



جلسه اول

آشنایی با فضای گرافیک نرم‌افزار MAXPLUS و ریختن در برد CIC 310

هدف از ارائه آزمایشگاه مدار منطقی، آشنایی دانشجویان با چگونگی کار با تراشه‌های دیجیتال و همچنین چگونگی طراحی و پیاده‌سازی مدارات ساده ترکیبی و ترتیبی دیجیتال است. قبل از شروع کار در آزمایشگاه لازم است آشنایی اولیه در مورد انواع تکنولوژی آی سی‌های منطقی و همچنین نحوه کار با آنها داشته باشیم.

آشنایی با IC های سری TTL و CMOS

طراحی مدارهای دیجیتال بر پایه عملیات منطقی چون NOR, NAND, OR, AND, ... انجام و پیاده‌سازی می‌شود. معمولاً تعدادی از دروازه‌های یکسان در یک مدار مجتمع (IC) و به صورت یک جا ساخته می‌شوند که با تکنولوژی TTL و CMOS به بازار ارائه می‌شوند. به منظور آشنایی با این قطعات و نحوه استفاده از دیتا شیت این نوع از مدارات مجتمع توجه شما را به نکات ذیل جلب می‌کنیم.

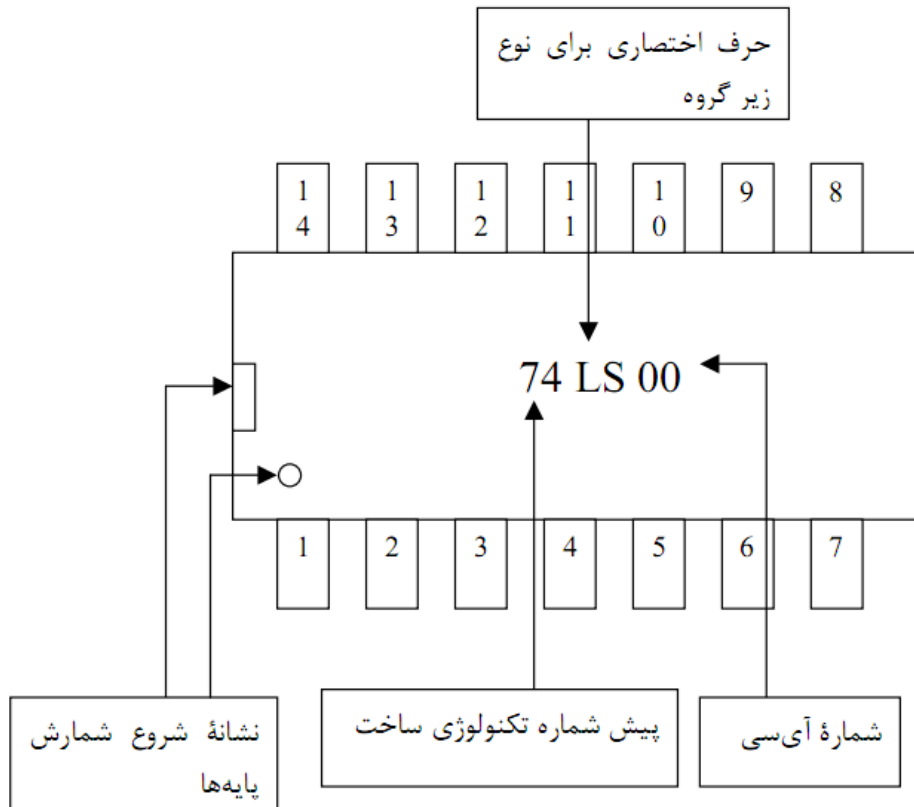
مفهوم آن	حرف اختصاری
نمونه CMOS	C
نمونه سریع	F
نمونه سریع و پر قدرت	H
نمونه شاتکی	S
نمونه سریع CMOS آی سی TTL که با CMOS سازگار است	HC
نمونه سریع CMOS آی سی TTL که با TTL سازگار است	HCT
کم مصرف	L
کم مصرف با ورودی شاتکی	LS
نمونه پیشرفته کم مصرف با ورودی شاتکی	ALS

سطح تغذیه در خانواده TTL ولتاژ نامی 5+ و در خانواده CMOS از 3+ تا 15+ ولت تغییر می‌کند. همچنین سطوح منطقی صفر و یک برای هر کدام از این دو خانواده در جدول زیر ذکر شده است.

CMOS	TTL	
بیش از دو سوم تغذیه	بیش از 2 ولت	سطح منطقی 1
کمتر از یک سوم تغذیه	کمتر از 0.8 ولت	سطح منطقی 0
بین یک سوم و دو سوم تغذیه	بین 0.8 - 2 ولت	نا معتبر

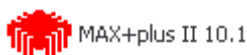


برای شناخت یک تراشه دیجیتال از نظر نوع تکنولوژی و نوع کاربرد آن ، می توان از شماره مخصوصی که روی هر آی سی نوشته شده است استفاده کرد و با مراجعه به کتابچه CMOS یا TTL با عملکرد آن تراشه بصورت کامل آشنا شد. متداولترین آی سی TTL با پیشوند 74 و آی سی CMOS با پیشوند 4 متمایز می گردند. در شکل نحوه شناخت تراشه و ترتیب قرارگرفتن پایه های آن آورده شده است.



دستور العمل سریع استفاده از MAX PLUS II

قسمت گرافیکی



- 1-RIGHT CLICK → ENTER SYNBOLE → SELECT ELEMENTS FROM LIBRARIES →CONNECT ELEMENTS
- 2- FILE →PROJECT → SET PROJECT TO CURRENT FILE
- 4- FILE →PROJECT → SAVE & CHECK
- 5- FILE →PROJECT → SAVE & COMPILE
- 6- ASSIGN → DEVICE → EPF8282ALC84-4 →OK
- 7- ASSIGN → GLOBAL PROJECT DEVICE OPTIONS... →
- 8- ASSIGN → GLOBAL PROJECT LOGIC SYNTHESIS →
- 9- MAX+PLUS II → FLOORPLAN EDITOR → DRAG & DROP YOUR DESIGN ELEMENTS TO DESIRED PIN WITH ATTENTION AT CIC – 310 BOARD FACILITIES
- 10- FILE →PROJECT → SAVE & COMPILE

دیدن مشخصه موجی شکل طراحی انجام شده با مقدار دهی به پین های ورودی.

- 1- MAX+PLUS II →WAVEFORM EDITOR → RIGHT CLICK → INSERT NODE → CLICK LIST → SELECT INPUT OR OUTPUT PINE NAME → CLICK OK
- 2- SELECT NAME → ASSIGN INPUT VALUE

در پایان توجه شما را به نکات زیر در خصوص استفاده از برد CIC-310 جلب می کنیم:

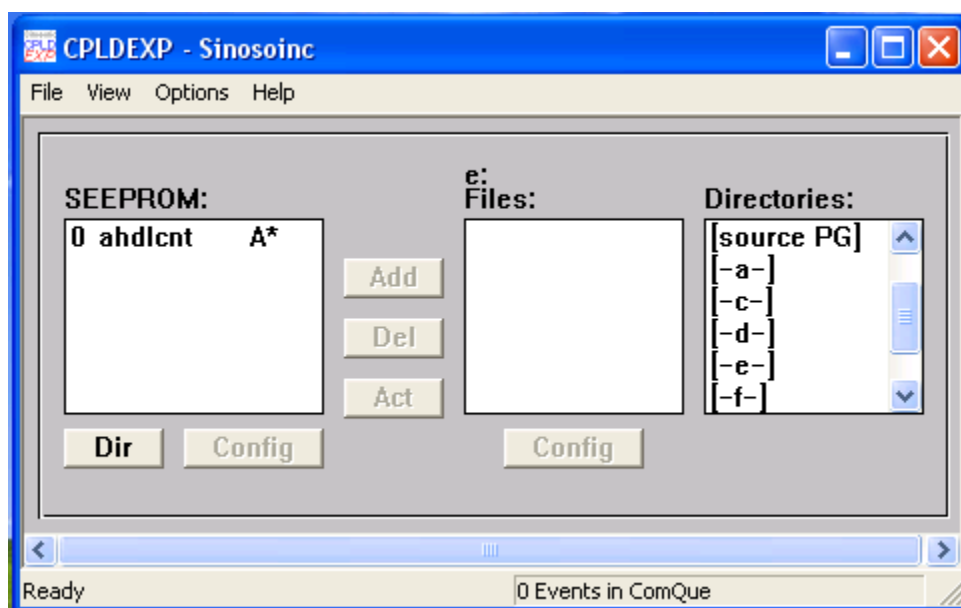
- ۱- پایه های نظیر چیپ FLEX برای هر قطعه در زیر یا کنار آن قطعه درج شده است که در هنگام اختصاص پایه های ورودی و خروجی در FLOOR PLAN می توانید از آن استفاده کنید.
- ۲- از دیپ سویچ ها ی موجود در برد CIC- 310 برای ورودی بیت ها به پایه های چیپ FLEX می توانید استفاده کنید.
- ۳- از LED ها می توانید برای نمایش خروجی استفاده کنید.
- ۴- از سون سگمنت ها که بصورت کاتد مشترک است و با جمپر به پایه های چیپ FLEX اتصال می یابد، برای نمایش اعداد استفاده کنید.



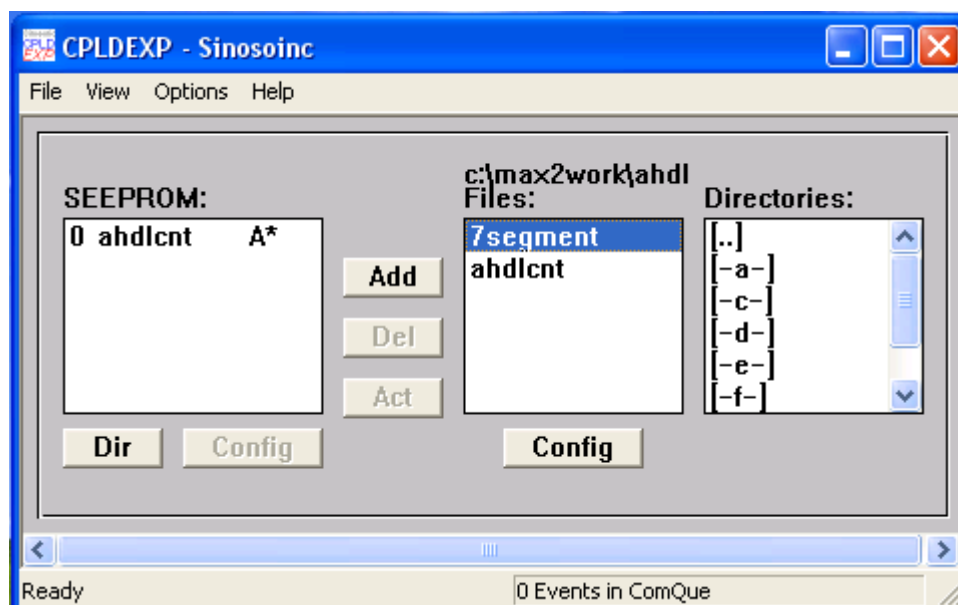
- ۵- کلاک به صورت دستی ، اتومات قابل تنظیم قابل ارائه به چیپ FLEX می باشد.
- ۶- از کی برد ۴*۴ برای ورودی می توان استفاده کرد.
- ۷- از دکمه Reset برای راه اندازی مجدد چیپ FLEX از روی برنامه انتخابی EEPROM استفاده می شود.
- ۸- برنامه که از طریق پروگرامر در چیپ FLEX ریخته می شود برنامه جاری آن می گردد. حال چنانچه بخواهیم این برنامه در راه اندازی مجدد درون چیپ FLEX اجرا گردد، باید آن را در EEPROM ذخیره (ADD) و آن را فعال کنیم. (ACT) در غیر این صورت با ری ست یا خاموش روشن شدن برنامه جاری از چیپ FLEX پاک می شود و برنامه فعال در EEPROM به عنوان برنامه جاری اجرا می گردد.
- ۹- چنانچه در هنگام ریختن برنامه در چیپ FLEX مشکل دارید استفاده از RESET سخت افزاری و نرم افزاری را از یاد نبرید.

ریختن برنامه آماده در ALTERA FLEX و دیدن خروجی بر روی برد CIC-310

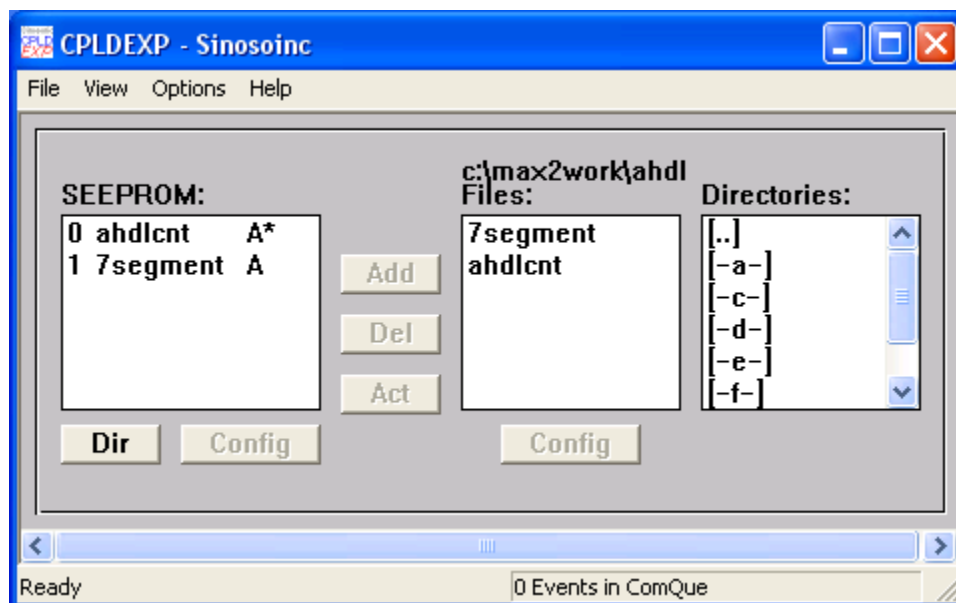
ابتدا برنامه را از روی دستکتاپ اجرا می‌کنیم و از Directories مسیر فایل کامپایل شده را که با پسوند HEX می‌باشد را انتخاب می‌کنیم.



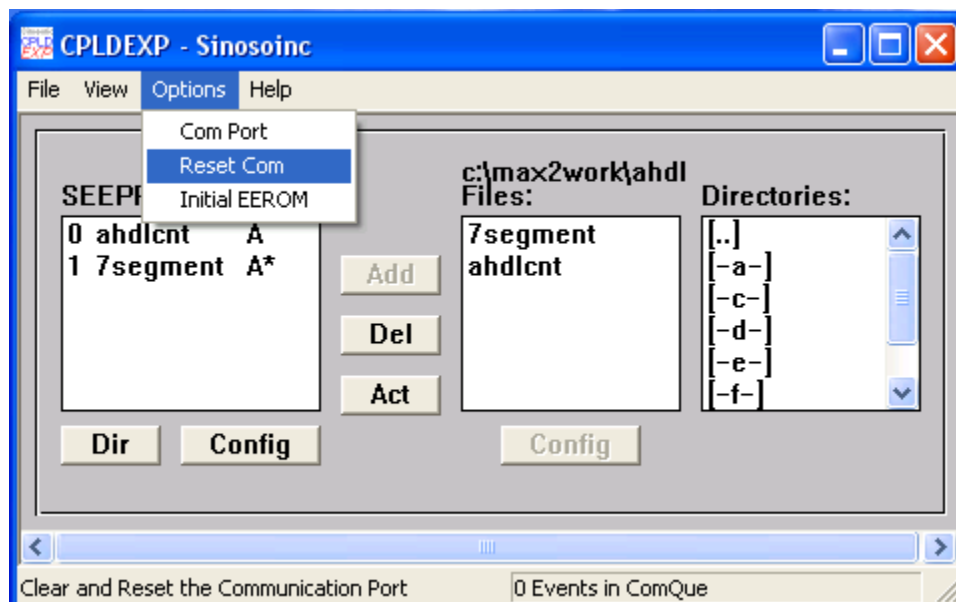
پس برنامه مورد نظر را انتخاب و در حالی که سخت افزار CIC-310 روشن است و تعداد EVENT پایین صفحه صفر می‌باشد را با فشردن CONFIG به چیپ ALTERA وارد می‌کنیم با توجه به این که، با این کار برنامه به حافظه جاری وارد می‌شود برای این که در EEPROM برد وارد شود می‌بایست دکمه ADD را بفشاریم.



برنامه به قسمت SEEPROM وارد می شود.



حال برنامه ای که با * مشخص شده برنامه ایست که پس از RESET برد آن برنامه در حافظه جاری FLEX بار می شود برای اینکه برنامه که ما ADD کردیم در EEPROM ریخته شود می بایست دکمه CONFIG و OK بعد آن را بفشاریم برای اینکه برنامه مورد نظر ما برنامه جاری بعد اولین RESET شود ابتدا آن را انتخاب (آبی رنگ می شود) و سپس دکمه ACT را می فشاریم.



ضمناً از OPTIONS به منظور انتخاب پورت سریال و RESET پورت سریال و همچنین مقدار دهی اولیه به EEPROM استفاده می شود.



آزمایش شماره ۱

دروازه های منطقی

مدارهای منطقی برنامه پذیر (PLD)، مدار مجتمعی با گیت های داخلی است که، از طریق فیوزهای الکترونیکی به هم مرتبط شده اند. برای بدست آوردن ساختار خاص، برنامه ریزی این مدارها از طریق سوزاندن فیوزها یا قطع آنها انجام می شود. کلمه برنامه پذیر مدار، در اینجا روش سخت افزاری است که، ساختار داخلی مدار را مشخص می نماید. PLD شامل آرایه ای از گیت های AND و OR می باشد که به گونه ای به هم متصل شده اند که ساختار AND-OR را، برای پیاده سازی مجموع حاصل ضرب ها، تولید نمایند.

مزیت بکار بردن PLD در طراحی سیستم های دیجیتال این است که، می توانند برای توابع پیچیده مدارهای LSI بکار روند.

Programmable Logic Device (PLD)

Programmable Read-Only Memory (PROM)

نکات زیر را در انجام آزمایشات این درس مد نظر داشته باشید:

- ۱- راهنمای استفاده از نرم افزار MAX+PLUS II را به دقت مطالعه کنید.
- ۲- از راهنمای سریع برای انجام مراحل آزمایش استفاده کنید.
- ۳- به منظور سهولت و تسریع در انجام آزمایشات موارد طراحی و کار در آزمایشگاه را در محیط گرافیکی نرم افزار MAX+PLUS II و قبل از آمدن به آزمایشگاه انجام دهید.
- ۴- در طراحی ها کشیدن جدول کارنو و ذکر مراحل طراحی مهم می باشد و می بایست قبل آمدن به آزمایشگاه در منزل انجام گیرد.
- ۵- شماره پایه های ورودی، خروجی، کلاک دستی، کلاک اتومات، سون سگمنت، پالس و صفحه کلید را در نظر داشته و در طراحی و انتصاب پایه ها در Floorplan آنها را اعمال کنید.
- ۶- خروجی طراحی خود را با استفاده از محیط Waveform مورد بررسی قرار دهید و اشکالات طرح خود را رفع کنید.
- ۷- بدیهی است طراحی که در آن با المان های کمتر توابعی دقیق تر ایجاد شده باشد دارای ارزش بیش تری در کاربری است.
- ۸- خروجی طراحی شما که به صورت HEX است را با توجه به راهنما ریختن برنامه در ALTERA FLEX درون چیپ ریخته و نتیجه را بر روی برد CIC-310 ببینید.

جلسه دوم

آزمایش شماره ۲

دروازه های منطقی

استفاده از زبان سخت افزاری Verilog

مثال: برنامه زیر یک **or2** ورودی را با **Vrilog** پیاده‌سازی می‌کند آن را در محیط متن نرم افزار MAX+PLUS II با پسوند **.v** ذخیره، **Wave Form** آن را ببینید.

```

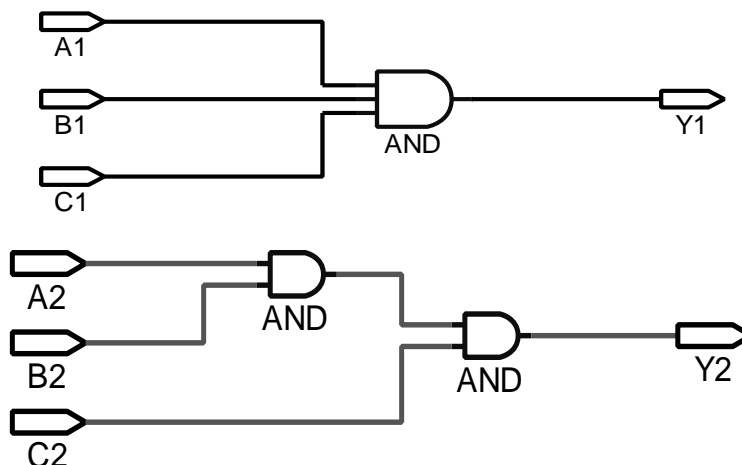
module test_or2 (Y, A, B);
output Y;
input A, B;
reg Y;
always @ (A or B )
if (A==1 || B==1)
Y = 1;
else
Y = 0;
endmodule

```

برنامه نویسی

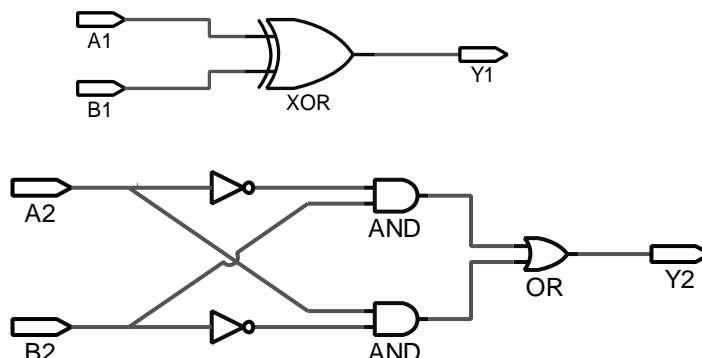
۱- برنامه فوق را به گونه‌ای اصلاح نمایید که **and2** ایجاد گردد. آن را در محیط متن نرم افزار MAX+PLUS II با پسوند **.v** ذخیره **Wave Form** آن را ببینید، سپس آن را در برد **CIC310** ریخته و نتیجه را ببینید.

۲- حال برنامه فوق را به گونه‌ای اصلاح نمایید تا به شکل مدارهای زیر درآید و **Wave Form** آن را با هم مقایسه نمایید. جدول زیر را کامل کنید.



اختیاری:

۳- حال برنامه‌ای بنویسید که کار مدارهای زیر را انجام دهد Wave Form آن را با هم مقایسه نمایید.



کار در آزمایشگاه

آزمایش ۱-۲:

مثال را در برد CIC310 ریخته و نتیجه را ببینید.

آزمایش ۲-۲:

برنامه یک برنامه‌نویسی را در برد CIC310 ریخته و نتیجه را ببینید.

آزمایش ۳-۲:

برنامه دو برنامه‌نویسی را در برد CIC310 ریخته، جدول مقابل را کامل کنید.

تابع Y1 و Y2 چه تفاوت و شباهتی با هم دارند؟

اختیاری:

آزمایش ۴-۲:

به خروجی Waveform نرم افزار برنامه سه برنامه‌نویسی به دقت نگاه کنید؛ تابع Y1 و Y2 چه تفاوت و شباهتی با هم دارند؟ مقدار تاخیر و اثر آن را مشاهده کنید.

A	B	C	Y2	Y1



جلسه سوم

آزمایش شماره ۳

مدارهای ترکیبی ۱

(انکدر و دیکدر)

مدارهای ترکیبی معمولاً توابع دیجیتال خاصی را که در اغلب سیستم‌های دیجیتال لازم است، اجرا می‌نمایند. مهم‌ترین آنها عبارتند از جمع‌کننده‌ها، تفریق‌گرها، مقایسه‌گرها، دیکدرها، انکدرها و مولتی‌پلکسرها. این مدارها خروجی یا خروجی‌های آنها تنها به ورودی در همان لحظه بستگی دارد و لذا این مدارها هیچگونه حافظه‌ای ندارند. مراحل طراحی را می‌توان به ترتیب زیر خلاصه کرد.

- ۱- تشخیص دقیق تعداد ورودی و خروجی و رابطه آنها.
- ۲- ایجاد جدول درستی هر یک از خروجی‌ها براساس ورودی‌ها.
- ۳- ساده‌سازی تابع منطقی ارتباط دهنده هر خروجی به ورودی‌های مربوطه با استفاده از جدول کارنو.
- ۴- تبدیل توابع ساده شده حاصل به شکلی که تنها با یکی از دروازه‌های NAND یا NOR قابل پیاده‌سازی باشند.
- ۵- نصب و آزمایش مدار ترکیبی حاصل.
- ۶- رسم به صورت گرافیکی در محیط گرافیکی نرم‌افزار Maxplus و آنالیز آن در Wave form این نرم‌افزار و اجرا بر روی برد CIC-310.
- ۷- نوشتن در محیط متنی نرم‌افزار Maxplus و آنالیز آن در Wave form این نرم‌افزار و اجرا بر روی برد CIC-310.

الف - رمزگشا (دیکدر):

رمزگشا مدار ترکیبی است با n ورودی که به ازای هر حالت ورودی (جمعا 2^n حالت) یکی از 2^n خط خروجی فعال می‌شود. اگر خط خروجی فعال شده یک و سایر خروجی‌ها صفر باشند خروجی‌ها را Active High و اگر خط خروجی فعال شده صفر و سایر خروجی‌ها یک باشند خروجی‌ها را Active Low می‌نامیم.

ب - رمز ساز (انکدر):

عملکرد رمزساز عکس عملکرد رمزگشا است. در این مدار ترکیبی در حالت کلی در هر لحظه تنها یکی از 2^n ورودی فعال است و بر روی n خط خروجی، کد دودویی معادل شماره خط ورودی فعال شده، یا نقیض آن را (بسته به اینکه Active High یا Active Low) را خواهیم داشت.

طراحی:

نکته: در کلیه آزمایشات طراحی یا برنامه‌نویسی را قبل از حضور در آزمایشگاه انجام داده در نرم‌افزار با Waveform چک بفرمایید برگه طراحی و فایل‌های آن را داخل پوشه‌ای بنام خودتان به آزمایشگاه بیاورید. مراحل کار در آزمایشگاه فقط با ارائه شکل موج و ریختن در برد انجام می‌شود.

- ۱- یک رمزگشا 4×2 با خروجی Active Low طراحی کنید به نحوی که علاوه بر دو ورودی $A0, A1$ ورودی دیگری به نام E (فعال ساز) داشته باشد. اگر E یک باشد رمزگشا غیر فعال بوده و اگر E صفر باشد رمزگشا فعال گردد.



- ۲- مداری با سه ورودی طراحی کنید، به نحوی که بتواند تعداد یک‌های موجود در ورودی را نشان دهد.
 ۳- عملکرد تراشه ۷۴۱۴۸ را با استفاده از دیتا شیت بررسی کنید.

کار در آزمایشگاه

آزمایش ۱-۳:

۱-۳-۱: سوال یک طراحی را به صورت گرافیکی در MAX+PLUS II رسم و ضمن بررسی صحت عملکرد طراحی خود، جدول زیر را پر کنید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

A1	A0	E	Y3	Y2	Y1	Y0
0	0	0				
0	1	0				
1	0	0				
1	1	0				
X	X	1				

۲-۱-۳: با دو عدد رمز گشای مشابه طراحی سوال یک، یک رمزگشا $8*3$ بسازید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

آزمایش ۲-۳:

آی سی ۷۴۱۴۸ یک انکدر $3*8$ می باشد که نقشه آن در ضمیمه آورده شده است جدول (۱) را پر کنید و عملکرد تراشه را توضیح دهید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

X7	X6	X5	X4	X3	X2	X1	X0	Y2	Y1	Y0
۰	۱	۱	۱	۱	۱	۱	۱			
۱	۰	۱	۱	۱	۱	۱	۱			
۱	۱	۱	۱	۰	۱	۱	۱			
۱	۱	۱	۱	۱	۱	۱	۰			
۱	۱	۰	۱	۱	۰	۱	۰			
۰	۰	۰	۱	۱	۱	۱	۱			

جدول شماره (۱)

آزمایش ۳-۳:

نتایج طرح سوال دوم را با استفاده از برد CIC-310 ببینید.

دانشکده مهندسی برق و کامپیوتر



آزمایشگاه سیستم‌های دیجیتال ۱

مولف: سیدحسین موسوی

دانشگاه بیرجند



جلسه چهارم

آزمایش شماره ۴

مدارهای ترکیبی ۱

(انکدر و دیکدر)

استفاده از زبان سخت افزاری Verilog

سوالات طراحی زیر را با **Vrilog** پیاده‌سازی کنید آن‌را در محیط متن نرم افزار MAX+PLUS II با پسوند .v ذخیره Wave Form آن را ببینید.

برنامه نویسی با استفاده از زبان سخت افزاری Verilog

- ۱- یک رمزگشا 2×4 با خروجی **Active Low** طراحی کنید به نحوی که علاوه بر دو ورودی **A0, A1** ورودی دیگری به نام **E** (فعال ساز) داشته باشد. اگر **E** یک باشد رمزگشا غیر فعال بوده و اگر **E** صفر باشد رمزگشا فعال گردد.
- ۲- مداری با سه ورودی طراحی کنید، به نحوی که بتواند تعداد یک‌های موجود در ورودی را نشان دهد.

کار در آزمایشگاه

آزمایش ۱-۴:

۱-۴-۱: ضمن بررسی صحت عملکرد طراحی یک خود، با استفاده از برد **CIC-310** آن را عملی ببندید و جدول زیر را پر کنید.

A1	A0	E	Y3	Y2	Y1	Y0
0	0	0				
0	1	0				
1	0	0				
1	1	0				
X	X	1				



۲-۱-۴: با زبان **Vrilog** دو عدد رمز گشای مشابه طراحی سوال یک، یک رمزگشا $8*3$ بسازید. با استفاده از برد **CIC-310** آن را عملی ببندید و نتایج را مانند جدول فوق پر کنید.

آزمایش ۳-۴:

طرح سوال دوم را ببندید و صحت عملکرد آن را بررسی کنید و شکل موج آنرا به ازای ورودی‌های متفاوت در نرم افزار را درج کنید. با استفاده از برد **CIC-310** آن را عملی ببندید و نتایج را ببینید.



جلسه پنجم

آزمایش شماره ۵

مدارهای ترکیبی ۲

(مالتی پلکسر و دی مالتی پلکسر)

الف – مالتی پلکسر (متمرکز کننده):

مداری است با n خط کنترل (به عنوان آدرس) و 2^n خط داده ورودی و تنها یک خروجی که به ازای هر حالت خط کنترل، یکی از ورودی‌ها به خروجی وصل می‌گردد.

ب – دی مالتی پلکسر (پخش کننده):

دارای یک خط ورودی، n خط کنترل و حداکثر 2^n خروجی است و به ازای هر حالت خطوط کنترل، ورودی به یکی از خروجی‌ها وصل می‌گردد.

طراحی

- ۱- یک مالتی پلکسر با چهار ورودی، یک خروجی و حداقل تعداد خطوط کنترل طراحی کنید. ورودی دیگری به نام E (فعال ساز) داشته باشد. اگر E صفر باشد مالتی پلکسر غیر فعال بوده و اگر E یک باشد مالتی پلکسر فعال گردد.
- ۲- یک دی مالتی پلکسر با چهار خروجی، یک ورودی و حداقل تعداد خطوط کنترل طراحی کنید. ورودی دیگری به نام E (فعال ساز) داشته باشد. اگر E یک باشد رمز گشا غیر فعال بوده و اگر E صفر باشد رمز گشا فعال گردد.
- ۳- عملکرد تراشه ۷۴۱۵۱ را با استفاده از دیتا شیت بررسی کنید.
- ۴- پیاده سازی توابع منطقی توسط مالتی پلکسر (متمرکز کننده) و دیکدر (رمزگشا) را توضیح دهید. آنگاه با آی سی ۷۴۱۵۱ تابع F را پیاده سازی کنید.

$$F(A,B,C,D)=\sum(0,2,6,8,11,15)$$

کار در آزمایشگاه**آزمایش ۱-۵:**

۱-۵-۱: طرح سوال اول را ببینید و صحت عملکرد آن را بررسی کنید و شکل موج آنرا به ازای ورودی‌های متفاوت در نرم افزار درج کنید. با استفاده از برد $CIC-310$ آن را عملی ببینید و نتایج را ببینید.

۲-۵-۱: با دو عدد مالتی پلکسر طرح یک و گیت‌های لازم یک مالتی پلکسر با هشت ورودی، یک خروجی فاقد E (فعال ساز) را ببینید و صحت عملکرد آن را بررسی کنید و شکل موج آنرا به ازای ورودی‌های متفاوت در نرم افزار درج کنید. با استفاده از برد $CIC-310$ آن را عملی ببینید و نتایج را ببینید.



آزمایش ۲-۵:

طرح سوال دوم را ببندید، در ورودی یک قطار پالس مربعی TTL اعمال کنید. با تغییر حالت خطوط کنترل، عملکرد مدار را بررسی نمایید. به وضعیت منطقی سایر خروجی ها هنگام وصل یکی از آنها به ورودی، توجه کنید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

آزمایش ۳-۵:

با آی سی ۷۴۱۵۱ در محیط گرافیکی تابع F را پیاده سازی کنید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

$$F(A,B,C,D)=\sum(0,2,6,8,11,15)$$



جلسه ششم

آزمایش شماره ۶

مدارهای ترکیبی ۲

(مالتی پلکسر و دی مالتی پلکسر)

پیاده ساز با زبان Verilog

برنامه نویسی با استفاده از زبان سخت افزاری Verilog

- ۱- یک مالتی پلکسر با چهار ورودی ، یک خروجی و حداقل تعداد خطوط کنترل طراحی کنید. ورودی دیگری به نام E (فعال ساز) داشته باشد. اگر E صفر باشد مالتی پلکسر غیر فعال بوده و اگر E یک باشد مالتی پلکسر فعال گردد.
- ۲- با دو عدد مالتی پلکسر برنامه یک و گیت های لازم یک مالتی پلکسر با هشت ورودی ، یک خروجی فاقد E (فعال ساز) بسازید و صحت عملکرد آن را با شکل موج به ازای ورودی های متفاوت در نرم افزار درج و بررسی کنید.
- ۳- یک دی مالتی پلکسر با چهار خروجی ، یک ورودی و حداقل تعداد خطوط کنترل طراحی کنید. ورودی دیگری به نام E (فعال ساز) داشته باشد. اگر E یک باشد رمز گشا غیر فعال بوده و اگر E صفر باشد رمزگشا فعال گردد.
- ۴- تابع منطقی $F(A,B,C,D)=\sum(0,2,6,8,11,15)$ می باشد. تابع F را پیاده سازی کنید.

کار در آزمایشگاه

آزمایش ۱-۶:

۱-۶-۱: نتایج برنامه سوال اول را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۲-۶:

۲-۶-۱: نتایج برنامه سوال دوم را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۳-۶:

نتایج برنامه سوال سوم را با استفاده از برد CIC-310 عملی ببندید؛ در ورودی یک قطار پالس مربعی TTL اعمال کنید. با تغییر حالت خطوط کنترل ، عملکرد مدار را بررسی نمایید. به وضعیت منطقی سایر خروجی ها هنگام وصل یکی از آنها به ورودی، توجه کنید.

آزمایش ۴-۶:

نتایج برنامه سوال دوم را با استفاده از برد CIC-310 عملی ببندید.

جلسه هفتم



آزمایش شماره ۷

مدارهای ترکیبی ۳

(جمع کننده و تفریق کننده)

در این آزمایش به گروه مهم دیگری از مدارهای ترکیبی یعنی جمع و تفریق کننده‌ها که پایه مدارهای محاسباتی به شمار می‌روند می‌پردازیم.

الف – جمع کننده‌ها

یک نیم جمع کننده مداری است با دو ورودی (بیت A0 و A1) و دو خروجی (مجموع و دو بر یک) و یک تمام جمع کننده مداری است با سه ورودی (بیت A0 و بیت A1 و C0 حاصل از جمع احتمالی بیت‌های قبلی) و دو خروجی (مجموع و دو بر یک).

ب – تفریق کننده‌ها

هر چند که برای تفریق نیز می‌توان از ایده نیم تفریق کننده و تمام تفریق کننده شروع نمود ولی عمل تفریق بیشتر به کمک جمع و با استفاده از ایده مکمل دو انجام می‌شود. در نمایش مکمل دو یک عدد دودویی، هر عدد مثبت با معادل دودویی خود (به شرط صفر بودن پرارزش ترین بیت) و هر عدد منفی با مکمل دو مقدار مثبت خود نشان داده می‌شود. (که در این صورت همواره پرارزش ترین بیت یک خواهد بود). مثلاً فرض کنید در یک سیستم رقمی برای ذخیره هر عدد تنها چهار بیت در اختیار است و لذا اعداد به صورت چهار بیتی قابل ذخیره اند. در این صورت تنها شانزده عدد متمایز قابل نمایش خواهد بود که جدول (۱) معادل دهنده‌ی و دودویی هر یک را نشان می‌دهد.

0	0000		
1	0001	-1	1111
2	0010	-2	1110
3	0011	-3	1101
4	0100	-4	1100
5	0101	-5	1011
6	0110	-6	1010
7	0111	-7	1001
		-8	1000

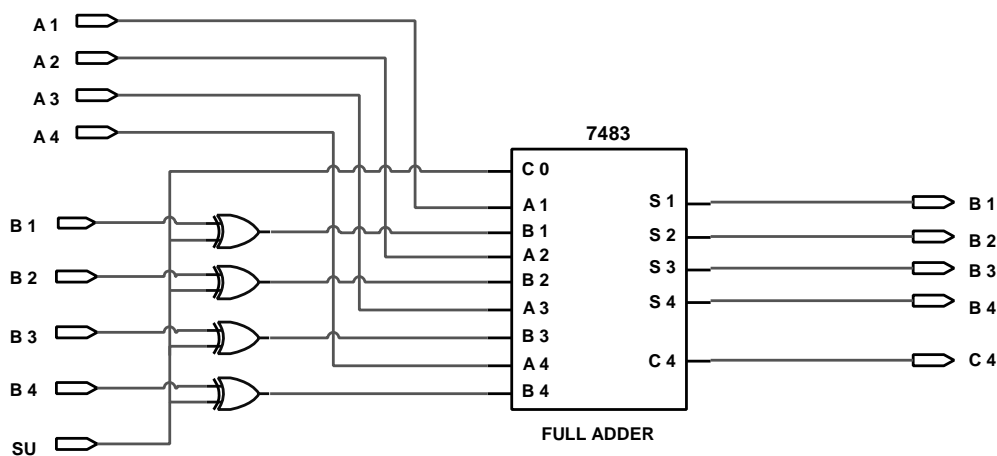
جدول شماره ۱

بنابراین اگر بخواهیم علامت را نیز در نظر آوریم تنها اعداد از 8- الی 7 قابل نمایش اند. لذا چنانچه تنها چهار بیت در اختیار باشد نتیجه محاسبات نیز در صورتی درست است که عدد حاصل بین 8- و 7 قرار گیرد. اکنون الگوریتم زیر را برای جمع و تفریق دو عدد چهار بیتی بیان می‌کنیم.

- ۱- هر عدد مثبت با معادل دودویی خود و هر عدد منفی با مکمل دو مثبت آن نشان داده می‌شود.
- ۲- در تفریق $M-N$ عدد M را با مکمل دو N جمع می‌کنیم.
- ۳- در جمع $M+N$ عدد M را با N جمع می‌کنیم.
- ۴- در جواب حاصل چهار بیت را در نظر گرفته و از Carry احتمالی همواره صرف‌نظر می‌کنیم.
- ۵- با توجه به مفهوم نمایش اعداد منفی براساس مکمل دو عدد حاصل را تفسیر می‌کنیم. یعنی اگر بیت پرارزش، 1 باشد عدد حاصل منفی است و لذا برای نمایش دهدهی آن باید از آن مکمل دو گرفته، یک علامت منفی جلو آن نهاد و اگر بیت پرارزش صفر باشد عدد حاصل مثبت است.

طراحی

- ۱- یک نیم جمع کننده با استفاده از دروازه NAND طراحی کرده و طرح خود را در MAX+PLUSII رسم و جدول صحت آن را با استفاده از شکل موج ورودی اعمالی نمایش دهید.
- ۲- یک تمام جمع کننده با استفاده از XOR, OR, AND طراحی کرده و طرح خود را در MAX+PLUSII رسم و جدول صحت آن را با استفاده از شکل موج ورودی اعمالی نمایش دهید.
- ۳- ضمن رسم و ذکر شماره پایه‌های آی سی ۷۴۲۸۳ عملکرد این جمع کننده را تشریح کنید.
- ۴- براساس الگوریتم تفریق بیان شده و با فرض حافظه چهار بیتی برای هر رقم تفریق‌های ۲-۳ و ۱+۳- و ۲-۳- را انجام داده و صحت الگوریتم را بررسی کنید.
- ۵- براساس الگوریتم بیان شده درباره جمع و تفریق اعداد، مدار شکل (۱) را توضیح دهید.



شکل (۱): جمع و تفریق کننده



کار در آزمایشگاه

آزمایش ۱-۷:

طرح سوال اول را ببندید و صحت عملکرد آن را بررسی کنید و شکل موج آنرا به ازای ورودی‌های متفاوت در نرم افزار را درج کنید.

آزمایش ۲-۷:

مدار طرح (۲) را با استفاده از برد CIC-310 عملی ببندید و ضمن آزمایش، جدول صحت آنرا رسم و بررسی نمایید.

آزمایش ۳-۷:

با استفاده از آی سی ۷۴۲۸۳ و نیم جمع کننده آزمایش های ۱-۷ و ۲-۷ یک جمع کننده ۶ بیتی بسازید و نتیجه جمع های زیر را بررسی کنید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

$$50 + 32 =$$

$$60 + 20 =$$

آزمایش ۴-۷:

مدار شکل (۱) جمع و تفریق اعداد اعم از مثبت و منفی را با توجه به نمایش اعداد منفی در شکل مکمل دو و بر اساس الگوریتم بیان شده انجام می دهد. جدول (۲) را براساس آزمایش، پر نموده و تفسیر کنید. به نتیجه سطر آخر دقت نمایید. با استفاده از برد CIC-310 آن را عملی ببندید و نتایج را ببینید.

نتیجه دهدهی	نتیجه دودویی	نوع عملیات	دودویی عدد دوم	دودویی عدد اول
		(3) - (-2)		
		(4) - (7)		
		(3) + (-7)		
		(-2) - (-3)		
		(5) - (-7)		

جدول شماره ۲



جلسه هشتم

آزمایش شماره ۸

مدارهای ترکیبی ۳

(جمع کننده و تفریق کننده)

با استفاده از زبان سخت افزاری Verilog

. برنامه نویسی با استفاده از زبان سخت افزاری Verilog

- ۱- برنامه یک نیم جمع کننده را بنویسید، جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۲- با استفاده از برنامه نیم جمع کننده بالا یک تمام جمع کننده جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۳- برنامه یک تمام جمع کننده هشت بیتی بنویسید، جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.

کار در آزمایشگاه

آزمایش ۱-۸:

نتایج برنامه سوال اول را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۲-۸:

نتایج برنامه سوال دوم را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۳-۸:

نتایج برنامه سوال سوم را با استفاده از برد CIC-310 عملی ببندید.



جلسه نهم

آزمایش شماره ۹

مدارهای ترکیبی ۴

(مقایسه کننده و مبدل های دودویی)

در این آزمایش دسته دیگری از مدارهای ترکیبی یعنی مقایسه کننده ها و مبدل های دودویی به نمایشگرهای هفت قسمتی مورد توجه قرار می گیرند و با بعضی از کاربردهای آنها آشنا می شویم. عمل مقایسه در بسیاری از زمینه ها مانند کنترل کننده ها، دستگاه های اندازه گیری رقمی و غیره بکار می رود. همچنین در بسیاری از مدارهای رقمی علاقه مندیم تا خروجی را به صورت اعداد یا علائم خاص مشاهده نماییم. یکی از قطعاتی که بدین منظور بکار می رود نمایشگر هفت قسمتی است که در دو نوع آند مشترک و کاتد مشترک موجود است. حال برای نمایش اعداد دودویی به معادل دهدهی آن باید تبدیل به نحوی که معادل دهدهی بروی نمایشگر هفت قسمتی قابل نمایش باشد انجام شود.

روش های تبدیل :

- ۱- استفاده از IC مبدل به شماره 7448 یا 7449 برای نمایشگر کاتد مشترک و IC مبدل به شماره 7446 و 7447 مبدل برای نمایشگر آند مشترک را فراهم می کنند.
- ۲- نوشتن کد تبدیل با MAX+PLUS

طراحی

- ۱- یک مقایسه کننده یک بیتی طراحی کنید. مدار شما باید دو عدد یک بیتی ورودی را مقایسه کرده و با 1 کردن یکی از سه خروجی وضعیت $A < B$ ، $A > B$ و $A = B$ را نشان دهد.
- ۲- آی سی ۷۴۸۵ یک مقایسه کننده است با مراجعه به دیتا شیت عملکرد پایه های شماره ۲ ، ۳ و ۴ (ورودی های Cascade Inputs) را توضیح دهید و با آن و طرح شماره یک، مقایسه کننده ای پنج بیتی بسازید.
- ۳- مداری با ۳ ورودی و ۷ خروجی چنان طراحی کنید که اگر ورودی باینری آن مضربی از 3 باشد در خروجی و بر روی یک نمایشگر کاتد مشترک عدد 3 و در غیر اینصورت علامت E نشان دهد.
- ۴- با مراجعه به دیتا شیت عملکرد پایه های شماره ۳ (Lt) ، شماره ۴ (Bo) ، شماره ۵ (Bi) آی سی ۷۴۴۸ را بررسی کنید.

کار در آزمایشگاه

آزمایش ۱-۹:

طرح شماره (۱) را ببندید و صحت عملکرد آن را با دادن دیتا از طریق دیپ سویچ ها بررسی کنید.

آزمایش ۲-۹:

طرح شماره (۲) را ببندید و صحت عملکرد آن را با دادن دیتا از طریق دیپ سویچ ها بررسی کنید.

آزمایش ۳-۹:



طرح (۳) را بسته و عملکرد آن را بررسی کنید.

آزمایش ۴-۹:

۱-۴-۹: با استفاده از آی سی ۷۴۴۸ مراحل زیر را انجام دهید. و جدول زیر را کامل کنید.

Inputs							Outputs							
Bi	Lt	RBi	D	C	B	A	RBo	a	b	c	d	e	f	G
0	0	0	0	0	0	0								
1	0	0	0	0	0	0								
1	1	0	0	0	0	0								
1	1	0	0	0	0	1								
1	1	1	0	0	0	0								
1	1	1	0	0	0	1								
1	1	1	0	0	1	0								
1	1	1	0	0	1	1								
1	1	1	0	1	0	0								

۲-۴-۹: سه تراشه ۷۴۴۸ را با توجه به نتایج قسمت های قبل به نحوی به یکدیگر مرتبط کنید که با حداکثر سه رقم صفرهای سمت چپ دیده نشود. (۲۵ را به صورت ۲۵ نمایش دهید).

۳-۴-۹: حال آزمایش فوق را برای اعداد اعشاری انجام دهید. (عدد ۰,۱۰۰ را به صورت ۰,۱ نمایش دهید).



جلسه دهم

آزمایش شماره ۱۰

مدارهای ترکیبی ۴

(مقایسه کننده و مبدل های دودویی)

با استفاده از زبان سخت افزاری Verilog

برنامه نویسی با استفاده از زبان سخت افزاری Verilog .

- ۱- برنامه مقایسه کننده چهار بیتی را بنویسید، مدار شما باید دو عدد چهار بیتی ورودی را مقایسه کرده و با 1 کردن یکی از سه خروجی وضعیت $A < B$ ، $A > B$ و $A = B$ را نشان دهد. جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۲- برنامه مداری با ورودی چهار بیتی و ۷ خروجی را چنان بنویسید، که اگر ورودی باینری آن مضربی از 3 باشد در خروجی و بر روی یک نمایشگر **کاتد** مشترک عدد 3 و در غیر اینصورت علامت E نشان دهد. جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۳- با مراجعه به دیتا شیت عملکرد پایه های شماره ۳ (Lt) ، شماره ۴ (Bo) ، شماره ۵ (Bi) آی سی ۷۴۴۸ را بررسی کنید؛ حال برنامه‌ای بنویسید که کار این آی سی را انجام دهد، جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۴- برنامه یک و سه را به گونه‌ای ترکیب کنید که دو عدد ورودی در سون سگمنت نمایش داده شود.

کار در آزمایشگاه

آزمایش ۱-۱۰:

نتایج برنامه سوال اول را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۲-۱۰:

نتایج برنامه سوال دوم را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۳-۱۰:

نتایج برنامه سوال سوم را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۴-۱۰:

نتایج برنامه سوال چهارم را با استفاده از برد CIC-310 عملی ببندید.

جلسه یازدهم

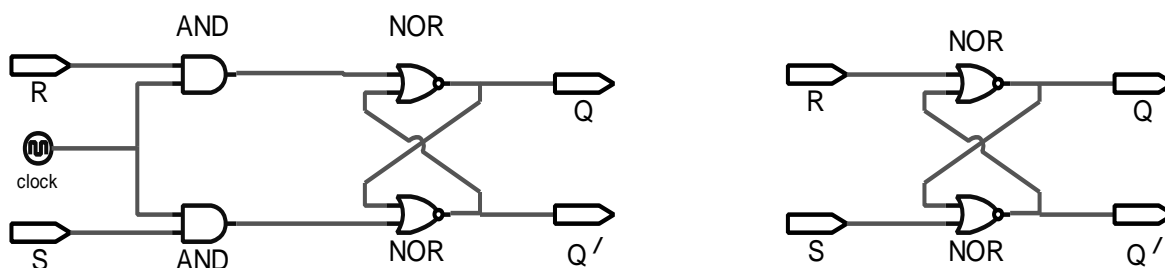
آزمایش شماره ۱۱ حافظه های دودویی

حافظه های دودویی (فلیپ فلاپ) یکی از اجزاء اصلی مدارهای ترتیبی می باشد. همانطور که می دانید فلیپ فلاپ انواع زیر را دارد: T, D, JK, RS که به عنوان عنصر اساسی حافظه در مدارات ترتیبی جهت ثبت یک حالت دودویی به کار می روند. در فلیپ فلاپ علاوه بر ورودی داده ورودی همزمان کننده ای به نام پالس ساعت موجود است که اگر فعال نباشد خروجی حتی با تغییرات ورودی هم تغییر نخواهد کرد. فعال شدن ساعت دو تعریف کلی دارد:

الف: اگر ورودی تا زمانی که پالس ساعت از صفر به یک یا برعکس بر خروجی اثر نمایند فلیپ فلاپ را حساس به لبه می نامیم.

ب: اگر ورودی فلیپ فلاپ تا زمانی که پالس ساعت یک منطقی (1) یا صفر منطقی (0) است بر خروجی اثر کنند فلیپ فلاپ حساس به سطح نامیده می شود.

شکل (۱) یک RSFF در طرح های بدون ساعت و با ساعت و جدول تحریک آن را نشان می دهد.



S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

طراحی

- ۱- طرح و جدول تحریک فلیپ فلاپ های RS, D, JK, T ساعت دار را رسم کنید. (راهنمایی: $Q(t+1) = ?$ بر حسب $Q(t)$ و مقادیر ورودی هر فلیپ فلاپ. بدیهی است که کلاک هم یک ورودی است)
- ۲- مدار یک JK را براساس روش پایه و پیرو ترسیم کنید و جدول تحریک آن را پر نمایید.
- ۳- با استفاده از فلیپ فلاپ JK فلیپ فلاپ های D و T بسازید.

برنامه نویسی با استفاده از زبان سخت افزاری Verilog

- ۱- برنامه یک فلیپ فلاپ JK را بنویسید، جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۲- برنامه یک فلیپ فلاپ T را بنویسید، جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۳- برنامه یک فلیپ فلاپ D را بنویسید، جدول صحت آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.

کار در آزمایشگاه:

آزمایش ۱-۱۱:

۱-۱-۱۱: یک عدد JK فلیپ فلاپ را فراخوان کنید. (JKFF) با ورودی ساعت و اعمال ورودی طبق جدول زیر آن را پر کنید. مشاهدات خود را یادداشت کنید. فلیپ فلاپ فوق به چه سطح یا لبه ای حساس است؟

CLK	J	K	Q(t+1)	CLK	J	K	Q(t+1)
0	1	0		0	0	1	
1	1	0		1	0	1	
0	1	0		0	0	1	
1	1	0		1	0	1	
0	0	0		0	1	1	
1	0	0		1	1	1	
0	0	0		0	1	1	
1	0	0		1	1	1	

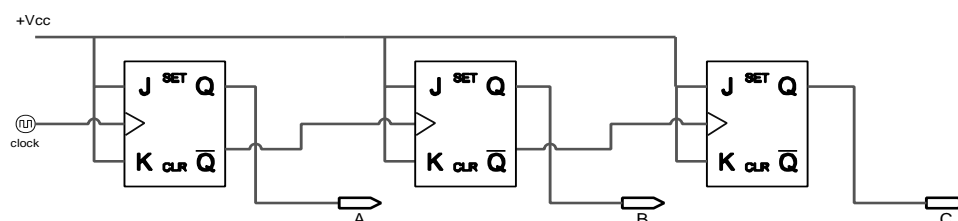
۲-۱-۱۱: JK آزمایش قبل را به یک DFF تبدیل کنید.

۳-۱-۱۱: مدار طراحی شماره ۲ (JK پایه و پیرو) را با کلاک سیستم ببندید و با توجه شکل موج در WAVE FORM نرم افزار MAX+PLUS II جدول حالت آن را پر کنید.

آزمایش ۲-۱۱:

۱-۲-۱۱: شکل (۲) یک مدار ترتیبی با استفاده از TFF را نشان می دهد. آن را ببندید و با اعمال یک پالس ساعت با فرکانس مناسب، آزمایش کنید. (خروجی را با ۷۴۴۸ به سون سگمنت وصل کنید). توضیح دهید:

- الف) کدام بیت پرارزش تر است؟
- ب) مدار چگونه کار می کند؟
- ج) شمارنده سنکرون است یا آسنکرون؟
- د) بالا رونده است یا پایین رونده؟



شکل شماره ۲: یک مدار ترتیبی

۱۱-۲-۲: مدار فوق را به گونه ای تغییر دهید که عکس حالت قبل بشمارد.

آزمایش ۳-۱۱:

نتایج برنامه سوال اول برنامه نویسی را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۴-۱۱:

نتایج برنامه سوال دوم برنامه نویسی را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۵-۱۱:

نتایج برنامه سوال سوم برنامه نویسی را با استفاده از برد CIC-310 عملی ببندید.



جلسه دوازدهم آزمایش شماره ۱۲ شمارنده های دودویی

شمارنده ها به دو دسته کلی شمارنده های سنکرون و آسنکرون تبدیل می شوند.

الف: شمارنده های سنکرون

در این نوع، پالس های ساعت به طور همزمان به تمامی فلیپ فلاپ ها وارد شده و تغییر هر یک تنها به ورودی و حالت فعلی مدار ربط دارد.

ب: شمارنده های آسنکرون

تغییر حالت هر فلیپ فلاپ ممکن است عمل راه اندازی فلیپ فلاپ دیگر را به عهده داشته باشد و پالس ساعت عموماً به CLOCK اولین فلیپ فلاپ وارد می شود. گاهی در ساخت شمارنده ها دو نوع فوق با یکدیگر ترکیب شده و بدین ترتیب امکان رسیدن به سرعت مناسب و سخت افزار ساده تر به طور یکجا فراهم می شود.

طراحی

۱- یک شمارنده دو بیتی همگام با استفاده از فلیپ فلاپ JK با دوره شمارش ۰ تا ۳ به همراه یک خط کنترل به نحوی طراحی کنید که اگر کنترل X، یک باشد، شمارش صعودی و در غیر این صورت نزولی انجام شود. همچنین اگر شمارش بدلیلی در یک حالت ناخواسته قرار گرفت، با اولین پالس بتواند به حلقه ذکر شده باز گردد.

۲- آی سی ۷۴۱۹۳ یک شمارنده با قابلیت های متعدد است. با رجوع به دیتا شیت و ضمن بررسی دیاگرام زمانی عملکرد تراشه فوق، به نقش پایه های ۱۱، ۴، ۵ و پایه های ۱، ۹، ۱۰، ۱۵ دقت کرده و عملکرد آن ها را توضیح دهید. همچنین نحوه عملکرد پایه ۱۲ (مفید هنگام شمارش صعودی) و پایه ۱۳ (مفید هنگام شمارش نزولی) را بررسی کنید. این شمارنده همگام است یا خیر؟

۳- با استفاده از تراشه قسمت قبل مداری طراحی کنید که بتواند دنباله ۷-۶-۵-۴-۳ را شمرده و تکرار کند.



کار در آزمایشگاه

(توجه: در تمام آزمایش‌ها خروجی‌های مورد نظر را بر روی سون سگمنت برد CIC-310 ببینید.)

آزمایش ۱-۱۲:

مدار طرح شماره (۱) را ببندید و صحت عملکرد آن را بررسی کنید.

آزمایش ۲-۱۲:

۱-۲-۱: آی سی ۷۴۱۹۳ را در MAX+PLUS II فراخوان کنید. مراحل زیر را طی کنید.

۲-۲-۱: شمارش به بالا و پایین را انجام دهد و وضعیت Carry و Borrow را همزمان با پالس ساعت در Wave Form نرم افزار MAX+PLUS II ببینید.

۳-۲-۷: بار شدن یک مقدار اولیه دلخواه را در شمارنده بررسی کنید.

آزمایش ۳-۱۲:

۱-۳-۱: مدار طرح (۳) را با در نظر گرفتن تجربه حاصل از آزمایش قبل ببندید و عملکرد آن را بررسی کنید.

۲-۳-۱۲: با استفاده از طراحی شماره (۱) و آی سی ۷۴۱۹۳ یک شمارنده با قابلیت شمارش ۰ الی ۳۹ بسازید.



جلسه سیزدهم
آزمایش شماره ۱۳
شمارنده های دودویی
با استفاده از زبان سخت افزاری Verilog

برنامه نویسی با استفاده از زبان سخت افزاری Verilog

- ۱- برنامه یک شمارنده چهار بیتی همگام با استفاده از فلیپ فلاپ JK با دوره شمارش ۰ تا ۱۵ به همراه یک خط کنترل به نحوی بنویسید، که اگر کنترل X، یک باشد، شمارش صعودی و در غیر این صورت نزولی انجام شود. همچنین اگر شمارش بدیلی در یک حالت ناخواسته قرار گرفت، با اولین پالس بتواند به حلقه ذکر شده باز گردد. خروجی آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.
- ۲- برنامه یک را به گونه‌ای تغییر دهید تا عدد خروجی حاصل از شمارش بر روی دو عدد سون سگمنت به صورت دهدهی نمایش داده شود.

کار در آزمایشگاه

آزمایش ۱-۱۳:

نتایج برنامه سوال اول را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۲-۱۳:

نتایج برنامه سوال دوم را با استفاده از برد CIC-310 عملی ببندید.



جلسه چهاردهم

آزمایش شماره ۱۴

ثبات های انتقالی

ثبات های انتقالی، یک جزء مهم در طراحی مدارهای دیجیتال می باشند که عموماً جهت ذخیره موقت اطلاعات برای استفاده بعدی بکار می روند. مثلاً در یک سخت افزار انجام دهنده عملیات محاسباتی همواره لازم است که بعضی بایت های اطلاعات را در جایی حفظ کنیم تا به موقع در عملیات وارد شوند یا در ارتباط یک پردازشگر با ابزار جانبی ابتدا اطلاعات به صورت یکجا در ثبات های رابط ریخته شده و سپس ضمن آنکه پردازشگر عملیات دیگر را انجام می دهد این اطلاعات به تدریج به ابزار جانبی منتقل می شوند. یک ثبات انتقالی در حالت کلی باید بتواند عملیات زیر را انجام دهد.

- ۱- ذخیره موازی یا همزمان اطلاعات.
- ۲- ذخیره بیت به بیت یا سریال اطلاعات.
- ۳- چرخش اطلاعات به چپ یا راست.
- ۴- ثابت نگهداشتن اطلاعات ذخیره شده.

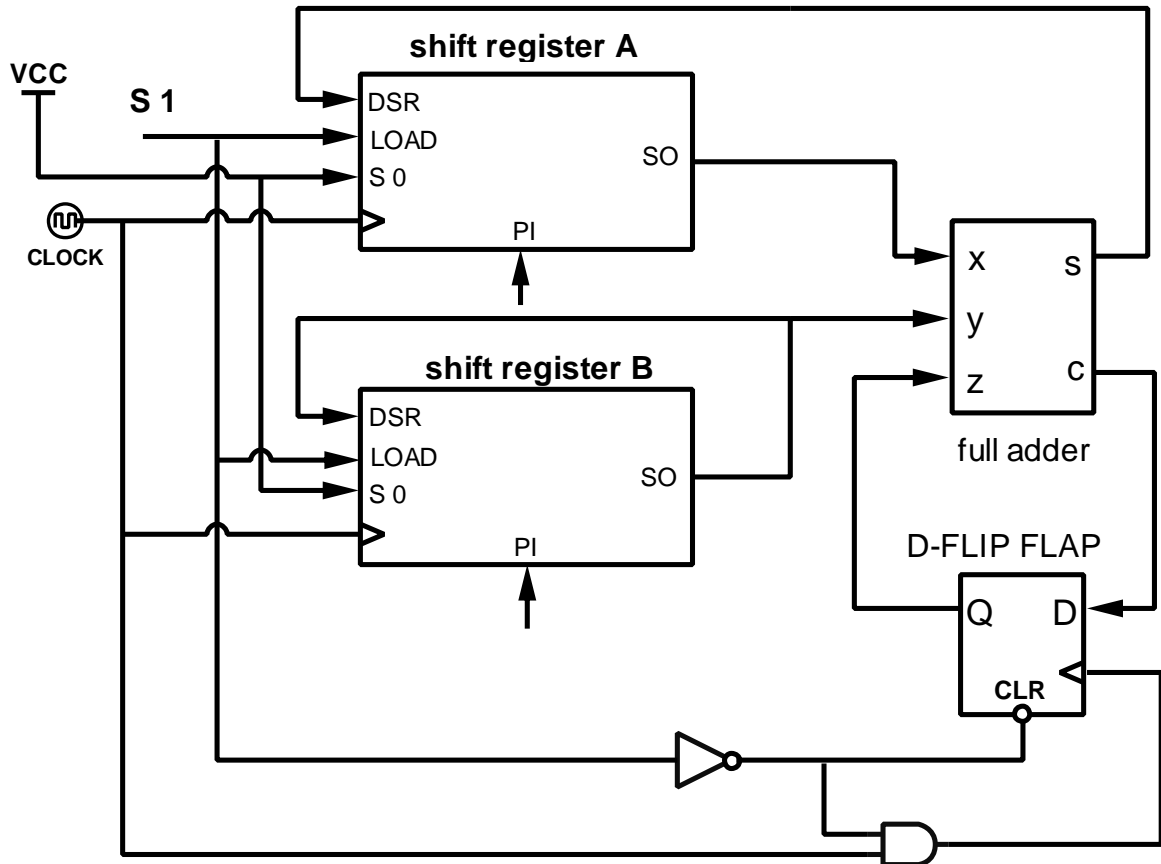
طراحی

۱- با سه عدد JK و تعداد لازم دروازه های OR و AND یک ثبات انتقالی با قابلیت بارشدن موازی و شیفت به راست طرح کنید. در این طرح یک ورودی کنترل، عملیات چرخش به راست و بار شدن موازی را کنترل می کند. Set و Reset شدن ثبات را بررسی کنید.

۲- آی سی ۷۴۱۹۴ یک ثبات انتقالی با قابلیت بار شدن موازی و چرخش به چپ و راست است. عملکرد این تراشه مخصوصاً پایه های شماره ۱، ۲، ۷، ۹ و ۱۰ و ورودی - خروجی آن را با مراجعه به دیتاشیت تحقیق کنید. تغییر اطلاعات در این تراشه در این تراشه با کدام وضعیت پالس صورت می گیرد؟ جدول ذیل را برای تراشه فوق پر کنید.

S0	S1	عملیات
		راست →
		چپ ←
		ذخیره ۸
		بی تغییر

۳- شکل شماره (۱) یک جمع کننده سریال را نشان می دهد. عملکرد آن را بررسی کرده و نقش فلیپ فلاپ D و دروازه AND در مدار چیست؟



شکل شماره (۱): طرح یک جمع کننده ۴ بیتی سریال

در مدار فوق پایه S1 جهت فرمان بار شدن موازی اطلاعات (خطوط Pi) با چرخش اطلاعات به راست به کار می‌رود و Pi خطوط ورودی موازی را نشان می‌دهد.

۴- طرح شکل شماره (۱) را با استفاده از دو عدد آی سی 74194 مدار یک تمام جمع کننده یک بیتی و تراشه ۷۴۷۴ به دقت در نرم افزار MAX+PLUS II رسم نمایید.

کار در آزمایشگاه

آزمایش ۱-۱۴:

مدار طرح شماره (۱) را ببندید و عملکرد آن را بررسی کنید.

آزمایش ۲-۱۴:



۱-۲-۱: با استفاده از تراشه ۷۴۱۹۴ که در محیط گرافیکی MAX+PLUS II فراخوان می شود بار شدن موازی ، سری ، و چرخش به راست و چپ را بررسی کنید. (با استفاده از دیپ سویچ ها و LED برد CIC-310)

۲-۲-۱: با دو عدد از تراشه فوق یک ثبات انتقالی ۸ بیتی ، بسازید و عملیات قسمت قبل را تکرار کنید.

آزمایش ۳-۱۴:

مدار طرح شماره (۴) (جمع کننده سریال) را ببینید و مراحل ذیل را دنبال کنید.

۱-۳-۱: ابتدا ثبات ها و فلیپ فلاپ D را Reset کنید و سپس ثبات ها را در حالت بار شدن موازی قرار داده و با اعمال یک پالس ساعت ، داده های مورد نظر را در آنها ذخیره کنید.

توجه : دقت کنید که در این حالت با اعمال هر پالس ساعت محتویات حافظه باید بدون تغییر بماند و ترکیب S1 و CP باید به گونه ای باشند که در حالت بار شدن موازی اطلاعات ، پالس ساعتی به فلیپ فلاپ D وارد نشود و لذا Q همچنان صفر بماند.

۲-۳-۱: ثبات ها را در وضعیت چرخش به راست قرار داده و با اعمال تعداد لازم ، پالس ساعت ، ضمن پر کردن جدول زیر ، عملکرد مدار را بررسی کنید. (دقت کنید که پایه های Preset و Clear به ولتاژهای مناسب وصل شده باشند)

شماره پالس	رجیستر B	رجیستر A
0		
1		
2		
3		
4		

آزمایش ۴-۱۴:

مدار آزمایش ۳-۱۴ را به نحوی اصلاح کنید که قادر باشید با اعمال پالس دستی لازم عمل $A+N*B$ را که در آن N یک عدد صحیح دلخواه است، انجام دهد و نتیجه عملیات را در A ذخیره کند.

جلسه پانزدهم

آزمایش شماره ۱۵

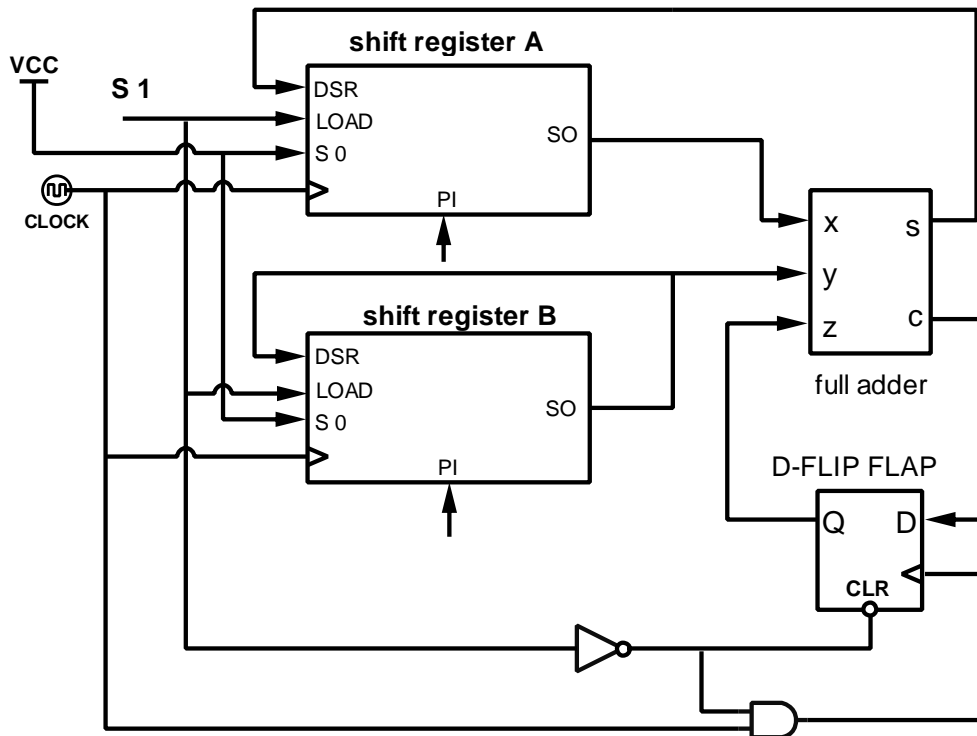
ثبات های انتقالی

با استفاده از زبان سخت افزاری Verilog

برنامه نویسی با استفاده از زبان سخت افزاری Verilog

۱- شکل شماره (۱) یک جمع کننده سریال را نشان می دهد. در این مدار پایه S1 جهت فرمان بار شدن موازی اطلاعات (خطوط Pi) با چرخش اطلاعات به راست به کار می رود و خطوط ورودی موازی را نشان می دهد. برنامه ای بنویسید که با یک شدن S1 ورودی چهاربیتی را در ثبات های A و B بار کند آنگاه با چهار پالس و استفاده از تمام جمع کننده تکبیتی و فلیپ فلاپ D با چهار پالس خروجی حاصل جمع دو عدد ورودی باشد. خروجی آن را با استفاده از شکل موج ورودی اعمالی در MAX+PLUSII رسم و نمایش دهید.

۲- برنامه یک را به گونه ای تغییر دهید تا عدد خروجی حاصل جمع را بر روی دو عدد سون سگمنت به صورت دهی نمایش داده شود. چنانچه سرریز اتفاق افتاد در سون سگمنت نماد E نمایش دهد.



شکل شماره (۱): طرح یک جمع کننده ۴ بیتی سریال



کار در آزمایشگاه

آزمایش ۱-۱۵:

نتایج برنامه سوال اول را با استفاده از برد CIC-310 عملی ببندید.

آزمایش ۲-۱۵:

نتایج برنامه سوال دوم را با استفاده از برد CIC-310 عملی ببندید.



پیوست

چند نمونه برنامه:

```
//*****  
module mux_2to1(Y, A, B, sel);  
//*****  
  
output [15:0] Y;  
input [15:0] A, B;  
input sel;  
reg [15:0] Y;  
  
always @(A or B or sel)  
if (sel == 1'b0)  
Y = A;  
else  
Y = B;  
  
endmodule  
  
module mux4_1(out, in, sel);  
output out;  
input [3:0] in;  
input [1:0] sel;  
  
reg out;  
wire [3:0] in;  
wire [1:0] sel;  
  
always @(in or sel)  
if (sel == 0)  
out = in[0];  
else if (sel == 1)  
out = in[1];  
else if (sel == 2)  
out = in[2];  
else  
out = in[3];  
endmodule
```



```
module mux4_1(out, in, sel);
output out;
input [3:0] in;
input [1:0] sel;

reg out;
wire [3:0] in;
wire [1:0] sel;

always @(in or sel)
    case (sel)
        0: out = in[0];
        1: out = in[1];
        2: out = in[2];
        3: out = in[3];
    endcase
endmodule
```



```

module testd_ff (qtest,q_bartest,d,clk);
input d,clk;
output qtest,q_bartest;
reg qtest, q_bartest;
always @ (posedge clk)
qtest = d;
q_bartest = !d;
endmodule

```

```

module jkff(J, K, clk, Q);
input J, K, clk;
output Q;
reg Q;
reg Qm;
always @ (posedge clk)
if(J == 1 && K == 0)
Qm <= 1;
else if(J == 0 && K == 1)
Qm <= 0;
else if(J == 1 && K == 1)
Qm <= ~Qm;
assign Q <= Qm;
endmodule

```



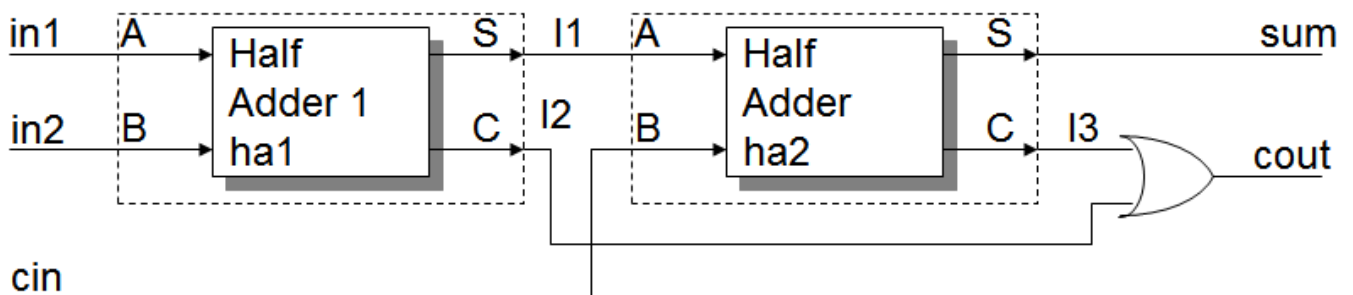
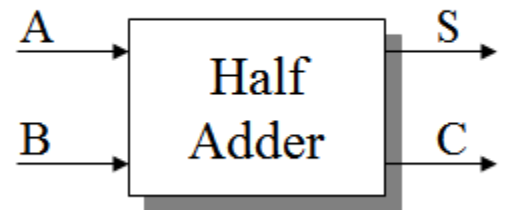
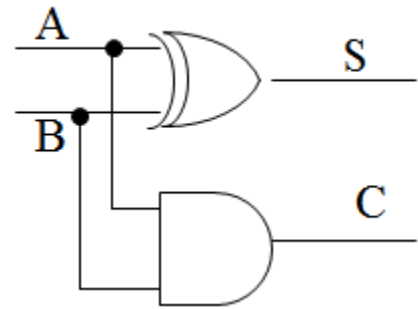
```

module half_adder(S, C, A, B);
output S, C;
input A, B;

wire S, C, A, B;

assign S = A ^ B;
assign C = A & B;

endmodule
    
```



```

module full_adder(sum, cout, in1, in2, cin);
output sum, cout;
input in1, in2, cin;

wire sum, cout, in1, in2, cin;
wire I1, I2, I3;

half_adder ha1(I1, I2, in1, in2);
half_adder ha2(sum, I3, I1, cin);

assign cout = I2 || I3;

endmodule
    
```