معرفی FPGA و کاربردهای آن

مهدی برج خانی
دانشگاه ارومیه
Mehdi_borjkhani@yahoo.com

چکیده :
در این مقاله سعي شده است مدارات مجتمع FPGA و مشخصات آنها توضیح داده شود . این مدارهاي مجتمع جدیداً بجاي اجراي مدارهاي دیجیتالي توسط آي سي هاي متداول (TTL یا CMES) مورد استفاده قرار مي گیرد و بسیاري از مشکلات ناشي از سیم کشي و خطا یابي را از بین برده است .

۱-مقدمه

شاید تا بحال مدارهاي منطقي را بوسیله گیتهاي NOT , OR , AND ، ساخته اید . براي ساخت چنین مدارهایي (از قبیل شمارنده ها ، کنترل کننذه ها ، و ...) ابتدا باید تعریفي از مدار در دسترس باشد سپس با توجه به منطق اعداد دودویي یك جدول صحت براي مدار تشکیل مي شود و حالتهاي مختلف مورد بررسي قرار مي گیرد سپس با توجه به جدول صحت مدار توسط گیتهاي منطقي مانند NAND , NOT , OR , AND ، . ... طراحي مي شود پس از این مرحله نوبت به پیاده سازي مدار برروي برد توسط آي سي هاي منطقي مي رسد و همانطور که مي دانید یکي از وقتگیرترین و خسته کننده ترین مرله ساخت یك مدار همین قسمت است . بعد از این مرحله نوبت به تست مدار جهت اطلاع از درستي مراحل کار کرد مدار مي رسد . اگر در یکي از مراحل قبل دچار اشتباه شده باشیم مطمئناً در مرحله تست مدار دچار مشکل مي شویم . در صورت اشتباه در مراحل قبل باید تمام مراحل را از آخر به اول یك به یك چك کنیم تا بتوانیم اشتباهات احتمالي موجود در نحوه بستن و سیم کشي مدار ، طراحي مدار از روي جدول صحت و درستي جدول صحت را برطرف کنیم . با توجه به مطالب گفته شده حتماً به این نکته اذعان خواهید داشت که بیشترین اشتباهات در مرحله سیم کشي و بستن مدار برروي برد پیش خواهد آمد . ممکن است سیمي در جاي اصلي وصل نشده باشد و یا ممکن است یك پایه به هیچ جا متصل نباشد و یا اشتباهات مشابه اینها . . . از طرف دیگر مي دانیم که هر چه مدار بزرگتر و پیچیده تر باشد اشتباهات بیشتر و عیب یابي مشکلتر خواهد بود . اینجاست که نقش آي سي هاي FPGA نمایانتر مي شود . آي سي هایي که با داشتن انواع گیتهاي مختلف درون خود

بسياري از مشكلات ناشي از عيب يابي مدارهاي منطقي را برطرف كرده است . در قسمتهاي بعد به بررسي و معرفي اين آي سي ها مي پردازيم .

۲- مدار مجتمع منطقي قابل برنامه ريزي :

اين آي سي ها كه بحث اصلي اين مقاله به شمار مي روند با نامهاي FPGA يا CPLD ها يا نام كلي FPLD ها شناخته مي شوند . FPLD ها شامل گيتهاي منطقي و ابزارهايي جهت اتصال آنها در يك مدار مجتمع واحد مي باشد . با استفاده از نرم افزار هاي موجود مشخص مي شود كه گيتهاي داخل يك device چگونه مي تواند جهت ساخت يك مدار منطقي به هم وصل مي شوند . خروجي برنامه يك فايل باينري است كه داخل FPLD ، download مي شود تا باعث شود FPLD مانند يك مدار منطقي عمل كند . سپس FPLD برنامه ريزي شده مي تواند داخل يك مدار بزرگتر قرار گيرد .

۳-اصول مدار داخلي FPLD ها :

اين device در واقع از PLA ها (programmable logic Array) نتيجه شده اند . ساختمان يك PLA شامل چند گيت AND ، چند گيت OR ، و inverter ها مي باشد كه از طريق آرايه سوئيچها به هم متصل مي شوند . در يك PLA هر ورودي و معكوس شدة آن توسط سيستمهاي افقي از داخل يك گيت AND عبور داده مي شوند . گيتهاي AND ورودیها را از طريق به هم بستن سيستمهاي افقي و عمودي دريافت مي كنند . سيستمهاي عمودي خروجيهاي گيتهاي AND را وارد آرايۀ گيتهاي OR مي كنند . در اين قسمتها سيمهاي افقي كه ورودیها را به گيتهاي OR مي برند به اين سيمهاي عمودي متل مي شوند . گاهي اوقات از آرايه منطقي قابل برنامه ريزي (PAL) ساده تري استفاده مي كنند .

مدارات PLA و PAL براي منطق تركيبي موثرند ولي براي منطق ترتيبي بدون اضافه كردن فليپ فلاپهاي خارجي قابل استفاده نيستند . در FPGA ها سه LUT و دو فليپ فلاپ و بعضي مدارهاي اضافي جهت ايجاد يك CLB به هم متصل مي شوند . سپس اين CLB ها توسط يك ماتريس سوئيچ قابل برنامه ريزي (programmable switch matrices) (PSM) به هم متصل مي شوند . پينهاي FPGA I/O مي توانند به PSM ها و CLB ها و يا حتي ماتريس مسير يابي خودشان متصل شوند . تعداد CLB نمي تواند يك ارتباط مستقيم با دنياي خارج داشته باشد . كار PSM ها بدين صورت است كه پس از طراحي مدار منطقي گيتهاي لازم جهت ايجاد مدار موردنظر را در CLB ها به هم متصل مي كند يعني تمام سيم بندي ها و اتصالات در FPGA ها در داخل IC انجام مي شوند . يعني بطور فيزيكي نمي توان اتصالي ايجاد يا آن را تغيير داد . در عوض اتصالات بطور الكتريكي برقرار مي شوند . در نسلهاي اوليه FPGA ها PSM ها با قرار دادن فيوزهايي در هر نقطة اتصال سيمهاي افقي و عمودي ذكر شده بوجود مي آمد با اعمال ولتاژ مثبت به سيمهاي عمودي و افقي مدار موردنظر حاصل مي شد . به اين ترتيب كه ولتاژ مثبت فيوز واقع در نقطه اتصال سيمهاي عمودي و افقي را مي سوزاند كه اين عمل بايد به دفعات زياد جهت از بين بردن اتصالات ناخواسته در مدار مورد نظر انجام مي شد . در انتهاي اين روند تنها اتصالاتي كه جهت ايجاد طرح مدار مورد نياز بودند باقي مي ماند ند . ايراد فيوزها اين است كه وقتي يك مرتبه سوختند قابل برگشت به وضعيت اوليه شان نيستند . بنابراين چنانچه خطايي رخ مي داد device قابل برنامه ريزي مزبور كنار گذاشته شده ، device جديدي برنامه ريزي مي شد لذا اگر اتصالات قابليت پاك شدن و دوباره برنامه ريزي شدن را داشته باشند بسيار ارزانتر و قابل قبول تر خواهد بود . به اين ترتيب كه اين device ها داراي سوئيچ هاي قابل برنامه ريزي به جاي

فيوزهاي مذكور هستند . هر سوئيچ توسط يك المان ذخيره سازي كه حالت باز بودن يا بسته بودن سوئيچ را در خود ذخيره مي كند كنترل مي شود . تغيير مقادير ذخيره شده در اين المانهاي ذخيره سازي باعث تغيير حالت سوئيچ و توابع device قابل برنامه ريزي مي شود . به اين ترتيب نياز به خريد IC هاي جديد جهت ايجاد هر طرح جديد كاملاً از بين مي رود .

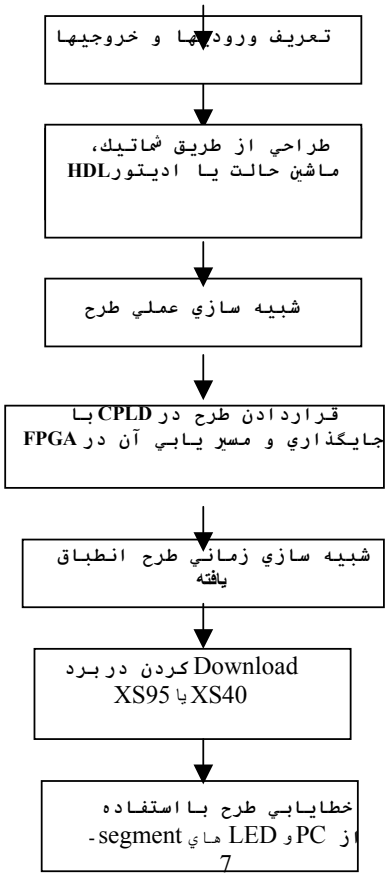
۴-روند طراحي يك مدار قابل برنامه ريزي:

نرم افزار XILINX Foundation محيطي جهت ايجاد برنامه هائي براي توصيف طرح منطقي مورد نظر مي باشد . روند طراحي با استفاده از نرم افزار Foundation به اين ترتيب است :

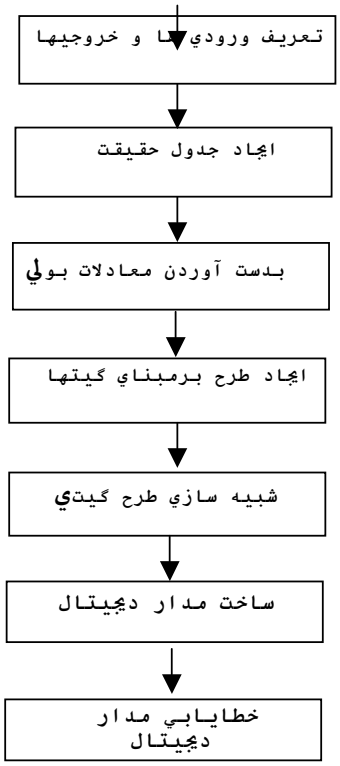
۴- ۱ طرحهاي ديجيتال مورد نظر ، با استفاده از اديتور شماتيك يا اديتور ماشين حالت يا ايتور متني HDL (ABEL يا VHDL) وارد مي شوند . طرح مورد نظر مي تواند توسط يكي از اين اديتور ها ي تركيبي از همة آنها ايجاد شود .

۴-۲ يك سيمولاتو (شبيه ساز) عملي عملكرد يك طرح كامپايل شده را چك مي كند و به شما اجازه مي دهد تا نتايج را ببينند و صحت يا عدم صحت نتايج را بررسي كنيد . در صورت بروز هر گونه خطائي مي توان به محيط اديتوري شماتيك ، HDL يا ماشين حالت برگشته خطاها را اصلاح كنيد .

روند طراحی برای XC9500/XC4000



مراحل کلی طرح



شكل ۱: تغييرات جريان طراحي ديجيتال هنگام استفاده از نرم افزار XILINX Foundation براي XC9500 CPLD يا XC4000 FPGA :

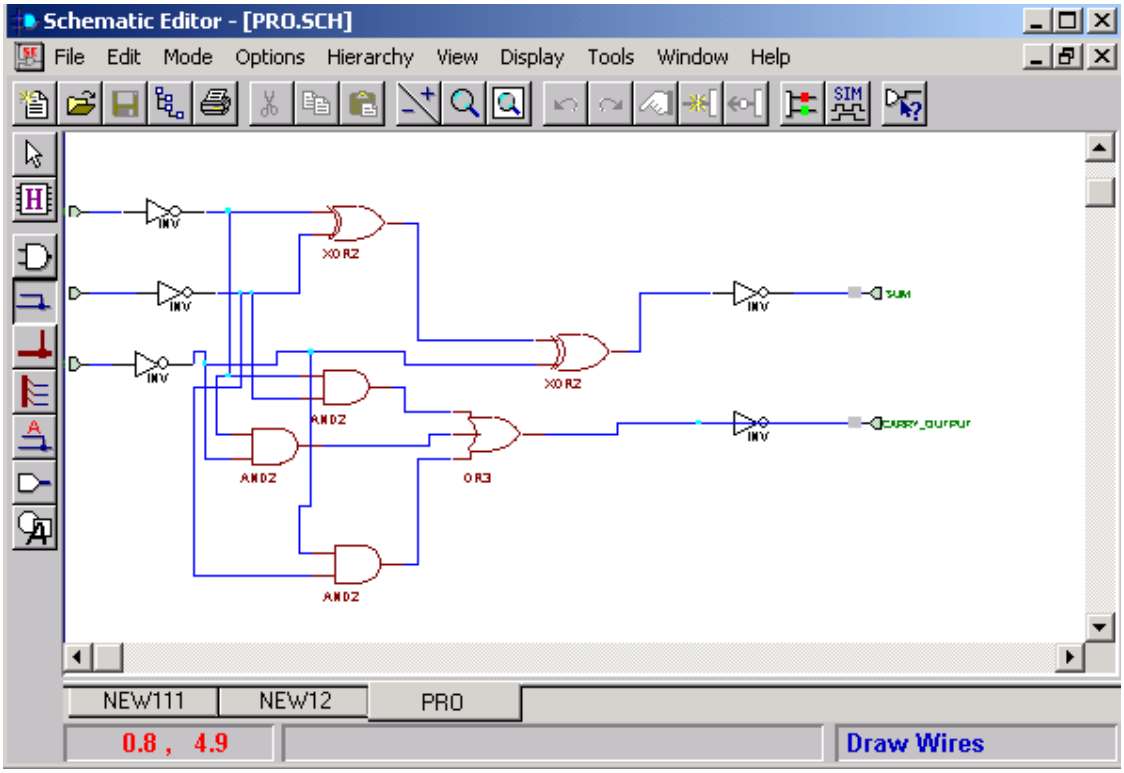
۳-۴ ابزار اجراي نرم افزار Foundation ، ابتدا ليست گيتها و اتصالات ايجاد شده را به يك فايل با فرمت باينري تبديل مي كند كه جهت برنامه ريزي FPLD استفاده خواهد شد . در اين مرحله است كه يك device خاص جهت برنامه ريزي كردن بايد مشخص شود ، مانند خانواده هاي XC95108 و XC400 ، 5XL . براي device هاي XC9500 ، برنامه مورد نظر طراح را داخل يك CPLD قرار مي دهيم ولي در XC4000 ، طرح داخل يك FPGA قرار مي گيرد . به اين ترتيب كه گيتها داخل CLB هاي مشخص قرار مي گيرند و مسير يابي و سيم پيچي با استفاده از PSM ها انجام مي شوند .
۴-۴ بعد ازاينكه ابزار اجرائي نرم افزار FOUNDATION تأخيرهاي مربوط به گيتها و عمل مسير يابي را مشخص كردند ، شبيه سازي خصوصيات زماني طرح با يك mapping مشخص در يك FPLD انجام مي شوند .
۵-۴ با وارد كردن ورودي ها به يك برد XS95 يا XS40 از طريق كابل پورت موازي كامپيوتر عمل خطايابي(Debugging) انجام مي شود .

۵-مثالي در مورد نحوة برنامه ريزي يك برد XS40 (جمع كنندة تك بيتي) :

جدول حقيقت براي يك جمع كنندة ۲ بيتي همراه باكري ورودي و خروجي :

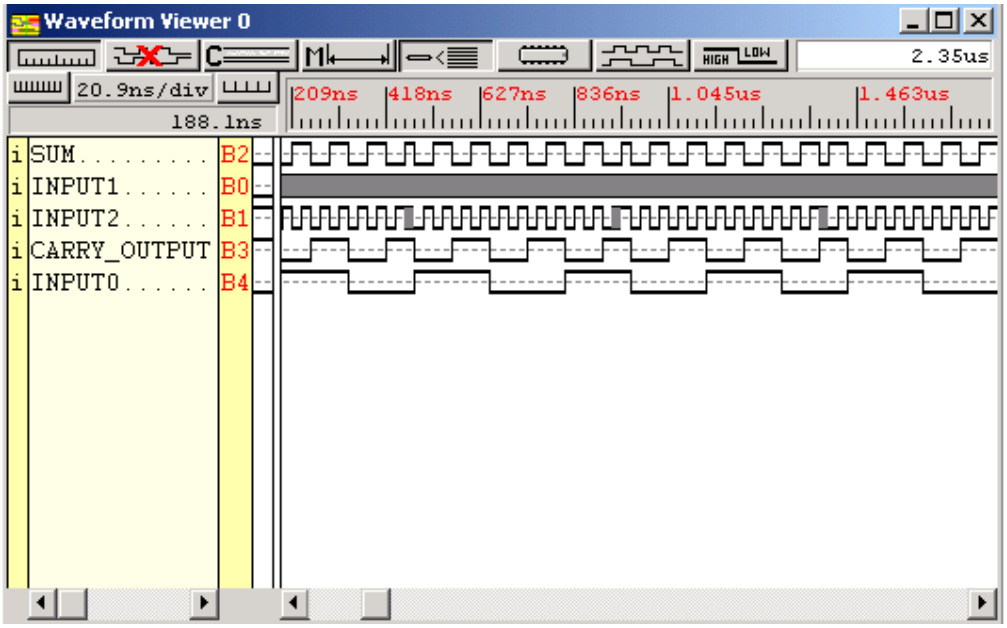
Input 1	Input 0	Carry input	Sum output	Carry output
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ابتدا يك پروژه در مورد شماتيك كه ADDL – 40 نام دارد تعريف مي كنيم كه طرح را براي XC4005XL ايجاد خواهد كرد . اديتور شماتيك را فعال مي كنيم و گيتها را مطابق شكل A متصل مي كنيم . شكل A يك جمع كنندة تك بيتي را با اضافه كردن با فرهاي ورودي و خروجي نشان مي دهد . وقتي شماتيك كامل شد ، آن را بعنوان ADDL – SCH ذخيره مي كنيم . سپس يك netlist را با انتخاب Options – create netlist از منو به وجود آورده و با Options – Export Netlist آن را در فرمت EDIF 200 ارسال مي كنيم . (همچنين با انتخاب Options – Integrity test مي توان درستي همه چيز را قبل از ادامة عمليات چك كرد .) سپس از اديتور روي كليد در Flow tab از صفحة project manager كليك مي كنيم .
با انتخاب Signal – Add Signals از منو ، سيگنالهاي ورودي و خروجي را به صفحة waveform viewer اضافه مي كنيم . با انتخاب signal – Add stimulators.. از منو ورودي ها را به پايين ترين سه بيت باينري شمارنده (BC0 , BC1 , BC2) متصل مي كنيم . در نهايت روي دكمة SIMULATOR در toolbar Logic كليك مي كنيم . شكل موجهاي نشان داده شده در شكل B ظاهر مي شوند



شکل A

.



شکل B

(مي توان از دكمة و scrollbar در صفحة waveform viewer جهت باز كردن و متمركز كردن شكل موجها استفاده كرد .) چنانچه با دقت بررسي كنيم مي بينيم كه نتيجه شبيه سازي با آنچه در جدول حقيقت نمايش داده شده بود ، مطابقت دارد . بعد از چك كردن عملكرد جمع كنندة تك بيتي

، مي توانيم از سيمولاتور (شبيه ساز) خارج شده و فايل را براي XC05xl FPGA كامپايل كنيم .
در ابتدا لازم است كه ترمينالهاي ورودي و خروجي طرح را به پينهاي فيزيكي I/O در XX4005CL اختصاص دهيم . به اين منظور از اديتور HDL جهت ايجاد فايل ADDL - 40 . UCF و افزودن خطوط زير به آن استفاده مي كنيم

```
NET INPUT 1                    LOC= P46;
NET INPUT 0                    LOC=P45;
NET CARRY - INPUT              LOC=P44;
NET SUM                        LOC=P25;
NET CARRY - OUTPUT             LOC=P26;
```

حال ، روي دكمه جهت شروع كامپايل كردن طرح كليك مي كنيم . سپس در صفحة IMPLEMMENT Design كه ظاهر خواهد شد . روي دكمة Option كليك مي كنيم . در صفحه Option ، در قسمت User Constraints ، نام UCF اي را كه ايجاد كرده ايم وارد مي كنيم .
مثلاً C:\XCPROJ\ADDL - 40\ADDL - 40 . UCF . (يا از دكمة Browse جهت انتخاب UCF خود استفاده مي كنيم .) سپس OK را براي بازگشت به صفحة implement Design كليك كرده و RUN را در اين مرحله ، بايد يك فايل ADDL - 40 . BIT داشته باشيم كه بتوانيم در برد download , XS40 كنيم .
1 ـ چه فضائي از XC4005XL FPGA جهت ساخت يك جمع كنندة تك بيتي اشغال مي شود ؟
2 ـ آيا كامپايلر ، ورودي ها و خروجي ها را به پينهائي كه ما درخواست كرده بوديم اختصاص مي دهد ؟
ما مي توانيم پاسخ اين سئوال را از فايلهاي گزارش كه توسط Foundation implementation ايجاد مي شود در يابيم . در قسمت سمت راست صفحة , reports tab project manager را انتخاب مي كنيم سپس روي آيكون implementation report files ، دوبار كليك مي كنيم . صفحة report browser ظاهر خواهد شد . روي هر كدام از آيكونها دوبار كليك مي كنيم . تا محتويات مربوط به آن گزارش را مشاهده كنيم . اين گزارشها ، اظلاعات زير را خلاصه مي كنند :

۵-۱ گزارش ترجمه

هر مشكلي كه در هنگام تبديل nelist به فرمت داخلي كه توسط ابزار foundation implementation استفاده مي شود ، مشاهده مي شود ، در اين قسمت ليست مي شود .
در اين قسمت چك كردن قوانين طراحي و فايل UCF جهت خطايابي ، نيز صورت مي گيرد . معمولاً بيشتر خطائي كه در اين فايل مي بينيد نتيجه عبارات قابل قبولي است كه وارد كرده ايد .

۵-۲ گزارش انطباق

در اينجا در مورد اينكه چه نوع بهينه سازي هائي روي netlist انجام شده است ، اظلاعاتي بدست خواهيد آورد . گيتهاي منطقي جابجا و اضافي مي شوند بطوريكه بدون تغيير عملكرد در مدار ، آن را بهينه سازي مي كنند. يك مثال از جابجائي منطقي وقتي اتفاق مي افتد كه يك گيت را در طرح قرار مي دهيد ولي خروجي را براي هيچ چيز استفاده نمي كنيد .
مثال ديگر يك گيت AND با يك ورودي متصل شده به "„ " منطقي (GND) است كه نتيجة خروجي آن همواره LOW است .

همچنين گزارش نشان مي دهد كه گيتهاي منطقي چگونه گروه بندي شده در LUT هاي CLB هاي FPGA قرار مي گيرند .

۵-۳ گزارش جابجايي و مسير يابي

اين گزارش ، انتخابهائي را كه ما جهت اثر گذاري به روند جايگذاري و مسير يابي انجام داده ايم ، يادداشت مي كند . همچنين خطاها و اخطارها را ليست كرده و زمان صرف شده در جايگذاري و مسير يابي هاي مختلف را ثبت مي كند . اين گزارش همچنين مشخصات آماري گوناگون عملكرد زماني مسير يابي را گزارش مي دهد .
ولي مهمتر از همه ، اين گزارش پاسخ سئوال اول ما را در مورد اينكه حجمي از XC4005XL جهت ساخت يك جمع كنندة ۱ بيتي مورد استفاده قرار مي گيرد ، خواهد داد .
خلاصة استفاده از فضاي device :

IO	5/112	4%used
	5/61	8% bonded
LOGIC	1/961	0% used
IBO	5/112	4% used
CLB	1/196	0% used

جمع كننده تك بيتي داراي ۳ ورودي و ۲ خروجي است . بنابراين جمع كنندة ٥ بلوك از ۱۱۲ بلوك I/O (IOB`s`) قابل دسترسي در XC4005XI FPGA را استفاده مي كند .
البته فقط ٦۱ بلوك از اين IOB ها واقعاً به پين هاي فيزيكي روي بستة ۸٤ پيني PLCC متصل هستند .
بنابراين تقريباً ۸% از I/O ها قابل دسترسي ، توسط جمع كننده هاي تك بيتي استفاده مي شوند .
هر دو مدار SUM و CARRY - OUTPUT در طرح ها ، يك خروجي و سه ورودي دارند . هر مدار بايد در يك 4LUT پيني قرار گيرد بنابراين انتظار داريم كه از دو LUT براي مداري جمع كنندة تك بيتي خود استفاده كنيم . و در هر CLB هم دو 4LUT پيني وجود دارد بنابراين فقط يك CLB جهت طراحي يك جمه كنندة تك بيتي كافي است .

۵-۴ گزارش PAD

پرسش دوم در اين گزارش پاسخ داده مي شود . اين گزارش در مورد محل ترمينالهاي I/O طرح ما با در نظر گرفتن پينهاي بسته توضيح مي دهد .
در اين جدول قسمتي از اطلاعات طرح ما قرار دارد

Comp name	Pin number
CARRY-INPUT	P44
CARRY-OUTPUT	P25
INPUTO	P45
INPUT	P46
SUM	P25

تخصيص پينها با آنچه در فايل UCF قرار داديم بايد يكي باشد .
در زير گزارش ليستي از تخصيص پينها وجود دارد كه مي تواند در يك (PCF) Physical File Constraint استفاده شود .

```
COMP  ` CARRY-INPU`T LOCATE = SITE `P44`;
COMP  `CARRY-OUTPUT`LOCATE = SITE `P26`;
COMP  `INPUT1`LOCATE = SITE `P45`;
COMP  `INPUT `LOCATE = SITE `P46 `;
COMP  `SUM ` LOCATE = SITE ` P25 `;
```

PCF ها از SYNTAX متفاوتي نسبت به UCF ها استفاده مي كنند .
Foundation implementation را RUN كنيد و سپس از اسامي پينهاي يافت شده در گزارش جهت ايجاد يك FUC استفاده كنيد . به احتمال ١٠٠% اختصاص پينها كه توسط كامپايلر صورت مي گيرد با آنچه شما مي خواهيد يكي نخواهد بود . ولي شما مي توانيد فايل را آنطور كه مي خواهيد ويرايش كنيد .

٥-٥ گزارش تأخير آسنكرون

تأخير توزيع براي هر سيگنال مسير يابي شده ، در اين گزارش ليست شده اند .

٥-٦ گزارش تأخير منتج شده از Layout

اين گزارش هر مسير جايگذاري و مسير يابي را كه محدوديت زماني مورد نظر را از بين ببرد گزارش مي دهد .
مثلاً ، چنانچه مدار شما بايد با فركانس 50MHZ كار كند (تمام عمليات منطقي در 20 ns يا كمتر بايد كامل شوند) و مسيري يافت شود كه داراي تأخير بزرگتر از 20^{ns} باشد ، در گزارش ذكر مي شود .

٥-٧ گزارش توليد فايل bit

اين گزارش تمام موارد و انتخابهاي مؤثر در هنگام توليد يك فايل باينري را يادداشت مي كند . هرخطائي كه در طي توليد باينري اتفاق مي افتد نيز در اين گزارش ليست مي شود .
حال مي توانيم مقتدير سه ورودي را به جمع كننده اعمال كنيم . انطباق ترمينال ورودي مدار جمع كننده و پينهاي XC4005XL در جدول زير نشان داده شده است .

ADDER TERMINAL	XC4005CL PIN	
CARRY-INPUT	44	
INPUTO	45	
INPUT1	46	
NOT Used	47	
NOT Used	48	
NOT Used	49	
NOT Used	32	
NOT Used	34	

همچنين فايل UCF . 40 – add1 ، خروجي هاي carry – out را به پينهاي 25 و 26 از XC4005XL FPGA اختصاص مي دهد . پين 25 به SO از Segment – 7 متصل مي شود در حاليكه پين 26 و S1 را device مي كند .
چنانچه همه چيز را به درستي انجام داده باشيد ، تست طرح download شدة شما بايد نتايج شبيه سازي و جدول حقيقت براي جمع كنندة تك بيتي منطبق باشند .
توضيحات اين قسمت نشان مي دهد كه چگونه مي توان بردهايي براساس FPGA و CPLD ساخت . اين بردها قابليت هاي بردهاي XC40 و XS95 تجاري در دسترس را تا حد امكان كپي مي كنند .
مهمترين جزء برد ، يك XC95108 CPLD يا XC4005XL FPGA مي باشد . FPLD با فايلهاي باينري كه توسط نرم افزار XILINX F1 ايجاد مي گردد ، بار مي شود . اين فايلهاي باينري از طريق پورت موازي كامپيوتر در برد ، download مي شود . اين پورت موازي همچنين براي اعمال سيگنالهاي تست

download مدار منطقي در برد ، استفاده مي شود . قسمت seven –segment وصل شده به FPLD يك تصوير از چگونگي كاركرد مدار ، آماده مي كند ، يك RAM استاتيك 32KB نيز به FPLD براي آماده كردن ديتاي ذخيره شدة خارجي ، متصل مي شود .
سپس اجزاي ديگري نيز براي آماده كردن كلاك ها و قدرت تنظيم شده براي باقي قسمتها وجود دارند .

۶- نتیجه گیری

به کمک طراحی با FPGA میتوان بسیاری از مشکلات ناشی از طراحی بر روی برد را از بین برد . علاوه بر از بین بردن مشکلات ناشی از سیم کشی اشتباه و یا جابجا گذاشتن قطعات باعث تسریع در ساختن مدار مورد نظر مي شود.

۷- مراجع

1. Van Den Bout, Prentice Hall, (February 1998),” The Practical Xilinx Designer Lab Book”pp.50-70,pp.90-120,pp.270-290
2. http://www.chipcenter.com/pld/products_001-500/pldp496.htm
3. http://www.fpga-faq.com/archives/authors_u.html
4. http://www.xilinx-china.com/xlnx/xil_prodcat_product.jsp?title=ss_vir
5. http://www.cic.edu.tw/research/Xilinx/xupmirror/XESS/ho04000.html
6. http://www.dacya.ucm.es/horten/dci/manualXs401.3.pdf