

## معرفی FPGA و کاربردهای آن

### چکیده :

در این مقاله سعی شده است مدارات مجتمع FPGA و مشخصات آنها توضیح داده شود . این مدارهای مجتمع جدیداً بجای اجرای مدارهای دیجیتال توسط آی سی های متداول ( TTL یا CMES ) مورد استفاده قرار می گیرد و بسیاری از مشکلات ناشی از سیم کشی و خطایابی را از بین برده است .

### ۱-مقدمه

شاید تا بحال مدارهای منطقی را بوسیله گیت های AND , OR , NOT ، ساخته اید . برای ساخت چنین مدارهایی ( از قبیل شمارنده ها ، کنترل کننده ها ، و ... ) ابتدا باید تعریفی از مدار در دسترس باشد سپس با توجه به منطق اعداد دودویی یک جدول صحت برای مدار تشکیل می شود و حالتهای مختلف مورد بررسی قرار می گیرد سپس با توجه به جدول صحت مدار توسط گیت های منطقی مانند AND , OR , NOT , NAND ، ... طراحی می شود پس از این مرحله نوبت به پیاده سازی مدار بر روی برد توسط آی سی های منطقی می رسد و همانطور که می دانید یکی از وقتگیرترین و خسته کننده ترین مرله ساخت یک مدار همین قسمت است . بعد از این مرحله نوبت به تست مدار جهت اطلاع از درستی مراحل کار کرد مدار می رسد . اگر در یکی از مراحل قبل دچار اشتباه شده باشیم مطمئناً در مرحله تست مدار دچار مشکل می شویم . در صورت اشتباه در مراحل قبل باید تمام مراحل را از آخر به اول یک به یک چک کنیم تا بتوانیم اشتباهات احتمالی موجود در نحوه بستن و سیم کشی مدار ، طراحی مدار از روی جدول صحت و درستی جدول صحت را برطرف کنیم . با توجه به مطالب گفته شده حتماً به این نکته اذعان خواهید داشت که بیشترین اشتباهات در مرحله سیم کشی و بستن مدار بر روی برد پیش خواهد آمد . ممکن است سیمی در جای اصلی وصل نشده باشد و یا ممکن است یک پایه به هیچ جا متصل نباشد و یا اشتباهات مشابه اینها . . . از طرف دیگر می دانیم که هر چه مدار بزرگتر و پیچیده تر باشد اشتباهات بیشتر و عیب یابی مشکلتر خواهد بود . اینجاست که نقش آی سی های FPGA نمایانتر می شود . آی سی های که با داشتن انواع گیت های مختلف درون خود

بسیاری از مشکلات ناشی از عیب یابی مدارهای منطقی را برطرف کرده است. در قسمتهای بعد به بررسی و معرفی این آی سی ها می پردازیم.

## ۲- مدار مجتمع منطقی قابل برنامه ریزی :

این آی سی ها که بحث اصلی این مقاله به شمار می روند با نامهای FPGA یا CPLD ها یا نام کلی FPLD ها شناخته می شوند. FPLD ها شامل گیتهای منطقی و ابزارهایی جهت اتصال آنها در یک مدار مجتمع واحد می باشد. با استفاده از نرم افزار های موجود مشخص می شود که گیتهای داخل یک device چگونه می تواند جهت ساخت یک مدار منطقی به هم وصل می شوند. خروجی برنامه یک فایل باینری است که داخل FPLD ، download می شود تا باعث شود FPLD مانند یک مدار منطقی عمل کند. سپس FPLD برنامه ریزی شده می تواند داخل یک مدار بزرگتر قرار گیرد.

## ۳- اصول مدار داخلی FPLD ها :

این device در واقع از PLA ها (programmable logic Array) نتیجه شده اند. ساختمان یک PLA شامل چند گیت AND ، چند گیت OR ، و inverter ها می باشد که از طریق آرایه سوئیچها به هم متصل می شوند. در یک PLA هر ورودی و معکوس شده آن توسط سیستمهای افقی از داخل یک گیت AND عبور داده می شوند. گیتهای AND ورودیها را از طریق به هم بستن سیستمهای افقی و عمودی دریافت می کنند. سیستمهای عمودی خروجیهای گیتهای AND را وارد آرایه گیتهای OR می کنند. در این قسمتها سیمهای افقی که ورودیها را به گیتهای OR می برند به این سیمهای عمودی متصل می شوند. گاهی اوقات از آرایه منطقی قابل برنامه ریزی (PAL) ساده تری استفاده می کنند.

مدارات PLA و PAL برای منطق ترکیبی موثرند ولی برای منطق ترتیبی بدون اضافه کردن فلیپ فلاپهای خارجی قابل استفاده نیستند. در FPGA ها سه LUT و دو فلیپ فلاپ و بعضی مدارهای اضافی جهت ایجاد یک CLB به هم متصل می شوند. سپس این CLB ها توسط یک ماتریس سوئیچ قابل برنامه ریزی (programmable switch matrices) (PSM) به هم متصل می شوند. پینهای FPGA I/O می توانند به PSM ها و CLB ها و یا حتی ماتریس مسیر یابی خودشان متصل شوند. تعداد CLB نمی تواند یک ارتباط مستقیم با دنیای خارج داشته باشد. کار PSM ها بدین صورت است که پس از طراحی مدار منطقی گیتهای لازم جهت ایجاد مدار موردنظر را در CLB ها به هم متصل می کند یعنی تمام سیم بندی ها و اتصالات در FPGA ها در داخل IC انجام می شوند. یعنی بطور فیزیکی نمی توان اتصالی ایجاد یا آن را تغییر داد. در عوض اتصالات بطور الکتریکی برقرار می شوند. در نسلهای اولیه FPGA ها PSM ها با قرار دادن فیوزهایی در هر نقطه اتصال سیمهای افقی و عمودی ذکر شده بوجود می آمد با اعمال ولتاژ مثبت به سیمهای عمودی و افقی مدار موردنظر حاصل می شد. به این ترتیب که ولتاژ مثبت فیوز واقع در نقطه اتصال سیمهای عمودی و افقی را می سوزاند که این عمل باید به دفعات زیاد جهت از بین بردن اتصالات ناخواسته در مدار مورد نظر انجام می شد. در انتهای این روند تنها اتصالاتی که جهت ایجاد طرح مدار مورد نیاز بودند باقی می ماندند. ایراد فیوزها این است که وقتی یک مرتبه سوختند قابل برگشت به وضعیت اولیه شان نیستند. بنابراین چنانچه خطایی رخ می داد device قابل برنامه ریزی مزبور کنار گذاشته شده ، device جدیدی برنامه ریزی می شد لذا اگر اتصالات قابلیت پاک شدن و دوباره برنامه ریزی شدن را داشته باشند بسیار ارزانتر و قابل قبول تر خواهد بود. به این ترتیب که این device ها دارای سوئیچ های قابل برنامه ریزی به جای

فیوزهای مذکور هستند . هر سوئیچ توسط يك المان ذخیره سازی که حالت باز بودن یا بسته بودن سوئیچ را در خود ذخیره می کند کنترل می شود . تغییر مقادیر ذخیره شده در این المانهای ذخیره سازی باعث تغییر حالت سوئیچ و توابع device قابل برنامه ریزی می شود . به این ترتیب نیاز به خرید IC های جدید جهت ایجاد هر طرح جدید کاملاً از بین می رود .

#### ۴- روند طراحی يك مدار قابل برنامه ریزی:

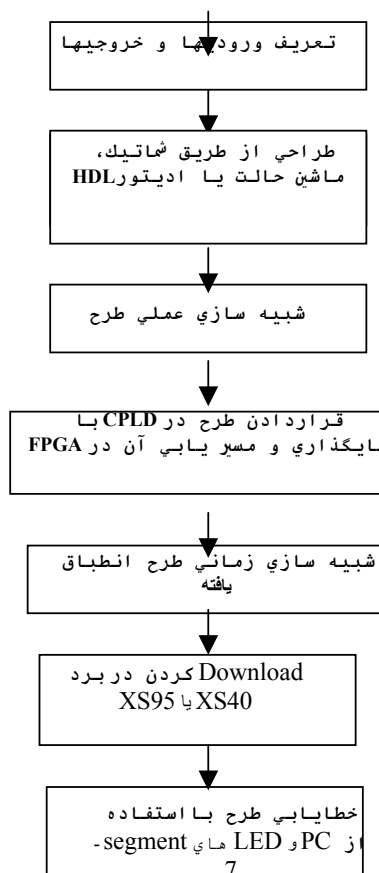
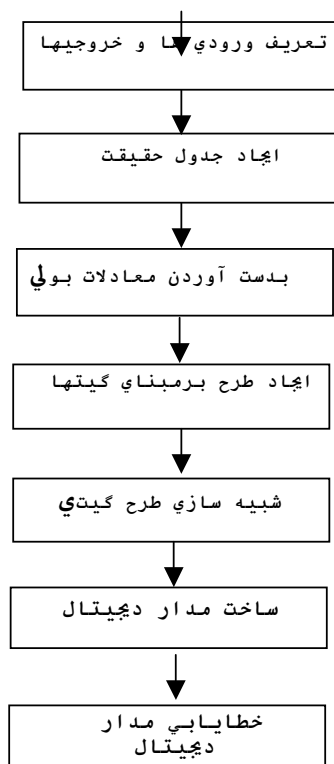
نرم افزار XILINX Foundation محیطی جهت ایجاد برنامه هائی برای توصیف طرح منطقی مورد نظر می باشد . روند طراحی با استفاده از نرم افزار Foundation به این ترتیب است :

۱-۴ طرحهای دیجیتال مورد نظر ، با استفاده از ادیتور شماتیک یا ادیتور ماشین حالت یا ایتور متنی HDL (ABEL یا VHDL) وارد می شوند . طرح مورد نظر می تواند توسط یکی از این ادیتور ها ی ترکیبی از همه آنها ایجاد شود .

۲-۴ يك سیمولاتو ( شبیه ساز ) عملی عملکرد يك طرح کامپایل شده را چک می کند و به شما اجازه می دهد تا نتایج را ببیند و صحت یا عدم صحت نتایج را بررسی کنید . در صورت بروز هر گونه خطائی می توان به محیط ادیتوری شماتیک ، HDL یا ماشین حالت برگشته خطاها را اصلاح کنید .

#### مراحل کلی طرح

#### روند طراحی برای XC9500/XC4000



شکل ۱: تغییرات جریان طراحی دیجیتال هنگام استفاده از نرم افزار XILINX Foundation برای XC9500 CPLD یا XC4000 FPGA :

۳-۴ ابزار اجرای نرم افزار Foundation ، ابتدا لیست گیتها و اتصالات ایجاد شده را به يك فایل با فرمت باینري تبدیل مي کند که جهت برنامه ریزی FPLD استفاده خواهد شد . در این مرحله است که يك device خاص جهت برنامه ریزی کردن باید مشخص شود ، مانند خانواده هاي XC95108 و XC400 ، 5XL . برای 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 این ترتیب که گیتها داخل CLB هاي مشخص قرار مي گیرند و مسیریابی و سیم پیچی با استفاده از PSM ها انجام مي شوند .

۴-۴ بعد از اینکه ابزار اجرائي نرم افزار FOUNDATION تأخیرهاي مربوط به گیتها و عمل مسیریابی را مشخص کردند ، شبیه سازی خصوصیات زمانی طرح با يك mapping مشخص در يك FPLD انجام مي شوند .

۵-۴ با وارد کردن ورودی ها به يك برد XS95 یا XS40 از طریق کابل پورت موازي کامپیوتر عمل خطایا بي ( Debugging ) انجام مي شود .

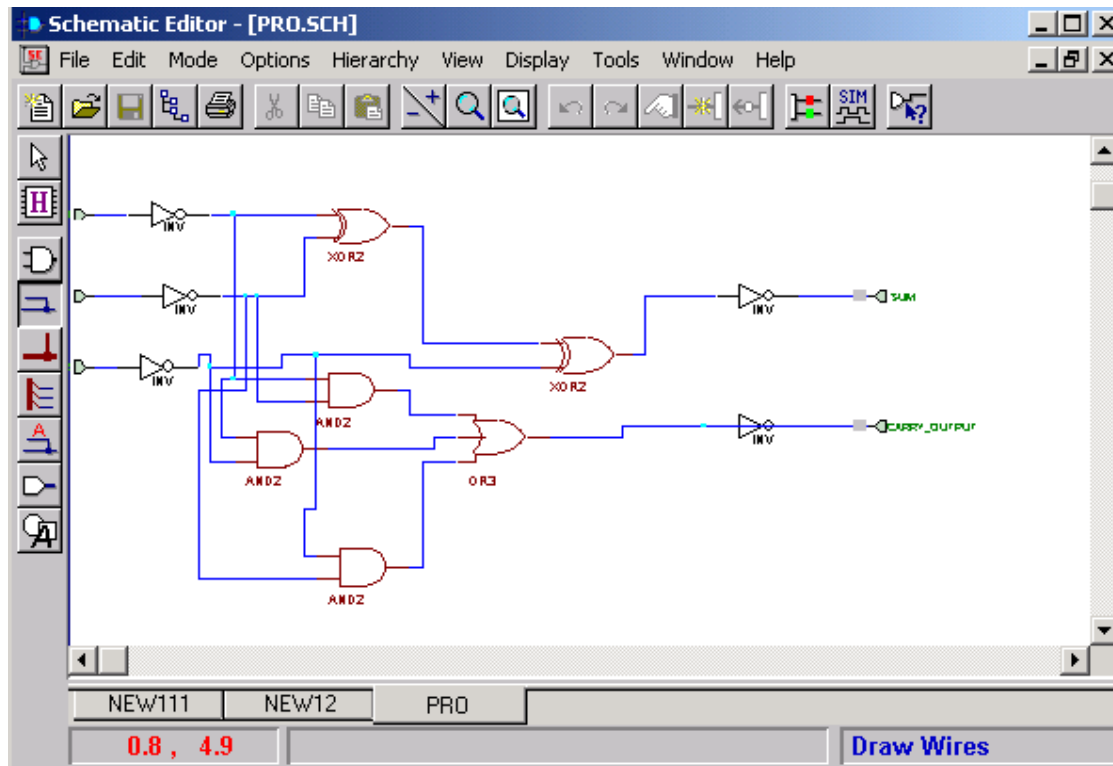
#### ۵- مثالی در مورد نحوه برنامه ریزی يك برد XS40 (جمع کننده تك بيتي) :

جدول حقیقت برای يك جمع کننده ۲ بيتي همراه باکری ورودی و خروجی :

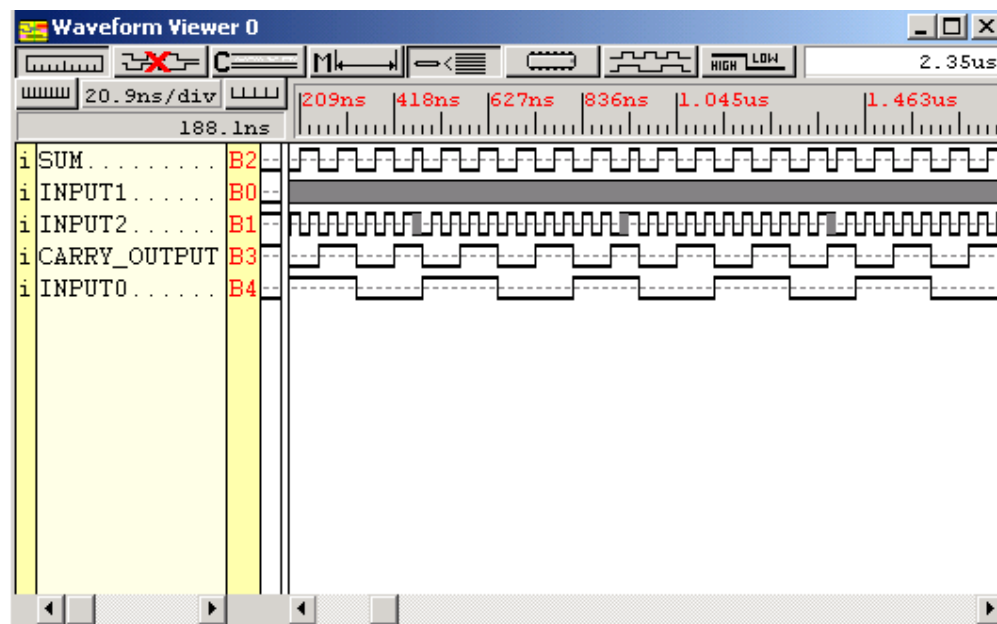
Input 1	Input 0	Carry input	Sum output	Carry output
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ابتدا يك پروژه در مورد شماتيك كه 40 - ADDL نام دارد تعريف مي كنيم كه طرح را براي XC4005XL ایجاد خواهد کرد . ادیتور شماتيك را فعال مي كنيم و گیتها را مطابق شکل A متصل مي كنيم . شکل A يك جمع کننده تك بيتي را با اضافه کردن با فرهای ورودی و خروجی نشان مي دهد . وقتی شماتيك كامل شد ، آن را بعنوان ADDL - SCH ذخیره 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 . ( همچنین با انتخاب Options - Integrity test مي توان درستي همه چیز را قبل از ادامه عملیات چك کرد . ) سپس از ادیتور روی كلید در Flow tab از صفحه project manager كلیك مي كنيم .

با انتخاب Signal - Add Signals از منو ، سیگنالهاي ورودی و خروجی را به صفحه waveform viewer اضافه مي كنيم . با انتخاب signal - Add stimulators.. از منو ورودی ها را به پایین ترین سه بیت باینري شماره ده ( BC0 , BC1 , BC2 ) متصل مي كنيم . در نهایت روی دکمه SIMULATOR در toolbar Logic كلیك مي كنيم . شکل موجهای نشان داده شده در شکل B ظاهر مي شوند



شکل A



شکل B

( می توان از دکمه و scrollbar در صفحه waveform viewer جهت باز کردن و متمرکز کردن شکل موجها استفاده کرد . ) چنانچه با دقت بررسی کنیم می بینیم که نتیجه شبیه سازی با آنچه در جدول حقیقت نمایش داده شده بود ، مطابقت دارد . بعد از چک کردن عملکرد جمع کننده تک بیتی

، مي توانيم از سيمولاتور ( شبیه ساز ) خارج شده و فايل را براي XC05xl FPGA کامپايل کنیم .  
در ابتدا لازم است که ترمینالهاي ورودی و خروجی طرح را به پینهاي فیزیکی I/O در XX4005CL اختصاص دهیم . به این منظور از ادیتور HDL جهت ایجاد فايل UCF . 40 - ADDL و افزودن خطوط زیر به آن استفاده مي کنیم

```
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 ، روی دکمه جهت شروع کامپايل کردن طرح کلیک مي کنیم . سپس در صفحه IMPLEMENT Design که ظاهر خواهد شد . روی دکمه Option کلیک مي کنیم . در صفحه Option ، در قسمت User Constraints ، نام UCF اي را که ایجاد کرده ایم وارد مي کنیم .

مثلاً UCF . 40 - ADDL - 40\XC\PROJ\ADDL - 40 . ( یا از دکمه Browse جهت انتخاب UCF خود استفاده مي کنیم . ) سپس OK را براي بازگشت به صفحه implement Design کلیک کرده و RUN را در این مرحله ، باید يك فايل . 40 - BIT . ADDL داشته باشیم که بتوانیم در برد XS40, download کنیم .

1 - چه فضایی از XC4005XL FPGA جهت ساخت يك جمع کننده تك بيتي اشغال مي شود ؟

2 - آیا کامپایلر ، ورودی ها و خروجی ها را به پینهاي که ما درخواست کرده بودیم اختصاص مي دهد ؟  
ما مي توانیم پاسخ این سئوال را از فايلهاي گزارش که توسط Foundation implementation ایجاد مي شود در يابیم . در قسمت سمت راست صفحه , reports tab project manager را انتخاب مي کنیم سپس روی آیكون implementation report files ، دوبار کلیک مي کنیم . صفحه report browser ظاهر خواهد شد . روی هر کدام از آیكونها دوبار کلیک مي کنیم . تا محتویات مربوط به آن گزارش را مشاهده کنیم . این گزارشها ، اطلاعات زیر را خلاصه مي کنند :

### ۵-۱ گزارش ترجمه

هر مشکلي که در هنگام تبدیل netlist به فرمت داخلی که توسط ابزار foundation implementation استفاده مي شود ، مشاهده مي شود ، در این قسمت لیست مي شود .

در این قسمت چک کردن قوانین طراحی و فايل UCF جهت خطايابی ، نیز صورت مي گیرد . معمولاً بیشتر خطایی که در این فايل مي بینید نتیجه عبارات قابل قبولى است که وارد کرده اید .

### ۵-۲ گزارش انطباق

در اینجا در مورد اینکه چه نوع بهینه سازی هائی روی netlist انجام شده است ، اطلاعاتی بدست خواهید آورد . گیتهاي منطقی جاچا و اضافی مي شوند بطوریکه بدون تغییر عملکرد در مدار ، آن را بهینه سازی مي کنند . يك مثال از جاچائی منطقی وقتی اتفاق مي افتد که يك گیت را در طرح قرار مي دهید ولی خروجی را براي هیچ چیز استفاده نمي کنید . مثال دیگر يك گیت AND با يك ورودی متصل شده به " 0 " منطقی ( GND ) است که نتیجه خروجی آن همواره LOW است .

همچنین گزارش نشان می دهد که گیت های منطقی چگونه گروه بندی شده در LUT های CLB های FPGA قرار می گیرند .

### ۳-۵ گزارش جایابی و مسیر یابی

این گزارش ، انتخاب های را که ما جهت اثر گذاری به روند جایگذاری و مسیر یابی انجام داده ایم ، یادداشت می کند . همچنین خطاها و اختلالات را لیست کرده و زمان صرف شده در جایگذاری و مسیر یابی های مختلف را ثبت می کند . این گزارش همچنین مشخصات آماری گوناگون عملکرد زمانی مسیر یابی را گزارش می دهد .

ولی مهمتر از همه ، این گزارش پاسخ سئوال اول ما را در مورد اینکه حجمی از XC4005XL جهت ساخت یک جمع کننده ۱ بیتی مورد استفاده قرار می گیرد ، خواهد داد .

خلاصه استفاده از فضای 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 کننده تک بیتی دارای ۳ ورودی و ۲ خروجی است . بنابراین جمع کننده ۵ بلوک از ۱۱۲ بلوک I/O ( IOB's ) قابل دسترسی در XC4005XI FPGA را استفاده می کند .

البته فقط ۶۱ بلوک از این IOB ها واقعاً به پین های فیزیکی روی بسته ۸۴ پینی PLCC متصل هستند .

بنابراین تقریباً ۸٪ از I/O ها قابل دسترسی ، توسط جمع کننده های تک بیتی استفاده می شوند .

هر دو مدار SUM و CARRY - OUTPUT در طرح ها ، یک خروجی و سه ورودی دارند . هر مدار باید در یک 4LUT پینی قرار گیرد بنابراین انتظار داریم که از دو LUT برای مدار جمع کننده تک بیتی خود استفاده کنیم . و در هر CLB هم دو 4LUT پینی وجود دارد بنابراین فقط یک CLB جهت طراحی یک جمع کننده تک بیتی کافی است .

### ۴-۵ گزارش PAD

پرسش دوم در این گزارش پاسخ داده می شود . این گزارش در مورد محل ترمینالهای I/O طرح ما با در نظر گرفتن پینهای بسته توضیح می دهد . در این جدول قسمتی از اطلاعات طرح ما قرار دارد

Comp name	Pin number
CARRY-INPUT	P44
CARRY-OUTPUT	P25
INPUT0	P45
INPUT	P46
SUM	P25

تخصیص پینها با آنچه در فایل UCF قرار دادیم باید یکی باشد . در زیر گزارش لیستی از تخصیص پینها وجود دارد که می تواند در یک ( Physical File Constraint ) استفاده شود .

```
COMP `CARRY-INPUT` 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 متفاوتی نسبت به UCF ها استفاده می کنند .  
 Foundation implementation را RUN کنید و سپس از اسامی پینهای یافت شده در  
 گزارش جهت ایجاد يك FUC استفاده کنید . به احتمال %۱۰۰ اختصاص  
 پینها که توسط کامپایلر صورت می گیرد با آنچه شما می خواهید یکی  
 نخواهد بود . ولی شما می توانید فایل را آنطور که می خواهید ویرایش  
 کنید .

#### ۵-۵ گزارش تأخیر آسنکرون

تأخیر توزیع برای هر سیگنال مسیر یابی شده ، در این گزارش لیست شده  
 اند .

#### ۶-۵ گزارش تأخیر منتج شده از Layout

این گزارش هر مسیر جایگذاری و مسیر یابی را که محدودیت زمانی مورد  
 نظر را از بین ببرد گزارش می دهد .  
 مثلاً ، چنانچه مدار شما باید با فرکانس 50MHZ کار کند ( تمام عملیات  
 منطقی در 20 ns یا کمتر باید کامل شوند ) و مسیری یافت شود که دارای  
 تأخیر بزرگتر از 20<sup>ns</sup> باشد ، در گزارش ذکر می شود .

#### ۷-۵ گزارش تولید فایل bit

این گزارش تمام موارد و انتخابهای مؤثر در هنگام تولید يك فایل  
 باینری را یادداشت می کند . هرخطایی که در طی تولید باینری اتفاق  
 می افتد نیز در این گزارش لیست می شود .  
 حال می توانیم مقتدیر سه ورودی را به جمع کننده اعمال کنیم .  
 انطباق ترمینال ورودی مدار جمع کننده و پینهای XC4005XL در جدول زیر  
 نشان داده شده است .

ADDER TERMINAL	XC4005CL PIN	
CARRY-INPUT	44	
INPUT0	45	
INPUT1	46	
***NOT Used***	47	
***NOT Used***	48	
***NOT Used***	49	
***NOT Used***	32	
***NOT Used***	34	

همچنین فایل UCF . add1 - 40 ، خروجی های carry - out را به پینهای 25  
 و 26 از XC4005XL FPGA اختصاص می دهد . پین 25 به SO از Segment - 7  
 متصل می شود در حالیکه پین 26 و S1 را device می کند .  
 چنانچه همه چیز را به درستی انجام داده باشید ، تست طرح download شده  
 شما باید نتایج شبیه سازی و جدول حقیقت برای جمع کننده تک بیتی  
 منطبق باشند .

توضیحات این قسمت نشان می دهد که چگونه می توان بردهایی براساس  
 FPGA و CPLD ساخت . این بردها قابلیت های بردهای XC40 و XS95 تجاری  
 در دسترس را تا حد امکان کپی می کنند .

مهمترین جزء برد ، يك CPLD XC95108 یا FPGA XC4005XL می باشد . FPLD  
 با فایل های باینری که توسط نرم افزار XILINX F1 ایجاد می گردد ، بار  
 می شود . این فایل های باینری از طریق پورت موازی کامپیوتر در برد  
 ، download می شود . این پورت موازی همچنین برای اعمال سیگنال های تست

download مدار منطقی در برد ، استفاده می شود . قسمت seven -segment وصل شده به FPLD یک تصویر از چگونگی کارکرد مدار ، آماده می کند ، یک RAM استاتیک 32KB نیز به FPLD برای آماده کردن دیتای ذخیره شده خارجی ، متصل می شود . سپس اجزای دیگری نیز برای آماده کردن کلاک ها و قدرت تنظیم شده برای باقی 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 جایجا گذاشتن قطعات باعث تسریع در ساختن مدار مورد نظر می 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](http://www.chipcenter.com/pld/products_001-500/pldp496.htm)
3. [http://www.fpga-faq.com/archives/authors\\_u.html](http://www.fpga-faq.com/archives/authors_u.html)
4. [http://www.xilinx-china.com/xlnx/xil\\_prodcat\\_product.jsp?title=ss\\_vir](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>