که سه ورودی تغذیه بایستی برای انتشار ریست POR معتبر باشد، می تواند در هر تغذیه عرضه شود و تعیین توالی ولتاژ خاص وجود ندارد. با این حال، استفاده از VCCAUX FPGA قبل از عرضه VCCINT کمترین جریان ICCINT را بکار می برد. هنگامی که هر سه منابع معتبر هستند، حداقل توان مورد نیاز در FPGA برابر است با بدترین حالت فعلی خاموشی، که در جدول ۳۳ برگه اطلاعات تراشه مشخص شده است[۷]. FPGAsهای اسپارتان۳ به Power-On Surge (POS) نیاز ندارد و در نتیجه پیکربندی با موفقیت انجام می شود.

ICCINT مازاد اگر VCCINT قبل از VCCAUX اعمال شود: اگر منبع VCCINT قبل از منبع VCCAUX اعمال شود، FPGA ممکن است ICCINT مازاد را علاوه بر ICCINT که در جدول ۳۳ برگه اطلاعات تراشه مشخص شده است[۷] ایجاد کند. با این حال، مازاد فعلی، زمانی که منبع VCCAUX اعمال می شود، بلافاصله ناپدید می شود و در پاسخ، FPGA's ICCINT تغذیه فعلی به

---

۱ - high-performance
۲ - Power-On Behavior

سطوحی که در جدول ۳۳ برگه اطلاعات تراشه مشخص شده است، می رود. FPGA از مازاد جاری برای موفقیت در توان و پیکربندی نه استفاده کرده و نه به آن نیاز دارد. اگر از VCCINT قبل از VCCAUX استفاده شود، این اطمینان حاصل می شود که تنظیم کننده از بسیاری ویژگی ها استفاده نمی کند.

۳-۳۲-۳- حداقل مجاز میزان شیب VCCO در دستگاه های اولیه

تجهیزات قابل حمل اساساً از سال ۲۰۰۶ هیچگونه محدودیت سرعتی در شیب VCCO ندارند. بررسی اسپارتان۳، یک محدودیت در سرعت تغذیه VCCO که می تواند شیب باشد را نشان داد. حداقل میزان شیب مجاز VCCO به عنوان TCCO در جدول ۳-۱۱ نشان داده شده است. در نتیجه، آرایه شبکه و مقیاس تراشه در بسته بندیهای (CP۱۳۲ ، FT۲۵۶، FG٤٥٦، FG٦٧٦ ، و FG۹۰۰) اجازه می دهد شیب با سرعت بیشتری از بسته های مسطح و چهار گوش صورت بگیرند (VQ۱۰۰ ، TQ۱٤٤ ، و PQ۲۰۸).

۳-۳۳- پیکربندی نگهداری داده ها

داده های پیکربندی FPGA در بافرهای[۱] CMOS قوی ذخیره می شود. داده ها در این بافرها زمانی که ولتاژ به حداقل سطح لازم برای حفظ محتوای RAM برسند، ذخیره می شود. این موضوع در جدول ۳-۱۲، مشخص شده است. اگر بعد از پیکربندی، تغذیه VCCAUX یا VCCINT پایین تر از ولتاژ داده های آن باشد، پیکربندی دستگاه با استفاده از یکی از روشهای زیر پاک می شود:

- ولتاژ تغذیه VCCAUX یا VCCINT را پایین تر از سطح ولتاژ (POR) آورید.
- PROG_B را در حالت Low قرار داد.

مدار POR تغذیه VCCO_٤ را پس از پیکربندی نظارت نمی کند. در نتیجه، ولتاژ VCCO_٤ دستگاه را توسط (POR) راه اندازی نمی کند.

۱ - latch

جدول ۳-۱۱- نیازمندیهای شیب زمان ولتاژ تغذیه

Symbol	Description	Device	Package	Min	Max	Units
T_{CCO}	V_{CCO} ramp time for all eight banks	All	All	No limit(4)	-	
T_{CCINT}	V_{CCINT} ramp time, only if V_{CCINT} is last in three-rail power-on sequence	All	All	No limit	No limit(5)	

جدول ۳-۱۲- سطوح ولتاژ مورد نیاز برای حفظ کردن محتوای RAM

Symbol	Description	Min	Units
V_{DRINT}	V_{CCINT} level required to retain RAM data	1.0	V
V_{DRAUX}	V_{CCAUX} level required to retain RAM data	2.0	V

۳-۳۴- پمپ های بدون شارژ داخلی یا نوسان سازهای آزاد

برخی از برنامه های کاربردی سیستم به منابع سر و صدای آنالوگ حساس هستند. مدارات اسپارتان ۳ به طور کامل استاتیک می باشند و از پمپ های شارژ داخلی استفاده نمی کنند. کلاک پیکربندی CCLK در طول روند پیکربندی FPGA فعال می باشد. پس از اینکه پیکربندی کامل شد، نوسان ساز CCLK به طور خودکار غیر فعال می شود مگر اینکه گزینه Bitstream Generator، Persist=Yes باشد. اسپارتان۳ به صورت اختیاری از ویژگی ای بنام کنترل امپدانس (DCI) پشتیبانی می کند. هنگامی که نرم افزار مورد استفاده قرار می گیرد، منطق DCI از نوسان ساز داخلی استفاده می کند. منطق DCI تنها در صورتی فعال می شود که کاربر FPGA نیاز به مشخص کننده I/O استاندارد DCI، (LVDCI_۲۵، LVDCI_۳۳) و غیره داشته باشد. اگر DCI استفاده نشود، نوسان ساز داخلی نیز غیر فعال می شود. به طور خلاصه، نرم افزار از گزینه Persist=Yes استفاده می کند و یا استاندارد DCI I/O را مشخص می کند، و FPGA بدون هیچ گونه سوئیچ خارجی به طور کامل استاتیک باقی می ماند.

فصل چهارم

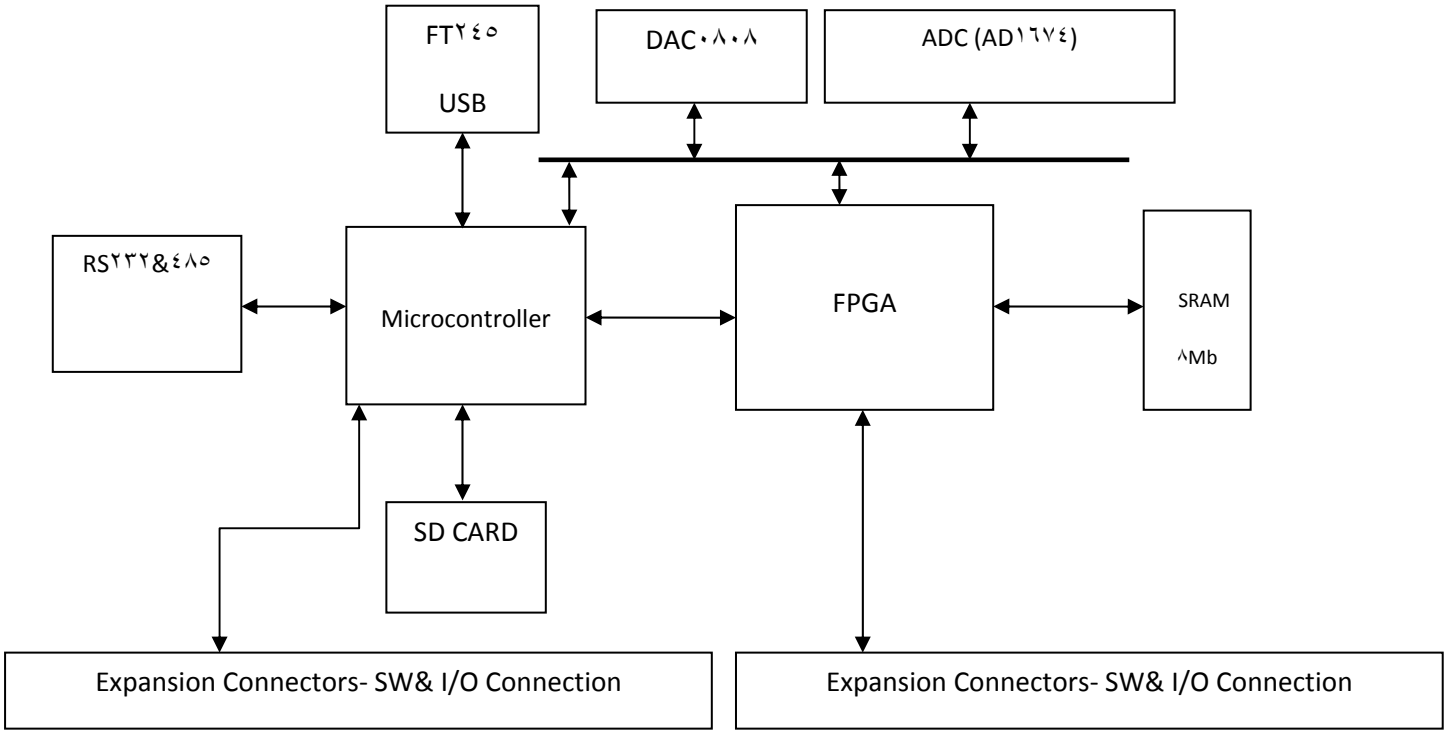
معرفی و راه اندازی سخت افزار SUT۲۰۰

۴-۱- مقدمه

در این فصل به معرفی و چگونگی راه اندازی سخت افزار خواهیم پرداخت. توضیحات تکمیلی راجع به برد طراحی شده، بلوک دیاگرام، معرفی بلوکهای مختلف و ارتباط آنها با یکدیگر، مراحل بکارگیری آن، قابلیت ها و مشکلات مواجهه شده با آن، از جمله مطالبی است که به آنها خواهیم پرداخت. در ادامه به بررسی تست سخت افزار و برنامه های نمونه آزمایشی پیاده شده به زبان VHDL و C بر روی برد پرداخته می شود. بررسی ارتباط بین کامپیوتر و سخت افزار از جمله مواردی است که در این فصل به آن اشاره می شود. در ادامه نیز به بررسی روند طراحی سیستم های میکروپروسسوری، مشکلات و چالش های روبرو خواهیم پرداخت. بررسی اجمالی نحوه کار با نرم افزار های مربوط به تراشه های اصلی سخت افزار نیز از جمله مباحث مطرح شده این فصل می باشد.

۴-۲- معرفی

هدف از انجام این پروژه طراحی و ساخت یک برد الکترونیکی پردازش سیگنالهای دیجیتال برای کاربردهای متنوع است. با توجه به اینکه کاربردهای پردازش سیگنال با تراشه های FPGA و DSP PROCESSOR روند رو به رشدی را داراست، استفاده و بهره برداری از این نوع تراشه ها در پروژه ها حائز اهمیت می باشد. در بسیاری از کاربردهای صنعتی و تجاری استفاده از چنین پردازنده هایی در کنار یکدیگر به منظور افزایش کارآیی و بازدهی سیستم حائز اهمیت می باشد. دستگاه های پیشرفته تجاری و صنعتی نظیر انواع پلیرهای پیشرفته، میکسرهای صدا و تصویر دیجیتال، کاربردهای نظامی- صنعتی و ... گویای مطلب فوق می باشد. طراحی این پروژه شامل دو بخش اساسی می باشد. بخش اول این فصل مربوط به طراحی سیستم سخت افزار و ارتباطات آن است و بخش دوم فصل ارتباط با کامپیوتر و نمایش اطلاعات مربوطه است. در شکل ۴-۱ نمای کلی طرح قابل مشاهده می باشد:



شکل ۴-۱- نمای کلی سخت افزار SUT۲۰۰

✓ شماتیک و مدار چاپی برد در بخش پیوست ضمیمه الف آورده شده است.

همانطور که در شکل ۴-۱ مشاهده می شود، پردازنده اصلی طرح از خانواده اسپارتان۳ شرکت Xilinx انتخاب شده است. این پردازنده به دلیل ویژگیها و قابلیت هایی که دارد در اکثر پروژه های صنعتی، برودکست، تجاری و بازرگانی مورد استفاده قرار می گیرد. خانواده اسپارتان ۳ از FPGA های بطور خاص برای کاربردهای با حجم بالا و مصارف حساس وگران قیمت الکترونیکی طراحی شده اند. هشت عضو خانواده با تراکم گیتهای متفاوت از ۵۰۰۰۰ الی پنج میلیون گیت ارائه می شوند. مشخصات و ویژگیهای این خانواده را در جدول ۴-۱ مشاهده می نمایید.

جدول ۴-۱: تراشه های خانواده اسپارتان۳

Device	System Gates	Equivalent Logic Cells(1)	CLB Array (One CLB = Four Slices)			Distributed RAM Bits (K=1024)	Block RAM Bits (K=1024)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs						
XC3S50(2)	50K	1,728	16	12	192	12K	72K	4	2	124	56
XC3S200(2)	200K	4,320	24	20	480	30K	216K	12	4	173	76
XC3S400(2)	400K	8,064	32	28	896	56K	288K	16	4	264	116
XC3S1000(2)	1M	17,280	48	40	1,920	120K	432K	24	4	391	175
XC3S1500	1.5M	29,952	64	52	3,328	208K	576K	32	4	487	221
XC3S2000	2M	46,080	80	64	5,120	320K	720K	40	4	565	270
XC3S4000	4M	62,208	96	72	6,912	432K	1,728K	96	4	633	300
XC3S5000	5M	74,880	104	80	8,320	520K	1,872K	104	4	633	300

تراشه های خانواده اسپارتان ۳ بر اساس برتریهای خانواده SpartanII-E و با افزایش مقدار منابع منطقی، ظرفیت حافظه رم داخلی، تعداد کل I/O ها و بهبود کارآیی کل بر اساس توابع مدیریت کلاک تراشه ساخته شده است. پیشرفتهای متعددی از فن آوری بستر Virtex® II مشتق می شود. این پیشرفتهای خانواده اسپارتان ۳ با فن آوری پردازش پیشرفته ای تر کیب می شود و منتج به کارآیی و پهنای باند بیشتری نسبت به سابق، به ازای هر دلار می شود. بدلیل هزینه فوق العاده کم آنها، خانواده اسپارتان۳ برای طیف وسیعی از لوازم الکترونیکی مصرفی و برنامه های کاربردی مناسب و ایده آل هستند. از جمله دسترسی به پهنای باند ، شبکه های خانگی، صفحه نمایش، پروژکتور و تجهیزات تلویزیون های دیجیتال است. خانواده اسپارتان۳ جایگزین برتری برای ASICs های قابل برنامه ریزی هستند. FPGA ها هزینه های بالای اولیه، چرخه توسعه طولانی و عدم انعطاف ذاتی متداول ASICs را ندارند. همچنین برنامه ریزیFPGA اجازه ارتقاء طراحی در زمینه جایگزینی با سخت افزار لازم را ندارد.

۴-۲-۱- ویژگیهای اصلی این تراشه FPGA

- ❖ کم هزینه، راه حل منطقی کارایی بالا برای با حجم بالا وبرای کاربردهای مصرف گرا
  - چگالی تا ۷۴۸۸۰ سلول منطقی
- ❖ رابط سیگنالینگ I/O
  - تعداد I/O تا ۶۵۵ پین
  - انتقال داده با نرخ ۶۲۲Mbps per IO
  - ۱۸ استاندارد سیگنال یک ترمینال(single-ended)

- ۸ استاندارد دیفرانسیلی I/O شامل LVDS, RSDS
- ختم شده توسط امپدانس دیجیتالی کنترل شده
- نوسان سیگنال در محدوده ۱.۱۴ الی ۳.۴۶۵ ولت
- پشتیبانی Double Data Rate (DDR)
- پشتیبانی DDR,DDR۲,SDRAM تا نرخ ۳۳۳Mbps

❖ منابع منطقی

- سلول های فراوان منطقی با قابلیت شیفت رجیستر
- گسترده و دارای مالتی پلکسرهای سریع
- دارای مالتی پلکسرهای اختصاصی ۱۸*۱۸
- دارای واسط JTAG مطابق با استاندارد IEEE۱۱٤۹.۱/۱۵۳۲

❖ بلوک رم تا ۱۸۷۲ کیلو بیت

❖ مدیر کلاک دیجیتال(Digital Clock Manager)

- حذف انحراف سیگنال کلاک
- سنتز فرکانس
- شیفت فاز

❖ دارای ۸ خط کلاک عمومی و مسیردهی مستقل

❖ پشتیبانی کامل توسط Xilinx ISE

۴-۲-۲- معرفی بلوکهای سخت افزار

ارتباطات بین تراشه ها و وسایل جانبی موجود در برد مطابق بلوک دیاگرام شکل ۴-۱ می باشد. تراشه های FT۲٤٥ و SD CARD مستقیماً به میکروکنترلر متصل هستند و مبدل ADC قابلیت

ارتباط به هر دو تراشه را دارد. مبدل DAC نیز مستقیماً به FPGA و میکروکنترلر ارتباط داده شده است. استاتیک رم نیز به FPGA متصل است. میکروکنترلر با یک پورت ۸ بیتی با FPGA متصل شده است. برای ارسال و دریافت داده ها بین میکروکنترلر و FPGA روشهای گوناگونی وجود دارد. می توان همزمان در این ۸ بیت ارسال و دریافت نمود که این مورد نیاز به سیگنالهای Handshaking می باشد و می توان از خود این بیت ها نیز بعنوان Handshaking استفاده نمود و یا اینکه از ۴ بیت برای ارسال و ۴ بیت برای دریافت بصورت مجزا استفاده نمود. این مورد بدلیل راحتی در پیاده سازی آن، بیشتر از مورد اول در طرح ها و پروژه ها استفاده شده است. روش پرکاربرد سومی نیز وجود دارد که بیشترین استفاده از آن می شود. روش کار بدین صورت است که انتقال داده ها بصورت سریال و از طریق یک پین ارتباطی بین دو تراشه صورت می پذیرد. یکی از تراشه ها بعنوان Master و دیگری بعنوان Slave عمل می کند. تراشه Master کلاک را تولید کرده و به Slave ارائه می دهد. Master داده را در خروجی قرار داده و سپس کلاک می زند. تراشه Slave نیز به محض دیدن لبه کلاک (بالا یا پایین رونده) داده را از روی پورت برداشت نموده و در حافظه ای ذخیره می کند. بعنوان مثال برای دریافت داده ها می توان یک شیفت رجیستر در FPGA پیاده سازی نمود. هر بیت داده جدید که با کلاک Master دریافت شد، در ورودی این شیفت رجیستر دریافت شده و ذخیره می شود. در این طرح، موارد اول و سوم در سخت افزار تست شده است.

برای ارتباط ADC به میکروکنترلر و FPGA، ۱۲ بیت خروجی، بعد از گذر از مقاومتهای کاهش دهنده سطح ولتاژ به گذرگاهی (انتخاب جامپر) می رسند که از آنجا با انتخاب پین های مربوطه، می توان خروجی را به میکروکنترلر یا FPGA متصل نمود. مقاومتها به این دلیل در مدار قرار گرفته اند تا سطح ولتاژ خروجی ADC را که ۵ ولت می باشد به سطح ولتاژ قابل قبول میکروکنترلر و FPGA که ۳.۳ ولت است برساند. رنج مقاومتها معمولاً حدود ۱K اهم انتخاب می شوند. البته این یک امر تجربی است و باید در مدار و شرایط تست، مقدار دقیق مقاومتها را انتخاب نمود.

مدار DAC با بکارگیری تراشه DAC۰۸۰۸ به همراه یک OP AMP با شماره LM۳۵۸ برای تبدیل سیگنال دیجیتال به سیگنال آنالوگ استفاده شده است. در DAC۰۸۰۸ ورودی های دیجیتال به جریان Iout تبدیل می شوند و با اتصال یک مقاومت به پایه Iout، جریان به ولتاژ تبدیل می شود. جریان Iout تابعی از اعداد دودویی ورودی D۰ تا D۷ روی پایه های BIT۰ تا BIT۷ در DAC۰۸۰۸ و جریان مرجع K است.

$$Iout = k(\frac{D۷}{۲} + \frac{D۶}{۴} + \frac{D۵}{۸} + \frac{D۴}{۱۶} + \frac{D۳}{۳۲} + \frac{D۲}{۶۴} + \frac{D۱}{۱۲۸} + \frac{D۰}{۲۵۶}) \qquad (۱-۴)$$

که در این رابطه K از برابری $K = \frac{V_{ref}}{R۳۴}$ به دست می آید. در اینجا Vref=۱۰v و R۳٤=۵K است. بنابراین در این مدار K=۲ خواهد شد. مقدار دیجیتال در تراشه LATCH به شماره ۷٤LS۵۷۳ نوشته می شود و خروجی های این تراشه به ورودی های مبدل DAC متصل است. اگر همه ورودی های متصل به DAC در سطح بالا باشند بیشینه جریان خروجی ۱.۹۹ میلی آمپر است. پایه خروجی Iout را برای تبدیل این جریان به ولتاژ به یک مقاومت متصل کرده و لازم بذکر است که این مقاومت قدری بی دقتی را بوجود می آورد زیرا ورود مقاومت بار که به آن وصل است نیز روی ولتاژ خروجی تأثیر خواهد گذاشت. به این دلیل جریان Iout را با اتصال به یک تقویت کننده عملیاتی با شماره LM۳۵۸ و مقاومت ۵K در مسیر پسخورد ایزوله می کنیم تا اثری روی ولتاژ خروجی نگذاشته باشیم.

برای ارتباط با کامپیوتر دو ارتباط USB و سریال در سخت افزار تعبیه شده است. بدلیل مزیت های این پورت در مقایسه با سایر مدارات واسط دیگر (موازی، سریال)، در واقع می توان گفت که پورتهای موازی و سریال، گویی کم کم دارند باز نشسته می شوند! در اکثر Mainboard های امروزی این پورتها کنار گذاشته شده و چندین پورت USB جایگزین آنها شده است. سرعت بالا، ارتباط آسان سخت افزاری برای کاربر، تشخیص دستگاه به صورت خودکار و اجرای توابع راه انداز و در انتها قابلیت اتصال دستگاه های زیاد به کامپیوتر (تا سقف ۱۲۷ دستگاه) به صورت همزمان همه از مزیت های USB است[۱۵]. برای ارتباط از طریق پورت USB نیاز به یکی از تراشه های کنتر لر USB است که نقش

ارتباط و ساماندهی بین میزبان (HOST) و سخت افزار مورد نظر ما که نقش دستگاه جانبی را برای آن دارد، ایفا می نماید. شرکت های زیادی اقدام به تولید این تراشه ها نموده اند. توجه به اینکه تنها دو نوع تراشه های USB در بازار ایران موجود می باشد، قدرت انتخاب بالتبع کمتر خواهد شد. این دو نوع عبارتند از :FT۲۳۲ (سریال)،FT۲۴۵ (پارالل) ساخت شرکت FTDIchip (Future Technology Device Interface) و همچنین تراشه ی PDIUSBD۱۲ ساخت شرکت فیلیپس. برای مقایسه ایندو تراشه برخی از ویژگیهای آنها در جدول ۴-۲ آمده است :

جدول ۴-۲ (مقایسه دو تراشه USB)

شماره تراشه	سرعت	سازگار با usb۱.۱	درایور آماده	ظرفیت حافظه
FT۲۴۵	۱Mbps	✓	✓	۳۸۴ بایت
PDIUSB	۲Mbps	✓	✗	۳۲ بایت

در این طرح از تراشه ی FT۲۴۵ استفاده شده است. همانطور که می دانیم، هر قطعه باید دارای درایور و فایل inf. باشد تا اینکه آن قطعه توسط host، سرشماری شده و شناسایی گردد. به همین منظور این شرکت این امکان را برای کاربران خود قرار داده تا بتوانند از طریق وب سایت شرکت به فایلها و درایور های مورد نیاز که شامل فایلهای ,(dynamic link library) dll. ,(information) inf. (header) h. ,(library)lib. می باشد به صورت رایگان دسترسی داشته باشند. بدین صورت ما می توانیم درایور دستگاه مورد نظرمان را در سیستم عاملهای مختلف نصب کنیم تا آن سیستم عامل نحوه ی ارتباط با آن دستگاه را بداند. تراشه ی USB بعنوان یک دستگاه جانبی به میزبان شناسانده می شود و از این به بعد قادریم که که از طریق اندپوینت[۱] موجود بر روی آن، با این قطعه ارتباط برقرار نمائیم. دو درایور برای این تراشه در سایت شرکت قرار داده شده است. یکی VCP (Virtual Com Port) و دیگری d۲xx است. VCP همانطور که از اسمش پیداست دستگاه را بعنوان یک کام پورت مجازی به ویندوز می شناساند و حداکثر تا سرعت ۳۰۰kbps انتقال دیتا را انجام می دهد. ولی درایور بعدی D۲XX (USB

۱- end point بخش مهمی از هر دستگاه جانبی که مبدا و مقصد هر ترنزکشن به آن ختم می شود. توضیحات بیشتر در [۱۵]

(direct driver می تواند تا سرعت ۱Mbps انتقال دیتا را نیز انجام دهد. برای آشنایی بیشتر با این تراشه، به معرفی و بررسی مشخصات آن پرداخته می‌شود :

این تراشه دومین تولید از سری تراشه های USB Controller این شرکت، با بافر FIFO است. در این تراشه سیگنال های Handshaking برای ارتباط با آن کمتر شده و طراحی مدارات با آن ساده تر می باشد.

برخی از ویژگیهای تراشه عبارتند از :

- سرعت انتقال دیتا با سرعت ۱Mbps با درایور D۲XX
- سرعت انتقال دیتا با سرعت ۳۰۰Kbps با درایور VCP
- دارای بافر FIFO دو جهته
- عدم نیاز به درایور برای راه اندازی دستگاه
- عدم نیاز برنامه نویسی برای پیاده سازی پروتکل USB
- اتصال راحت به MCU،PLD، FPGA و ارتباط با آنها از طریق سیگنالهای Handshaking
- ۳۸۴ بایت بافر برای ارسال دیتا (tx)
- ۱۲۸ بایت بافر برای دریافت دیتا (rx)
- سازگار با نسخه USB۱.۱ و USB.۲
- قابلیت ارائه ی PID و VID،توصیف گر محصول و شماره ی سریال دستگاه در حافظه ی خارجی
- تغذیه ۴.۳۵ تا ۵.۲۵
- پشتیبانی برای USB Suspend/Resume از طریق پینهای #PWREN و WAKEUP

کار برد های این تراشه نیز در موارد زیر بعنوان نمونه آمده است :

- ادوات USB
- USB Industrial Control
- انتقال سیگنالهای صدا و تصویر (با پهنای باند کم)

- MP۳ Player Interface
- فلش کارت Reader & Writer
- دوربین دیجیتال
- سخت افزار مودم و مودم های بی سیم (Wireless)
- قابلیت ارتقاء بخشیدن ادوات جانبی به USB Interface

تراشه ی USB را می توان در دو نوع Bus Powered و Self Powered راه اندازی کرد. Bus Powered، مواقعی استفاده می گردد که دستگاه ما تغذیه نداشته باشد (مانند flash memory ها) و تغذیه ی آن از طریق پورت تأمین می شود. در این حالت ما محدودیت جریان داریم و نمی توان جریان زیادی را از پورت کشید (بیشینه ۱۰۰ میلی آمپر). Self Powered نیز مواقعی استفاده می گردد که تغذیه USB از طریق مدار خود سخت افزار تأمین گردد و نه از طریق پورت. که در این حالت محدودیت جریان را نیز نخواهیم داشت. در این پروژه USB از نوع دوم یعنی Self Powered راه اندازی شده است.

۴-۲-۳- سیگنالهای Handshake تراشه و نحوه ی ارتباط با آن

۱- نوشتن در تراشه (ارسال از MCU به تراشه)

این حالت با وضعیت پایه ی TXE مشخص می شود. بصورتی که LOW بودن این پایه نشانگر آماده بودن تراشه برای دریافت اطلاعات است. در این وضعیت برای ارتباط با تراشه باید پس از اطمینان از LOW بودن این پایه، باید با قرار دادن بایت اطلاعات بر روی گذرگاه I/O تراشه، اطلاعات را با سیگنال WR (لبه ی پایین رونده) بر روی تراشه نوشت. پس از نوشتن یک بایت در بافر تراشه، پایه ی TXE برای مدت کوتاهی High شده و در صورتی که بافر هنوز ظرفیت داشته باشد مجدداً این پایه LOW خواهد شد. در صورتی که بافر پر باشد، این پایه به نشانه ی پر بودن، High باقی خواهد ماند. در اینصورت امکان نوشتن در بافر آن وجود ندارد.

۲- خواندن از تراشه (ارسال از تراشه به MCU)

این وضعیت با سیگنال RXF تعیین می شود. در زمانی که این پایه در وضعیت LOW قرار می گیرد نشانه ی این می باشد که حداقل یک بایت درون بافر وجود دارد، که توسط سیگنال RD (لبه ی بالا رونده) خوانده می شود. و اگر این پایه High باشد به دستگاه جانبی اعلام می دارد که درون بافر RX، داده ای وجود ندارد. برای کسب اطلاعات بیشتر در مورد ارتباط USB به مرجع [۱۵] رجوع شود.

برای ارتباط سریال نیز دو تراشه MAX۲۳۲ و ADM٤٨٥ در مدار در نظر گرفته شده است. این تراشه ها مستقیماً به پورت USART میکروکنترلر متصل شده اند. برای ارتباط با وسایل جانبی می توان با تعیین تنظیمات این ارتباطات در میکروکنترلر به برقراری ارتباط پرداخت. در این طرح بدلیل اینکه نیاز به درایور و فایل های کتابخانه ای برای راه اندازی ارتباط سریال نمی باشد و همچنین نرم افزار مورد نظر ما در این طرح MATLAB است، از این ارتباط برای اتصال به کامپیوتر استفاده شده است. بدین منظور می توان مستقیم خروجی تراشه MAX۲۳۲ را به پورتهای COM کامپیوتر متصل نمود و یا اینکه از یک مبدل USB به سریال استفاده کرد. در این طرح از مورد دوم یعنی از یک مبدل USB به سریال استفاده شده است. سیستم عامل کامپیوتر این مبدل را بعنوان یک پورت COM شناسایی کرده و برقراری ارتباط با آن همانند ارتباط سریال است. همچنین نرم افزار MATLAB امکان برقراری با پورتهای سریال را داراست و این کار با باز کردن پورت و برقراری ارتباط با آن صورت می پذیرد.

۴-۲-۴- میکروکنترلر

تراشه میکروکنترلر، از خانواده Atmega AVR شرکت Atmel می باشد. AVRها، میکروکنترلرهای ۸ بیتی از نوع CMOS با توان مصرفی پایین هستند که معماری آنها بر اساس ساختار RISC می باشد. اولین نسخه ی AVR، در سال ۱۹۹۶روانه ی بازار شد و در حال حاضر به یکی از پر مصرف ترین میکروکنترلرها تبدیل شده است. هر میکروکنترلر کاربردها و ویژگیهای خاص خود را دارد و نمی توان گفت که بهترین نوع میکروکنترلرها است. ولی به دلیل مزیت هایی که دارند می توان گفت، بر بقیه ی میکروکنترلرها ارجحیت دارند. بدلیل معماری که برای آنها طراحی شده است دستورات را در یک کلاک

انجام می دهند. به عنوان مثال به ازای کلاک ۸MHz، ۸MIPS قدرت انجام دستورات میکرو است. که باعث ازدیاد سرعت پردازش می شود. علی رغم اینکه حداکثر فرکانس داخلی در اکثر AVR ها ۱٦MHz است ولی در بعضی از انواع، (مثلاً ATiny۱۳) حداکثر فرکانس تایمر تا ٦٤MHz نیز افزایش یافته است. AVR ها ۳۲ رجیستر همه منظوره GPR[۱]، و دستورات قدرتمندی را شامل می شوند.

میکروکنترلرهای AVR به سه دسته تقسیم می شوند :

۱. Tiny AVR (ATiny)
۲. Classic AVR (AT۹۰s)
۳. (x) Mega AVR (AT(x)mega)

تفاوت این سه نوع به امکانات آنها مربوط می گردد. ATiny ها غالباً تراشه هایی با تعداد پایه ها و مجموعه دستورات کمتری نسبت به (x) mega AVR ها می باشند و حداقل امکانات را دارا هستند. Xmega AVR ها حداکثر امکانات را دارا هستند. Classic AVR ها جایی بین ایندو هستند. باید توجه داشت که اصول همه ی میکروکنترلرهای AVR یکی است و تفاوت آنها در میزان حافظه‌ی موجود بر روی تراشه و امکانات جانبی آنها می باشد.

یکی از ویژگیهای جالب میکروکنترلرهای AVR در مقایسه با سایر میکروکنترلرها فیوز بیت ها هستند. فیوز بیتها همانطور که از اسمشان پیداست، فیوز هایی هستند که در زمان برنامه ریزی تراشه، قابل برنامه ریزی هستند. هریک می تواند یکی از امکانات جانبی میکروکنترلر را فعال یا غیر فعال کند و یا نحوه‌ی استفاده از آنرا مشخص نماید. ضمناً فیوز بیتها با پاک کردن میکرو پاک نمی شوند و حتی می توان آنها را قفل نمود. خصوصیات دیگر میکروکنترلرهای AVR عبارتند از :

- دارای قابلیت پروگرام شدن درون سیستم ISP (In System Programming)
- نوشتن برنامه ی میکرو به زبان C از طریق نرم افزار Codevision[۲]
- فعال سازی آسان امکانات جانبی میکرو (Adc, Dac, Usart, Analog Comparator,...)
- دارای ۱۲۸Kbyte حافظه ی داخلی از نوع فلش (ATmega۱۲۸)

۱- General Purpose Register

۲-در ضمیمه ج با این نرم افزار آشنا خواهیم شد.

- امکان خط به خط اجرا کردن برنامه و همچنین دیباگ کردن آن توسط نرم افزارAVR Studio
- دارا بودن حافظه داخلی از نوع EEPROM و همچنین SRAM

برای افزایش کارآیی سیستم یک تراشه رم از نوع استاتیک به شماره ٤٨LC٢M٣٢B٢ ساخت شرکت Micron در کنار FPGA تعبیه شده است تا اگر به فضای بیشتری برای کار با رم نیاز بود، این امکان وجود داشته باشد. این رم بصورت ٥١٢K x ٣٢ x ٤ banks پیکربندی می شود. بسته به نوع بسته بندی تا ٢٠٠ مگاهرتز می توان فرکانس کلاک رم را بالا برد. برای ارتباط تراشه های اصلی با دنیای بیرون کانکتورهایی تعبیه شده تا این امر نیز امکان پذیر باشد.

برای پیکربندی تراشه FPGA دو مد استفاده از حافظه فلش XC٠F و ارتباط JTAG نیز در نظر گرفته شده است. اطلاعات بیشتر در مورد پیکر بندی تراشه در فصل ٣ قابل مشاهده است.

جهت ذخیره سازی اطلاعات بصورت دائمی و ماندگار از یک کارت حافظه در مدار استفاده شده است. بدلیل استفاده روزافزون از کارتهای SD و پرکاربردی این کارتها، مختصراً در این بخش به بررسی و نحوه برقراری ارتباط با آنها می پردازیم: (توضیحات تکمیلی در بخش پیوست ضمیمه ب قابل مشاهده می باشد).

٤-٣- بررسی SD CARD

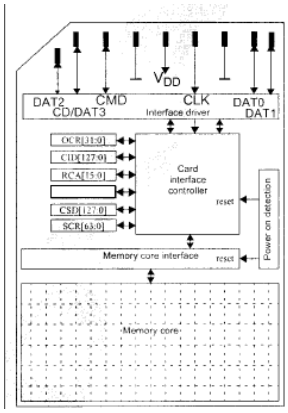
٤-٣-١- مقدمه

کارت های حافظه امروزه به یکی ازمتداول ترین وسایل برای جابه جایی اطاعات بین دستگاههای دیجیتال تبدیل شده اند. جابجائی حجم زیادی از اطلاعات، به همراه قابلیت ارتباط با پرتکل SPI، این کارت ها را به حافظه هائی ایده ال برای کار بامیکروکنترلرهای کوچک تبدیل کرده است.

استاندارد SD CARD توسط SD CARD ASSOCIATON طراحی گردید و توسط شرکت های SAN DISK،TOSHIBA وMEI مورد حمایت قرار گرفت و کامل تر شد، امروزه استاندارد SD CARD

فقط محدود به کارت های حافظه نیست و در کلاس های مختلفی از دستگاهها ی دیجیتال به کار می رود که از جمله آنها می توان به بلوتوث و مودم ها اشاره نمود.

این کارت های حافظه دارای کنترلر داخلی می باشند که کلیه ی اعمال کنترلی (پاک کردن، خواندن، نوشتن و خطا یابی) را در داخل کارت انجام می دهند. به کمک رجیسترها می توان وضعیت فعلی کارت، مشخصه های ذاتی آن واطلاعات مفید دیگری راکسب کرد.

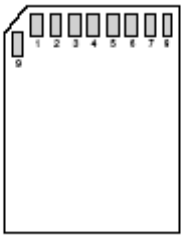


شکل ۴-۲: ساختارداخلی SD CARD

۴-۳-۲-سخت افزار SD CARD

ترتیب پایه ها و نحوه عملکرد آنها

در مجموع ۹ خط ارتباطی برای انتقال توان وداده دارد، سه خط برای انتقال توان و شش خط برای انتقال داده به کار می رود. ترتیب پایه ها وعملکرد آنها مطابق شکل زیر می باشد:



Pin	Name	Function (SD Mode)	Function (SPI Mode)
1	DAT3/CS	Data Line 3	Chip Select/Slave Select ($\overline{SS}$)
2	CMD/DI	Command Line	Master Out Slave In (MOSI)
3	VSS1	Ground	Ground
4	VDD	Supply Voltage	Supply Voltage
5	CLK	Clock	Clock (SCK)
6	VSS2	Ground	Ground
7	DAT0/DO	Data Line 0	Master In Slave Out (MISO)
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

شکل۴-۳: پایه هایSD CARD در مدهای SPI وSD

رنج دقیق تغذیه باید از رجیستر OCR[1] خوانده شود ، معمولاً اکثر کارت ها تغذیه بین ۲.۷ تا ۳.۶ ولت را می پذیرند، جریان کارت های حافظه می تواند تا چندین میلی آمپر باشد. لازم است کنترلر خارجی بتواند تا ۱۰۰ میلی آمپر را تأمین نماید.

جدول۴-۳: ولتاژ وجریان مصرفی کارت حافظه

VDD (ripple: max, 60 mV peak to peak)	2.7 V – 3.6 V

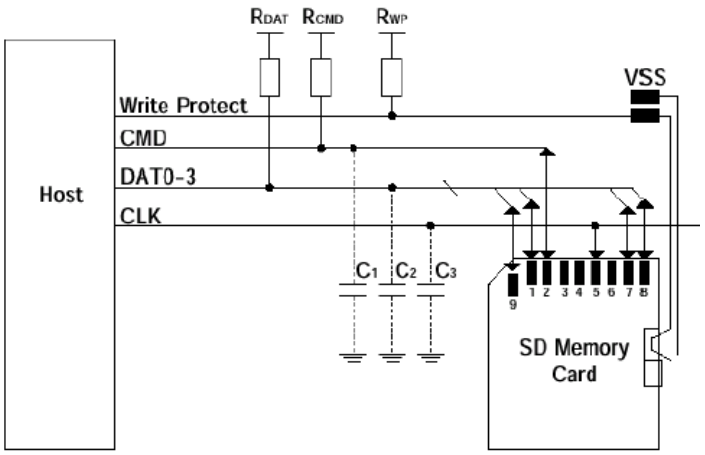
(Ta = 25°C @3 V)

	Value	Measurement	Notes
Sleep	250	uA	Max
Read	50	mA	Typical
Write	75	mA	Typical

۴-۳-۳- نحوه ارتباط SD CARD با میکروکنترلر

پین های دیتا را باید به پین های GIO[2]میکروکنترلروصل نمود، درصورتی که میکرو حاوی واحد سخت افزاری SPI باشد، می توان از آن برای ارتباط با SD CARD استفاده نمود. خطوط داده باید با مقاومت pull up شوند.

۱ - Operating Conditions Register
۲ - General Input Output



شکل ۴-۴: نحوه ارتباط SD CARD با میکروکنترلر

۴-۳-۴- پروتکل های ارتباطی

SD CARD از دو پروتکل ارتباطی اصلی حمایت می کند SD PROTOCOL و SPI PROTOCOL.

که به صورت زیر می باشد:

SD ۱ BIT PROTOCOL:

یک پروتکل سریال سنکرون می باشد که شامل یک خط دیتا برای انتقال دیتای اصلی، یک خط کلاک برای سنکرون سازی و یک خط دستور برای ارسال فریم های دستور می باشد .

SD ۴ BIT PROTOCOL:

شبیه استادارد فوق می باشد، با این تفاوت که انتقال دیتا توسط ۴ خط به صورت موازی انجام می شود، با یک طراحی صحیح به کمک این شیوه می توان سرعت انتقال داده را ۴ برابر نمود.

هر دو پروتکل بالا نیازمند محاسبه CRC[۱] می باشند.

خطوط ارتباطی مد SD:

CMD: خط ارتباطی دوطرفه

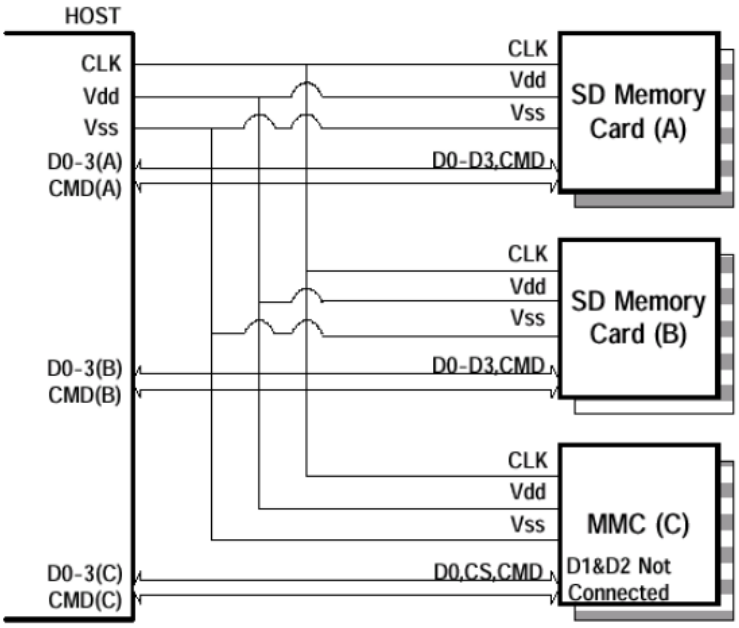
DAT۰-۳: خط ارتباطی دوطرفه

۱ -Cycle Redundancy Code

CLK: خط یک طرفه از کنترلر به کارت

این پروتکل از اشتراک باس دیتا حمایت می کند و می توان به کمک یک Master و یک خط کلاک چندین کارت حافظه را کنترل و انتقال داده را انجام داد .در هنگام Initialize دستورات به هر کارت جداگانه فرستاده می شود، سپس می توان به هر کارت آدرس جداگانه ای اختصاص داد و به وسیله آن با کارت ارتباط برقرار نمود. داده به هر کارت جداگانه فرستاده می شود ،اما می توان برای سادگی یک دستور را برای همه ی کارت ها به صورت همزمان فرستاد. اطلاعات مربوط به آدرس دهی در پاکت دستور قرار می گیرد .

پس از روشن شدن دستگاه SD CARD به صورت پیش فرض تنها از DAT۰ به عنوان خط داده استفاده می کند، بعد از Initialize کنترلر خارجی می تواند پهنای باس داده را تا چهار خط داده افزایش دهد، با استفاده از این روش می توان به سرعت انتقال صد مگا بیت داده در ثانیه بین حافظه و کنترلر خارجی دست پیدا کرد.



شکل ۴-۵: پروتکل باس باقابلیت ارتباط باچندکارت حافظه باسرعت ۱۰۰ mbps

۴-۳-۵- راه اندازی اولیه SD CARD

۴-۳-۵-۱- مراحل اولیه برای ورود به مد SPI

پس از آنکه تغذیه SD CARD وصل شود، کارت وارد مد SD می شود، برای انتقال آن به مد SPI باید مراحل زیر طی شود:

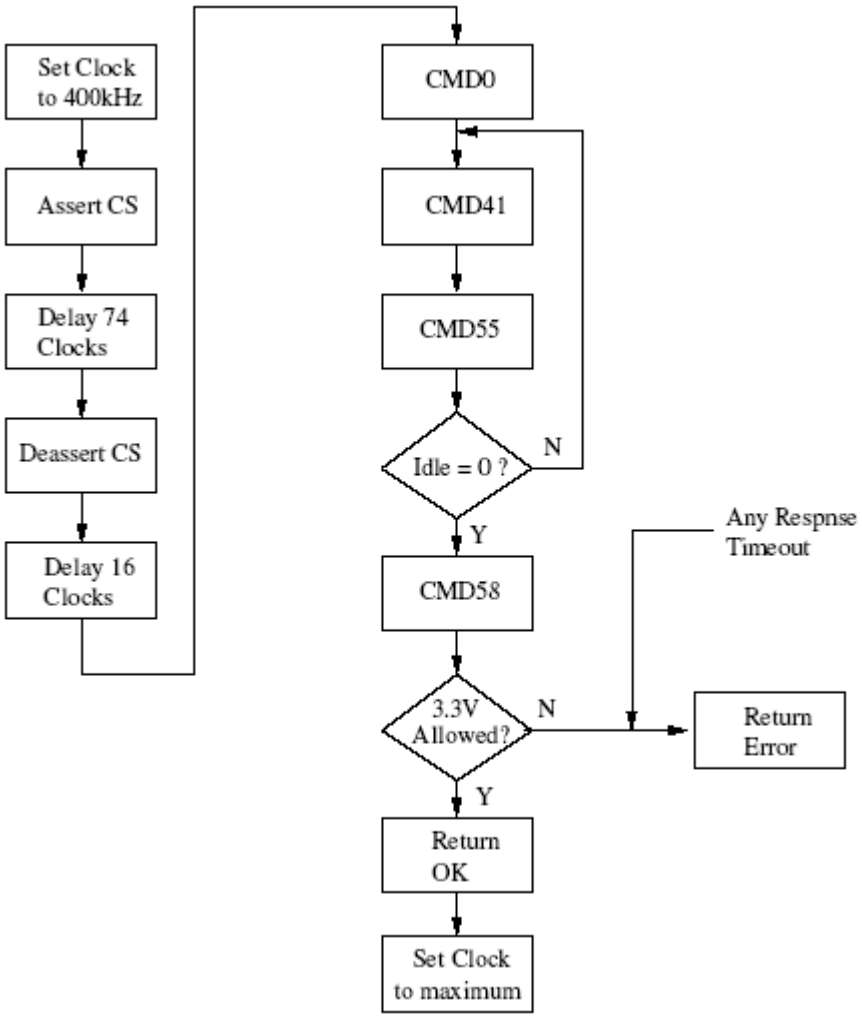
پس از روشن کردن دستگاه باید یک میلی ثانیه صبر شود، خطوط CS و DI را بالا نگه داشته و حداقل ۷۴ پالس به SCLK بزنیم، سپس CMD۰ را با CS به صورت LOW ارسال نماییم. در این حال SD CARD از خط CS نمونه برداری می کند و هنگامی که CMD۰ دریافت می شود اگر CS، LOW باشد، کارت وارد مد SPI می شود. فقط در همین یک بار لازم است تا CRC با مقدار صحیح ارسال شود و پس از ورود به مد SPI، CRC غیر فعال می شود و مقدار آن قابل صرف نظر کردن می باشد.

پس از آنکه CMD۰ پذیرفته می شود کارت وارد مد idle می شود و پاسخ R۱ را با مقدار ۱ ارسال می کند، CRC را می توان با CMD۵۹ دوباره فعال نمود.

هنگامی که کارت در مد idle می باشد، تنها CMD۰، CMD۱وCMD۵۸ پذیرفته می شود و مابقی دستورات توسط کارت Reject می گردد. هنگامی که CMD۱ ارسال می شود، کارت شروع به Initialize کردن خود می کند، بسته به ظرفیت حافظه این کار ممکن است صد ها میلی ثانیه طول بکشد، در طول این مدت کنترلر خارجی باید مدام CMD۱ را بفرستد و پاسخ R۱ را چک کند، هر گاه این پاسخ از ۱ به ۰ تغییر کند، یعنی عمل Initialize تمام شده و نوشتن و خواندن از کارت حافظه با پروتکل SPI قابل انجام می باشد. در این هنگام می توان رجیستر های OCR و CSD[۱] برای اطلاع از مشخصه های SD CARD قرائت نمود.

روند نمای راه اندازی اولیه مطابق شکل۴-۶ است.

۱ - Card Specific Data



شکل۴-۶- روندنمای راه اندازی

شرح کامل این بخش در بخش پیوست، ضمیمه ب آورده شده است.

۴-۴- ملاحظات عملی در اجرای طرح

همیشه شروع یک کار و پروژه جدید، از آنجایی که انسان با آن موضوع آشنایی ندارد و به جوانب کار آگاهی ندارد و نمی داند ملزومات کار، منابع و مراجع و ... را از کجا تهیه کند و نمی داند که کجای کار است نگرانی هایی را بدنبال دارد. بدین جهت مناسب است به افراد خبره در آن کار مراجعه و او را راهنمایی کنند و در کوتاهترین مدّت اطلاعات اولیه لازم را کسب کند. زیرا اگر خود فرد بخواهد به تنهایی این اطلاعات را بدست آورد بسیار وقت گیر خواهد بود. متأسفانه دسترسی به افرادی که خود

عملاً کار کرده باشند و مسلط به کار عملی باشند کم هستند و دسترسی به آنها ساده نیست در ادامه سعی شده است تا جایی که ممکن است و تجربه نگارنده اجازه می دهد به این موضوع پرداخته شود.

۴-۵- مراحل طراحی ساخت یک پروژه

برای طراحی و ساخت یک دستگاه که یک عمل خاص را انجام دهد بایستی مطابق روند زیر طراحی را کامل و مراحل مختلف پروژه را که در ادامه توضیح داده شده است طی نمود تا در کوتاهترین مدت بتوان پروژه مورد نظر را به نتیجه رساند و قبل از اینکه روند اجرای پروژه به نقطه مشخصی برسد و امکاناتی لازم داشته باشد آن امکانات قبلاً تهیه شده باشد و اجرای پروژه برای تهیه آن مقدمات با تأخیر نیفتد. برای این منظور روند زیر قابل تصور می باشد:

۱- تهیه ابزار و وسایل لازم نظیر کامپیوتر و نرم افزارهای لازم

۲- طراحی سخت افزار

برای این منظور بایستی برگه های اطلاعات تراشه های کاندید را جمع آوری و آنها را مطالعه و شناخت دقیقی از قابلیتها و محدودیتهای آن بدست آورد و با دید باز نوع پردازنده و میکروکنترلر را مشخص کرد. سپس به طراحی سخت افزار پرداخت و تمامی جوانب و امکانات لازم را مد نظر قرار داد و حتی در طراحی تا جایی که ممکن باشد حالت عمومی را در نظر گرفت و قابلیتهای بیشتری برای سخت افزار در نظر گرفت. زیرا ممکن است با گذشت زمان و تجربه بیشتر لازم باشد امکانات دیگری به سیستم اضافه گردد و این باعث نگردد کل طراحی به هم بریزد. استفاده از قطعات جانبی، کانکتورهای توسعه ای و ارتباطی در این مورد راهگشا خواهد بود.

۳- برنامه ریزی نرم افزار

ابتدا برنامه نویس صورت مسئله را دقیقاً برای خود تشریح نموده و کمبودها را که در صورت مسئله عنوان نشده با پرس و جو و یا با فرض های معقولی مرتفع می کند و سپس در حالی که هیچ مجهولی وجود ندارد روند نمای مسئله را برای خود ترسیم می نماید، ترسیم روند نما این حسن را دارد که برنامه نویس در آن واحد فقط با یک مشکل یعنی الگوریتم برنامه یا نوشتن برنامه درگیر است لذا می تواند با

مشکلات راحت تر برخورد نماید. پس از تنظیم روند نما نوشتن برنامه را بصورت جزء به جزء انجام داده و هر مرحله را تست می کند و بایستی در نظر داشته باشد که تست یک باره برنامه مخصوصاً موقعی که حجم برنامه طولانی است اصلاً کار درست و معقولی نیست و کمتر به نتیجه خواهد رسید و در صورتیکه به این صورت عمل کند پس مدتی خود به این نتیجه می رسد که قسمتهای کوچک برنامه را جدا کرده آنها را تست و برای جزء بعد اقدام نماید.

۴- تست برنامه ها و تکمیل سخت افزار

در طراحی یک سخت افزار که پیاده سازی انواع پروژه های متنوع در آن پیش بینی شده است (سخت افزار مشمول پردازنده های گوناگون، مبدل ها و واسطه های سریال، حافظه و ...) پیشنهاد می گردد چهار چوب کلی پروژه از جمله حافظه مورد نیاز، پورتهای مورد نیاز و امکانات مورد نیاز در طراحی پیش بینی شده و از لحاظ منطقی اشکالی نداشته باشد سپس به طراحی مدار نهایی اقدام نمود. بدین منظور در این طرح دو شماتیک و مدار چاپی طراحی گردید تا در طرح اولیه اشکالات سیستم مشخص شده و در طرح نهایی با چنین مشکلاتی مواجه نباشیم. در طرح اولیه از قطعات اصلی استفاده شده و عناصر فرعی در طرح نهایی به سخت افزار افزوده شدند. در طراحی لازم است که بخش به بخش مدار را تکمیل نمود و در پایان هر بخش صحت ارتباطات و عملکرد آن بخش را تست نمود. بعنوان مثال ابتدا تغذیه سیستم که تمامی قسمتها به آن مربوطند طراحی شده و تست گردد و ولتاژ خواسته شده را بدست آورد سپس به طراحی مینیمم سیستم شامل پردازنده و حافظه و قطعات اصلی پرداخت و در صورت نیاز قسمت دیگری از مدار را تکمیل کرد و هر مرحله را تست نمود. اگر به یکباره کل مدار بسته شود هر قسمتی از مدار ممکن است اشکالی را باعث شده باشد که روی عملکرد کل تأثیر گذارد، لذا بهترین کار این است که هر مرحله را تکمیل و تست نمود و در صورت بروز اشکال، مکان آن و سپس علت آن بسادگی مشخص می گردد و این روند تا آنجایی ادامه پیدا می کند که کل طراحی پیاده شده و در صورت تغییر در طراحی اصلاحات لازم صورت گرفته باشد. بایستی در نظر داشت که با یک تست ساده نمی توان از درست بودن طراحی مطمئن بود و بایستی تمامی جوانب حالات مختلف را در نظر

داشت. بعنوان مثال اگر خطوط چهاردهم و پانزدهم آدرس که به حافظه ها اعمال می گردند با هم اتصالی پیدا کرده باشند این اشکال بروز پیدا نمیکند مگر اینکه آدرس قرار گیری برنامه روی حافظه باعث تغییر وضعیت این خطوط گردد.

۵- تهیه برد مدار چاپی

پس از اینکه مدار موردنظر تکمیل و انواع تستها روی آن انجام گرفت و نقشه های مدارات بدرستی کشیده شد و عیبها مرتفع گردید به طراحی Lay out مدار چاپی می پردازیم. در این طراحی تجربه و حوصله حرف اول را می زند. مخصوصاً زمانیکه تیراژ نسبتاً بالا مد نظر باشد. زیرا کوچک در آوردن برد و کمتر Via زدن، کشیدن شیار ها و ارتباطات بهینه، به کیفیت مدار و سریعتر مونتاژ کردن و در نهایت به هزینه کمتر می انجامد که ممکن در مرحله اول به چشم نیاید ولی با گذشت زمان خود را نشان خواهد داد. لازم است که جهت تست اینکه ارتباطات درست وصل شده اند یا نه از کامپیوتر کمک گرفته شود. بدین صورت از مدار طراحی شده NET LIST گرفته و از آن در در نرم افزار طراحی مدار چاپی استفاده کرد (از Auto Route بهره می گیریم) ممکن است در ابتدای امر و کم تجربه بودن، این امر تا اندازه ای مشکل بنظر برسد اما بایستی در نظر داشت که فرا گیری این موضوع ارزش آنرا دارد و وجود یک اشکال در مدار چاپی و پیدا کردن و اصلاح آن وهمچنین ساخت مجدد آن قطعاً وقت بیشتری را از طراح خواهد گرفت و زمان بیشتری را باید گذاشت.

اغلب نرم افزارها قسمتی بنام نقشه کشی اتوماتیک دارند (Auto Route) که در طرح های بزرگ اغلب جوابگو نبوده و می بایستی بصورت دستی (Manual) اقدام به طراحی مدار چاپی نمود.

بنابراین برای طراحی یک مدار چاپی لازم است که:

- یک نقشه خوب از مدار مورد نظر تهیه شود.
- از نقشه مورد نظر NET LIST سازگار با نرم افزار طراحی Lay out را استخراج کرد.
- حتی المقدور Lay out توسط یک فرد با تجربه و صاحب ذوق و سلیقه کشیده شود. به این وسیله می توان ظاهر برد مدار چاپی را شکیل و در هزینه نهایی صرفه جویی کرد.

- برای تست Lay out طراحی شده از کامپیوتر کمک گرفت.
- اگر محیط قرار گیری برد مرطوب است و بمدت طولانی قرار است کار کند حتماً از چاپ سبز و برای مونتاژ راحت برد، از مارکاژ استفاده گردد.
- خطوط VCC و مخصوصاً GND تا جایی که ممکن است پهن انتخاب گردند (حداقل ۴۰ میل). همچنین در سخت افزار هایی که سیگنال آنالوگ و دیجیتال تؤاماً در مدار هستند می بایستی زمین دیجیتال و آنالوگ از یکدیگر جدا شوند. این مورد در مبحث فرکانس بالا از اهمیت ویژه ای برخوردار است. تجربه نگارنده و بررسی های صورت پذیرفته حاکی از این است که در برد های دولایه معمولی، در صورتی که فرکانس کلاک سیستم کمتر از ۱۰ مگاهرتز باشد نیازی به جداسازی زمین ها نیست و در فرکانس های بالاتر از این مقدار این موضوع اهمیت پیدا می کند.

۶- تست قطعات و مونتاژ آنها

صحت عملکرد قطعات قبل و خصوصاً بعد از عملیات لحیم کاری از اهمیت ویژه ای برخوردار است. آسیب دیدن قطعات با بسته بندی های کوچکتر مانند SMD,QFP,... هنگام عملیات لحیم کاری بدلیل حرارت بالای آن بسیار مترتب است. لذا تراشه های اصلی برد، توسط دستگاههای لحیم کاری مونتاژ شده اند.

۷- تست نهایی سخت افزار و نرم افزار

اصولاً گرفتن جواب اولیه و نسبتاً خام از مدارات الکترونیکی مشکل نیست. اما هنگامی که تعداد این دستگاهها زیاد و یا تحت شرایط محیطی متفاوتی کار می کنند ساده نیست و احتیاج به انجام تستهای متعدد و اصلاحات لازم دارد که نباید آنرا از نظر دور داشت. عوامل زیادی این مشکلات را باعث می گردند. قطعات تهیه شده مشخصات یکسانی ندارند. قطعات الکترونیکی امکان سوختن آنها در ابتدای امر که استفاده می گردند محتمل تر است بدین صورت که اگر مداری الکترونیکی بمدت چند ساعت درست کار کرد امکان اینکه بعدها بسوزد بسیار کم است، مگر اینکه شرایط اعمالی به مدار تغییر کند. بعنوان مثال پایه ای ورودی وجود داشته باشد که ولتاژ روی آن حداکثر ۳ ولت باشد و مداری محافظ پیش

بینی نشده باشد اگر در شرایط تست ولتاژ بیش از ۳ ولت اعمال نگردد و در شرایط کاری ولتاژی بیش از ۳ ولت اعمال گردد امکان سوختن وجود دارد. البته تستی که انجام می گیرد بایستی تمامی شرایط را در نظر گرفته باشد در غیر این صورت خواه نا خواه این گونه موارد مشکلاتی را بوجود خواهند آورد که بایستی برای رفع آنها از ابتدا پیش بینی های لازم را انجام داد.

۴-۶- عیب یابی مدارات

از آنجایی که دلایل وجود خطا در سیستمهای میکروپروسسوری بسیارند یکی از مشکل ترین کارها هنگام اجرای این پروژه ها عیب یابی سخت افزاری آنها می باشد و در این بین اشکالاتی وجود دارند که گاهی ظاهر و گاهی ظاهر نمی گردند که این حالت بمراتب مشکل تر است و امکان وجود آن بسیار زیاد است. کمتر کسی وجود دارد که با سیستمهای میکروپروسسوری همچون کامپیوتر کار کرده باشد و با این مشکلات برخورد نکرده باشد که این اشکالات گاهی نرم افزاری و گاهی سخت افزاری می باشند که سعی می گردد منبع این خطا بیان گردد. از عمده مشکلات موجود عبارتند از:

- صاف نبودن و تغییرات منبع تغذیه و عدم جریان دهی مناسب آن: اگر در حین اجرای برنامه ولتاژ تغذیه به دلایلی کم گردد روند اجرای برنامه مختل شده و سیستم کار خود را انجام نمی دهد در این رابطه بایستی جریان دهی منبع مناسب بوده و از صافی مناسبی استفاده گردد. استفاده از خازنهای صافی در خروجی منبع تغذیه و همچنین استفاده از خازنهای دی کوپلاژ مفید است. لازم به توضیح است آی سی ها و میکروپروسسورها یک جریان ثابت از تغذیه نکشیده و این جریان مرتب در حال تغییر است و ممکن است شرایطی پیش بیاید که در آن واحد چند آی سی جریان زیادی لازم داشته باشند که اگر خازنها مربوطه پیش بینی شده باشند جریان لحظه أی لازم را تأمین می کنند در غیر اینصورت ولتاژ افت کرده و اگر از حدی معینی کمتر گردد مدار با مشکل برخورد می کند.
- اشکالات زمانی (Timing) مدارات: این نیز یکی از مشکلاتی است که عیب یابی آن ساده نیست. این مشکل ناشی از تأخیر آی سی ها می باشد. ممکن است مدار با یک آی سی با یک

شماره مشخص ساخت یک شرکت کار کند و با یک آی سی دیگر ساخت شرکت دیگر کار نکند. ممکن است اگر مدار سرد باشد کار کند و اگر مدار گرم شد کار نکند.

در این رابطه بایستی کلاک انتخابی برای پردازنده را تاجایی که ممکن است کم انتخاب کرد و یا از قطعات سریعتر استفاده کرد و از گیتهای پشت سر هم که هر کدام تأخیری به مدار اعمال می کنند اجتناب کرد.و در طراحی به حداکثر تأخیر گیتها و زمانبندی[1] آی سی ها توجه داشت. این مورد مخصوصا برای ارتباط با رم حائز اهمیت می باشد. در طراحی سخت افزار زمانبندی طرح بسیار مهم می باشد. در واقع یک طراح سخت افزار خوب کسی است که به مبحث زمانبندی آشنا باشد.

- وجود قطعات سوخته در مدار: اگر سیستم میکرو پروسسوری از قطعات زیادی تشکیل شده باشد و یکی از آنها سوخته شده باشد بسته به اینکه در کجای مدار قرار گرفته باشد عیب یابی آن بسادگی ممکن نیست. راه حل ساده و مطمئنی که پیشنهاد می گردد تعویض یکباره تمامی آی سی های قابل تعویض آن است. لذا اگر اشکالی در قطعات باشد مدار با قطعات جدید جواب خواهد داد که پس از آن می توان قطعات را یکی یکی در آورده و قطعه اولی جایگزین گردد تا قطعه سوخته پیدا شود. البته اگر امکان چنین کاری وجود داشته باشد. فرضاً در قطعات SMD عملیات جابجا کردن قطعات بسیار مشکل است. زیرا باعث صدمه دیدن شیارها و مدار چاپی شده و حتی تست قطعه جایگزین شده بسیار سخت تر از قبل خواهد شد. بدلیل اینکه تست ارتباطی شیارها نیز علاوه بر تست های قبلی اضافه شده است. البته استفاده از اسیلوسکوپ در بعضی از موارد کمک خوبی می کند. به شرط آنکه وضعیت درست سیگنالها مشخص باشد. از آنجایی که تعداد پایه ها بسیار زیاد و سیستم از روند عادی خود خارج شده است عیب یابی با اسیلوسکوپ بسیار وقت گیر و پردردسر است.
- ترک خوردگی، خراشیدگی یا پوسیدگی برد مدار چاپی: عیب های فیزیکی مدار چاپی ممکن است هنگام مونتاژ برد وجود داشته باشند و یا بدلیل عدم کیفیت مناسب و شرایط جوی بد و

۱- Timing

تحت فشار قرار گرفتن برد بعدها بوجود بیایند. لذا اولین قدم در عیب یابی یک سیستم، بازدید دقیق بردهای آن بوده و اگر اشکالی مشاهده نشد دیگر احتمالات بررسی می گردد.

- درست جا نخوردن قطعات روی سوکتها و برقرار نشدن اتصال در اثر لحیم کاری: شل شدن ارتباط سوکتها و آی سی ها به جهت اینکه برنامه روی سیستم بایستی قابل تغییر باشد استفاده از سوکت برای حافظه های EPROM اجتناب ناپذیر است.
- تأثیر ولتاژهای ناخواسته روی مدار: در صورتیکه مدارات بصورت مناسب شیلد نشده باشند و در محیط نویزی قرار گیرند این امکان وجود که مدار تحت تأثیر قرار گرفته در حالاتی درست عمل نکند.
- استفاده از نرم افزارهایی جهت عیب یابی

بایستی توجه داشت هر برنامه نرم افزاری برای تست و عیب یابی مناسب نیست، لذا بایستی برنامه ای مناسب برای این منظور نوشت بنحوی که صد در صد درست بوده و نتیجه مشخصی که قابل دیدن و ارزیابی باشد ارائه دهد.

- عدم اتصال مناسب کانکتورها

ارتباطات بین برد و محیط خارج از طریق کانکتورها تأمین می گردد. اگر این ارتباطها شل باشند عملکرد سیستم دستخوش تغییرات شده و جواب مطلوب حاصل نمی شود. لذا بایستی از کانکتورهای مناسب که از حالت فنریت خوبی برخوردارند استفاده کرد. لذا می بایست تغذیه روی تمامی آی سی ها چک گردد، سالم بودن تمامی قطعات بررسی، ارتباطهای مهم و اساسی همچون خطوط دیتا، آدرس و کنترل بصورت اهمی چک شوند.

۴-۷- مراحل اجرای پروژه

برای اجرای هر پروژه تحقیقاتی و کاربردی بایستی مراحل مختلفی را طی کرد تا به نتیجه مطلوب رسید. ابتدا بایستی مشخصات پروژه تعیین، راه حلهای ممکن تعیین و از بین آنها بهترین راه انتخاب و طراحی های لازم انجام و پس از بستن مدار و انجام تستها و عیب یابیهای مختلف، آنرا به نتیجه رساند.

برای پروژه مورد نظر نیز این مراحل که در ادامه با جزئیات بیشتری آورده شده اند طی شده و جواب نهایی که طراحی و ساخت سیستم پردازش موازی مبتنی بر FPGA بوده است حاصل شده است.

۴-۷-۱- تعریف دقیق پروژه و اهداف مورد نظر

در پروژه های کاربردی و عملی یکی از موارد مشکل و وقت گیر تعریف صورت مسئله است. اگر پروژه از جانب فرد، گروه یا شرکتی پیشنهاد شده باشد و تمامی جزئیات در آن منعکس شده باشد طراح بدون هیچ دغدغه ای به طراحی سخت افزار و نوشتن نرم افزار پرداخته و تنها مشکل او بعد علمی و عملی بوده و کار او مشخص و روتین وار می باشد. اما همیشه اینطور نیست. در این صورت کار طراح یا برنامه نویس مشکل بوده و بایستی با انجام تحقیقات و بررسی های مختلف سخت افزار ها و سیستم های مشابه، با در نظر گرفتن محدودیتها و امکانات پیش بینی شده کلیه جزئیات لازم را استخراج و در طراحی و برنامه نویسی لحاظ نماید. در هر صورت با در نظر گرفتن موارد بالا طراح بایستی بداند جهت انجام یک پروژه نرم افزاری یا سخت افزاری دانستن کلیه جزئیات لازم و ضروری بوده و بایستی قبل از هر اقدامی به مشخص کردن آنها اقدام کند. وجود این نوع کم و کاستی ها هنگام اجرای یک پروژه عملی کاملاً طبیعی بوده که تنها با مطالعه و تحقیق و انجام تحقیقات متعدد می توان به قسمتی از آنها پاسخ گفت. لذا قبل از هر اقدامی بایستی صورت مسئله دقیقاً تشریح و خواسته ها مشخص گردند.

یکی از مشکلات موجود در اجرای این پروژه نیز تعیین مشخصات لازم برای سخت افزار و تعیین قابلیتهای آن بود که پس از بررسی های فراوان سخت افزار مورد نظر ارائه گردید.

۴-۷-۲- انجام مطالعات و تحقیقات لازم

پس از اینکه صورت مسئله و خواسته های پروژه مشخص گردید بایستی راه حلهای مختلف و ابزار لازم جهت رسیدن به هدف تعیین گردند. تعیین راه حلهای ممکن و ابزار لازم و اینکه چگونه پروژه پیاده سازی گردد ساده نبوده و به عوامل زیادی وابسته است در این بین بایستی اطلاعات موجود، وجود قطعات و عناصر لازم، هزینه سخت افزار و نرم افزار و ... مد نظر قرار گیرد. در اولین مرحله بایستی

اطلاعات لازم در آن زمینه را حد امکان تکمیل نمود و اگر کار یا فعالیتی قبلاً انجام شده است شناسایی و مطالعه گردد و پس از بدست آوردن اطلاعات لازم تصمیم گیری لازم انجام گیرد.

در پروژه مورد نظر نیز این مطالعات انجام شده و پاسخ مناسبی برای آنها دریافت گردید. این مطالعات بطور کلی روی دو موضوع مبانی و ساخت متمرکزند که هریک بایستی بدقت مورد بررسی و مطالعه قرار گیرد و مناسب ترین انتخاب صورت گیرد. بدین منظور بررسی بر روی پروژه هایی که با تراشه FPGA در برخی از دانشگاه ها و مراکز تحقیقاتی انجام شده بود، صورت پذیرفت و قابلیت ها و امکانات سخت افزاری آنها مورد مطالعه قرار گرفت.

۴-۷-۳- تعیین راه حلهای ممکن و انتخاب بهترین راه

برای حل یک مسئله راه حلهای مختلفی ممکن است وجود داشته باشد. بایستی ابتدا این راه حلها شناسایی و باتوجه به بررسی های انجام شده مناسب ترین راه با توجه به هزینه، قابلیت اجرا و سادگی انتخاب و اقدامات بعدی صورت گیرد. در این پروژه می بایست ساختار سخت افزار، نوع پردازنده ها، ساختار نرم افزار و کامپایلرهای مناسب و امکانات لازم برای سخت افزار و نرم افزار مشخص گردند و سپس از بین پاسخهای مختلف موجود، مناسب ترین راه با توجه به امکانات موجود انتخاب گردد.

۴-۷-۴- تهیه قطعات و طراحی سخت افزار

پس از انجام مطالعات و تعیین راه حلهای ممکن برای سخت افزار و دریافت پاسخ مناسب برای هر سؤال بایستی قطعات و عناصر مورد استفاده تهیه گردند. طراحی و تهیه قطعات بایستی توأم انجام گیرد بدون در اختیار بودن قطعات و عناصر لازم، طراحی قابلیت اجرا نخواهد داشت. این مسئله کار را مشکل نموده و باعث به عقب افتادن طراحی و ساخت می گردد. در اجرای پروژه های منجر به ساخت با موارد زیاد برخورد می شود که قطعات مورد استفاده در طراحی در بازه زمانی مشخص در اختیار نیست، لذا بایستی از قطعات و عناصری استفاده کرد که از منابع مختلف بتوان آنها را تهیه کرد و یا موارد زیادی پیش می آید که در حین طراحی و ساخت به عناصر و قطعات جدیدتر و بهتری می توان دست یافت که

به بهبود طراحی بینجامد. مجموعه این عوامل کار طراحی و ساخت و در اختیار بودن قطعات را مشکل می سازد و بایستی در هر صورت آنرا در نظر داشت.

۴-۷-۵- استفاده از برنامه های تست و آموزشی

پس از ساخت سخت افزار بایستی برنامه های مختلفی جهت تست قسمتهای مختلف سیستم انجام داد. در این زیر برنامه ها هدف تنها تست قسمتی از سیستم است که نشان می دهد ارتباط آن با سخت افزار واقعی برقرار شده است. از آنجایی که طراح در این مرحله با دو مشکل همزمان یعنی درست بودن اتصال سخت افزار و برنامه تست نوشته شده درگیر می باشد بایستی سعی کرد تا جایی که ممکن است برنامه تست از ساختار ساده و مطمئنی برخوردار باشد بنحوی که از صحت آن مطمئن و عیب سخت افزار را بدرستی نشان دهد. در طراحی سیستمهای میکروپروسسوری بسیار پیش می آید که طراح اشکال را در سخت افزار جستجو می کند در حالی که اشکال در نرم افزار است و همچنین مواقعی پیش می آید که اشکال در نرم افزار است و عیب را در نرم افزار جستجو می کند. این مشکل بزرگی است که تنها با تجربه و سعی و خطا و نوشتن برنامه تست ساده ممکن است قابل حل باشد.

پس از ساخت برد و تست قسمتهای مختلف سیستم و عیب یابی سخت افزار و رفع نواقص، نوشتن برنامه اصلی با توجه به مطالعات انجام شده و روندنمای ترسیم شده شروع می گردد. در نوشتن این برنامه ها از برنامه های تست نوشته شده کمک گرفته می شود. انتخاب زبان برنامه نویسی و رعایت قواعد برنامه نویسی از اهمیت خاصی برخوردار است. نوشتن برنامه های آموزشی به دو بخش برنامه نویسی با زبان C و برنامه نویسی با VHDL تقسیم می گردد که قسمت مهمی از کار را به خود اختصاص می دهد.

۴-۷-۶- طراحی Layout مدار چاپی

پس از اینکه طراحی انجام شد و بررسی های مربوطه صورت پذیرفت، بایستی اقدام به ساخت مدار چاپی نمود. ساخت مدار چاپی هنگامی که تعداد قطعات و عناصر زیاد باشد و طراحی پیچیده باشد مشکل است و به تجارب خاصی نیاز دارد. در این پروژه با توجه به سوابق قبلی و اینکه طراحی با استفاده

از Auto Route نا ممکن است، طراحی بصورت دستی انجام پذیرفت. بدین منظور از نرم افزار Protel DXP استفاده شد و برد مدار چاپی طراحی گردید.

۴-۷-۷- سفارش ساخت برد مدار چاپی

پس از طراحی مدار چاپی با نرم افزار Protel DXP بایستی اقدامات لازم جهت ساخت آنرا انجام داد. این برد بصورت دولایه و متالیزه بهمراه چاپ سبز و مارکاژ سفارش داده شده است. ساخت این بردها نسبتاً وقت گیر و هزینه بر بوده و همچنین ممکن است بردهای ساخته شده اشکالاتی نیز داشته باشند و لازم باشد آن اشکالات مرتفع گردند.

۴-۷-۸- مونتاژ برد مدار چاپی و تست آن

پس از ساخت برد مدار چاپی و تهیه قطعات لازم بایستی قطعات و عناصر مورد استفاده در مکانهای خود نصب گردند. هنگام طراحی و ساخت برد مدار چاپی اشکالات متعددی ممکن است وجود داشته باشد که تنها با مونتاژ و تست برد ساخته شده مشخص خواهند شد. این اشکالات می تواند موادر زیر باشند:

- قطر سوراخهای در نظر گرفته شده برای قطعات مناسب نباشد.
- جای در نظر گرفته شده برای قطعات کافی نباشد و لازم باشد قطعات جابجا گردند.
- ترتیب پایه های در نظر گرفته شده برای قطعات مناسب نباشد.
- نقشه شماتیک اشکالاتی داشته باشد و متناسب با آن برد مدار چاپی درست نباشد.بعموان مثال تغذیه برای تمامی آی سی ها اعمال نشده باشد.
- طراحی نقص داشته باشد و مواردی در آن لحاظ نشده باشد.

۴-۷-۹- اصلاح نقشه های شماتیک و مدار چاپی

پس از مونتاژ و مشخص شدن اشکالات که به احتمال زیاد وجود خواهند داشت نقشه های شماتیک و مدار چاپی اصلاح و برد نهایی ساخته می شود. در این برد سخت افزاری تاجایی که ممکن است بایستی

از فضای روی برد استفاده کرد و فضای خالی روی برد کم باشد. تجربه طراح و نرم افزار مورد استفاده در این رابطه تعیین کننده است. هرچقدر سطح برد کمتر باشد هزینه ساخت کمتر بوده و به توجیه اقتصادی طرح کمک می کند. در صورتی که تعداد بردهای درخواستی زیاد باشد بایستی یک یا چند نمونه از برد مدارچاپی اصلاح شده تهیه و مونتاژ شود و تستهای لازم انجام گیرد و در صورت وجود هیچگونه نقصی سفارش به تعداد مورد نیاز صورت گیرد. در این طرح یک طرح اولیه و طبیعتاً یک مدار چاپی تهیه شد و با بررسی اشکالات آن مدار نهایی تهیه گردید.

۴-۷-۱۰- ساخت برد نهایی

پس از اصلاح نقشه شماتیک و اصلاح مدارچاپی بایستی برد مورد نظر را جهت سفارش نهایی آماده کرد.

۴-۷-۱۱- تست و ارزیابی نهایی

پس از مونتاژ برد نهایی بایستی آنرا مورد ارزیابی و تست قرار داده و جواب مطلوب را از آن استخراج کرد. بدلایل متعددی ممکن است برد ساخته شده جوابگوی نیازها نباشد و لازم باشد به مراحل قبل برگشته و روند را تکرار نمود در هر صورت بایستی بردنهایی با مشخصات خواسته شده ساخته شده و مورد تأیید قرار گیرد.

۴-۷-۱۲- نوشتن برنامه های لازم روی کامپیوتر

پس از تهیه سخت افزارها و نوشتن برنامه های لازم در محیط CODE VISION، ISE و انجام تستهای مختلف، نوشتن برنامه لازم روی کامپیوتر جهت برنامه ریزی و مدیریت در محیط مناسب شروع می گردد. تست ارتباطی بین برد و کامپیوتر نیز از جمله موارد دیگر در این طرح است که می بایستی مورد توجه قرار گیرد. در این طرح برنامه های متنوعی بر روی برد پیاده سازی شده است. بعنوان مثال می توان به پیاده سازی شیفت رجیستر، حافظه FIFO، برنامه های تست ارتباطی نظیر روشن/خاموش کردن LED، ارتباط با میکروکنترلر و رم، پیاده سازی برخی فیلترهای دیجیتال از جمله برنامه های آزمایشی پیاده سازی شده بر روی سخت افزار می باشد. همچنین سخت افزار قابلیت پیاده سازی انواع

فیلترها، دیکدر و انکدرها، انواع کارتهای ترکیب کننده صدا و تصویر، سوئیچ های سرعت پایین و متوسط (حدود ۵۰Mbps) و پروژه های کنترلی را داراست.

۴-۷-۱۳- مستند سازی و گزارش نهایی

پس از اینکه برنامه کامل و جواب مورد نظر حاصل شد ممکن است لازم باشد تغییراتی در برنامه ایجاد گردد. برای این منظور بایستی شرح برنامه، نمودار ساختار برنامه، منطق برنامه، لیست برنامه همراه با توضیحات لازم بهمراه طراحی قسمتهای مختلف سخت افزار مورد بررسی قرار گرفته و در دسترس باشد و این ممکن نیست مگر اینکه پس از تکمیل سخت افزار و برنامه ها اقدامات لازم جهت مستند سازی صورت گیرد.

۴-۷-۱۴- نتایج بدست آمده

برای هر پروژه تحقیقاتی اهداف اولیه خاصی مد نظر قرار می گیرد که در مدت اجرای آن سعی می گردد آن اهداف تعقیب گردند. همانطور که اشاره شد طراحی و ساخت سیستمی مد نظر است که قابلیت پردازش موازی را داشته باشد و همچنین تراشه های متنوعی برای پیاده سازی پروژه های گوناگون در سخت افزار باشند. در این بین کاربردی بودن سیستم، هزینه قابل قبول و قابل اجرا بودن از اهمیت خاصی برخوردار است. پس از طراحی، قسمتهای مختلف سیستم ساخته شده، نرم افزارهای تست و آموزشی روی سخت افزار و کامپیوتر نوشته و جواب لازم دریافت گردید. نتایج بدست آمده از اجرای پروژه به دو مقوله نتایج پژوهشی و نتایج عملی تقسیم بندی می گردد. در نتایج پژوهشی ابتدا مطالعات و تحقیقات لازم در مورد تراشه های مرتبط از جمله FPGA های خانواده SPARTAN۳ انجام شده، مشخصات نرم افزارهای واسطه و سخت افزاری مرتبط مطالعه شده و طراحی های لازم جهت ساخت برد انجام می گیرد که نتایج حاصل از آن بصورت خلاصه در گزارش ارائه شده و علاوه بر این، سخت افزاری مناسب برای انجام آزمایشها و تحقیقات مختلف با ساخت این سیستم، فراهم می گردد. نتایج عملی نیز بیشتر به بررسی افزایش تجارب عملی طراح پروژه می پردازد و در این بین می توان به موارد زیر اشاره کرد:

- طراحی مدار چاپی و کارکرد بیشتر با نرم افزار های طراحی مدار
- آشنایی با فوت پرینت قطعات با پایش قطعات موجود در بازار
- محدودیت های طراحی مدار چاپی از لحاظ فرکانس کاری، شیارهای ارتباطی و ...
- ملاحظات طراحی شماتیک مدار (تغذیه و ارتباطات تراشه ها، استفاده از جامپرها، ارتقاء برد و...)
- اندازه ی پهنای خطوط و تنظیمات Clearence بین خطوط
- انتخاب پکیج قطعات
- مباحث مرتبط با خطوط انتقال (خطوط سیگنال نباید زیاد طولانی باشند. حتی در مواردی که اختلاف فاز و تأخیر در فرکانس های نه چندان بالا محدودیت ایجاد نکند. در واقع می بایستی مسیر زمین از دید خط انتقال کوتاه ترین مسیر باشد. این کار باعث کاهش امپدانس خط می شود. همچنین خطوط سیگنال نباید بیش از اندازه باریک باشند. اگر در مدارمان سیگنال RF هم داشتیم، باید بحث انعکاس موج و تأثیر فیلترینگ خطوط و همشنوایی[۱] ها هم در نظر گرفته شود. در انتها باید توجه نمود که بار خازنی روی پایه ها نباید از یک حدی بیشتر شود)

۱ - Cross talks

فصل پنجم

پیاده سازی الگوریتم نمونه تشخیص لبه

۵-۱- مقدمه

در این فصل به بررسی پیاده سازی یکی از الگوریتم های پردازش تصویر بر روی سخت افزار پرداخته می شود. آشنایی با روشهای تشخیص لبه تصاویر بعنوان زیر مجموعه ای از بخش بندی تصاویر از جمله مباحثی است که در این فصل به آن پرداخته می شود. در واقع برای تست سخت افزار این الگوریتم بعنوان برنامه ای نمونه و تست، مورد بررسی قرار گرفته است. در ادامه به بررسی نحوه پیاده سازی بر روی FPGA، ارسال اطلاعات تصویر از کامپیوتر به برد و دریافت نتیجه، آشنایی با نحوه ارتباط با FPGA و پیکربندی آن پرداخته می شود.

۵-۲- معرفی

معمولاً مرحله اول در تحلیل تصویر، بخش بندی است. با عمل بخش بندی، تصویر به قسمت های تشکیل دهنده‌اش تقسیم می‌شود. میزان بخش بندی، به موضوع مورد نظر بستگی دارد. یعنی وقتی اشیای مورد علاقه کاربرد مورد نظر از هم جدا شدند، باید بخش بندی متوقف شود[۱٤].

مفهوم بخش بندی تصویر بر اساس ناپیوستگی یا شباهت مقادیر سطوح خاکستری را می‌توان برای هر دو نوع تصویر ثابت و متحرک به کار برد. از الگوریتم‌های بخش بندی تصاویر تک رنگ، برای آشکارسازی خطوط و لبه‌های تصاویر استفاده می شود. در یک تصویر، مرز میان یک شیء و زمینه و یا مرز میان اشیاء همپوشان را لبه تعریف می‌کنند. در حالت ایده‌آل اگر فرض شود که مقدار شدت روشنایی هر تصویر یکنواخت و با مقادیر شدت روشنایی اشیاء مجاورش متفاوت باشد، آنگاه هرگونه تغییر قابل ملاحظه در مقدار شدت روشنایی را می‌توان لبه در نظر گرفت.

با این تعریف، اگر لبه‌های یک تصویر مشخص شوند، مکان تمام اشیاء موجود در تصویر مشخص شده و خواص اساسی آنها از قبیل سطح، محیط و غیره به راحتی قابل اندازه‌گیری خواهند بود. بخش بندی، تصویر را به ناحیه های سازنده آن تقسیم می کند. این امر یکی از مشکل ترین وظیفه ها در پردازش تصویر است. دقت بخش بندی، موفقیت یا شکست نهایی رویه های تحلیل کامپیوتری را تعیین می کند.

بخش بندی تصویر مبتنی بر دو روش است. در روش اول تصویر بر اساس تغییرات سریع شدت، مثل لبه ها است. در روش دوم مبتنی بر تقسیم بندی تصویر به ناحیه هایی است که بر اساس مجموعه ای از معیارهای از پیش تعریف شده شبیه اند. مانند آستانه گیری، رشد ناحیه، تقسیم ناحیه و ... . در این طرح یک الگوریتم نمونه ای برای پیاده سازی بر روی FPGA مد نظر بوده است. بدین منظور تشخیص لبه تصاویر برای پیاده سازی انتخاب شده است. تصویر ورودی توسط نرم افزار MATLAB خوانده شده و از طریق پورت سریال به سخت افزار منتقل می شود. میکروکنترلر با دریافت اطلاعات، آنها را در بافری ذخیره نموده و پس از دریافت تمام داده ها، آنها را برای پردازش به FPGA ارسال می نماید. FPGA نیز با دریافت داده و پردازش بر روی داده های ورودی، نتایج حاصله را مجدداً به میکروکنترلر ارسال می نماید. روند کلی طرح بصورت فوق می باشد. در ادامه مختصراً به بررسی مبانی و روشهای تشخیص لبه پرداخته شده، سپس به بررسی نحوه پیاده سازی الگوریتم اشاره می شود.

الگوریتم های بخش بندی برای تصاویر تک رنگ، معمولاً بر اساس یکی از دو دسته ای هستند که با مقادیر شدت سروکار دارند: ناپیوستگی و شباهت. در دسته اول، فرض بر این است که مرزهای ناحیه ها کاملاً متفاوت از هم و متفاوت از پس زمینه اند. در نتیجه تشخیص مرز بر اساس ناپیوستگی های محلی در شدت امکان پذیر است. بخش بندی مبتنی بر لبه روش اصلی مورد استفاده در این دسته است. روش های بخش بندی مبتنی بر ناحیه در دسته دوم، مبتنی بر تقسیم بندی تصویر به ناحیه ای است که بر اساس مجموعه ای از معیار از پیش تعریف شده مشابه اند[۱۴].

۵-۳- تشخیص لبه

پیکسل های لبه، پیکسل هایی هستند که در آنها شدت تابع تصویر به شدت تغییر می کند، و لبه ها یا بخش های لبه مجموعه هایی از پیکسل های لبه متصل هستند. تشخیص دهنده های لبه روش های پردازش تصویر محلی هستند که برای تشخیص پیکسل های لبه طراحی شدند. خط را می توان به عنوان

یک بخش لبه دانست که در آن، شدت پس زمینه در هر دو طرف خط، خیلی بالاتر یا خیلی کمتر از شدت پیکسل های خط است. در حقیقت خطوط به لبه های سقف ارتقاء می یابند.

همانطور که می دانیم، میانگیری محلی تصویر را هموار می کند. با توجه به این که میانگین گیری مشابه جامعیت است، تعجبی ندارد که تغییرات محلی شدید در شدت توان می تواند با استفاده از مشتق ها تشخیص داده شود[۱۴]. مشتق های مرتبه اول و دوم، برای این منظور مناسب اند. مشتق های تابع دیجیتال بر حسب تفاضل ها تعریف می شوند و راههای مختلفی برای تخمین این تفاضل ها وجود دارد و لازم است که مشتق ها شرایطی را برای وجود دارا باشند. برای مشتق گیری با FPGA نیز از تفاضل استفاده می نمائیم. اعم از اینکه در ناحیه هایی با شدت ثابت، صفر باشد، در شروع مرحله شدت یا شیب غیرصفر باشد و در نهایت اینکه باید در نقاطی در امتداد شدت غیرصفر باشد.

مشتق مرتبه اول در شروع و در امتداد کل شیب شدت غیر صفر است، در حالی که مشتق دوم فقط در شروع و پایان شیب غیر صفر است. چون لبه های تصاویر دیجیتال شبیه این نوع تغییر است نتیجه می گیریم که مشتق های مرتبه اول، لبه های ضخیم و مشتق های مرتبه دوم لبه های ریزتری را تولید می کنند. بعلت اینکه مشتق مرتبه دوم نسبت به مشتق مرتبه اول در ارتقاء های تغییر تیز، شدید تر است. بنابراین مشتقات مرتبه دوم نسبت به مشتقات مرتبه اول جزئیات ریز از جمله نویز را ارتقاء می دهند. مشتق مرتبه دوم وقتی که به داخل یا خارج یک لبه منتقل می شود، علامت های مثبت و منفی دارد. این اثر دو لبه ای ویژگی مهمی است که می تواند برای یافتن لبه ها استفاده شود. علامت مشتق دوم برای تعیین این نکته به کار می رود که آیا لبه گذاری از روشن به تاریک یا از تاریک به روشن است که وقتی داخل لبه می رویم این علامت مشاهده می شود.

در نتیجه مشتقات مرتبه اول لبه های ضخیم تری در تصویر ایجاد می کند و مشتقات مرتبه دوم پاسخ قوی تری به جزئیات ریز، مانند خطوط باریک، نقاط مجزا و نویز دارند. مشتقات مرتبه دوم پاسخ

دولبه ای در گذارهای مرحله و شیب در شدت تولید می کنند و علامت مشتق دوم می تواند تعیین کند که آیا انتقال به داخل لبه از روشنایی به تاریکی یا از تاریکی به روشنایی است.

روش انتخاب برای مشتقات اول و دوم در هر مکان پیکسل در یک تصویر استفاده از فیلترهای مکانی است. برای نقاب ۳*۳ روش کار محاسبه مجموع حاصلضربهای ضرایب نقاب با مقادیر شدت در ناحیه ای است که توسط نقاب در برگرفته شده است. یعنی پاسخ نقاب در نقطه مرکزی ناحیه عبارت است از:

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{k=1}^{9} w_k z_k \qquad (۱-۵)$$

که z_k شدت پیکسلی است که مکان متناظر با مکان k امین ضریب در نقاب است. به عبارت دیگر محاسبه مشتق ها بر اساس نقاب مکانی فیلتر مکانی تصویر با آن نقاب است[۱۴].

همانطور که اشاره شد تشخیص تغییرات در شدت به منظور یافتن لبه ها می تواند با استفاده از مشتق های مرتبه اول و دوم انجام گیرد. در این بخش به بررسی مشتق های مرتبه اول پرداخته می شود.

۵-۴- گرادیان تصویر

ابزار انتخاب برای یافتن قدرت و جهت لبه در مکان (x,y) در تصویر f، گرادیان نام دارد و با ∇f نشان داده می شود و بصورت بردار زیر تعریف می شود:

$$\nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad (۲-۵)$$

همانطور که می دانید، گرادیان دارای این خاصیت است که به جهت بیشترین نرخ تغییر f در مکان (x,y) اشاره دارد. اندازه بردار ∇f با $M(x,y)$ نشان داده می شود که عبارت زیر مقدار نرخ تغییر در جهت بردار گرادیان است:

$$M(x,y) = mag(\nabla f) = \sqrt{g_x{}^{۲} + g_y{}^{۲}} \qquad (۳-۵)$$

g_x ، g_y و $M(x,y)$ تصاویری با اندازه اصلی اند و وقتی ایجاد شدند که x و y اجازه داشتند روی تمام مکان های پیکسل f تغییر کنند. تصویر آخری را تصویر گرادیان یا وقتی که معنای آن مشخص باشد، گرادیان می نامند.

جهت بردار گرادیان با زاویه زیر مشخص می گردد که نسبت به محور X اندازه گیری می شود:

$$\propto (\mathrm{x},\mathrm{y}) = \tan^{-۱}\left[\frac{\mathrm{g_x}}{\mathrm{g_y}}\right] \qquad (۴-۵)$$

همانند تصویر گرادیان، $\propto (\mathrm{x},\mathrm{y})$ تصویری با اندازه اصلی است که با تقسیم آرایه ای تصویر $\mathrm{g_y}$ بر $\mathrm{g_x}$ ایجاد شد. جهت لبه ای در نقطه اختیاری (x,y) عمود بر این جهت بردار گرادیان در آن نقطه است.$(\propto (\mathrm{x},\mathrm{y}))$.

بدست آوردن گرادیان تصویر نیازمند محاسبه مشتقات جزیی $\frac{\partial f}{\partial x}$و$\frac{\partial f}{\partial y}$ در هر مکان پیکسل در تصویر است. با توجه به اینکه با کمیت های دیجیتال سروکار داریم، لذا نیاز به تخمین دیجیتال مشتقات جزیی روی خروجی یک همسایگی حول یک نقطه می باشیم.

داریم:

$$g_x = \frac{\partial f(x,y)}{\partial x} = f(x+۱,y) - f(x,y) \qquad (۵-۵)$$

$$g_y = \frac{\partial f(x,y)}{\partial y} = f(x,y+۱) - f(x,y) \qquad (۶-۵)$$

این دو معادله می تواند برای تمام مقادیر مربوط به x و y، بوسیله فیلترکردن f(x,y) با نقاب های یک بعدی پیاده سازی شود. وقتی جهت لبه قطری مطلوب باشد، به دو نقاب دو بعدی نیاز داریم. همانند عملگرهای دوبعدی روبرتز، پرویت و سوبل.

عملگرهای روبرتز مبتنی بر پیاده سازی تفاضلهای قطری زیر هستند:

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \quad (۷-۵)$$

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6) \quad (۸-۵)$$

این مشتق ها می توانند با فیلترکردن یک تصویر با نقاب هایی دودویی پیاده سازی شوند.

ساده ترین تخمین های دیجیتال از مشتقات جزیی با استفاده از نقاب هایی به اندازه ۳*۳ به صورت زیر مشخص می گردد:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad (۹-۵)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \quad (۱۰-۵)$$

در این فرمول ها تفاضل بین سطرهای سوم و اول ناحیه ۳*۳، مشتق را در جهت x، و تفاضل بین ستونهای سوم و اول، مشتق در جهت y را تخمین می زند. معادلات فوق می توانند بوسیله فیلتر کردن f با دو نقاب معرفی شده روی کل تصویر پیاده سازی شوند. این نقاب بنام عملگرهای پرویت (Prewit) خوانده می شود.

شکل دیگری از دو معادله قبلی از وزن ۲ از ضریب مرکزی استفاده می کنند:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (۱۱-۵)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (۱۲-۵)$$

Z۱	Z۲	Z۳
Z٤	Z٥	Z٦
Z۷	Z۸	Z۹

استفاده از وزن ۲ در مکان مرکزی، هموار سازی تصویر را فراهم می کند[۱۴]. این نقاب به عملگرهای سوبل (Sobel) معروف است. شکل ۵-۱ نقاب های مورد استفاده برای پیاده سازی معادلات فوق الذکر را نشان می دهد.

نقاب روبرتز

۰	-۱
۱	۰

-۱	۰
۰	۱

نقاب پرویت

-۱	-۱	-۱
۰	۰	۰
۱	۱	۱

۱	۰	-۱
۱	۰	-۱
۱	۰	-۱

نقاب سوبل

-۱	-۲	-۱
۰	۰	۰
۱	۲	۱

۱	۰	-۱
۲	۰	-۲
۱	۰	-۱

شکل ۵-۱: نقابهای گوناگون برای محاسبه گرادیان تصویر

با انتخاب یکی از این عملگرها و فیلتر کردن تصویر با این معادلات، تصویر نهایی، لبه های تصویر اصلی خواهد بود. در این طرح از عملگر سوبل برای تشخیص لبه و پیاده سازی بر روی سخت افزار استفاده شده است.

۵-۵- ارتباط با FPGA و پیاده سازی بر روی آن

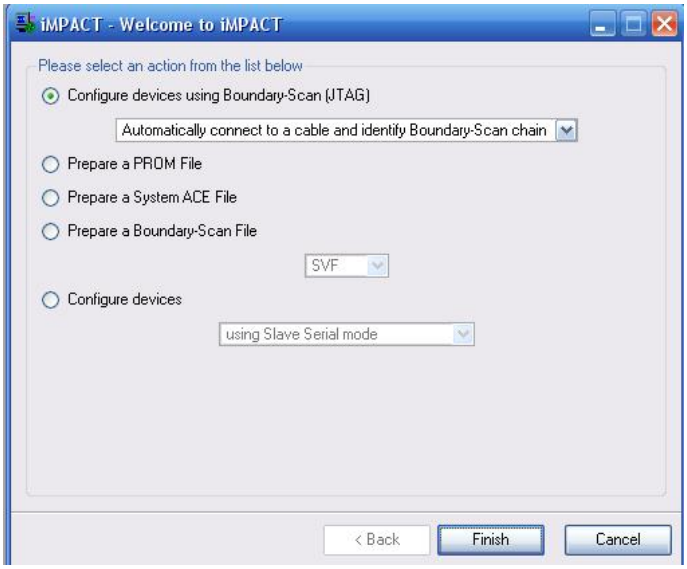
ارتباط با FPGA از طریق نرم افزار Xilinx ISE صورت می پذیرد. برای آشنایی با این محیط، ابزارهای Xilinx و ابزارهای پیاده سازی، شبیه سازی و سنتز طرح به بخش پیوست ضمیمه ه مراجعه نمائید. در ادامه به نحوه محاسبه و مشاهده نتیجه پرداخته خواهد شد.

۵-۵-۱- نحوه پیکربندی FPGA

بعد از اینکه فایل نهایی برای برنامه ریزی FPGA آماده شد، می بایستی آنرا از طریق درگاه خاصی پیکربندی نمود. با اجرای به ترتیب هر یک از برگه های Synthesize-XST، Implement Design و Generate Programming File و رفع خطاها و انجام تنظیمات مربوطه، برای برنامه ریزی و پیکربندی FPGA آماده می شویم. بدین منظور حالتی برای پیکربندی می بایست انتخاب گردد (اطلاعات بیشتر در بخش ۳-۳۱). در این طرح از ارتباط JTAG استفاده شده است. FPGA های امروزی مانند اغلب قطعات مدرن یک درگاه JTAG (Joint Test Action Group) دارند. درگاه JTAG در اصل برای پیاده سازی روش پویش مرزی (boundary scan) برای آزمایش بردهای مدار و آی سی ها طراحی شد که رسماً با IEEE ۱۱۴۹.۱ شناخته می شود.

FPGA چند پایه دارد که بصورت یک درگاه JTAG استفاده می شوند. یکی از این پایه ها برای ورود داده های JTAG و دیگری برای خروج آن داده استفاده می شود. ایده ی پویش مرزی این است که بوسیله ی درگاه JTAG می توان داده ها را همگام با ساعت به ثبات های JTAG مربوط به پایه های ورودی فرستاد و به FPGA این امکان را داد که روی داده ها عملیات انجام دهد. قابل توجه است که پین های پیکربندی را باید در حالت JTAG تنظیم نمود. برای کسب اطلاعات بیشتر در مورد پیکر بندی به بخش ۳-۳۱ مراجعه نمایید.

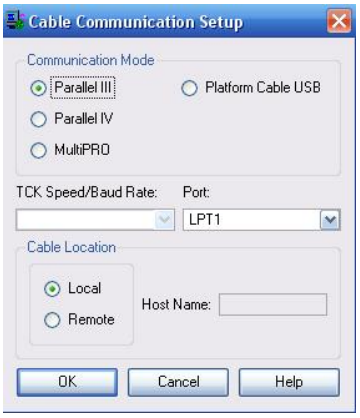
پیکربندی FPGA توسط نرم افزار Xilinx iMPACT صورت می پذیرد. با اجرای نرم افزار (قسمت انتهایی سنتز) دیالوگ زیر (شکل ۵-۲) نمایش داده می شود. که در آن می توان نوع برنامه ریزی قطعه و تنظیمات مربوط به آنرا انجام داد.



شکل ۵-۲: تنظیمات ورودی iMPACT

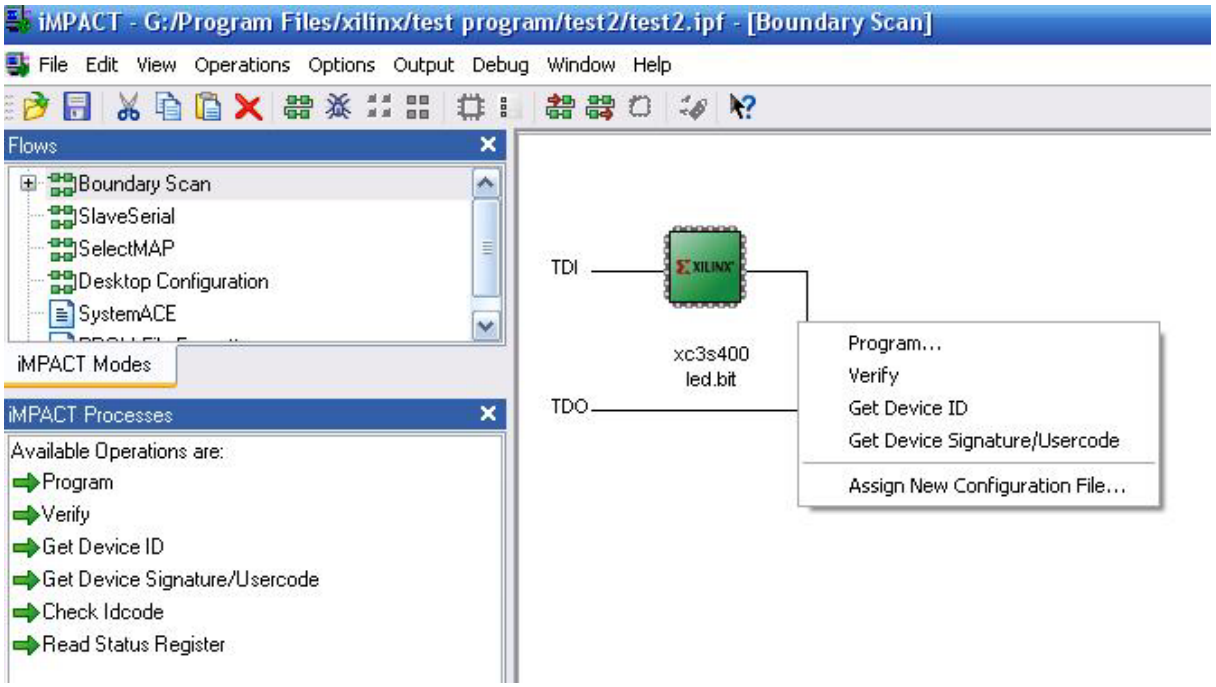
با ورود به این محیط تنظیماتی را می بایست انجام داد. FPGA ها برای برنامه ریزی نیاز به یک پروگرامری دارند که به کامپیوتر متصل شده تا آنها را برنامه ریزی نماید. پروگرامر های مختلفی در بازار و توسط شرکت های مختلفی ارائه شده که در این طرح از پروگرامر NFP۱۰۲ که بوسیله پورت موازی به کامپیوتر متصل می گردد استفاده شده است. خود این پروگرامر ها بسته به قطعه استفاده شده نیز دارای تنظیماتی هستند که توسط شرکت سازنده ارائه می شود.

تنظیم نوع پروگرامر: ادر نرم افزار Impact از منوی Output گزینه Cable Setup را انتخاب و در پنجره باز شده گزینه Parallel III را انتخاب می کنیم.

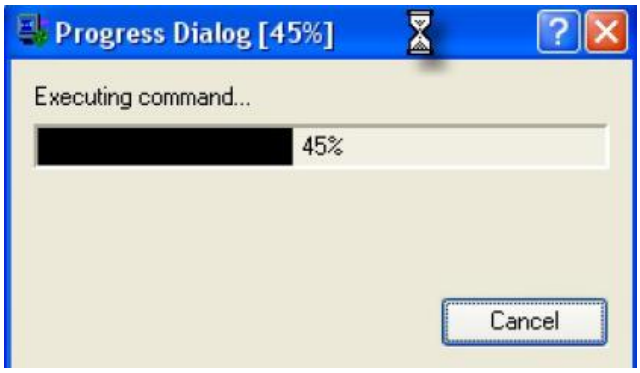


شکل ۵-۳: تنظیمات پروگرامر

در صورتی که ارتباط بین کامپیوتر و FPGA برقرار شود، پیغام Cable Connection Established نمایش داده می شود. با انجام تنظیمات پروگرامر و برقراری اتصالات مربوطه، در صورت صحت انجام مراحل ذکر شده، پنجره زیر (شکل ۵-۴) نمایش داده می شود که با راست کلیک کردن بر روی تراشه و انتخاب Program می توان تراشه را برنامه ریزی نمود. در هنگام پیکربندی، پنجره برنامه ریزی (شکل ۵-۹) را می توان مشاهده نمود.



شکل ۵-۴: پیکربندی تراشه

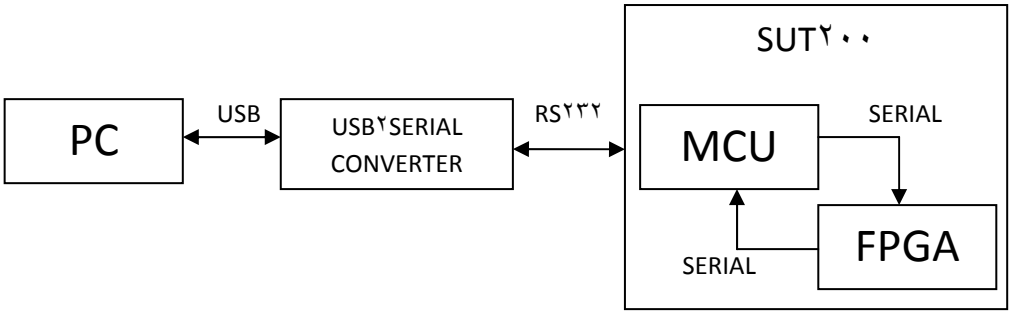


شکل ۵-۵: پنجره برنامه ریزی تراشه

در صورت درست انجام شدن مراحل برنامه ریزی پیغام Program Succeeded نمایش داده می شود.

۵-۶- نحوه ورود و خروج داده ها به سخت افزار

همانطور که قبلاً گفته شد، ارتباط با سخت افزار از طریق پورت سریال انجام می پذیرد. روند کلی ارتباط بین کامپیوتر و سخت افزار بصورت شکل ۵-۶ می باشد.



شکل ۵-۶: روند کلی ورود و خروج اطلاعات بین کامپیوتر و برد

بدین منظور یک تصویر ۱۵*۱۵ نمونه توسط نرم افزار MATLAB خوانده شده و با برقراری ارتباط با پورت سریال مد نظر، اطلاعات را با نرخ معینی و بصورت آسنکرون برای برد ارسال می شود. میکروکنترلر با دریافت داده ها و بصورت بایت به بایت، آنها را در بافری ذخیره می نماید. پس از تکمیل دریافت، میکروکنترلر آنها را برای پردازش در اختیار FPGA قرار می دهد. FPGA پس از دریافت کامل داده ها و پردازش بر روی آنها، نتیجه را در اختیار میکروکنترلر قرار داده و آن نیز داده های دریافت شده را در بافری دیگری ذخیره نموده و برای ارسال به کامپیوتر و نمایش آماده می نماید. در این مرحله بایستی یک همزمانی برای ارسال داده ها از میکروکنترلر و دریافت توسط کامپیوتر صورت پذیرد.

برای برقراری ارتباط FPGA با دنیای بیرون از خود چندین روش وجود دارد. می توان سخت افزاری کاملاً مستقل که مجهز به دوربین، مبدلهای A/D، D/A، نمایش دهنده ها و ... باشد، تهیه نمود و یا اینکه با قرار دادن برد FPGA بر روی کامپیوتر و استفاده از اسلاتها این ارتباط را برقرار کرد. روش بعدی، دریافت و ارسال اطلاعات توسط تراشه های جانبی نظیر پورت سریال، USB و یا SD کارت می باشد. در

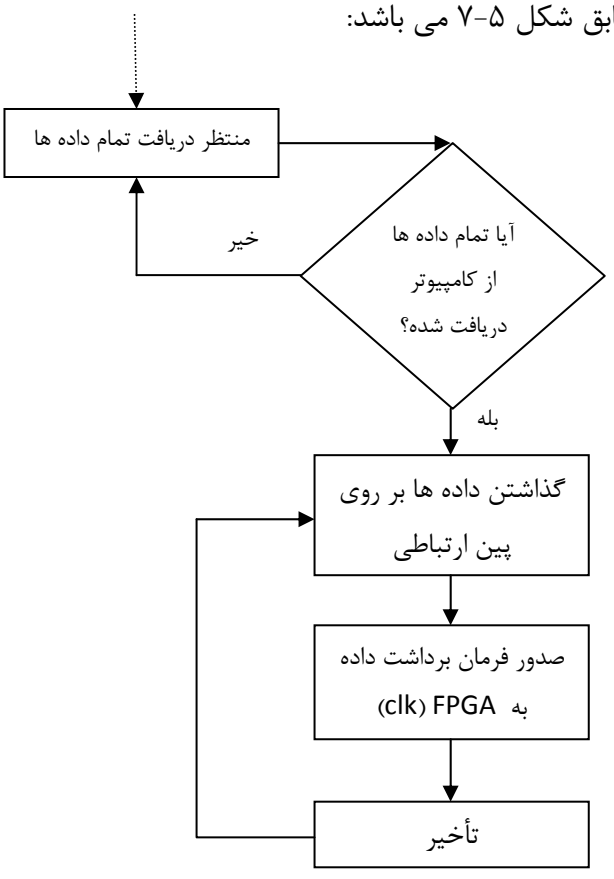
این پروژه بدلیل سادگی نوع سوم که ساده ترین حالت است انتخاب شده و ایده اصلی این است که سخت افزاری از نوع اول داشته و طراحی نماییم.

۵-۷- پیاده سازی و سنتز طرح

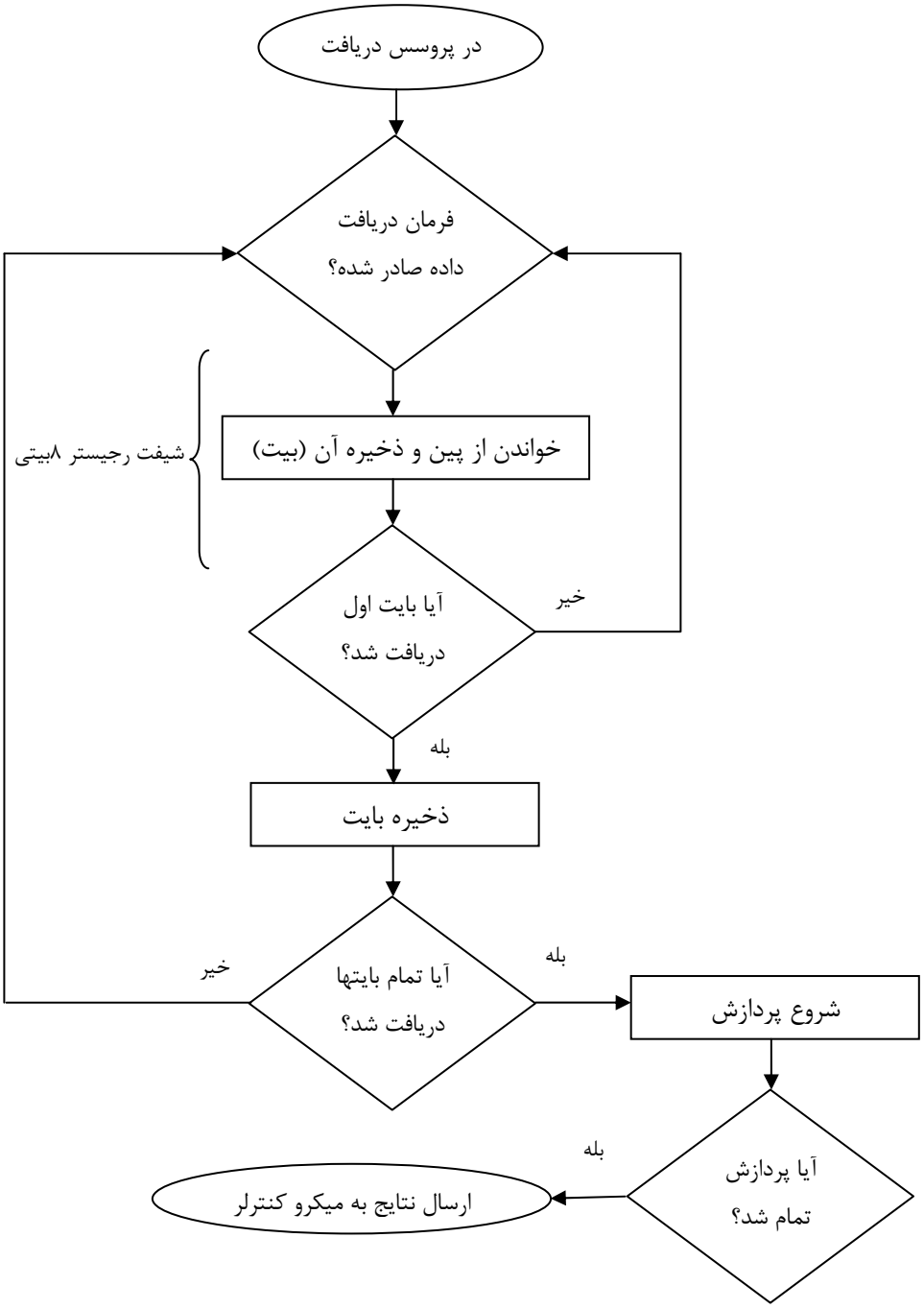
ایده انتقال داده بین میکرو کنترلر و FPGA همانند استفاده از ویژگی SPI میکروکنترلر است. در این ارتباط میکروکنترلر بعنوان Master بوده و FPGA بعنوان Slave عمل کرده، داده ها بر روی دو پین منتقل می شوند (همانند MISO و MOSI) و کلاک را میکروکنترلر تولید می کند. عملیات تکمیل ارسال و دریافت نیز با چک کردن وضعیت دو پرچم انجام می پذیرد. روند کلی بدین صورت است که میکروکنترلر داده ها را بصورت بیت به بیت در پین مربوطه قرار می دهد، فرمان برداشت داده از پین را به FPGA ارسال کرده (کلاک) و FPGA با دیدن لبه کلاک (بالا یا پایین رونده) داده را از روی باس برداشت می کند و آنها را در یک بافر بصورت شیفت رجیستر ذخیره می نماید. پس از دریافت کامل داده ها (۲۲۵ بایت) پردازش را بر روی آنها انجام داده و نتایج را در اختیار میکروکنترلر قرار می دهد.

روند نمای ارتباط بین میکروکنترلر و FPGA مطابق شکل ۵-۷ می باشد:

- ارسال توسط میکروکنترلر



- در یافت توسط FPGA



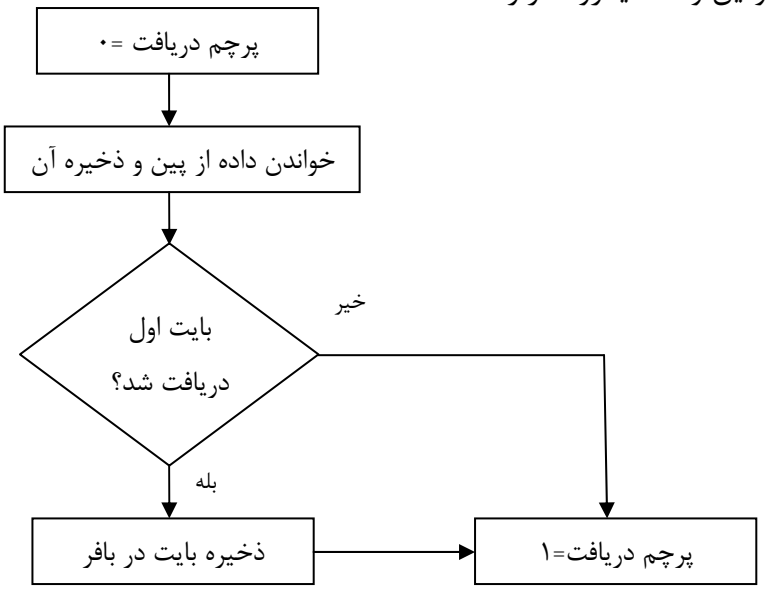
- ارسال توسط FPGA

در این حالت FPGA هنگام ارسال هر بیت به میکروکنترلر یک وقفه (هر تغییر وضعیتی) را به میکروکنترلر داده و میکروکنترلر در روتین وقفه داده ها را دریافت و ذخیره نموده و دریافت داده را در

پایان روتین به FPGA اعلام می نماید. FPGA با دریافت این علامت شروع به ارسال داده بعدی می نماید تا تمامی داده ها را (بایت به بایت) ارسال نماید.

- دریافت توسط میکروکنترلر

در روتین وقفه میکروکنترلر:



شکل۵-۷: روند نمای انتقال داده بین میکروکنترلر و FPGA

آنچه در این طرح و پروژه های مشابه از اهمیت خاصی برخوردار است، استفاده از سرعت و قدرت پردازنده های متنوع برای رسیدن سریعتر به جواب مطلوب می باشد. برنامه و الگوریتم بهینه شده، افزایش کلاک سیستم، استفاده از چندین پردازنده و تقسیم الگوریتم بین آنها از جمله مواردی است که طراح می تواند با انتخاب بهینه آنها اقدام به انجام دادن پروژه نماید. بعنوان مثال در این طرح با توجه به نوع تراشه انتخابی (XC۳S٤۰۰) بیشینه تا ۲۸۰ مگاهرتز می توان فرکانس کلاک ورودی را بالا برد. که در اینصورت مدار چاپی طراحی شده و مسائل دیگری از جمله برنامه پیاده سازی شده می بایستی امکان و اجازه این کار را بدهند. لازم بذکر است که کامپایلر در انتهای سنتز با توجه به برنامه نوشته شده و

محاسبه بیشینه تأخیر ممکنه، حداکثر فرکانسی را که می توان به FPGA اعمال کرد را نشان می دهد. در تراشه های سری جدید تر نظیر kintex-۷ فرکانس کلاک را تا ۶۴۰ مگاهرتز نیز می توان افزایش داد[۷]. البته با توجه به حلقه های قفل فاز درون تراشه این فرکانس را می توان چندین برابر نمود و سرعت نهایی پردازش را بالاتر نیز برد.

۵-۸- نتایج اعمال تصاویر

بعد از اینکه پردازش مورد نظر بر روی داده های ورودی صورت پذیرفت، نتایج حاصله به کامپیوتر برای نمایش ارسال می شود. نمونه ای از انجام عملیات تشخیص لبه در شکل ۵-۱۲ قابل مشاهده است.

تصویر ورودی: تصویر خروجی:

شکل ۵-۸: نتایج حاصل از اعمال الگوریتم تشخیص لبه بر روی تصویر ۱۵*۱۵

برای محاسبه سرعت پردازش و همچنین مقایسه سخت افزار با نرم افزار (کامپیوتر)، پیاده سازی نرم افزاری نیز توسط MATLAB و به زبان C صورت پذیرفته است. این مقایسه حاکی از آن است که پیاده سازی سخت افزاری سرعت بالاتری دارد، زیرا ریزپردازنده کامپیوتر به صورت ترتیبی عملیات پردازش را انجام می دهد در حالیکه سخت افزار به صورت موازی و همزمان اطلاعات را پردازش می کند و فرکانس ساعت کمتری را برای رسیدن به نتیجه نیاز خواهد داشت [۲۱]. جدول ۵-۱ نتایج حاصله از مقایسه سرعت پردازش بین سخت افزار و نرم افزار را نشان می دهد.

جدول ۵-۱: مقایسه سرعت نرم افزار و سخت افزار در برخی از الگوریتمهای پردازش تصویر

	Xilinx SpartanIII FPGA		Pentium IV	
	Freq [MHz]	Time~ [ms]	Freq [MHz]	Time~ [ms]
Edge Detection	۱۰	۴	۲۴۰۰	۱۲
Median Filter	۱۰	۲	۲۴۰۰	۷

البته این نکته حائز اهمیت است که زمان اجرای الگوریتم در نرم افزار به الگوریتم برنامه نویسی، قدرت پردازشگر و تعداد دفعات ارجاع به حافظه سیستم بستگی دارد و می توان با بهبود هریک از موارد فوق سرعت پردازش را افزایش داد. این موضوع در سخت افزار مورد نظر با افزایش فرکانس کلاک سیستم، تکنیک های موازی سازی و بهبود الگوریتم صادق است. در مورد دقت محاسبات نیز می بایستی اشاره نمود که در مورد تصویر مورد نظر، اعداد حاصله توسط نرم افزار و FPGA یکسان بوده و نتیجه مشابهی را حاصل کرده است.

نتایج تجربی حاکی از آن است که زمان کلی I/O بسیار بیشتر از زمان محاسبات بوده و در نتیجه ارسال داده بین حافظه کامپیوتر و سخت افزار و همچنین خواندن داده ها از داخل خود حافظه کامپیوتر به داخل حافظه نرم افزار، زمان بسیاری را به خود اختصاص می دهد که این زمان می بایستی از زمان اجرای الگوریتم حذف شود.

همانطور که اشاره شد برنامه مورد نظر با نرم افزار XilinxISE ۸.۱i سنتز شده است که پارامترهای مهم حاصله از آن در جدول ۵-۲ آمده است.

جدول ۵-۲: خلاصه سنتز طرح و ماژول های اصلی آن

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slices	۱۷۱	۳۵۸۴	۴٪
Number of Flip Flops	۲۱۸	۷۱۶۸	۳٪
Number of LUTs	۲۵۲	۷۱۶۸	۳٪
Number of IOBs	۱۸	۱۴۱	۱۲٪
Number of GCLKs	۱	۸	۱۲٪

خلاصه زمان بندی طرح نیز بصورت زیر خواهد بود:

Speed Grade: -٤

Minimum period: ٧.١٧٨ns (Maximum Frequency: ١٣٩.٣١٥MHz)

Minimum input arrival time before clock: ٣.٤٠٣ns

Maximum output required time after clock: ٧.١٦٥ns

Maximum combinational path delay: ٤.١٨٧ns

Delay: ٤.١٨٧ns (١.٣٧٢ns logic (٣٢.٨%), ٢.٨١٥ns route (٦٧.٢%))

Total memory usage is ١١٦٤٦٨ kilobytes

با توجه به گزارش فوق حداکثر فرکانس اعمالی به سیستم حدود ۱۴۰ مگاهرتز می بایست باشد.

فصل ششم

نتیجه گیری و پیشنهادات

۶-۱- مقدمه

در ابتدا جمع بندی و نتیجه گیری کلی در مورد فعالیت های انجام شده در این پروژه صورت می پذیرد و در ادامه در مورد پیشنهادات و فعالیت های آتی بحث خواهیم نمود.

۶-۲- نتیجه گیری

در این پروژه، هدف طراحی و ساخت یک برد سخت افزاری با قابلیت بالا جهت کاربردهای گوناگون و متنوع بوده است. کاربردهای مورد نظر این طرح شامل انواع سیستم های پردازشگر سیگنال آزمایشگاهی نظیر پیاده سازی انواع فیلترها و الگوریتم ها، سیستم های حفاظتی، سیستم های پردازش موازی، بکار گیری در رباتهای هوشمند، سیستم های مانیتورینگ، کاربردهای نظامی و ... می باشد.

سعی بر آن بوده است که از وسایل جانبی متنوعی برای افزایش کارآیی سیستم در سخت افزار استفاده شود و قابلیت اضافه شدن پردازنده های دیگر به سیستم جهت افزایش قدرت پردازش سیستم مد نظر باشد. قطعات اصلی استفاده شده در سخت افزار عبارتند از:

FPGA خانواده اسپارتان۳ ساخت شرکت Xilinx، میکروکنترلر Atmega۱۲۸، مبدل آنالوگ به دیجیتال ۱۲ بیتی با نرخ نمونه برداری ۱۰۰Ksps، مبدل دیجیتال به آنالوگ هشت بیتی، تراشه FT۲٤٥ برای برقراری ارتباط USB، SD CARD، ۸Mb SRAM و وسایل ارتباطی و نمایشی دیگری نظیر ارتباط سریال، LCD و... .

امروزه با کمک یک یا چند FPGA به همراه قطعات جانبی سخت افزار هایی ارائه می گردد که قادرند الگوریتمهایی با بار محاسباتی بالا را اجرا کنند. استفاده از FPGA ها با قابلیتهای فراوانی که در پیاده سازی الگوریتمهای پیچیده خصوصاً پردازش سیگنالهای تصویر دارند روز به روز در حال گسترش و توسعه است. پیاده سازی الگوریتم ها و روشهای بررسی شده در مقالات سالهای اخیر بر روی این پردازنده ها، قابلیت پردازش موازی و استفاده زیاد این تراشه ها در سیستمهای پردازش بلادرنگ همگی بر قدرت و قابلیت این تراشه ها اشاره دارند. لذا بهره گیری و استفاده از این پردازنده ها بسیار حائز اهمیت می باشد.

بدلیل اینکه طرح و پروژه مورد نظر بعد از شبیه سازی توسط کامپیوتر نهایتاً می بایستی در یک سخت افزار پیاده سازی شود و نمود عملی پیدا کند. چیزی که در این نوع محاسبات حائز اهمیت می باشد بهبود سرعت پردازش الگوریتم می باشد. محاسبات و پردازشهای یکسان با پردازنده های متفاوت، مقیاس مناسبی جهت بررسی کارآیی و سرعت پردازش می باشد. بررسی مقالات اخیر در این مورد شاهد برتری چنین سیستم هایی است [۱۷].

تجربیات عملی حاکی از آن است که پیاده سازی سخت افزاری سرعت بالاتری دارد، زیرا ریزپردازنده کامپیوتر به صورت ترتیبی عملیات پردازش را انجام می دهد در حالیکه سخت افزار به صورت موازی و همزمان اطلاعات را پردازش می کند و فرکانس ساعت کمتری را برای رسیدن به نتیجه نیاز خواهد داشت. مقالات و پژوهشهای اخیر در زمینه های مشابه نیز مؤید مطلب فوق الذکر است [۲۱].

همانطور که اشاره شد، زمان اجرای الگوریتم در نرم افزار به الگوریتم برنامه نویسی، قدرت پردازشگر و تعداد دفعات ارجاع به حافظه سیستم بستگی دارد و می توان با بهبود هریک از موارد فوق سرعت پردازش را افزایش داد. این موضوع در سخت افزار مورد نظر با افزایش فرکانس کلاک سیستم، تکنیک های موازی سازی و بهبود الگوریتم صادق است.

در این طرح نیز برای تست و آزمایش سخت افزار چند برنامه نمونه نظیر چند برنامه محاسباتی به زبان VHDL، فیلتر دیجیتال، برنامه های تست ارتباطی و... بر روی آن پیاده سازی شده است. برای پیاده سازی تشخیص لبه یک تصویر ساده با ابعاد کوچک، بعنوان ورودی در نظر گرفته شده است و پردازش توسط FPGA صورت می پذیرد. بدلیل قابلیت های این پردازنده در اجرای محاسبات، مقایسه ای با اجرای برنامه توسط نرم افزار صورت پذیرفته و نتایج منتجه حاکی از آن است که سرعت پردازش در چنین سخت افزارهایی بسیار سریعتر از نرم افزار می باشد. بدیهی است که با افزایش کلاک سیستم، بهبود الگوریتم و استفاده از تکنیک های پردازش موازی می تواند ما را در رسیدن به این مهم کمک نماید. بدلیل طراحی سخت افزار و مدار چاپی با مشکلات و مسائل مرتبط با آن مواجه هستیم که کار را کمی با مشکل روبرو می کند. آنچه که در این امر حائز اهمیت می باشد، آشنایی با قطعات و چگونگی

استفاده از آنها می باشد. راه اندازی سخت افزار و برقراری ارتباط با FPGA و پیاده سازی طرحی بر روی آن، کار با مدارات واسط نظیر USB، سریال، ارتباط یک سخت افزار جانبی با کامپیوتر و کار با نرم افزار های مرتبط با طراحی سخت افزار و قطعات مورد استفاده، از جمله دستاوردهای این پروژه می باشد. همانگونه که اشاره شد، در ابتدای امر یک بررسی جامعی بر روی سخت افزارهای مشابه موجود در بازار و برد پیشنهادی Xilinx صورت پذیرفت تا برد مورد نظر قابلیتهای سخت افزارهای مشابه را داشته و در برخی از موارد از آنها برتر نیز باشد. واسط بودن با کامپیوتر نیز باعث افزایش کارآیی سیستم خواهد شد بگونه ای که جهت پایش وضعیت سخت افزار، بررسی خطاها و یک سیستم جانبی می تواند مورد استفاده قرار گیرد. بدین ترتیب سخت افزار مورد نظر می تواند در پیاده سازی مسائل مختلف مورد استفاده قرار گیرد. بعنوان مثال برخی از کاربردهای عملی و درحال استفاده سخت افزار در پروژه های صنعتی و تجاری عبارتند از:

- مبدل های A/D,D/A (SDI۲PAL) سیگنال تصویر دیجیتال
- انواع کارتهای Distributer سیگنالهای دیجیتال
- انواع مدولاتور/دمدولاتور ها
- سیستم های پردازش رادار و سونار
- استفاده در سوئیچ های سرعت بالا (Gigabit Ethernet) و ...

۶-۳- پیشنهادات و تحقیقات آتی

همانگونه که اشاره شد طراحی یک سخت افزار با قابلیت های بالا از جمله اهداف این پروژه می باشد. به منظور افزایش کارآیی سیستم می توان از دو یا چند پردازنده FPGA در کنار یکدیگر بعنوان هسته محاسباتی استفاده نمود بگونه ای که این پردازنده ها با تقسیم محاسبات فی ما بین یکدیگر باعث افزایش سرعت پردازش گردند. به عنوان مثال استفاده از معماری آرایه های سیستولیک[۱] دو بعدی یکپارچه برای انجام عملیات پردازش سیگنال از جمله موضوعهای مطرح شده این مبحث است. بدلیل قابلیت همزمان

۱ - systolic

اجرا کردن محاسبات توسط FPGA (Concurrent) و ترکیب شدن آن با قطعات دیگر، می توان به یک سخت افزار قوی برای اجرا و انجام محاسبات مربوطه دست یافت. ایده ای که در ساخت و طراحی ابر رایانه ها و سیستم های پردازش موازی از آن استفاده می شود. نحوه تقسیم محاسبات بین دو پردازنده و ارتباط آنها با یکدیگر از جمله مسائلی است که باید به آنها توجه نمود. این موضوع و مباحث مرتبط با آن در FPGA-Based Computers بررسی می شود [۲۲]. در برخی از دانشگاهها (نظیر خواجه نصیر الدین طوسی) نیز تحقیقات و اقداماتی برای استفاده از این پردازنده ها و ترکیب آنها (پردازش موازی) برای انجام محاسبات خاص صورت پذیرفته است. اتصال مستقیم دوربین و LCD به سخت افزار، پیاده سازی کامل تشخیص لبه و انجام الگوریتم های دیگری همانند پیاده سازی و طراحی انواع فیلترهای دیجیتال FIR,IIR، از فعالیت های آتی می تواند باشد. امکان دریافت تصاویر با رزولوشن متغیر برای FPGA نیز از جمله مواردی است که بر قابلیت های این سیستم خواهد افزود.

از جمله مباحثی که در سخت افزار ها مورد توجه می گیرد تغذیه روی برد، داشتن وسایل ارتباطی متنوع با دنیای بیرون از برد و داشتن اکثر ابزارهای کاری با سخت افزار می باشد. لذا برای افزایش قابلیت سیستم می توان انواع دیگر پورت ها نظیر کنترلر ارتباط شبکه ای (تراشه های Ethernet Controller)، ارتباط VGA، PS۲ و داشتن تعداد بیشتری از پورت های پر کاربرد نظیر USB را به سیستم اضافه کرد.

بمنظور بکارگیری سخت افزار در طرح هایی مانند روبات های فوتبالیست، افزودن ماژول های مخابراتی و ارتباطی از راه دور، اتصال دوربین با خروجی دیجیتال به FPGA و مدارات واسط دیگر نیز می تواند مفید واقع شود. برای طراحی و ساخت سیستم های حرفه ای پردازش سیگنال، با بررسی های انجام شده در این طرح مشخص شده است که اکثر این سیستم ها بدلایل گوناگون بصورت مدارهای ASIC طراحی می شوند. از جمله این دلایل مجزا بودن از یک سیستم عامل خاص و در نتیجه ایمن و امنیت بالا و پایداری این سیستم ها است.

خلاصه پیشنهادات در این بخش شامل موارد زیر می باشد:

- توسعه سخت افزار با استفاده از پردازنده های قوی تر و قطعات جانبی بیشتر

- پردازش بلادرنگ با استفاده از سخت افزار طراحی شده
- پیاده سازی موازی الگوریتم ها و مقایسه عملکرد با سیستم های پردازشی دیگر
- مطالعه، بررسی و استفاده از FPGA در کنار پردازندهای DSP بمنظور بهبود کارآیی سیستم

مراجع

[۱] Adrien E. Desjardins, Real-Time FPGA Processing for High-Speed Optical Frequency Domain Imaging, IEEE Transactions on Medical Imaging, September 2009

[2] Alfredo Gardel Vicente, Embedded Vision Modules for Tracking and Counting People, IEEE Transactions on Instrumentation and Measurement, September 2009

[3] Sungchan Oh and Gyeonghwan Kim, FPGA-based Fast Image Warping with Data-parallelization Schemes, IEEE Transactions on Consumer Electronics, November 2008

[4] Dave Landis, Programmable Logic and Application Specific Integrated Circuits, Pub2000

[5] Stephen Trimberger, Field Programmable Gate Array Technology, Kulwer academic Pub.1994

[6] Beauchamp, Underwood. Architechtal Modification to enhance in floating point performance of FPGA, IEEE Transaction ,Feb. 2008

[7] www.xilinx.com

[8] www.wikipedia.com

[9] www.top500.org

[10] Andrej Trost Baldomir Zajc, Design of Real-Time Edge Detection Circuits on FPGA, International Conference on Electrical and Electronics Engineering, 2002

[11] شکاری زاده، فرزاد (۱۳۸۹)**"مرجع کامل طراحی با FPGA"** چاپ اول، انتشارات نص، ص ۲۱۰

[12] کلیو مکس فیلد، (۱۳۸۸)**"مفاهیم و مبانی FPGA"** مترجم علی مالکی، چاپ اول، انتشارات ناقوس

[13] سید رضی حسن (۱۳۹۰) ، **"طراحی مدارهای دیجیتال با FPGA"** چاپ سوم، انتشارات ناقوس

[14] گونزالز سی رافائل، (۱۳۸۷) **"پردازش تصویر دیجیتال"** مترجم جعفر نژاد قمی، چاپ اول، انتشارات علوم رایانه

[15] امیر ره افروز (۱۳۸۵) "میکروکنترلرهای AVR و کاربردهای آنها" ،انتشارات نص

[16] J. Maddocks & R. Williams, VHDL Image Processing Source Modules Reference Manual, Hunt Engineering, 2006

[17] Muthukumar Venkatesan and Daggu Venkateshwar Rao, Hardware Acceleration of Edge Detection Algorithm on FPGAs, Department of Electrical and Computer Engineering, University of Nevada Las Vegas.

[18] Tom Chen, Real-time high performance Edge detector for computer vision applications Proceedings of ASP-DAC '1997

[19] I. Pitas, Parallel algorithms: for digital image processing, computer vision and neural networks, Johan Wiley& Sons, 1993

[20] D.L Perry, "VHDL: Programming by example", McGraw hill, 2002

[21] Daggu Venkateshwar Rao, Shruti Patil: Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture, International Journal of Theoretical and Applied Computer Sciences, Department of Electrical and Computer Engineering University of Nevada Las Vegas. (2006)

[22] CHUAN HE, FPGA-Based Computers (Numerical Solutions of Differential Equations on FPGA-Enhanced Computers), Shandong University, May 2007

[23] Roger Woods, John McAllister, FPGA-based Implementation of Signal Processing Systems, John Wiley and Sons, Ltd., Publication 2008

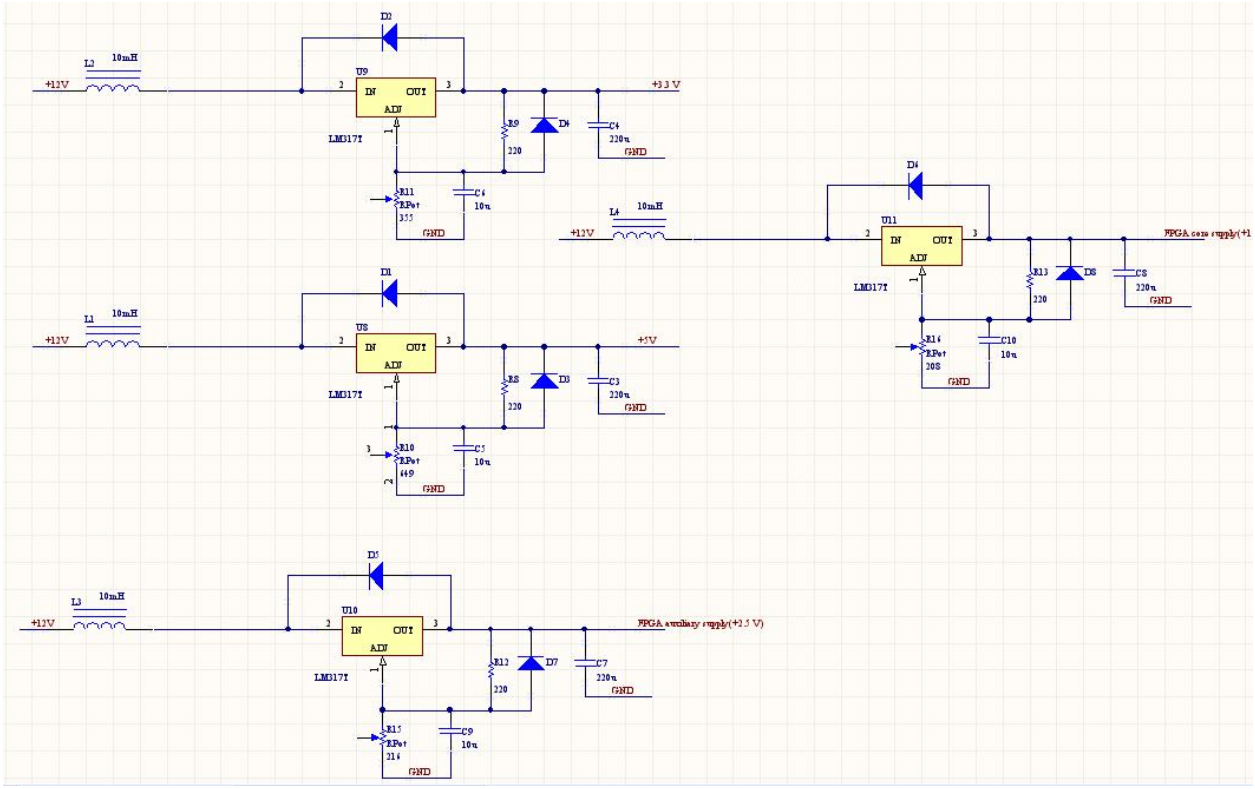
[24] Peter J. Ashenden, "the VHDL Cookbook" Dept. Computer Science University of Adelaide South Australia, July 1990

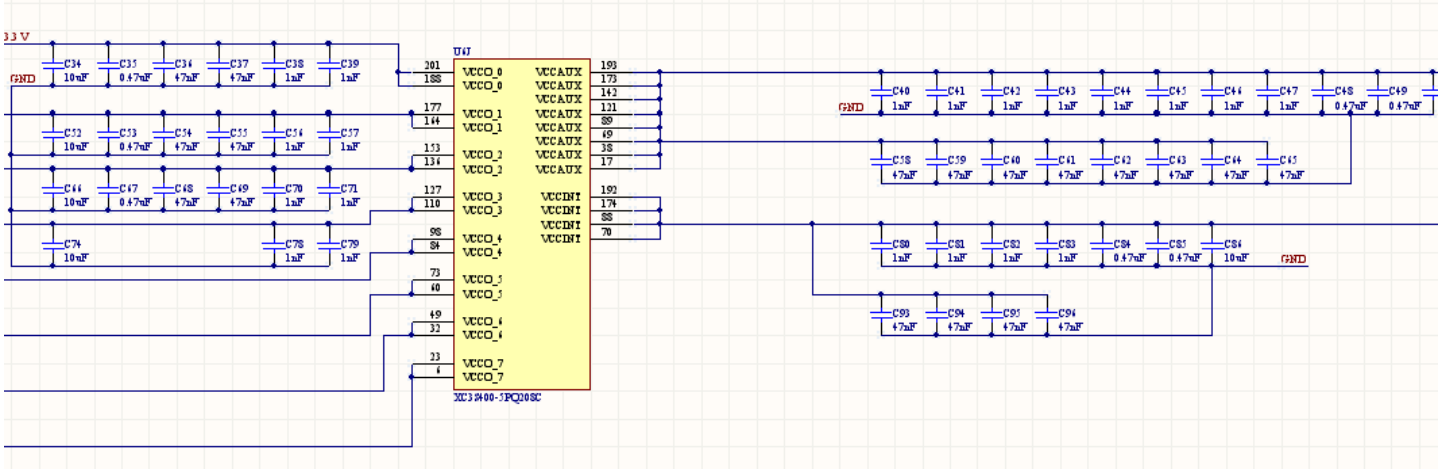
[25] VHDL Tutorials, VHDL and FPGA Resources on the Web, 2010, http://equipe.nce.ufrj.br/gabriel/vhdlfpga.html

پیوست

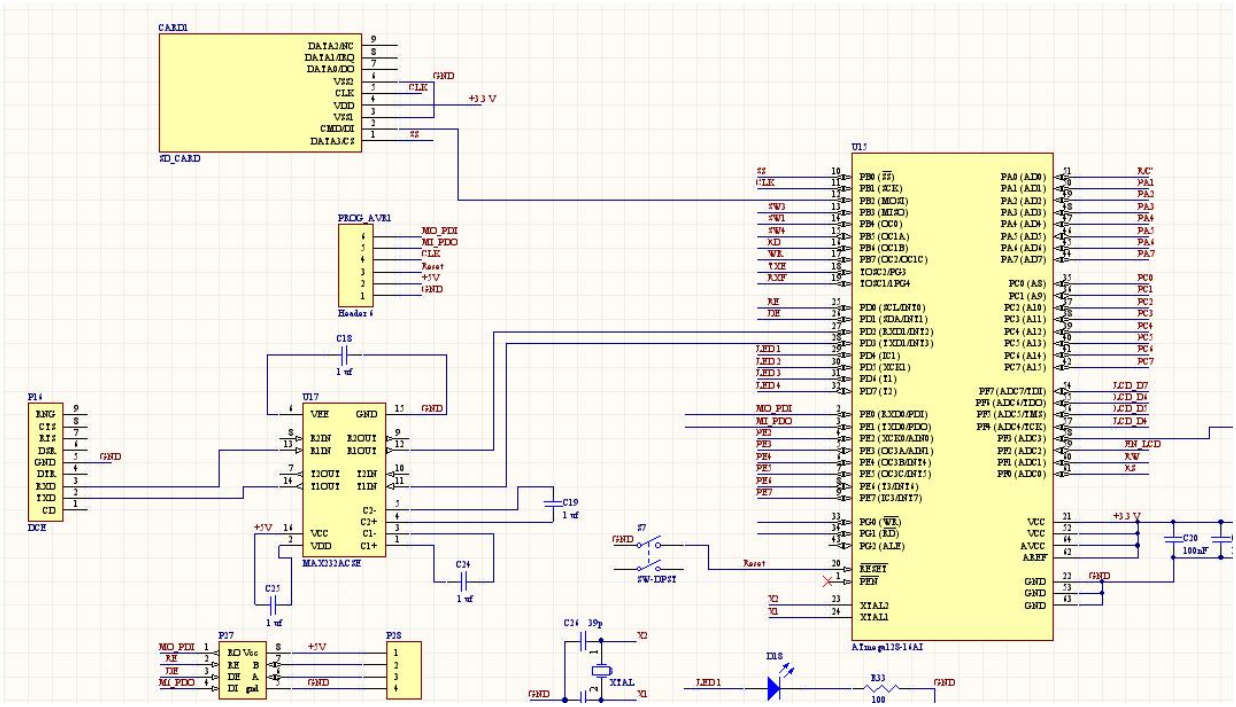
ضمیمه الف- شماتیک برد

- تغذیه مدار- ۱۲+ ولت(DC)

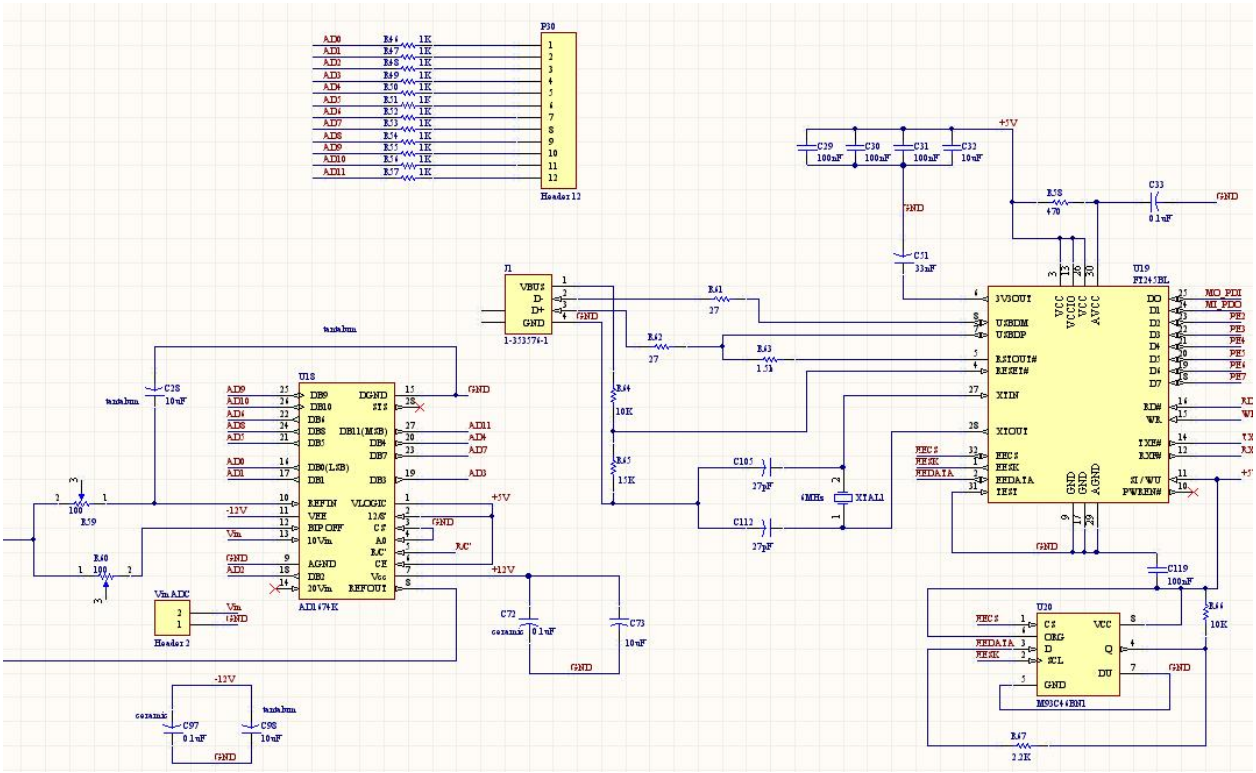




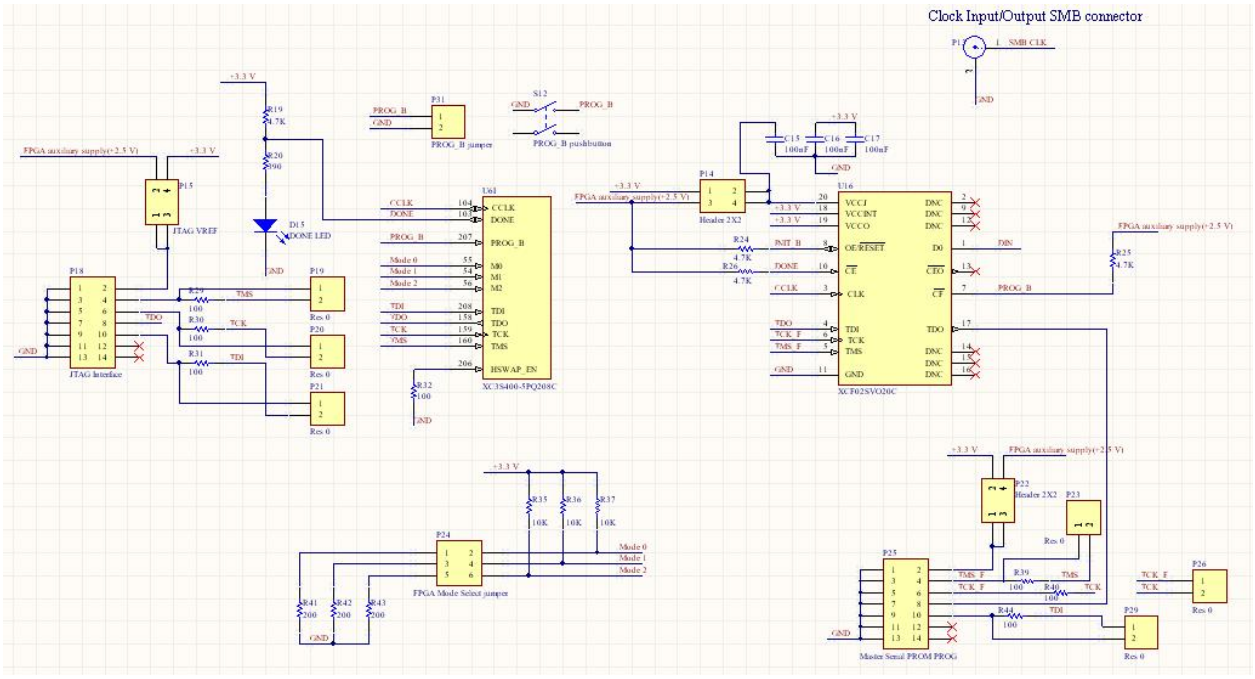
- میکروکنترلر بهمراه ارتباطات سریال و SD CARD



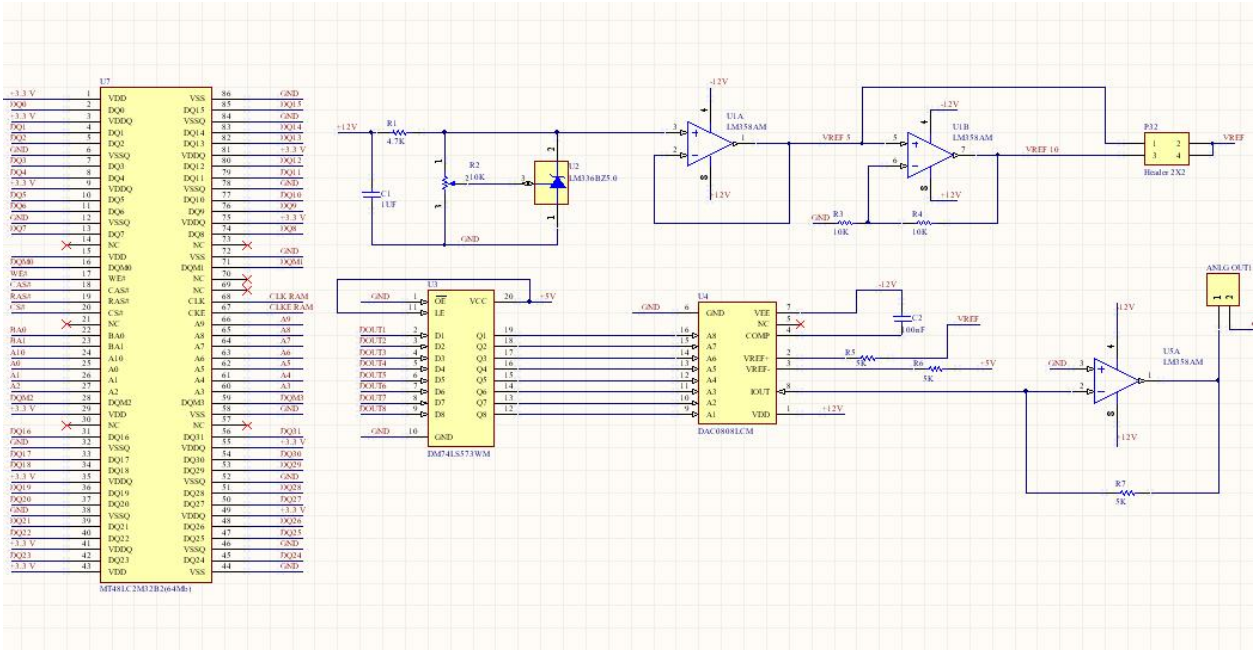
- مدار ADC بهمراه مبدل USP TO PARALLEL



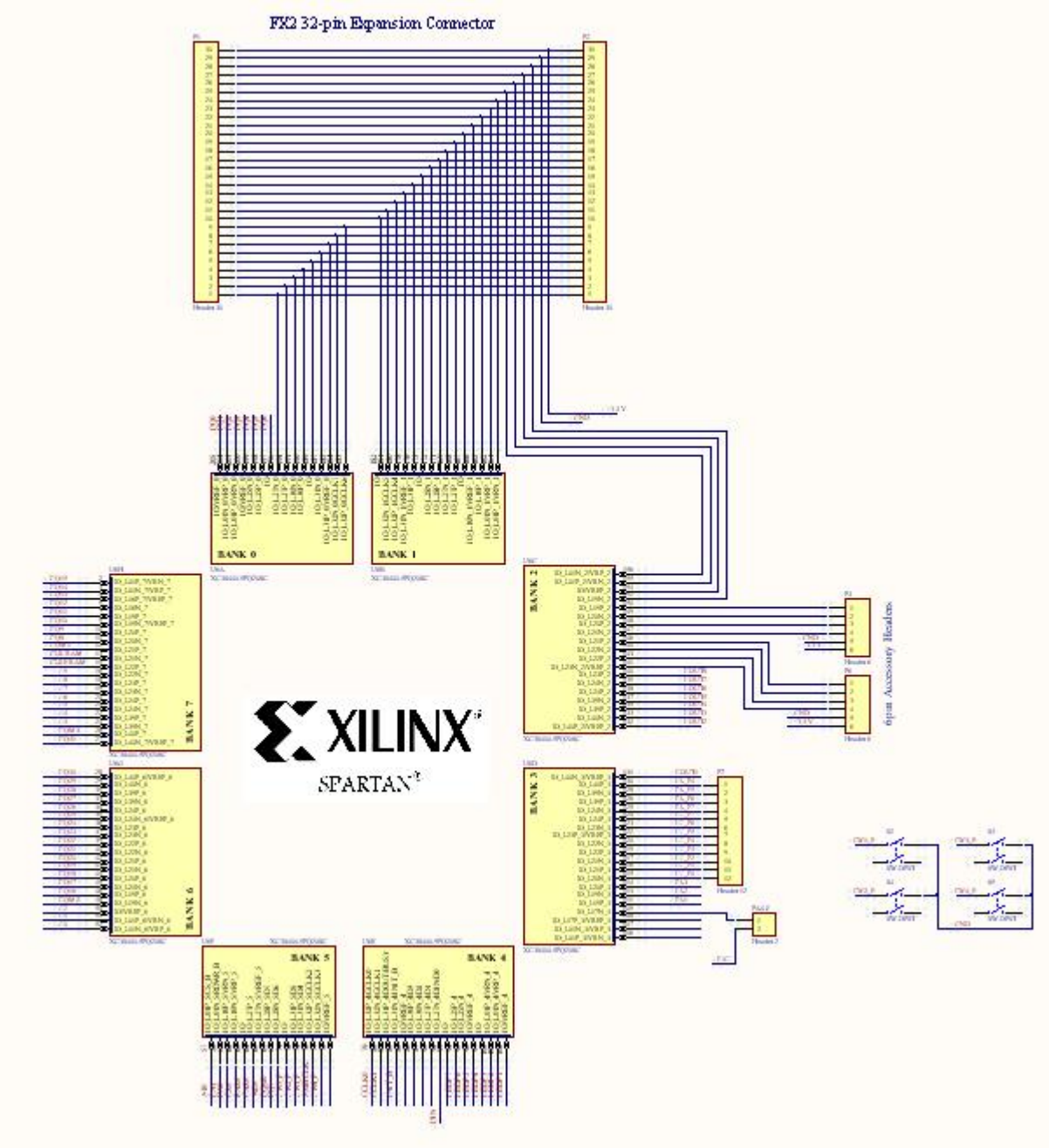
- مدار پیکر بندی FPGA



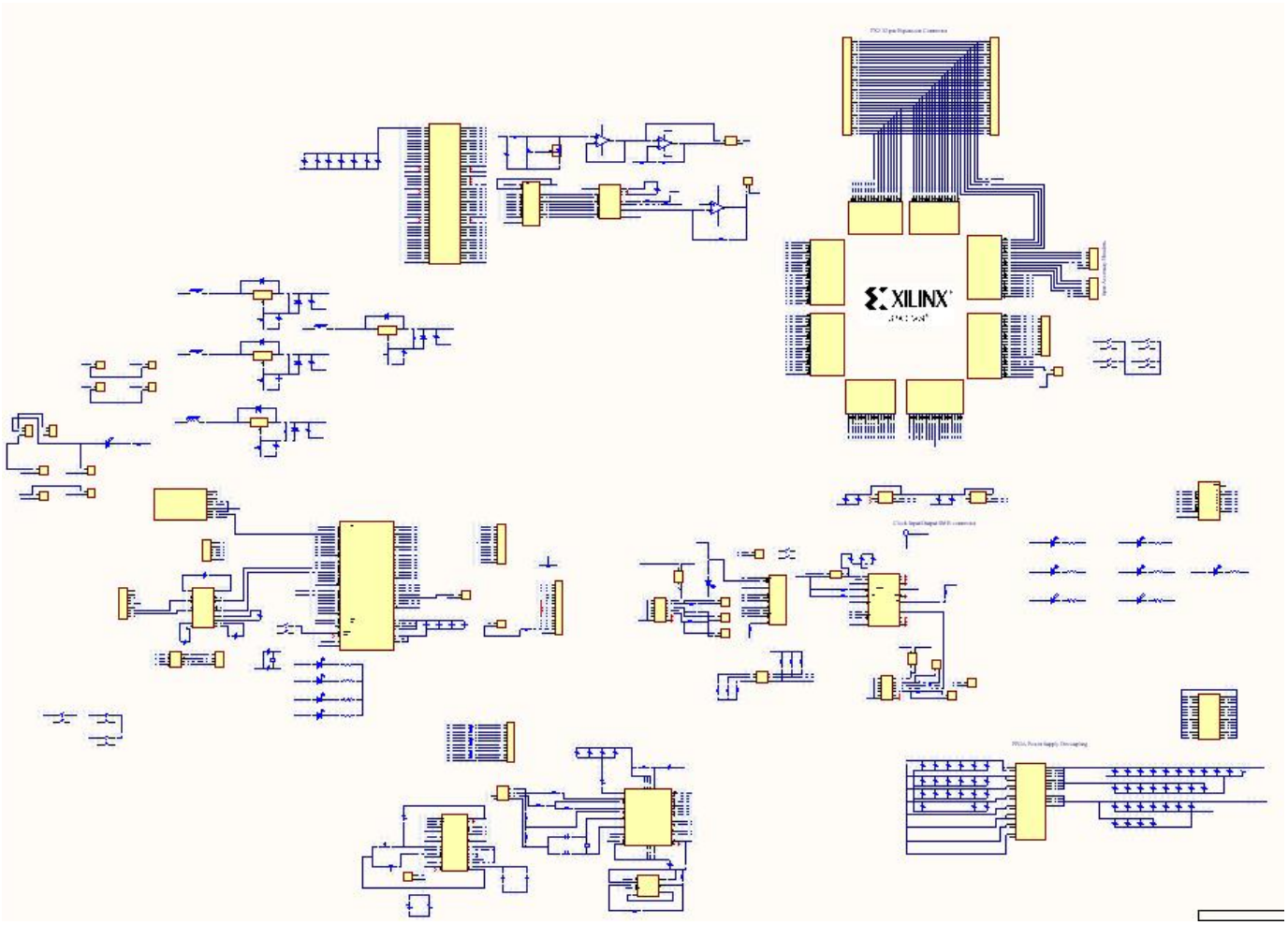
- مدار DAC بهمراه رم



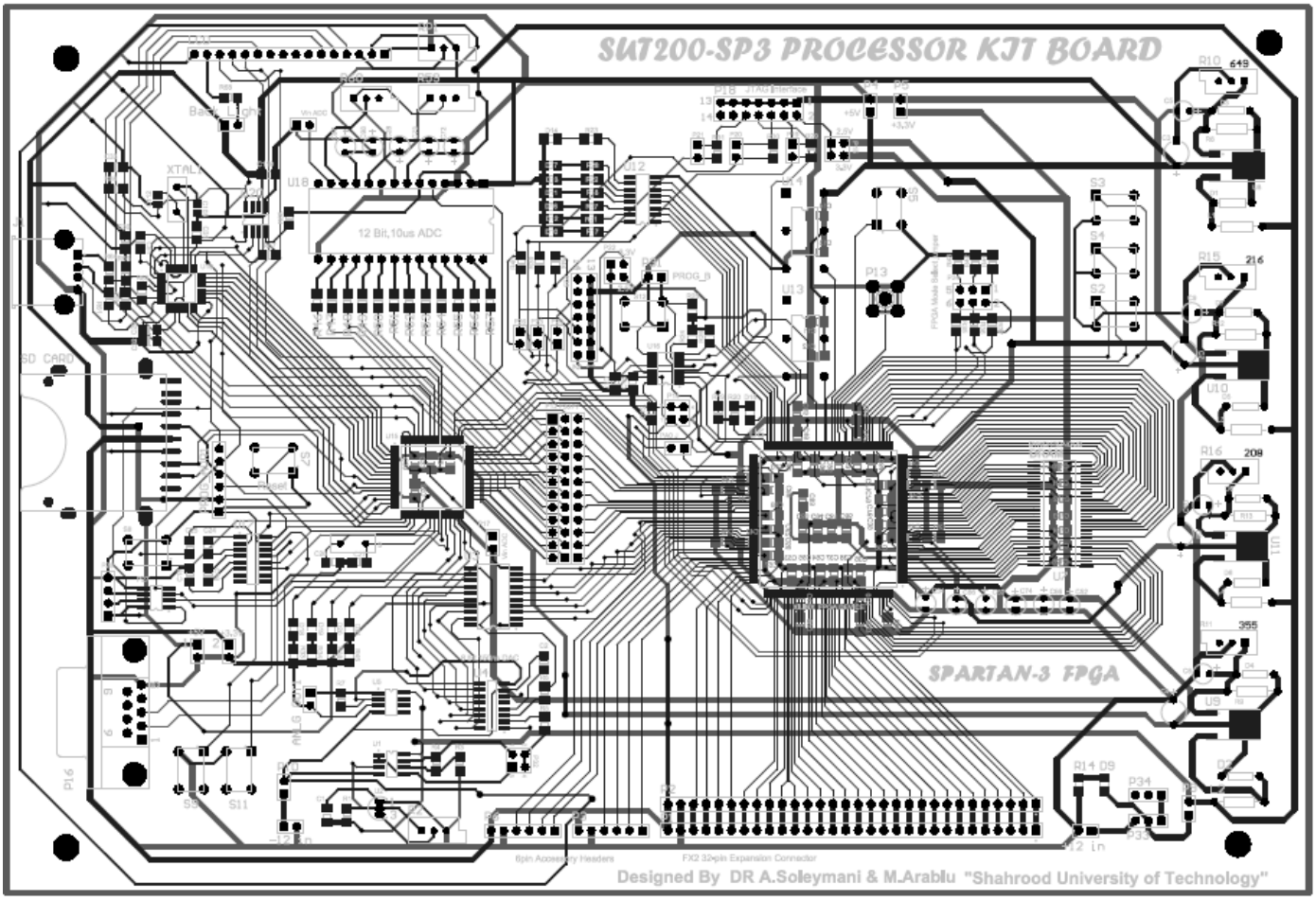
- ارتباطات FPGA



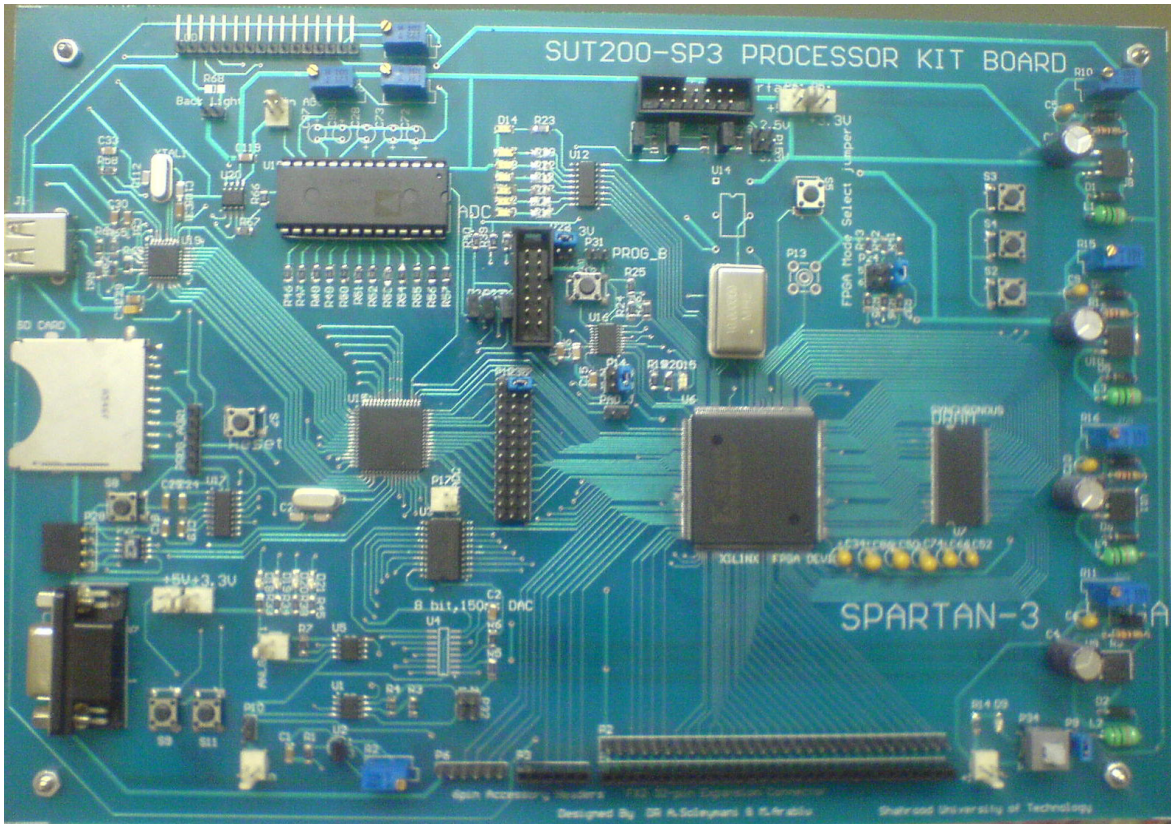
- نمای کلی سخت افزار

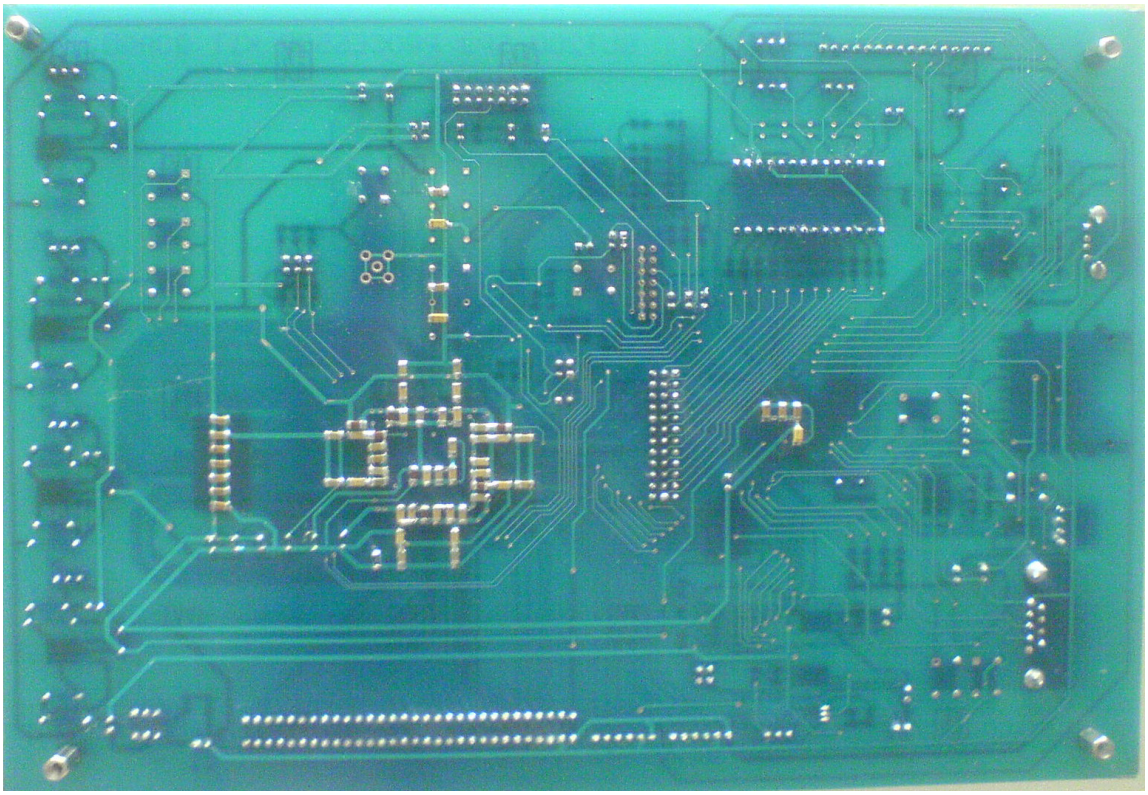


- **مدار چاپی سخت افزار**



- تصویر سخت افزار ساخته شده





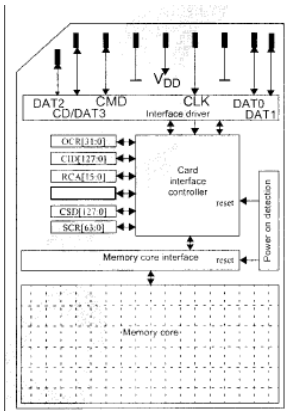
ضمیمه ب- بررسی SD CARD

مقدمه:

کارت های حافظه امروزه به یکی ازمتداول ترین وسایل برای جابه جایی اطاعات بین دستگاههای دیجیتال تبدیل شده اند. جابجائی حجم زیادی از اطلاعات، به همراه قابلیت ارتباط با پرتکل SPI، این کارت ها را به حافظه هائی ایده ال برای کار بامیکروکنترلرهای کوچک تبدیل کرده است.

استاندارد SD CARD توسط SD CARD ASSOCIATON طراحی گردید و توسط شرکت های SAN DISK،TOSHIBA وMEI مورد حمایت قرار گرفت و کامل تر شد، امروزه استاندارد SD CARD فقط محدود به کارت های حافظه نیست و در کلاس های مختلفی از دستگاهها ی دیجیتال به کار می رود که از جمله آنها می توان به بلوتوث و مودم ها اشاره نمود.

این کارت های حافظه دارای کنترلر داخلی می باشند که کلیه ی اعمال کنترلی (پاک کردن، خواندن، نوشتن و خطا یابی) را در داخل کارت انجام می دهند. به کمک رجیسترها می توان وضعیت فعلی کارت، مشخصه های ذاتی آن واطلاعات مفید دیگری راکسب کرد.

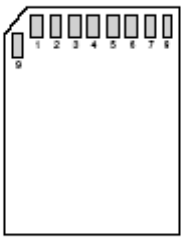


شکل ۱: ساختارداخلی SD CARD

سخت افزار SD CARD

ترتیب پایه ها و نحوه عملکرد آنها

در مجموع ۹ خط ارتباطی برای انتقال توان وداده دارد، سه خط برای انتقال توان و شش خط برای انتقال داده به کار می رود. ترتیب پایه ها وعملکرد آنها مطابق شکل ۲ می باشد:



Pin	Name	Function (SD Mode)	Function (SPI Mode)
1	DAT3/CS	Data Line 3	Chip Select/Slave Select ($\overline{SS}$)
2	CMD/DI	Command Line	Master Out Slave In (MOSI)
3	VSS1	Ground	Ground
4	VDD	Supply Voltage	Supply Voltage
5	CLK	Clock	Clock (SCK)
6	VSS2	Ground	Ground
7	DAT0/DO	Data Line 0	Master In Slave Out (MISO)
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

شکل ۲: پایه هایSD CARD در مدهای SPI و SD

رنج دقیق تغذیه باید از رجیستر OCR[۱] خوانده شود ، معمولاً اکثر کارت ها تغذیه بین ۲.۷ تا ۳.۶ ولت را می پذیرند، جریان کارت های حافظه می تواند تا چندین میلی آمپر باشد. لازم است کنترلر خارجی بتواند تا ۱۰۰ میلی آمپر را تأمین نماید.

جدول۱ : ولتاژ وجریان مصرفی کارت حافظه

۱ - Operating Conditions Register

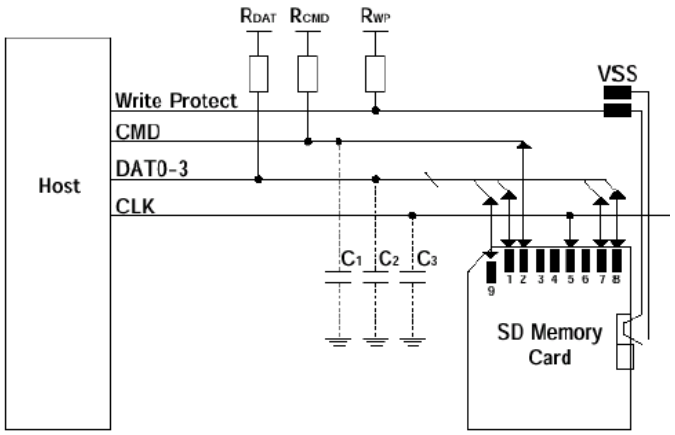
VDD (ripple: max, 60 mV peak to peak)	2.7 V – 3.6 V

(Ta = 25°C @3 V)

	Value	Measurement	Notes
Sleep	250	uA	Max
Read	50	mA	Typical
Write	75	mA	Typical

نحوه ارتباط SD CARD با میکروکنترلر

پین های دیتا را باید به پین های GIO[۲]میکروکنترلروصل نمود، درصورتی که میکرو حاوی واحد سخت افزاری SPI باشد، می توان از آن برای ارتباط با SD CARD استفاده نمود. خطوط داده باید با مقاومت pull up شوند.



شکل ۳: نحوه ارتباط SD CARD با میکروکنترلر

پروتکل های ارتباطی

۲ - General Input Output

SD CARD از دو پروتکل ارتباطی اصلی حمایت می کند SD PROTOCOL و SPI PROTOCOL. که به صورت زیر می باشد:

SD PROTOCOL:

SD ۱ BIT PROTOCOL:

یک پروتکل سریال سنکرون می باشد که شامل یک خط دیتا برای انتقال دیتای اصلی، یک خط کلاک برای سنکرون سازی و یک خط دستور برای ارسال فریم های دستور می باشد .

SD ٤ BIT PROTOCOL:

شبیه استادارد فوق می باشد، با این تفاوت که انتقال دیتا توسط ۴ خط به صورت موازی انجام می شود، با یک طراحی صحیح به کمک این شیوه می توان سرعت انتقال داده را ۴ برابر نمود.

هر دو پروتکل بالا نیازمند محاسبه CRC[۳] می باشند.

خطوط ارتباطی مد SD:

CMD: خط ارتباطی دوطرفه

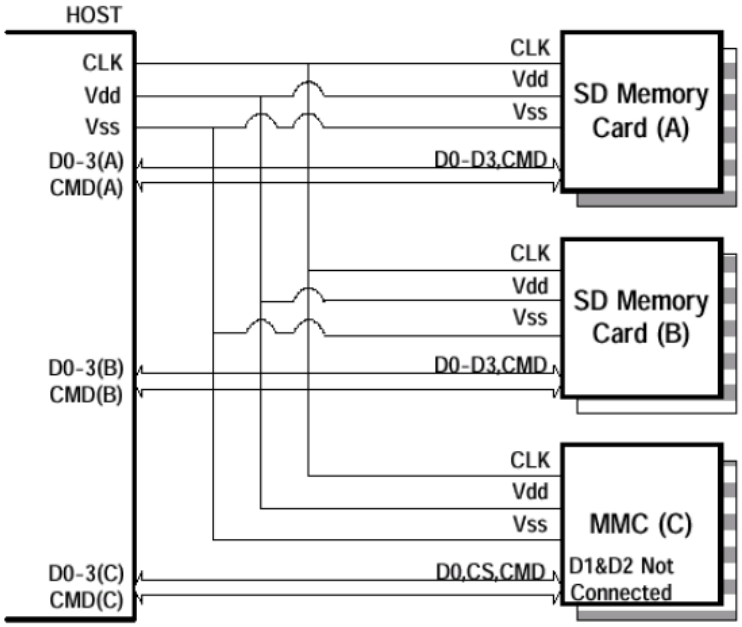
DAT۰-۳: خط ارتباطی دوطرفه

CLK: خط یک طرفه از کنترلر به کارت

این پروتکل از اشتراک باس دیتا حمایت می کند و می توان به کمک یک Master و یک خط کلاک چندین کارت حافظه را کنترل و انتقال داده را انجام داد .در هنگام Initialize دستورات به هر کارت جداگانه فرستاده می شود، سپس می توان به هر کارت آدرس جداگانه ای اختصاص داد و به وسیله آن با کارت ارتباط برقرار نمود. داده به هر کارت جداگانه فرستاده می شود ،اما می توان برای سادگی یک دستور را برای همه ی کارت ها به صورت همزمان فرستاد. اطلاعات مربوط به آدرس دهی در پاکت دستور قرار می گیرد .

۳ -Cycle Redundancy Code

پس از روشن شدن دستگاه SD CARD به صورت پیش فرض تنها از DAT۰ به عنوان خط داده استفاده می کند، بعد از Initialize کنترلر خارجی می تواند پهنای باس داده را تا چهار خط داده افزایش دهد، با استفاده از این روش می توان به سرعت انتقال صد مگا بیت داده در ثانیه بین حافظه و کنترلر خارجی دست پیدا کرد .



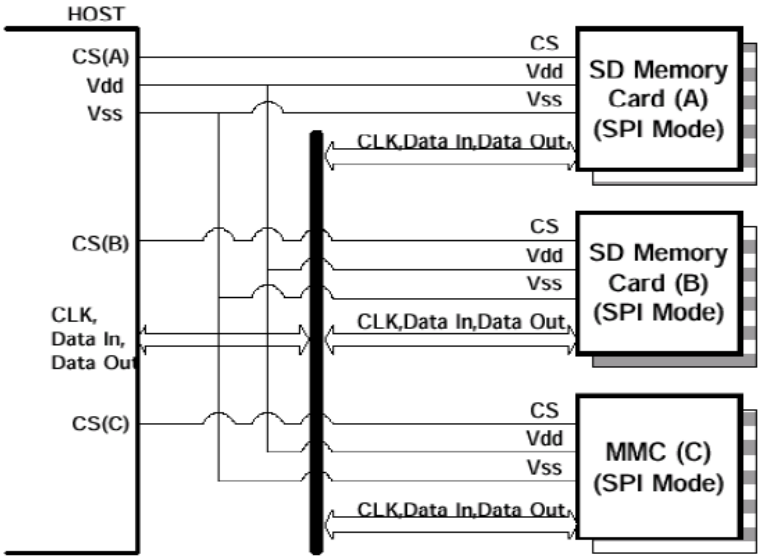
شکل ۴: پروتکل باس باقابلیت ارتباط باچندکارت حافظه باسرعت ۱۰۰ mbps

پروتکل ارتباطی SPI

این مد کاملا مجزا نسبت به مد قبلی می باشد. SPI یک پروتکل معروف ارتباطی سریال برای ارتباط وسایل جانبی با میکروکنترلرها می باشد. بخشی از پروتکل SD CARD توسط SPI حمایت می شود و عملکرد قابل قبولی از کارت ها را می توان به کمک ارتباط SPI انجام داد .

در این پروتکل تنها یک کانال (شامل دوخط) برای انتقال دیتا وجوددارد، می توان چندکارت حافظه را به این کانال وصل نمود و توسط یک خط انتخاب با هرکدام ارتباط داشت. در این حالت حداکثر سرعت بیست و پنج مگابیت در ثانیه است.

برای ارتباط در این مد می توان از واحدسخت افزاری SPI که دربرخی میکروکنترلرها وجوددارد، استفاده نمود.



شکل ۵: پروتکل SPI باقابلیت ارتباط باچندکارت حافظه باسرعت ۲۵ mbps

این مد های مختلف ارتباطی دست طراح را برای عملکرد های متفاوت باز می گذارد تا بر اساس نیاز طراحی مناسب را انجام دهد، اگر سرعت ارتباط از اهمیت زیادی برخوردار باشد از مد پارالل ۴ بیتی استفاده کند، یا بنابر نیاز مد SPI با سرعت کمتر را انتخاب نماید .

شرح پروتکل SPI

اساس ارتباط بر مبنای یک پروتکل دستور- پاسخ ساده می باشد، همه دستورات توسط Master ارسال می شود، SD با فریم پاسخ به آنها جواب می دهد، که این پاسخ با توجه به دستورات مختلف متفاوت می باشد.

شرح کامل دستورات SD CARD در ضمیمه د آورده شده است. شکل کلی دستورات به صورت CMDxx می باشد که xx بیانگر شماره دستور است. این دستورات همراه پاره ای اطلاعات دیگر به صورت یک فریم اطلاعاتی ۶ بایتی برای کارت فرستاده می شود.

ابتدا یک صفر به عنوان بیت شروع، یک به عنوان بیت فرستنده و سپس شماره دستور قرار می گیرد، سپس ۴ بایت به عنوان آرگومان دستور و در نهایت CRC و یک بیت بعنوان بیت خاتمه قرار می گیرد.

جدول ۲: فرمت فریم دستور SD CARD

First Byte			Bytes 2-5	Last Byte	
0	1	Command	Argument (MSB First)	CRC	1

جدول۳: برخی دستورات مهمSD CARD

Command	Argument	Type	Description
CMD0	None	R1	Tell the card to reset and enter its idle state.
CMD16	32-bit Block Length	R1	Select the block length.
CMD17	32-bit Block Address	R1	Read a single block.
CMD24	32-bit Block Address	R1	Write a single block.
CMD55	None	R1	Next command will be application-specific (ACMDXX).
CMD58	None	R3	Read OCR (Operating Conditions Register).
ACMD41	None	R1	Initialize the card.

فرمت پاسخ های SD CARD

سه نوع پاسخ R۱،R۲وR۳ وجود دارد که بنابر نوع دستور ارسالی یکی از آنها توسط کارت برای کنترلر خارجی ارسال می شود.

فرمت پاسخ R۱: پاسخ اکثر دستورات به این فرمت می باشد، اگر R۱ برابر صفر باشد یعنی دستور مورد نظر با موفقیت انجام شده است اگر نه بر اساس خطای رخ داده شده یکی از بیت های این پاسخ یک می شود.

جدول ۴ پاسخ نوع R۱

Byte	Bit	Meaning
1	7	Start Bit, Always 0
	6	Parameter Error
	5	Address Error
	4	Erase Sequence Error
	3	CRC Error
	2	Illegal Command
	1	Erase Reset
	0	In Idle State

فرمت پاسخ R۲:

این پاسخ ۲بایتی در جواب به دستور SEND_STATUS ارسال می شود، که به فرمت جدول ۵ می باشد:

جدول ۵:پاسخ نوع R۲

Byte	Bit	Meaning
1	7	Start Bit, Always 0
	6	Parameter Error
	5	Address Error
	4	Erase Sequence Error
	3	CRC Error
	2	Illegal Command
	1	Erase Reset
	0	In Idle State
2	7	Out of Range, CSD Overwrite
	6	Erase Parameter
	5	Write Protect Violation
	4	Card ECC Failed
	3	Card Controller Error
	2	Unspecified Error
	1	Write Protect Erase Skip, Lock/Unlock Failed
	0	Card Locked

فرمت پاسخ R۳:

این پاسخ ۵ بایتی در جواب به دستور Read_OCR ارسال می شود، بایت اول همان پاسخ R۱ و چهار بایت بعدی محتویات رجیستر OCR می باشد.

جدول ۶: پاسخ نوع R۳

Byte	Bit	Meaning
1	7	Start Bit, Always 0
	6	Parameter Error
	5	Address Error
	4	Erase Sequence Error
	3	CRC Error
	2	Illegal Command
	1	Erase Reset
	0	In Idle State
2-5	All	Operating Condition Register, MSB First

راه اندازی اولیه SD CARD

مراحل اولیه برای ورود به مد SPI

پس از آنکه تغذیه SD CARD وصل شود، کارت وارد مد SD می شود، برای انتقال آن به مد SPI باید مراحل زیر طی شود :

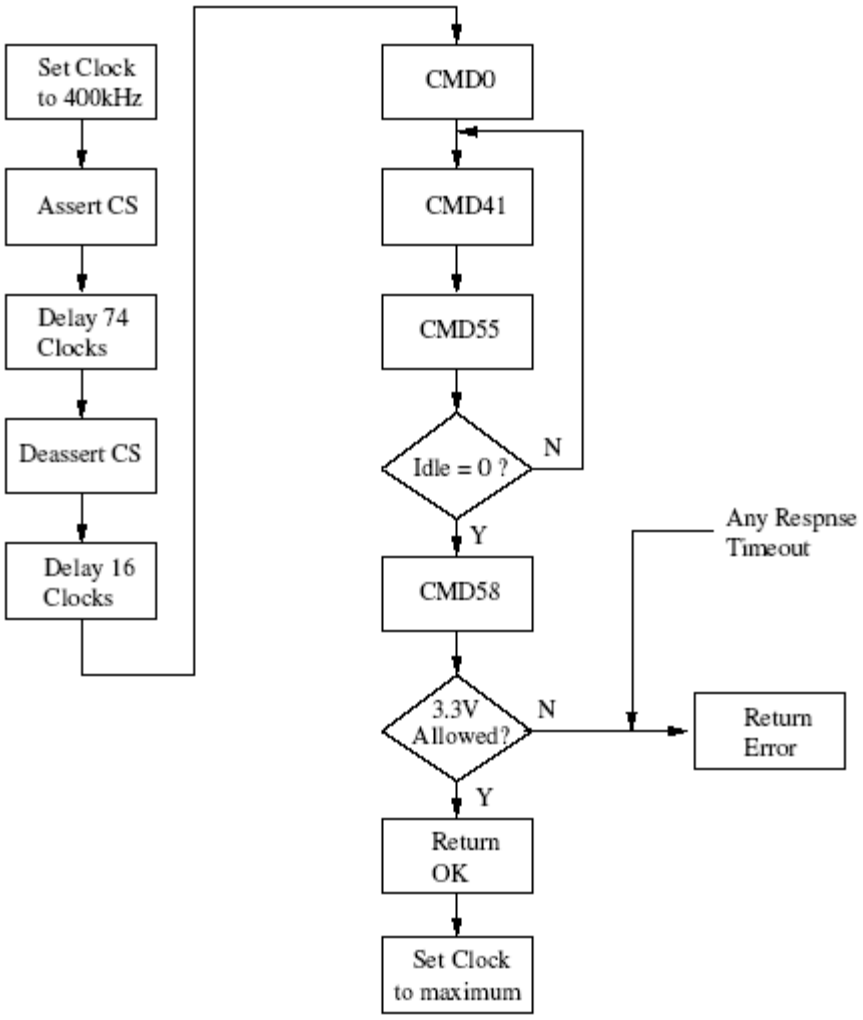
پس از روشن کردن دستگاه باید یک میلی ثانیه صبر شود، خطوط CSو DI را بالا نگه داشته و حداقل ۷۴ پالس به SCLK بزنیم، سپس CMD۰ را با CS به صورت LOW ارسال نماییم. در این حال SD CARD از خط CS نمونه برداری می کند و هنگامی که CMD۰ دریافت می شود اگر CS، LOW باشد ، کارت وارد مد SPI می شود. فقط در همین یک بار لازم است تا CRC با مقدار صحیح ارسال شود و پس از ورود به مد SPI ، CRC غیر فعال می شود و مقدار آن قابل صرف نظر کردن می باشد.

پس از آنکه CMD۰ پذیرفته می شود کارت وارد مد idle می شود و پاسخ R۱ را با مقدار ۱ ارسال می کند، CRC را می توان با CMD۵۹ دوباره فعال نمود.

هنگامی که کارت در مد idle می باشد، تنها CMD۰ ،CMD۱وCMD۵۸ پذیرفته می شود و مابقی دستورات توسط کارت reject می گردد. هنگامی که CMD۱ ارسال می شود، کارت شروع به Initialize کردن خود می کند، بسته به ظرفیت حافظه این کار ممکن است صد ها میلی ثانیه طول بکشد، در طول این مدت کنترلر خارجی باید مدام CMD۱ را بفرستد و پاسخ R۱ را چک کند، هر گاه این پاسخ از ۱ به ۰ تغییر کند، یعنی عمل Initialize تمام شده و نوشتن و خواندن از کارت حافظه با پروتکل SPI قابل انجام می باشد، در این هنگام می توان رجیستر های OCR و CSD[۴] برای اطلاع از مشخصه های SD CARD قرائت نمود.

روند نمای راه اندازی اولیه مطابق شکل ۶ است.

٤- Card Specific Data



شکل ۶: روندنمای راه اندازی

انتقال دیتا

در انتقال دیتا یک یا تعداد بیشتری بلوک داده بعد از پاسخ ارسال شده توسط کارت ، ارسال یا دریافت می شود. بلوک داده شامل Token ، Data Block و CRC می باشد.

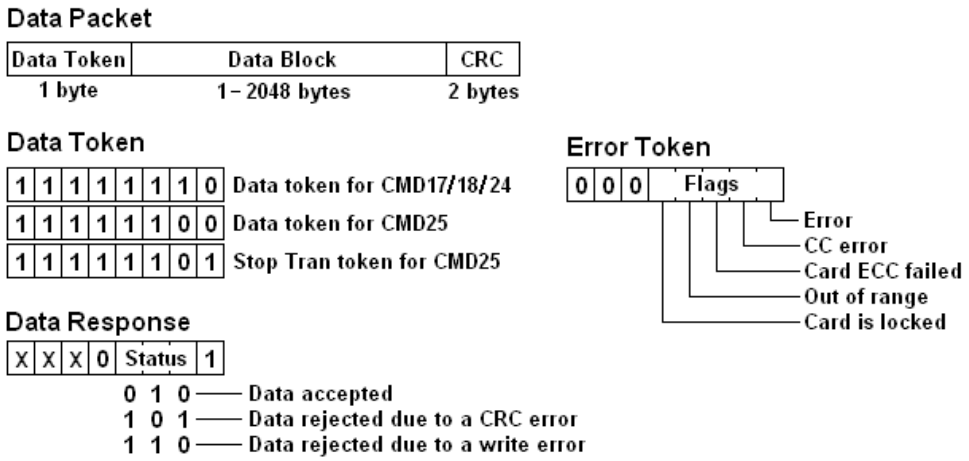
:Tokens

۱- Data Response Token هر بلوک داده که در کارت نوشته می شود با یک Data Response Token شناخته می شوند که توسط کارت ارسال می گردد، ۰۱۰ به معنای اینست که بلوک داده به درستی در داخل SD CARD نوشته شده است.

۲- Data Token: به منظور خواندن و نوشتن بلوک های داده بسته به نوع دستور یک Data Token در ابتدای پاکت داده قرار می گیرد.

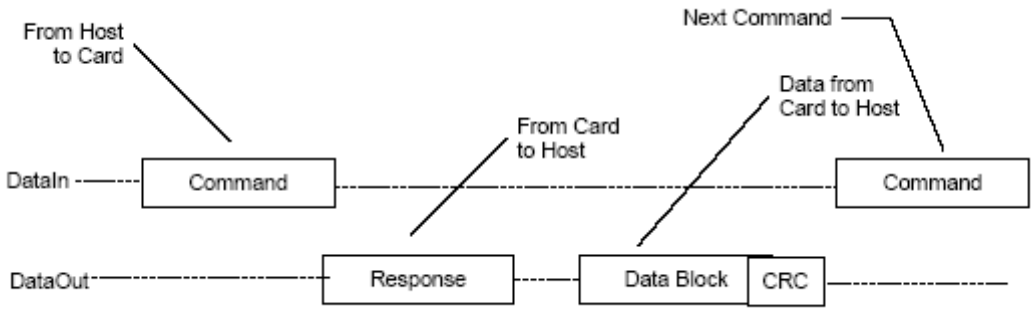
۳- Error Token: هرگاه SD CARD نتواند به درستی اطلاعات مورد نیاز برای خواندن را در اختیار کنترلر قرار دهد، این Token توسط SD CARD فرستاده می شود .

جدول ۷: فرمت پاکت داده



خواندن تک بلوکی

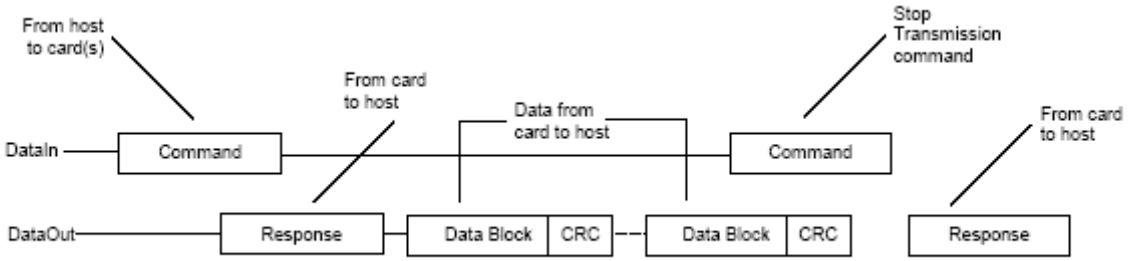
به کمک CMD۱۷ می توان یک بلوک (معمولاً ۵۱۲ بایت) دیتا را از آدرس مطلوب خواند، آدرس مورد نظر باید به عنوان آرگومان دستور و در واحد بایت قرار گیرد. پس از ارسال این دستور، کنترلر باید منتظر دریافت یک Data Token معتبر باشد، اطلاعات خوانده شده به دنبال Data Token توسط کارت فرستاده می شود، اگر هر خطایی در هنگام خواندن رخ دهد، به جای پاکت داده یک Error Token ارسال خواهد شد. به کمک CMD۱٦ می توان طول بلوک دیتا را تغییر داد.



شکل۷: خواندن تک بلوکی

خواندن چند بلوکی

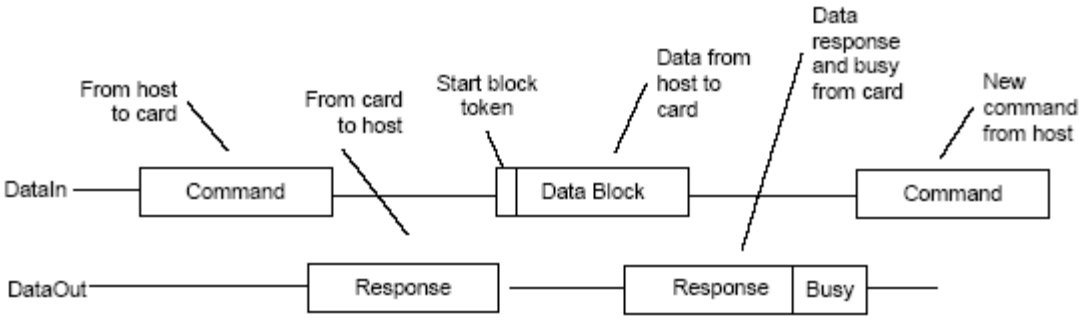
CMD۱۸ چندین بلوک پشت سر هم را از آدرس مشخص شده توسط آرگومان دستور می خواند، اگر قبل از این دستور تعداد بلوکهایی که باید خوانده شوند مشخص نشود، خواندن بلوکها پشت سرهم ادامه پیدا می کند تا زمانی که توسط CMD۱۲ دستور اتمام خواندن داده شود.



شکل۸: خواندن چند بلوکی

نوشتن تک بلوکی

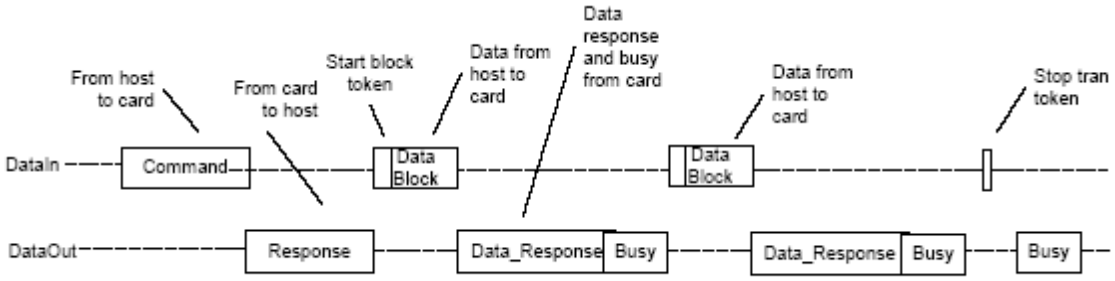
پس از پذیرش CMD۲٤ توسط SD CARD کنترلر بیرونی باید یک بلوک دیتا را پس از یک بایت تأخیر برای کارت حافظه ارسال نماید، بلافاصله پس از دریافت بلوک دیتا، کارت یک Data Response برای کنترلر ارسال می کند که بیان گر صحت یا عدم صحت دریافت بلوک داده است.



شکل ۹: نوشتن تک بلوکی

نوشتن چند بلوکی

CMD۲۵ چندین بلوک پشت سر هم را از آدرس مشخص شده توسط آرگومان دستور می خواند، اگر قبل از این دستور تعداد بلوکهایی که باید نوشته شوند مشخص نشود، نوشتن بلوکها پشت سرهم ادامه پیدا می کند تا زمانی که توسط Stop Tran Token دستور اتمام نوشتن داده شود.



شکل ۱۰: نوشتن چند بلوکی

■ خواندن رجیسترهای CID وCSD

همانند خواندن یک تک بلوکی است با این تفاوت که طول بلوک در آنها ۱۶ بایتی می باشد

رجیسترهای SD CARD

SD CARD حاوی۷رجیستر اصلی می باشد که ۵ رجیستر از مهمترین آنها عبارتند از: OCR،CID،RCA،CSDو SCR که حاوی اطلاعات مفیدی درباره ی وضعیت کارت و پیکر بندی آن است.

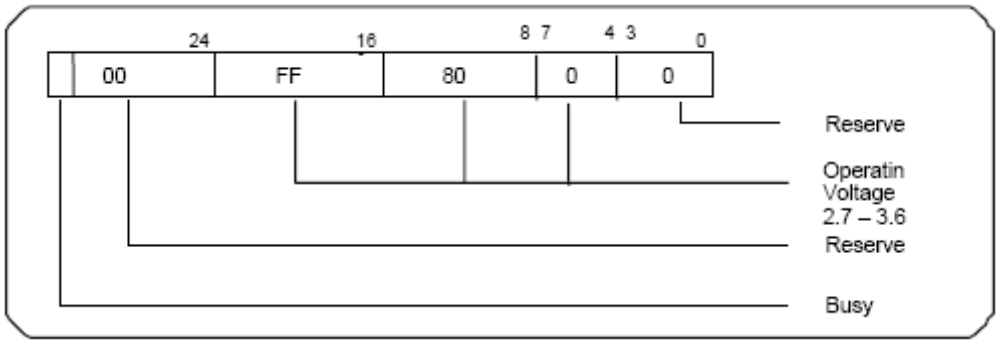
جدول ۸: رجیسترهای کارت حافظه

Name	Width	Description
CID	128	Card identification number: individual card number for identification.
RCA[1]	16	Relative card address: local system address of a card, dynamically suggested by the card and approved by the host during initialization.
CSD	128	Card specific data: information about the card operation conditions.
SCR	64	SD Configuration Register: information about the SD Card's special features capabilities.
OCR	32	Operation Condition Register

NOTE: 1) The RCA register is not available in SPI Mode.

OCR (Operating Conditions Reg)

یک رجیستر ۳۲ بیتی حاوی اطلاعاتی راجع به رنج تغذیه SD CARD می باشد. OCR محدوده ولتاژی را که دیتا ی کارت حافظه قابل دسترسی می باشد را نشان می دهد. شرح کامل این رجیستر در ضمیمه آورده شده است. اگر بیت ۳۲، یک شود به معنای آن است که روند بالا آمدن تغذیه تمام شده است.



شکل ۱۱: OCR Reg

CID (Card Identification Reg)

یک رجیستر ۱۶ بایتی می باشد که حاوی یک کد شناسایی یکتا برای SD CARD می باشد، این رجیستر در هنگام ساختن کارت حافظه، برنامه ریزی می شود و محتویات آن توسط کاربر قابل تغییر نمی باشد. شرح کامل این رجیستر در ضمیمه آورده شده است.

CSD (Card Specific Data)

حاوی برخی اطلاعات مورد نیاز برای دسترسی به اطلاعات SD CARD است . شرح کامل محتویات این رجیستر در ضمیمه آورده شده است. ستون CELL TYPE نوع Read Only(R) ، One Time(R/W) یا Erasable(R/W/E) را نشان می دهد.

برخی از اطلاعات مهم در این رجیستر عبارتند از:

READ_BL_LEN: بیشینه طول بلاک قابل خواندن عبارت است از $\mathbf{2}^{\text{READ_}\boldsymbol{BL_LEN}}$ که از ۵۱۲ تا ۲۰۴۸ بایت قابل تنظیم می باشد. باید توجه نمود در SD CARD، WRITE_BL_LEN همواره برابر READ_BL_LEN می باشد.

جدول ۹: طول بلوک دیتا

READ_BL_LEN	Block Length
0-8	Reserved
9	2^9 = 512 Bytes
.....	
11	2^{11} = 2048 Bytes
12-15	Reserved

SECTOR_SIZE: یک عدد باینری ۷ بیتی که تعداد بلوک های قابل نوشتن را مشخص می کند.

FILE_FORMAT_GROUP: فرمت فایل های داخل SD CARD را مطابق جدول زیر مشخص می کند.

جدول ۱۰: فرمت فایل

FILE_FORMAT_GRP	FILE_FORMAT	Type
0	0	Hard disk-like file system with partition table
0	1	DOS FAT (floppy-like) with boot sector only (no partition table)
0	2	Universal File Format
0	3	Others/Unknown
1	0, 1, 2, 3	Reserved

SCR (SD CARD Configuration Reg)

یک رجستر ۸ بایتی می باشد که حاوی اطلاعات پیکربندی SD CARD می باشد که توسط کارخانه سازنده در کارت حافظه برنامه ریزی می شود، شرح کامل این رجیستر در ضمیمه آورده شده است.

RCA (Relative Card Address)

یک رجیستر ۲ بایتی است که حاوی آدرسی است که توسط کارت در کاربرد ارتباط کنترلر با چند کارت حافظه ، بعد از Initialize در نظر گرفته می شود.

SD Status

یک رجیستر ۶۴ بایتی می باشد که حاوی اطلاعات خاصی در مورد SD CARD می باشد، این رجیستر در پاسخ به CMD ۵۵ و CMD ۱۳ (پشت سر هم) توسط کارت حافظه ارسال می شود. شرح کامل این رجیستر در ضمیمه آورده شده است.

ارسال دستورات

برخلاف مد SD، در مد SPI کارت همواره به تمامی دستورات پاسخ می دهد. پاسخ قبول یا ردکردن دستور داده شده را مشخص می کند. یک دستور ممکن است به دلایل زیر رد شود:

- هنگامی که کارت درحال عمل خواندن است، دستور ارسال شود.
- هنگامی که کارت BUSY است دستور ارسال شود.
- دستور حمایت نشود (opcode).
- محاسبه CRC درست نباشد.

- حاوی عملوند غیر معتبر باشد.

رجیسترها و برخی پارامترهای مهم SD CARD

DC Characteristics:

Item		Symbol	Condition	MIN.	Typ.	MAX.	Unit	Note
Supply Voltage 1		V_{DD}	-	2.0	-	3.6	V	For CMD0, 15,55, ACMD41 Only
Supply Voltage 2			-	2.7	-	3.6	V	For All commands
Input Voltage	High Level	V_{IH}	-	VDD*0.625	-	-	V	
	Low Level	V_{IL}	-	-	-	VDD*0.25	V	
Output Voltage	High Level	V_{OH}	VDD = 2V IOH = -100uA	VDD*0.75	-	-	V	
	Low Level	V_{OL}	VDD = 2V IOL = 100uA	-	-	VDD*0.125	V	
Standby Current		I_{CC1}	3.6V Clock 25MHz	-	-	30	mA	
			2.7V Clock Stop	-	-	0.2		
Operation Voltage		I_{CC2}	3.6V/25MHz	-	-	80	mA	Write
			2.7V/25MHz	-	-	80		Read
Input Voltage Setup Time		Vrs	-	-	-	250	ms	

AC Characteristics:

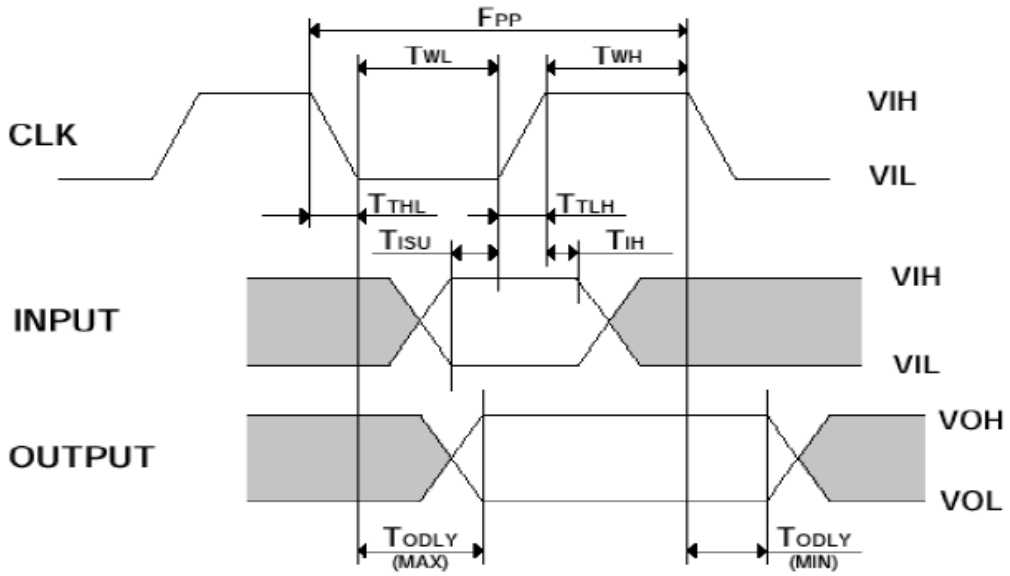


Table : AC Characteristics

Item	Symbol	Min.	Max.	Unit	Note
Clock Frequency (In any Sates)	F_{sty}	0	25	MHz	C_L<100pF (7Cards)
Clock Frequency (Data transfer Mode)	F_{PP}	0.1	25	MHz	C_L<100pF (7Cards)
Clock Frequency (Card identification Mode)	F_{OD}	100	400	kHz	C_L<250pF (21Cards)
Clock Low Time	T_{WL}	10	-	ns	C_L<100pF (7Cards)
Clock High Time	T_{WH}	10	-	ns	
Clock Rise Time	T_{TLH}	-	10	ns	
Clock Fall Time	T_{THL}	-	10	ns	
Clock Low Time	T_{WL}	50	-	ns	C_L < 250pF (21Cards)
Clock High Time	T_{WH}	50	-	ns	
Clock Rise Time	T_{TLH}	-	50	ns	
Clock Fall Time	T_{THL}	-	50	ns	
Input Setup Time	T_{ISU}	5	-	ns	C_L < 25pF (1Cards)
Input Hold Time	T_{IH}	5	-	ns	
Output Delay Time	T_{ODLY}	0	14	ns	

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets the SD Card
CMD1	Yes	None	R1	SEND_OP_COND	Activates the card's initialization process.
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	Reserved				
CMD6	Reserved				
CMD7	No				
CMD8	Reserved				
CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD).
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its card identification (CID).
CMD11	No				
CMD12	Yes	None	R1b	STOP _TRANSMISSION	Forces the card to stop transmission during a multiple block read operation.
CMD13	Yes	None	R2	SEND_STATUS	Asks the selected card to send its status register.
CMD14	No				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read & write).[1]
CMD17	Yes	[31:0] data address	R1	READ_SINGLE _BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[2]
CMD18	Yes	[31:0] data address	R1	READ_MULTIPLE _BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_ TRANSMISSION command.
CMD19	Reserved				
CMD20	No				
CMD21 ... CMD23	Reserved				
CMD24	Yes	[31:0] data address	R1[3]	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[4]
CMD25	Yes	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a stop transmission token is sent (instead of 'start block').

1) The only valid block length for write is 512 bytes. The valid block length for read is 1 to 512 bytes. A set block length of less than 512 bytes will cause a write error. The card has a default block length of 512 bytes. CMD16 is not mandatory if the default is accepted.
2) The start address and block length must be set so that the data transferred will not cross a physical block boundary.
3) Data followed by data response plus busy.
4) The start address must be aligned on a sector boundary. The block length is always 512 bytes.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD26	No				
CMD27	Yes	None	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28[1]	Yes	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29[4]	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_ PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits.[2]
CMD31	Reserved				
CMD32	Yes	[31:0] data address	R1	ERASE_WR_BLK_START _ADDR	Sets the address of the first write block to be erased.
CMD33	Yes	[31:0] data address	R1	ERASE_WR_BLK_END _ADDR	Sets the address of the last write block in a continuous range to be erased.
CMD34 .... CMD37	Reserved				
CMD38	Yes	[31:0] don't care*	R1b	ERASE	Erases all previously selected write blocks.
CMD39	No				
CMD40	No				
CMD41 ... CMD54	Reserved				
CMD55	Yes	[31:0] stuff bits	R1	APP_CMD	Notifies the card that the next command is an application specific command rather than a standard command.
CMD56	Yes	[31:0] stuff bits [0]: RD/WR.[3]	R1	GEN_CMD	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose/application specific commands. The size of the Data Block is defined with SET_BLOCK_LEN command.
CMD57	Reserved				
CMD58	Yes	None	R3	READ_OCR	Reads the OCR register of a card.
CMD59	Yes	[31:1] don't care* [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A '1' in the CRC option bit will turn the option on, a '0' will turn it off.
CMD60-63	No				

* The bit places must be filled but the values are irrelevant.

1) These features are not currently supported in the SanDisk SD Card.
2) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line.
3) RD/WR_: "1"=the host will get a block of data from the card. "0"=the host sends a block of data to the card.

OCR Register -

-

OCR Bit	VDD Voltage Window
0-3	Reserved
4	1.6-1.7
5	1.7-1.8
6	1.8-1.9
7	1.9-2.0
8	2.0-2.1
9	2.1-2.2
10	2.2-2.3
11	2.3-2.4
12	2.4-2.5
13	2.5-2.6
14	2.6-2.7
15	2.7-2.8
16	2.8-2.9

OCR Bit	VDD Voltage Window
17	2.9-3.0
18	3.0-3.1
19	3.1-3.2
20	3.2-3.3
21	3.3-3.4
22	3.4-3.5
23	3.5-3.6
24-30	reserved
31	Card power up status bit (busy)

- ***CID Register***

Name	Type	Width	CID—Slice	Comments	CID Value
Manufacturer ID (MID)	Binary	8	[127:120]	The manufacturer IDs are controlled and assigned by the SD Card Association.	0x03
OEM/Application ID (OID)	ASCII	16	[119:104]	Identifies the card OEM and/or the card contents. The OID is assigned by the 3C.*	SD ASCII Code 0x53, 0x44
Product Name (PNM)	ASCII	40	[103:64]	5 ASCII characters long	SD128, SD064, SD032, SD016, SD008
Product Revision** (PRV)	BCD	8	[63:56]	Two binary coded decimal digits	Product Revision (30)
Serial Number (PSN)	Binary	32	[55:24]	32 Bits unsigned integer	Product Serial Number
Reserved		4	[23:20]		
Manufacture Date Code (MDT)	BCD	12	[19:8]	Manufacture date–yym (offset from 2000)	Manufacture date(for example: Apr 2001 = 0x014)
CRC7 checksum*** (CRC)	Binary	7	[7:1]	Calculated	CRC7
Not used, always '1'		1	[0:0]		

* 3C = The 3 SDA founding companies: Toshiba, SanDisk, and MEI.

** The product revision is composed of two Binary Coded Decimal (BCD) digits, four bits each, representing an "n.m" revision number. The "n" is the most significant nibble and the "m" is the least significant nibble. Example: The PRV binary value filed for product revision "6.2" will be: 0110 0010.

*** The CRC Checksum is computed by the following formula:
CRC Calculation: G(x)=x7+3+1
M(x)=(MID-MSB)*x119+...+(CIN-LSB)*x0
CRC[6...0]=Remainder[(M(x)*x7)/G(x)]

- ***CSD Register***

Name	Field	Width	Cell Type	CSD-Slice	CSD Value	CSD Code
CSD structure	CSD_STRUCTURE	2	R	[127:126]	1.0	00b
Reserved	-	6	R	[125:120]	-	000000b
data read access-time-1	TAAC Binary MLC	 8 8	 R R	 [119:112] [119:112]	 1.5msec 10msec	 00100110b 00100110b
data read access-time-2 in CLK cycles (NSAC*100)	NSAC	8	R	[111:104]	0	00000000b
max. data transfer rate	TRAN_SPEED	8	R	[103:96]	25MHz	00110010b
card command classes	CCC	12	R	[95:84]	All (incl. WP, Lock/unlock)	1F5h
max. read data block length	READ_BL_LEN	4	R	[83:80]	512byte	1001b
partial blocks for read allowed	READ_BL_PARTIAL	1	R	[79:79]	Yes	1b
write block misalignment	WRITE_BLK_MISALIGN	1	R	[78:78]	No	0b
read block misalignment	READ_BLK_MISALIGN	1	R	[77:77]	No	0b
DSR implemented	DSR_IMP	1	R	[76:76]	No	0b
Reserved	-	2	R	[75:74]	-	00b
device size	C_SIZE	12	R	[73:62]	SD128=3843 SD064=3807 SD032=1867 SD016=899 SD008=831	F03h EDFh 74Bh 383h 33Fh
max. read current @VDD min	VDD_R_CURR_MIN	3	R	[61:59]	25mA	100b
max. read current @VDD max	VDD_R_CURR_MAX	3	R	[58:56]	25mA	011b
max. write current @VDD min	VDD_W_CURR_MIN	3	R	[55:53]	25mA	100b
max. write current @VDD max	VDD_W_CURR_MAX	3	R	[52:50]	35mA	100b
device size multiplier	C_SIZE_MULT	3	R	[49:47]	SD128=64 SD064=32 SD032=32 SD016=32 SD008=16	100b 011b 011b 011b 010b
erase single block enable	ERASE_BLK_EN	1	R	[46:46]	Yes	1b
erase sector size	SECTOR_SIZE	7	R	[45:39]	32blocks	0011111b
write protect group size	WP_GRP_SIZE	7	R	[38:32]	128sectors	1111111b
write protect group enable	WP_GRP_ENABLE	1	R	[31:31]	Yes	1b
Reserved for MultiMediaCard compatibility		2	R	[30:29]	-	00b
write speed factor Binary MLC	 R2W_FACTOR R2W_FACTOR	 3 3	 R R	 [1:16] [1:4]	 X16 X16	 100b 010b

max. write data block length	WRITE_BL_LEN	4	R	[25:22]	512Byte	1001b
partial blocks for write allowed	WRITE_BL_PARTIAL	1	R	[21:21]	No	0
Reserved	-	5	R	[20:16]	-	00000b
File format group	FILE_FORMAT_GRP	1	R/W(1)	[15:15]	0	0b
copy flag (OTP)	COPY	1	R/W(1)	[14:14]	Not Original	1b
permanent write protection	PERM_WRITE_PROTECT	1	R/W(1)	[13:13]	Not Protected	0b
temporary write protection	TMP_WRITE_PROTECT	1	R/W	[12:12]	Not Protected	0b
File format	FILE_FORMAT	2	R/W(1)	[11:10]	HD w/partition	00b
Reserved		2	R/W	[9:8]	-	00b
CRC	CRC	7	R/W	[7:1]	-	CRC7
not used, always'1'	-	1	-	[0:0]	-	1b

ضمیمه ج- محیط برنامه نویسی Code Vision AVR

انواع کامپایلر ها

امروزه کامپایلر های مختلفی برای کار با میکروکنترلرهای AVR به بازار عرضه شده اند که از جمله مهمترین آنها، می توان به کامپایلر های CodeVision AVR ،Imagecraft، GCC و IAR به زبان C و C++ و کامپایلرهای Bascom و FastAVR به زبان پاسکال اشاره کرد. که از میان آنها کامپایلرهای CodeVision و Imagecraft از اهمیت بیشتری برخوردارند.

کامپایلر های C

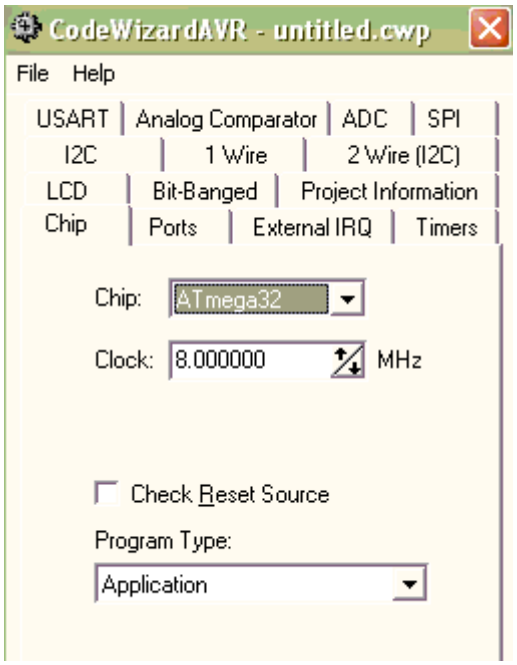
امروزه زبان C هرچه بیشتر برای برنامه نویسی میکرو کنترلرها عمومیت یافته است. مزایای استفاده از C بجای اسمبلی، کاهش زمان برنامه نویسی، نگهداری ساده تر، استفاده ی مجدد از کدها و ساده تر شدن فهم برنامه است. عیب آن هم افزایش حجم کدها و در نتیجه کاهش سرعت برنامه می باشد. برای رفع این معایب، ساختار AVR به گونه ای طراحی شده تا دستورات تولید شده توسط کامپایلرهای C را به صورت بهینه دیکد و اجرا نماید. علاوه بر این زبان C در مقایسه با زبانهای سطح بالای برنامه نویسی دیگر، به زبانهای سطح پایین نزدیکی بیشتری دارد و مسلماً حجم کدهای تولید شده از یک برنامه ی C نسبت به کدهای تولید شده از زبان سطح بالای دیگر کمتر خواهد بود.

آشنایی با نرم افزار CodeVision AVR

CodeVision AVR یک کامپایلر C، محیط توسعه یافته ی گرافیکی یکپارچه(IDE) و تولید کننده ی خودکار کدهای برنامه است که برای کار با میکروکنترلرهای AVR، طراحی شده است.

این برنامه یک پروگرامر ISP را نیز شامل می شود که امکان انتقال کدهای برنامه به میکرو را بعد از انجام موفق عمل کامپایل فراهم می آورد. علاوه بر این، نرم افزار دارای کتابخانه ی قدرتمندی است که کار با LCD، ارتباط SPI، I۲C وکار با تراشه های سنسور دما، همچنین EEPROM های سریال را میسر می سازد. CodeVision AVR برنامه ی CodeWizard را شامل می گردد. این ویزارد به ما اجازه می دهد که بدون نوشتن کد برنامه تنظیمات امکانات مختلف تراشه را از قبیل تایمرها، پورتها، ADC، وقفه های خارجی،

USART و... را به صورت یک محیط گرافیکی انجام دهیم. در این صورت CodeWizard کدهای مورد نیاز را به صورت خودکار تولید می کند. و به این ترتیب زمان برنامه نویسی کاهش پیدا می کند.



شکل ۱

ایجاد پروژه ی جدید

برای تولید یک پروژه ی جدید از منوی فایل،New را انتخاب کرده وسپس از پنجره ی ایجاد شده، گزینه یProject را انتخاب می کنیم. در اینجا سوال می کند که آیا می خواهید کد ویزارد را به کار بنندید؟

با فشردن yes پنجره ی کد ویزارد نمایش داده خواهد شد.(شکل۱) البته لازم به ذکر است که تقریباً تمام پنجره های مهم دارای

آیکونی در پنجره ی اصلی برنامه می باشند که با فشردن آنها پنجره ی مورد نظر باز خواهد شد.

در این پنجره همانطور که در شکل ۱ مشخص است tab های مختلفی وجود دارد. در chip، tab تراشه را انتخاب کرده، در قسمت کلاک، مقدار آنرا مشخص می نمائیم و در قسمت program type ، application را انتخاب می کنیم. بیش از ۹۹ درصد برنامه در قسمت application قرار می گیرد. Boot loader برای زمانی است که بخواهیم میکرو را توسط خودش پروگرام کنیم و برای دیباگ کردن آن نیز به کار برده می شود. در tab پورت، تنظیمات پورتهای میکرو را انجام داده و در واقع ورودی(pull up-۳ state) یا خروجی بودن آنها را مشخص می نمائیم. در tab تایمر وقفه ای که باید رخ دهد، همچنین مد کاری تایمر و مقدار اولیه ی آنرا تنظیم می کنیم. بدین ترتیب با کلیک کردن بر روی هر tab می توان تنظیمات مربوط به آنرا انجام داد و آنها را فعال نمود. بعد از اتمام کار، از منوی فایل، Generate,Save and Exit را کلیک کرده و فایلهای تولید شده توسط کد ویزارد را که سه فایل به پسوند های .C، .cwp. وprj. است را در محلی ذخیره می

نمائیم. سپس نرم افزار پنجره اصلی برنامه را باز می کند که در آن تنظیماتی که در کدویزارد انجام داده بودیم (به صورت کد و در رجیستر های مخصوص آن) قابل ملاحظه است. این پنجره فایل اصلی برنامه (Source) می باشد و برنامه مان را باید در این پنجره نوشته و آنرا تکمیل نمائیم. برای کامپایل کردن برنامه از منوی project گزینه ی compile را انتخاب می کنیم تا برنامه کامپایل گردد. در اینصورت پنجره ای باز می گردد که در آن یکسری اطلاعات اعم از حجم برنامه،خطاها(در صورت بودن).... را در اختیار کاربر می گذارد. بعد از اتمام برنامه و کامپایل کردن آن در صورتی که برنامه مان با خطایی مواجه نشده باشد باید یکسری تنظیمات دیگری را نیز انجام دهیم. برای تنظیم پروگرامر از منوی setting گزینه ی programmer را اتخاب کرده و در پنجره ی باز شده stk۲۰۰/۳۰۰ را انتخاب می کنیم. اگر پروگرامر مان از نوع دیگری بود باید آنرا در این قسمت معین کنیم. سپس از منوی project گزینه ی configure راکلیک کرده و در پنجره ی باز شده در tab، after make جلوی نام program the chip را تیک می زنیم. مشاهده می شود که قسمتهای جدیدی باز می شوند. در اینجا باید فیوز بیت های میکرو را تنظیم نمائیم. در اینجا باید کاربردهای فیوز بیت ها و نحوه ی فعال سازی آنها را از برگه ی اطلاعات میکرو استخراج کرده و در مقابل هر کدام در صورت لازم علامت بزنیم. لازم به ذکر است که فیوز بیت ها در اثر پروگرام شدن مقدار صفر را به خود می گیرند. CKSEL۰ الی CKSEL۳ مربوط به این است که منبع کلاک میکروکنترلر از کجاست. که با توجه به کلاکی که در مدارمان داریم کد مربوطه را از دیتا شیت استخراج کرده و آنها را تیک می زنیم. SUT مربوط به زمان راه اندازی میکروکنترلر (startup time) است. که در حالت معمولی این زمان برابر ۶۳ms می باشد. اگر که بخواهیم این مقدار کمتر باشد باید دیتا شیت را مطالعه نمائیم.Clkopt نیز سوییینگ کلاک را زیاد می کند تا اثر نویز بر روی آن کم گردد. این آپشن مواقعی استفاده می شود که فرکانس کلاک میکرو بالا باشد. در tab های دیگر نیز برخی دیگر از تنظیمات را می بینیم که غالباً تغییر داده نمی شوند. به طور مثال در tab، c compiler می توانیم کد خروجی برنامه را بهینه سازیم (از لحاظ سرعت یا از لحاظ حجم برنامه). بعد از اتمام این تنظیمات از منوی Project، گزینه ی Make را کلیک می کنیم. این گزینه برنامه را مجدداً کامپایل کرده و برنامه را با تمام تنظیمات انجام شده، برای پروگرام شدن در حافظه ی فلش میکرو آماده می کند. مشاهده می شود که در پنجره ی باز شده، دکمه ی program the chip فعال شده است و با فشردن آن (در صورتی که تمام تنظیمات درست صورت گرفته باشد و پروگرامر نیز به میکرو

متصل باشد) میکرو کنترلر را پروگرام می کند. اگر پنجره جدیدی دال بر وقوع خطا در حین پروگرام شدن باز نشد، نشان دهنده ی صحت عملیات پروگرامینگ می باشد.

لازم به ذکر می باشد که بعد از کامپایل کردن برنامه، می توان با نرم افزار های دیگری نیز میکرو را پروگرام کرد. از جمله AVR Studio و یا pony prog. که AVR Studio امکان شبیه سازی برنامه ی نوشته شده، بررسی وضعیت رجیستر ها در حین اجرای برنامه، نوشتن برنامه اصلی به زبان اسمبلی و چند قابلیت دیگر را دارا می باشد.

ضمیمه د- ابررایانه های برتر دنیا (June ۲۰۱۱)

Rank	Site	Manufacturer	Computer	Country	Year	Processor
۱	RIKEN Advanced Institute for Computational Science (AICS)	Fujitsu	K computer, SPARC۶۴ VIIIfx ۲.۰GHz, Tofu interconnect	Japan	۲۰۱۱	SPARC٦٤ VIIIfx ۲۰۰۰ MHz
۲	National Supercomputing Center in Tianjin	NUDT	NUDT TH MPP, X۵۶۷۰ ۲.۹۳Ghz ۶C, NVIDIA GPU, FT-۱۰۰۰ ۸C	China	۲۰۱۰	Intel EM۶۴T
۳	DOE/SC/Oak Ridge National Laboratory	.Cray Inc	Cray XT۵-HE Opteron ۶-core ۲.۶ GHz	United States	۲۰۰۹	AMD x۸۶_۶۴
۴	National Supercomputing Centre in Shenzhen ((NSCS	Dawning	Dawning TC۳۶۰۰ Blade, Intel X۵۶۵۰, NVidia Tesla C۲۰۵۰ GPU	China	۲۰۱۰	Intel EM۶۴T
۵	GSIC Center, Tokyo Institute of Technology	NEC/HP	HP ProLiant SL۳۹۰s G۷ Xeon ۶C X۵۶۷۰, Nvidia GPU, Linux/Windows	Japan	۲۰۱۰	Intel EM۶۴T
۶	DOE/SC/LBNL/NERSC	Cray Inc.	Cray XE۶ ۱۲-core ۲.۱ GHz	United States	۲۰۱۰	AMD x۸۶_۶۴

۷	Commissariat a l'Energie Atomique ((CEA	Bull SA	Bull bullx super-node S۶۰۱۰/S۶۰۳۰	France	۲۰۱۰	Intel EM۶۴T
۸	DOE/NNSA/LANL	IBM	BladeCenter QS۲۲/LS۲۱ Cluster, PowerXCell ۸i ۳.۲ Ghz / Opteron DC ۱.۸ GHz, Voltaire Infiniband	United States	۲۰۰۹	Power
۹	National Institute for Computational Sciences/University of Tennessee	Cray Inc.	Cray XT۵-HE Opteron ۶-core ۲.۶ GHz	United States	۲۰۰۹	AMD x۸۶_۶۴
۱۰	Forschungszentrum (Juelich (FZJ	IBM	Blue Gene/P Solution	Germany	۲۰۰۹	Power
۱۱	DOE/NNSA/LANL/SNL	Cray Inc.	Cray XE۶ ۸-core ۲.۴ GHz	United States	۲۰۱۰	AMD x۸۶_۶۴
۱۲	Moscow State University - Research Computing Center	T-Platforms	Xeon X۵۵۷۰/X۵۶۷۰ ۲.۹۳ GHz, Nvidia ۲۰۷۰ GPU, Infiniband QDR	Russia	۲۰۱۱	۵۵۷۰Xeon X
۱۳	DOE/NNSA/LLNL	IBM	BlueGene/L - eServer Blue Gene Solution	United States	۲۰۰۷	PowerPC ۴۴۰ ۷۰۰ MHz
۱۴	DOE/SC/Argonne National Laboratory	IBM	Intrepid - Blue Gene/P Solution	United States	۲۰۰۷	PowerPC ۴۵۰ ۸۵۰ MHz
۱۵	Sandia National Laboratories / National Renewable Energy Laboratory	Oracle	Red Sky - Sun Blade x۶۲۷۵, Xeon X۵۵xx ۲.۹۳ Ghz, Infiniband	United States	۲۰۱۰	Intel EM۶۴T Xeon X۵۵xx
۱۶	Texas Advanced Computing	Oracle	Ranger -	United States	۲۰۰۸	AMD x۸۶_۶۴ Opteron

	Center/Univ. of Texas		SunBlade x۶۴۲۰, Opteron QC ۲.۳ Ghz, Infiniband			Quad Core ۲۳۰۰ MHz
۱۷	DOE/NNSA/LLNL	IBM	Dawn - Blue Gene/P Solution	United States	۲۰۰۹	PowerPC ۴۵۰ ۸۵۰ MHz
۱۸	Air Force Research Laboratory	Cray Inc.	Raptor - Cray XE۶ ۸-core ۲.۴ GHz	United States	۲۰۱۰	AMD x۸۶_۶۴ Opteron ۸ Core ۲۴۰۰ MHz
۱۹	Korea Meteorological Administration	Cray Inc.	Haedam - Cray XE۶ ۱۲-core ۲.۱ GHz	Korea, South	۲۰۱۰	AMD x۸۶_۶۴ Opteron ۱۲ Core ۲۱۰۰ MHz
۲۰	Universitaet Frankfurt	Clustervision/Supermicro	LOEWE-CSC - Supermicro Cluster, QC Opteron ۲.۱ GHz, ATI Radeon GPU, Infiniband	Germany	۲۰۱۱	AMD x۸۶_۶۴ Opteron ۱۲ Core ۲۱۰۰ MHz

ضمیمه ه- محیط نرم افزار ISE

در این ضمیمه با ویژگیهای Xilinx ISE آشنا خواهیم شد. رابطه بین ابزارهای وارد کردن طرح، ابزارهای Xilinx و ابزارهای پیاده سازی نیز نشان داده می شود. معرفی بخش های زیر از مباحث اصلی این قسمت می باشد:

- ایجاد یک پروژه جدید ISE.
- ایجاد یک طرح HDL سطح بالا با املای درست و بازبینی آن.
- ایجاد یک فایل محدودیت های کاربر.
- شبیه سازی، سنتز و پیاده سازی طرح.

بدین منظور می بایستی نرم افزار ISE را در کامپیوتر نصب نمود. پس از نصب نرم افزار را از مسیر زیر می توان آنرا اجرا نمود:

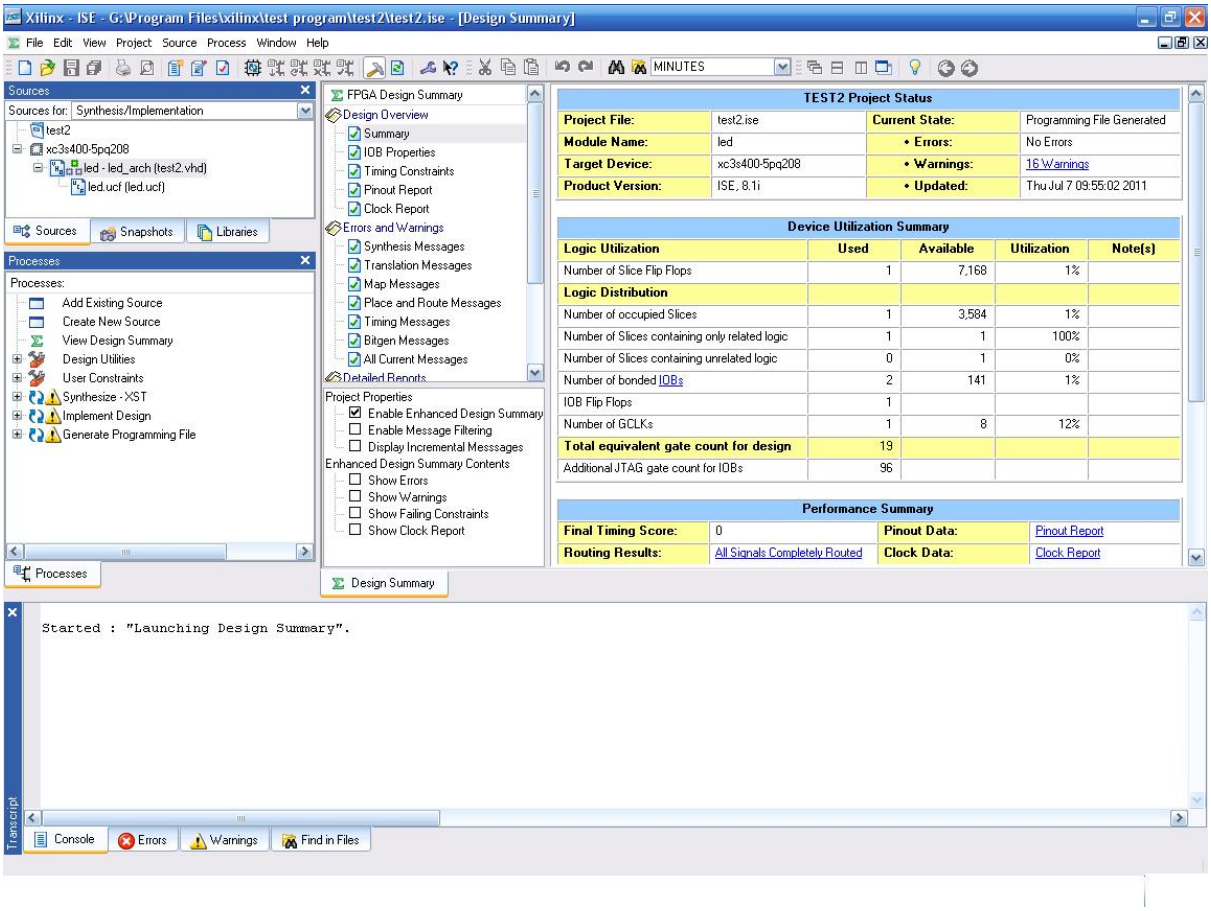
Start/ All Programs/ Xilinx ISE/ Project Navigator

با انتخاب گزینه فوق هدایت گر پروژه ISE باز می شود (شکل ۱). در این واسط می توانید منابع و پروسه های پروژه ISE را مدیریت کنید. ISE تمام جنبه های روند طراحی را کنترل می کند. از طریق واسط هدایت گر پروژه می توان به ابزارهای مختلف وارد کردن طرح و پیاده سازی و نیز به مستندات و فایل های مربوط به پروژه دسترسی داشت.

واسط هدایت گر پروژه (Project Navigator)

همانطور که در شکل۱ نشان داده شده است، واسط هدایت گر پروژه به چهار زیر پنجره تقسیم می شود. پنجره Source عناصر اضافه شده به پروژه را بصورت سلسله مراتبی نشان می دهد. در زیر آن پنجره Process قرار دارد که پروسه های موجود را نشان می دهد. پنجره Transcript پیام های وضعیت، خطاها و هشدارها را نمایش می دهد و طی تمام عملیات های پروژه به روز می شود. پنجره

چهارم یک واسط چند سندی است که فضای کاری (Workspace) نامیده می شود. در این واسط می توان گزارش های HTML، فایل های اسکی، نقشه های شماتیک و شکل موج های شبیه سازی شده را مشاهده کرد.



شکل ۱: واسط هدایت گر پروژه

پنجره Source Project: این پنجره بطور پیش فرض سه برگه دارد که اطلاعاتی در اختیار کاربر قرار می دهد: برگه Source، Snapshot و برگه Library.

برگه Source: در این برگه نام پروژه، مستندات کاربر، نوع قطعه مشخص شده، روند طراحی/ ابزار سنتز و فایل های منبع طرح نمایش داده می شود. هر فایل آیکون مخصوص به خود دارد که برای مشاهده لیست کامل انواع فایل ها و آیکونهای مربوط به آنها می توان از قسمت Help/ ISE Help Content بهره جست.

برگه Snapshot: این برگه تمام نسخه های فوری مربوط به پروژه ای را که در هدایت گر پروژه باز است نمایش می دهد. نسخه فوری یک کپی از پروژه است؛ تمام فایل های درون دایرکتوری کاری و زیر دایرکتوری سنتز و شبیه سازی. تمام اطلاعاتی که در این برگه ارائه می شود فقط خواندنی هستند.

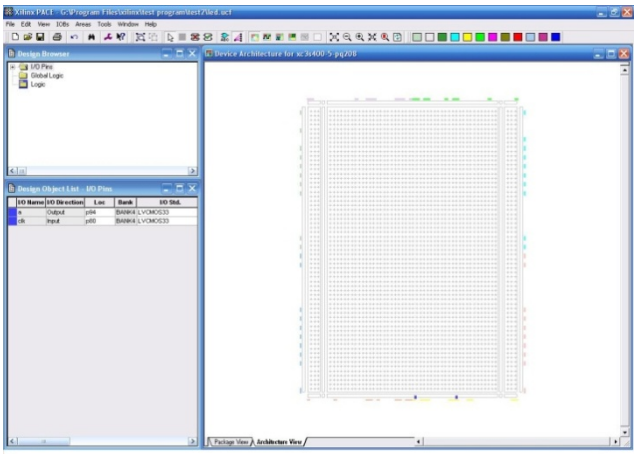
برگه Library view: این برگه تمام کتابخانه های متصل شده به پروژه ی باز شده در هدایت گر پروژه را نمایش می دهد.

پنجره Process:

- اضافه کردن فایل های منبع موجود: با دابل کلیک روی گزینه Add Existing Source دیالوگ مربوطه باز شده که در آن می توان فایلی در دایرکتوری پروژه یا دایرکتوری دیگری برای افزودن به پروژه انتخاب کرد.
- ایجاد فایل های منبع جدید: با استفاده از گزینه Create New Source می توان به ویزارد New Source دسترسی یافت که به کمک آن می توان یک فایل منبع جدید ایجاد کرده و آنرا به پروژه اضافه کرد.
- مشاهده خلاصه طرح: با انتخاب گزینه View Design Summary دیالوگ مربوطه باز شده و در آن اطلاعات سطح بالای پروژه از جمله اطلاعات کلی، خلاصه ای از میزان مصرف قطعه، اطلاعات مربوط به کارآیی حاصل از گزارش جانمایی و مسیر یابی، اطلاعات محدودیت ها و خلاصه ی اطلاعات تمام گزارش ها با پیوند به گزارش های جداگانه را لیست می کند.
- امکانات طرح: گزینه ی Design Utilities امکان تولید نماد، الگوهای نمونه سازی، مبدل HDL، نمایش خط فرمان فایل گزارش و کامپایل کتابخانه شبیه سازی را فراهم می کند.
- محدودیت های کاربر: گزینه ی User Constraints دسترسی به محل ویرایش و زمان بندی و محدودیت ها را ممکن می سازد. قسمت مهم این بخش Assign Package Pin است. با دابل کلیک بر روی آن دیالوگ مربوطه باز شده و در آن پنجره ورودیها، خروجی و سایر تنظیمات

مربوط به IOB را انجام داده و تنظیمات را ذخیره می نمائیم. اینکه هر IO بصورت ورودی است یا خروجی و از چه استانداردی استفاده می کند در این بخش صورت می پذیرد (شکل ۲)

- سنتز کردن: با راست کلیک کردن بر روی گزینه Synthesize و انتخاب گزینه ی Run یا Rerun از منو، امکان بررسی املا، سنتز کردن، دسترسی به نمای RTL یا شماتیک فناوری و گرفتن گزارش های سنتز میسر می شود. این گزینه ها بسته به ابزار سنتز مورد استفاده تغییر می کند.
- پیاده سازی طرح: گزینه ی Implement Design دسترسی به ابزارهای پیاده سازی، گزارش های مربوط به روند طراحی و ابزارهای اشاره را فراهم می کند.
- تولید فایل برنامه ریزی: گزینه ی Generate Programming File دسترسی به ابزارهای پیکربندی و تولید بیت استریم را فراهم می کند.

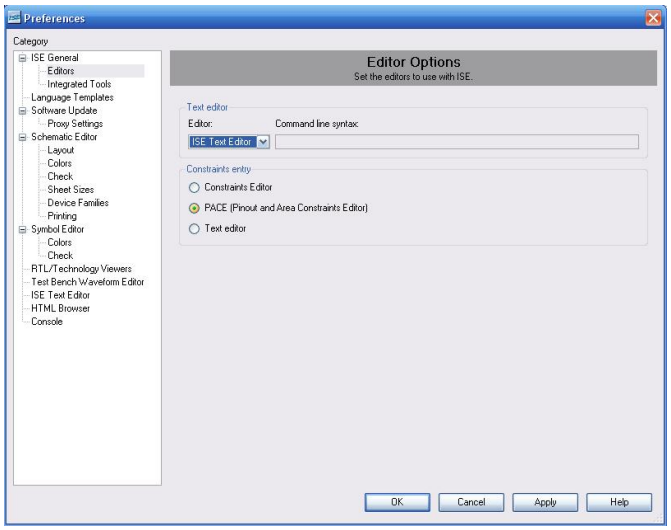


شکل ۲: تنظیمات پایه های I/O

- **فضای کاری**

برگه Design Summary: این برگه اطلاعات سطح بالای مربوط به پروژه، از جمله اطلاعات کلی، خلاصه ای از مصرف قطعه، داده های کارآیی جمع آوری شده از گزارش جانمایی و مسیریابی، اطلاعات محدودیت ها و اطلاعات خلاصه تمام گزارش ها همراه با پیوندهای گزارش های مجزا را لیست می کند.

- ویرایشگر متن: فایل های منبع و دیگر مستندات متنی را می توان در یک ویرایشگر طراحی شده توسط کاربر باز کرد. کاربر می تواند با باز کردن Edit/ Preferences تنظیمات مربوط به ویرایشگر را انجام دهد. با انتخاب این گزینه، دیالوگ Preferences باز می شود. در لیست سمت چپ گزینه ISE General را توسعه داده و روی Editors کلیک کنید. ویرایشگر پیش فرض ISE Text Editor است که به کمک آن می توان فایل های منبع را ویرایش کرد و به الگوهای زبان، یک کاتالوگ زبان ABEL، وریلاگ و VHDL و الگوهای فایل محدودیت کاربر دسترسی داشت و از آنها در طرح استفاده کرد. این مورد در شکل ۳ قابل مشاهده است.



شکل ۳: در ISE ویرایشگر پیش فرض ISE Text Editor است

- ویرایشگر شبیه ساز/ شکل موج ISE: این ویرایشگر یک ابزار ایجاد آزمونه مبتنی بر PC است که در هدایت گر پروژه گنجانده شده است. از ویرایشگر شکل موج می توان برای وارد کردن گرافیکی تحریک و پاسخ مورد انتظار استفاده کرد و سپس یک آزمونه VHDL را تولید کرد.
- ویرایشگر شماتیک: از این ویرایشگر می توان برای ایجاد گرافیکی طرح های منطقی و مشاهده آن ها استفاده کرد.
- ایجاد نسخه فوری: به کمک نسخه های فوری می توان روی نسخه های اصلاح شده طرح کنترل داشت. هر نسخه فوری حاوی یک کپی از تمام فایل های درون دایرکتوری پروژه است. برای ایجاد این نسخه می توان از مسیر زیر استفاده کرد:

۱. گزینه Project/ Take Snapshot را انتخاب کرد.

۲. در دیالوگ Take a Snapshot of the Project نام نسخه فوری و هرگونه توضیح مربوط به آن را وارد کرد.

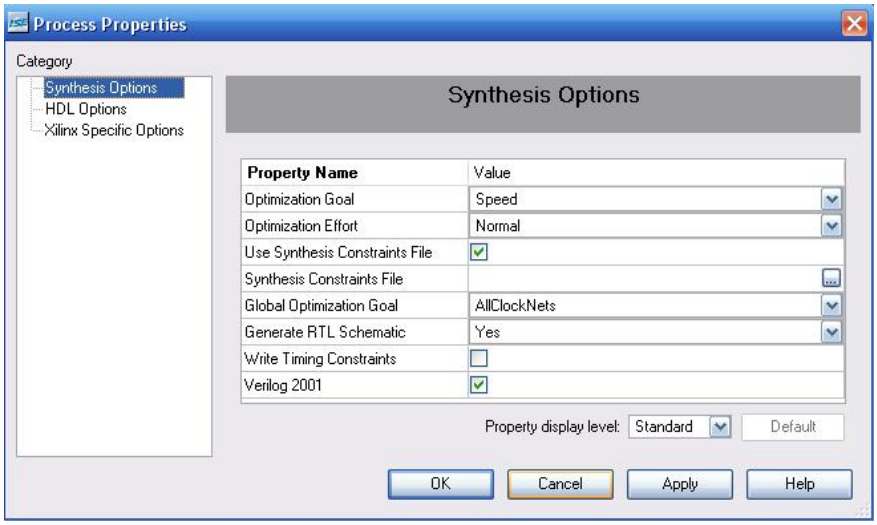
- **ابزارهای سنتز**

هر طرح را می توان با ابزارهای مختلفی سنتز کرد که از مشهورترین آنها می توان به Leonardo Spectrum، Precision و XST اشاره کرد. خصوصیت های متفاوتی توسط هر کدام از این ابزار پشتیبانی شده، که بسته به نوع کاربرد می توان از آنها بهره جست. لازم بذکر است که تمام این ابزار در نسخه های معمولی ISE در دسترس نیست و باید بصورت جداگانه تهیه شود.

ابزار سنتز XST: ابزار سنتز XST (Xilinx Synthesis Technology) قسمتی از بسته نرم افزاری ISE است و برای هر دو روند طراحی مبتنی بر HDL یا شماتیک قابل استفاده است.

دیالوگ Process Properties برای XST:

دیالوگ خواص پروسه برای ابزار سنتز XST در شکل ۴ نشان داده شده است. در این دیالوگ می توانید نتایج سنتز XST را کنترل کنید. از این طریق می توانید نتایج سنتز را برای ناحیه یا سرعت و زمان اجرای Synthesizer کنترل نمود.



شکل ۴: دیالوگ Process Properties برای ابزار سنتز XST

Abstract:

The fast processing of data is an important issue which has been desired from the past. It is obtained with special hardware's, algorithms and programming methods including parallel processing. One of the newest hardware technologies that have made practical usage is called FPGA's. Nowadays has provided hardware with help of one or more processors including special peripheral devices, Able to run algorithms with high computational and complexity.

One Application of this Real-Time signal processing is the implementing of image and video processing algorithms. Implementation of algorithms and methods on these processors represents Parallel processing capability and usage importance of these chips in large systems and real-time processing systems.

In this thesis after the hardware design it has been attempted to build it. Then focus on implementation of the edge detection algorithm as a case study and sample programs and then try improving performance of designed system. In fact, in this project we seek to design hardware with processor core of FPGA. In addition it has been considered using of the chips and the other components for communicating with outside of the board.

One of the important targets of this project is design and manufactures an evaluation board with high compatibility, Hardware test and demonstrates its ability to implement an edge detection algorithm as a sample program. finally compare the performance of designed hardware with software (PC) in terms of speed process. Microprocessor system design considerations, schematic and PCB design and hardware design are the main topics discussed in the plan.

Keywords: Hardware Design, Signal Processing, Configurable Devices, Parallel Processing, FPGA

Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

Design and Implementation of Parallel Processing System Based on FPGA

By: Mohammad Arablu

Supervisor:

Dr. Ali Soleamani

September 2011