

شماره آزمایش: <b>1</b>	موضوع: طراحی مدارهای ترکیبی با استفاده از امکانات شماتیک	تاریخ آزمایش: 16 اسفند 99
عرشیا اخوان <b>97110422</b>	محمدحسین عبدی <b>97110285</b>	علیرضا ایلامی <b>97101286</b>

آزمایش اول: طراحی مدارهای ترکیبی با استفاده از امکانات شماتیک

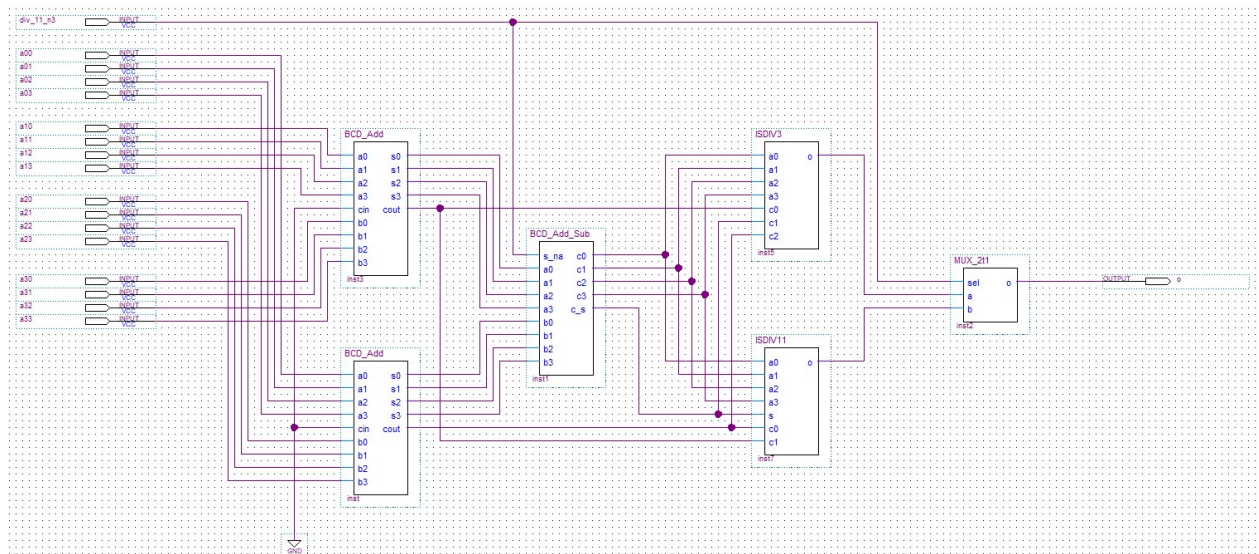
صورت آزمایش: مدار ترکیبی طراحی کنید که یک خروجی یک بیتی داشته باشد و خروجی آن وقتی 1 شود که ورودی BCD چهار رقمی آن ضربی از 3 باشد. همین آزمایش را برای اعداد ضرب 11 نیز تکرار کنید. در طراحی این مدار فقط مجاز به استفاده از گیت‌های پایه هستید.

ابتدا ذکر این نکته لازم است که ما مدار را در نرم افزار ModelSim شبیه سازی کرده و همچنین تمام شماتیک ها را در نرم افزار Quartus به کد وریلاگ تبدیل کرده ایم. فایل وریلاگ تست بنچ هم با نام testbench.v در مسیر اصلی پروژه موجود است. عکس تست بنچ به شرح زیر است:

VSIM 9> run 300

# A: 3 2 3 5	mode:0 -> res:0
# A: 3 2 3 5	mode:1 -> res:0
# A: 3 2 3 4	mode:0 -> res:1
# A: 3 2 3 4	mode:1 -> res:1
# A: 8 5 5 8	mode:1 -> res:1
# A: 8 3 8 2	mode:1 -> res:1
# A: 9 9 9 9	mode:0 -> res:1
# A: 9 9 9 9	mode:1 -> res:1
# A: 0 0 0 0	mode:0 -> res:1
# A: 0 0 0 0	mode:1 -> res:1
# A: 9 8 9 9	mode:0 -> res:0
# A: 9 8 9 9	mode:1 -> res:0

مدار اصلی: (لایه اول)



شرح آزمایش:

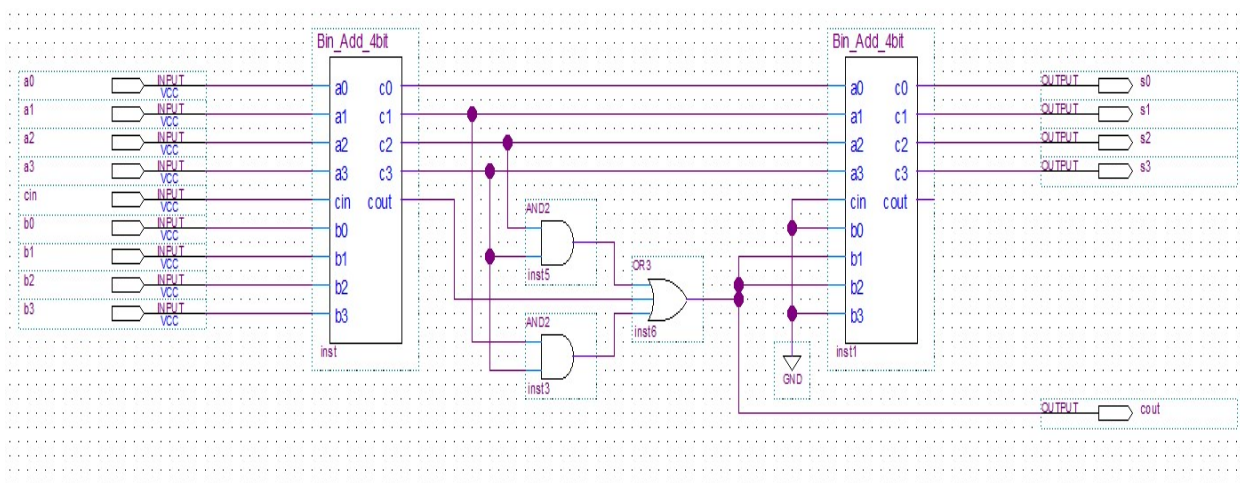
ورودی های مسئله یک عدد BCD چهار رقمی است. به همراه یک سیگنال که مشخص میکند میخواهیم بخش پذیری بر 3 را بررسی کنیم یا بر 11.

برای حل این مسئله یک مدار ترکیبی داریم که در مرحله اول ارقام اول و سوم را با هم و ارقام دوم و چهارم را با یکدیگر جمع میکند.

علت این مسئله جدا کردن بیت های 1 و 3 از بیت های 2 و 4 به استراتژی تشخیص بخش پذیری اعداد 3 و 11 برمیگردد. میدانیم اعدادی بر 3 بخش پذیرند که مجموع ارقامشان بر 3 بخش پذیر باشد. برای بررسی بخش پذیری بر 11 بین ارقام یکی در میان منفی و مثبت قرار میدهیم و سپس آنها را باهم جمع میکنیم و چک میکنیم که آیا حاصل ضربی از 11 است یا خیر. به عبارت دیگر، اعداد را یکی در میان با یکدیگر جمع میکنیم و در نهایت بخش پذیری تفاضل دو جمع بدست آمده بر 11 را چک میکنیم.

بنابراین ما ارقام اول و سوم را از ارقام دوم و چهارم در ابتدا جدا میکنیم. هرکدام از این دو بیت را به یک FullAdder میدهیم که آنها را با هم جمع کند.

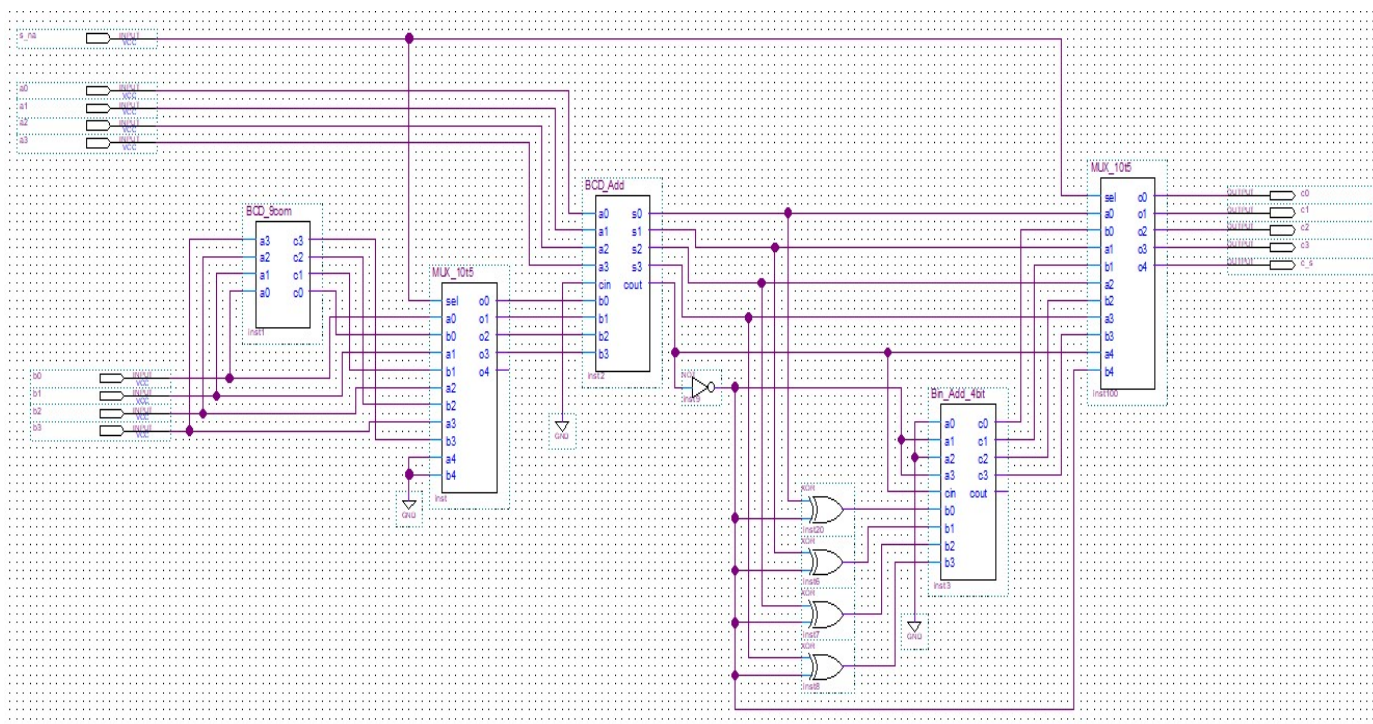
مدار BCD FullAdder به شکل زیر است:



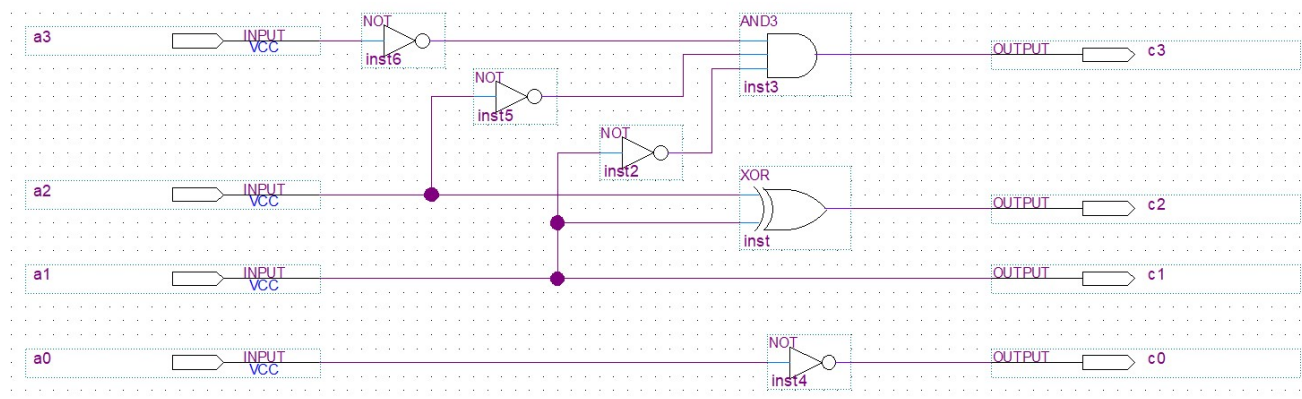
پس تا اینجا دو تا خروجی BCD و دو خروجی Carry داریم.

حال مجددا این دو خروجی BCD را به همراه سیگنال مشخص کننده مقسوم علیه به یک Full Adder/Subtractor میبریم. در صورتی که مقسوم علیه 3 بود آنها را باهم جمع و در صورتی که 11 بود آنها را از هم تفریق میکنیم.

مدار داخلی این BCD\_ADD/SUB به شکل زیر است:

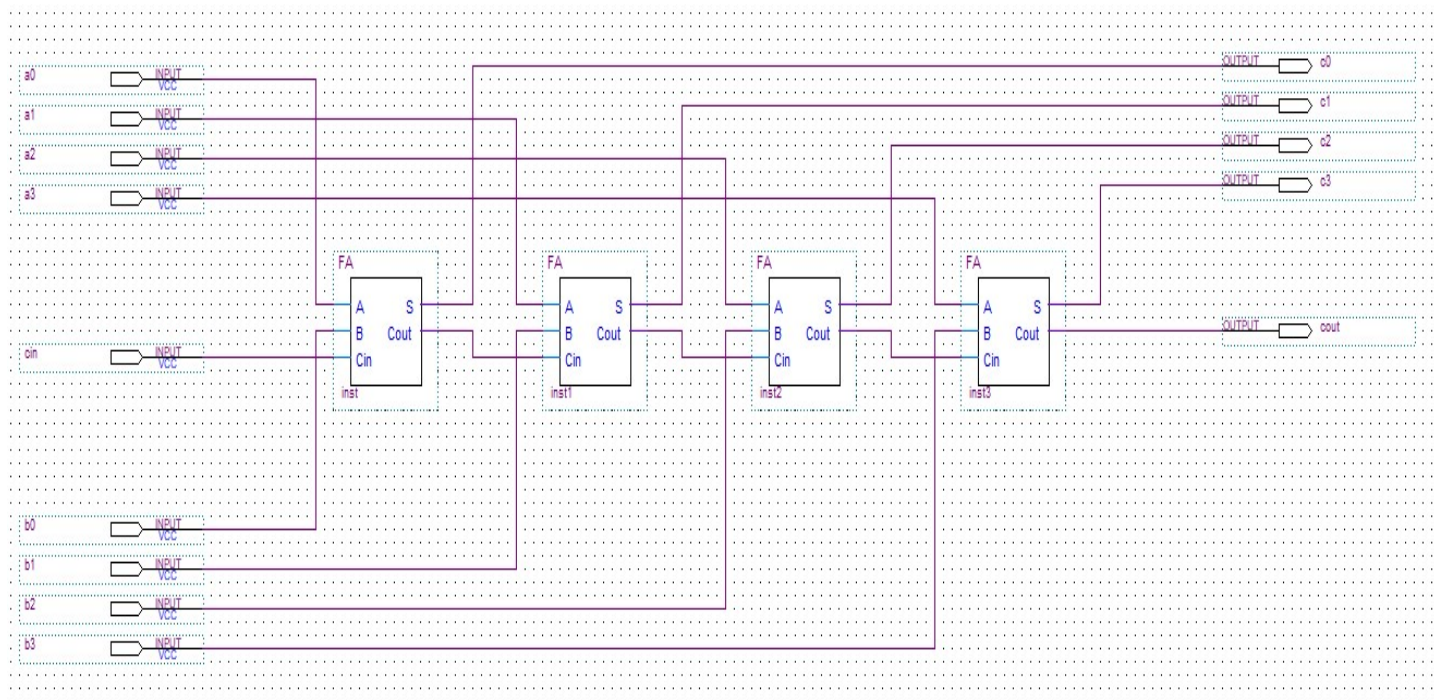


در مدار شکل بالا از ماژول های دیگری مانند BCD\_9com استفاده کرده ایم که مدار داخلی آن به شکل زیر است:

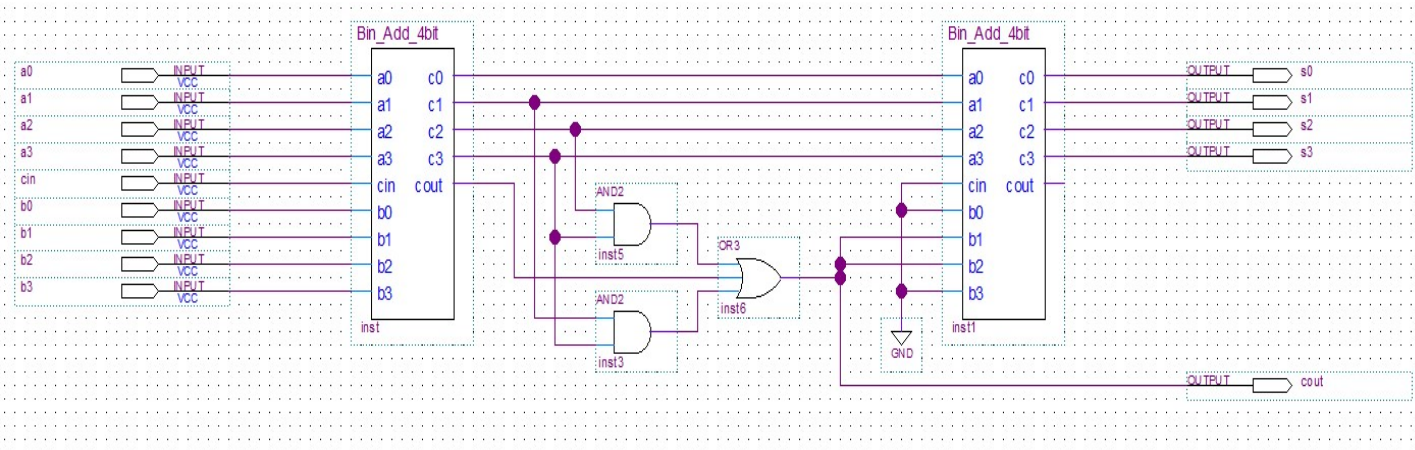


همچنین از ماژولهای BCD\_add و Bin\_add\_4bit نیز استفاده شده که مدار آنها مطابق شکل زیر می باشد:

مدار داخلی Bin\_add\_4bit:



مدار داخلی BCD\_Add:





خروجی این ماژول نیز یک BCD و یک Carry خواهد بود. پس تا اینجا یک BCD و سه تا Carry داریم.

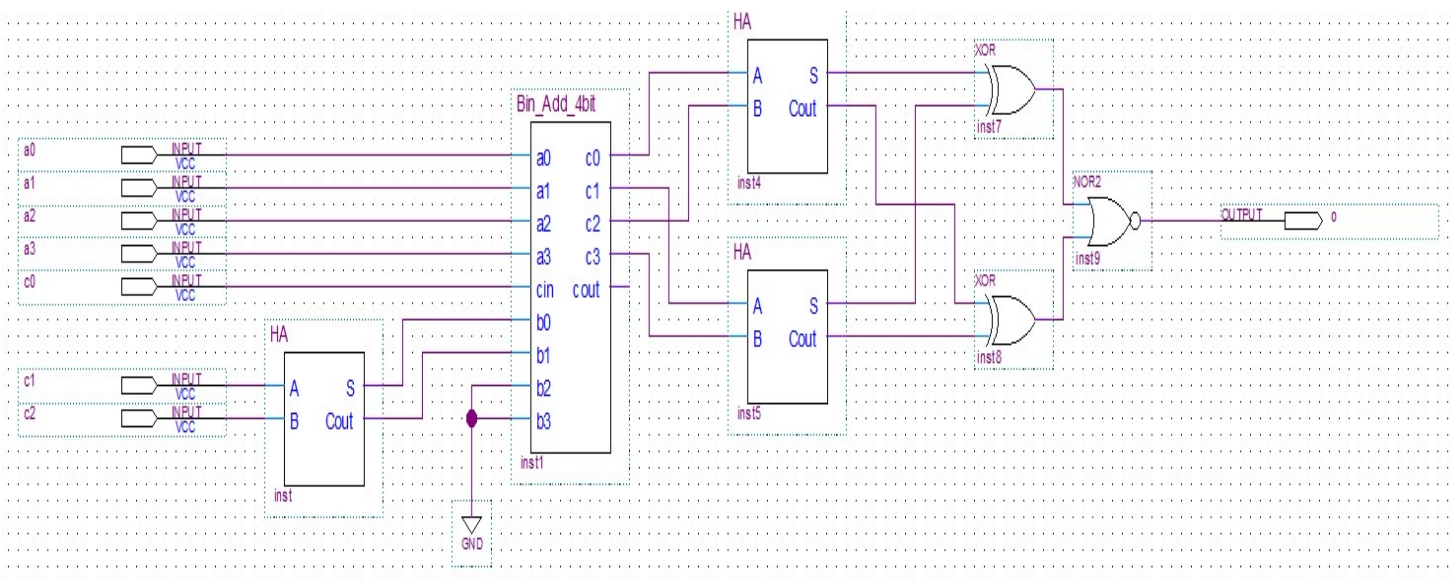
حالت ۳:

اگر در حالت 3 باشیم، Carry ها هر کدام در مرتبه دوم قرار دارند. یعنی یا 00 هستند یا 10

و چون برای بخش پذیری بر 3 این جایگاه اهمیتی ندارد، می توانیم صرفاً خود رقم را در نظر بگیریم. یک Full Adder Binary برای جمع این سه بیت در نظر گرفته شده که بخش پذیری بر سه را چک کند. خروجی این ماژول یک عدد دو بیتی است که حداکثر 3 می باشد. خروجی این ماژول را با خروجی BCD که حداکثر 9 است جمع میکنیم. پس خروجی حداکثر 12 است و در همان 4 بیت جا می شود و دیگر Carry ندارد.

حال باید به این نکته توجه کرد که بخش پذیری بر 3 در مبنای 2 مانند بخش پذیری بر 11 در مبنای 10 است. بنابراین لازم است این بیت ها را یکی در میان جمع و تفریق کنیم تا بخش پذیری بر 3 را چک کنیم. کافیهست آنها را دو به دو با یکدیگر جمع کنیم. و در نهایت آنها را از هم تفریق کنیم. با توجه به اینکه حاصل هر مجموع حداقل 0 و حداکثر 2 است، حاصل این تفریق در بازه 2- تا 2 محدود است. در این بازه تنها جوابی که بر 3 بخش پذیر است زمانی رخ میدهد که حاصل تفریق دقیقاً صفر باشد. صفر شدن را با کمک گیت های xor و nor چک کرده ایم و خروجی را به عنوان سیگنال خروجی نهایی مدار داده ایم.

مدار ماژول تشخیص بخش پذیری بر 3 به شکل زیر است:



## حالت 11:

برای حالت 11، Carry سوم که آخرین مرحله بدست آمد، نشان دهنده علامت BCD است. آن را S مینامیم و دو Carry دیگر را c0, c1 نامگذاری میکنیم.

در حالت قبل c0, c1 را با یکدیگر جمع میکردیم. برای بخش پذیری بر 11 بید برعکس عمل کنیم و تفاضل آنها را لازم داریم. ضمناً نباید فراموش کنیم که c0, c1 از مرتبه دهگان هستند. بنابراین تفاضل آنها را در 10 ضرب میکنیم و با خروجی BCD جمع میکنیم.

جدول بررسی بیت های c0, c1, S برای بکارگیری گیت مناسب:

S	C0	C1	sub	b0
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	0

$$SUB = (S \text{ xnor } C0) \text{ xor } C1$$

$$b0 = C1 \text{ xor } C2$$

برای این جمع، یک عدد با علامت BCD داریم و یک عدد 0 یا -10 یا +10 که اختلاف  $c0, c1$  را نشان میدهد.

پس علامت هر دو عدد مهم است.

اگر هر دو عدد منفی بودند، انگار دو عدد مثبت را جمع کرده ایم با Sign منفی خروجی نهایی BinaryAdder جمع  $c0, c1$  و BCD خواهد بود. اما بدون علامت.

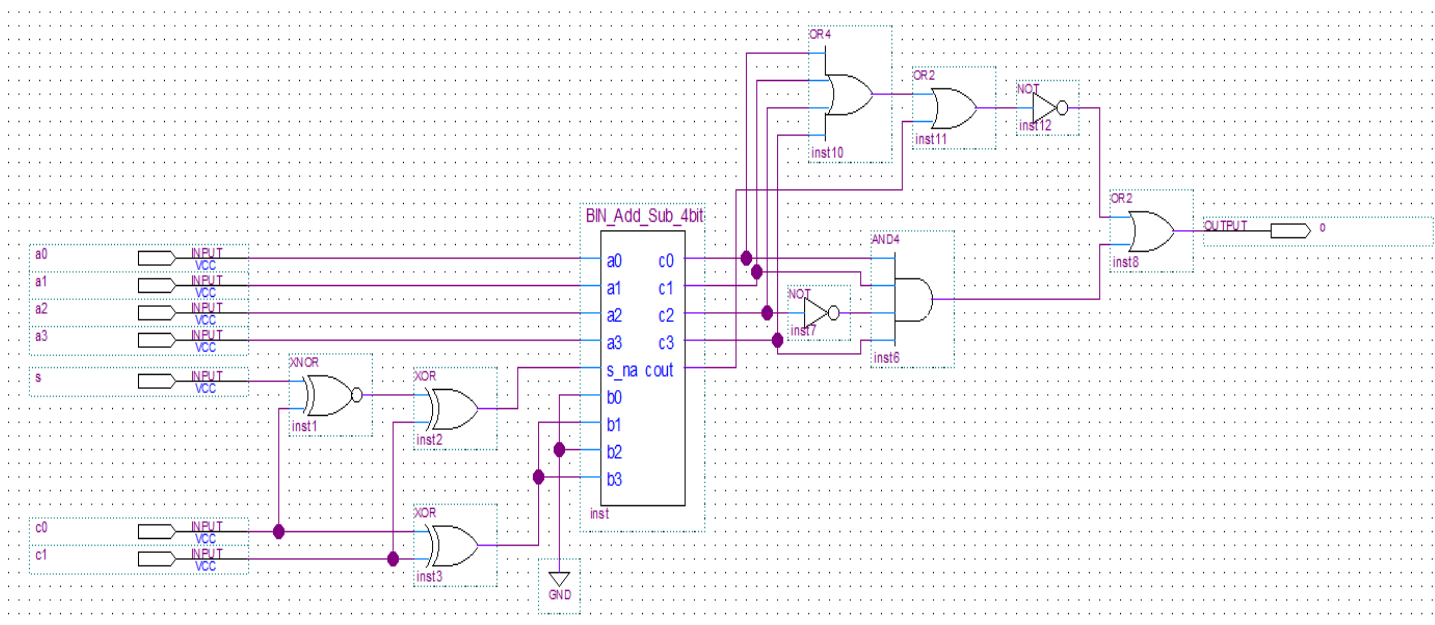
این عدد را  $x$  می نامیم. دامنه  $x$  بین 0 و 19 است.

بین 0 تا 19 تنها اعدادی که بر 11 بخش پذیرند، 0 و 11 هستند.

برای چک کردن 0، تمام بیت ها را با یکدیگر به یک گیت NOR داده ایم و برای چک کردن 11 نیز چک کرده ایم که آیا خروجی نهایی برابر با 1011 (همان 11 در مبنای 2) است یا خیر.

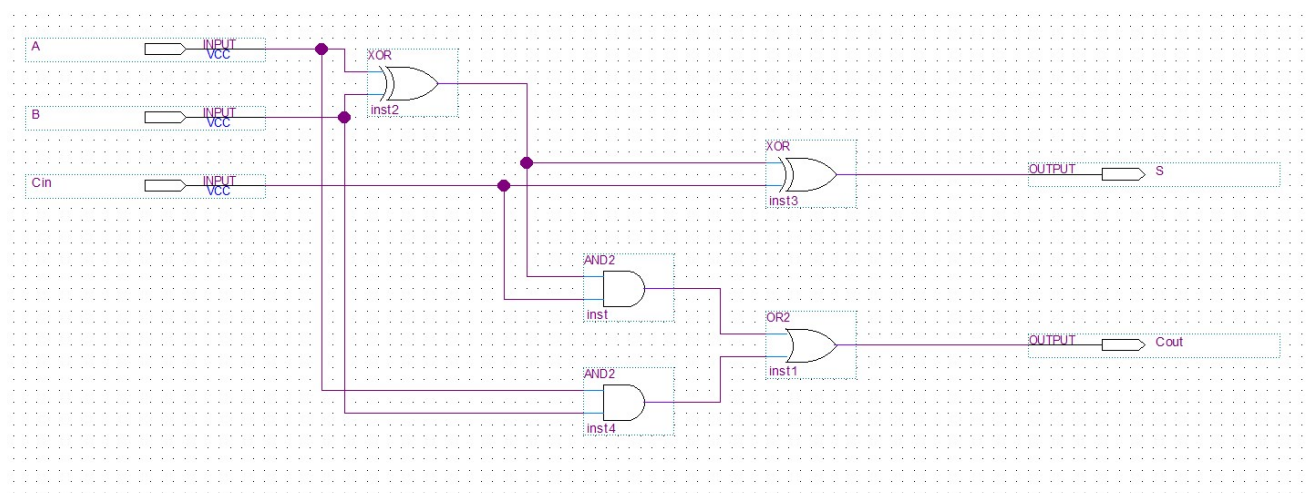
مدار داخلی ماژول تشخیص بخش پذیری بر 11 به شکل زیر است:



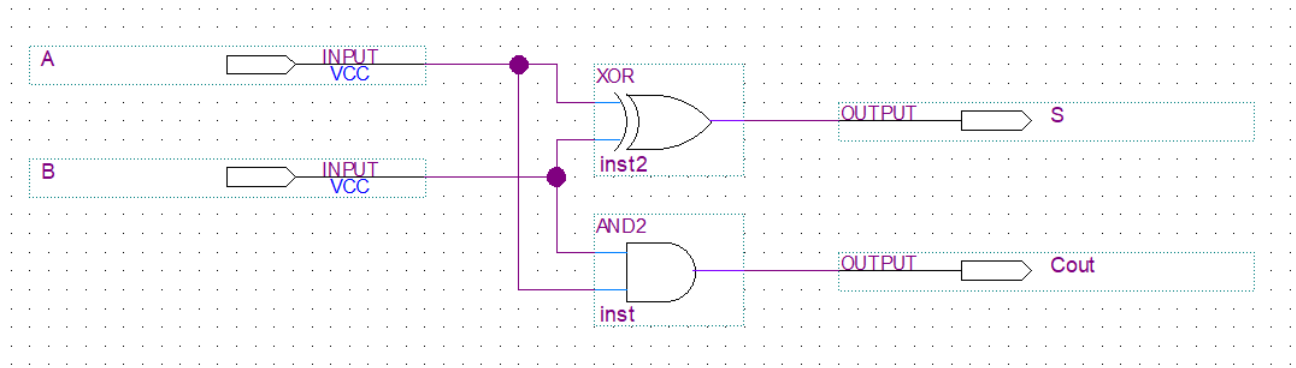


در پایان تصویر مدار داخلی ماژولهای پایه ای تر را نشان میدهیم:

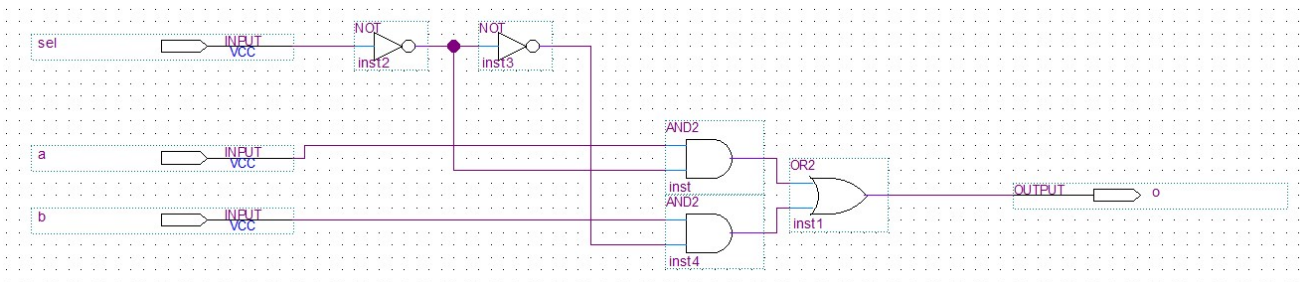
مدار FA:



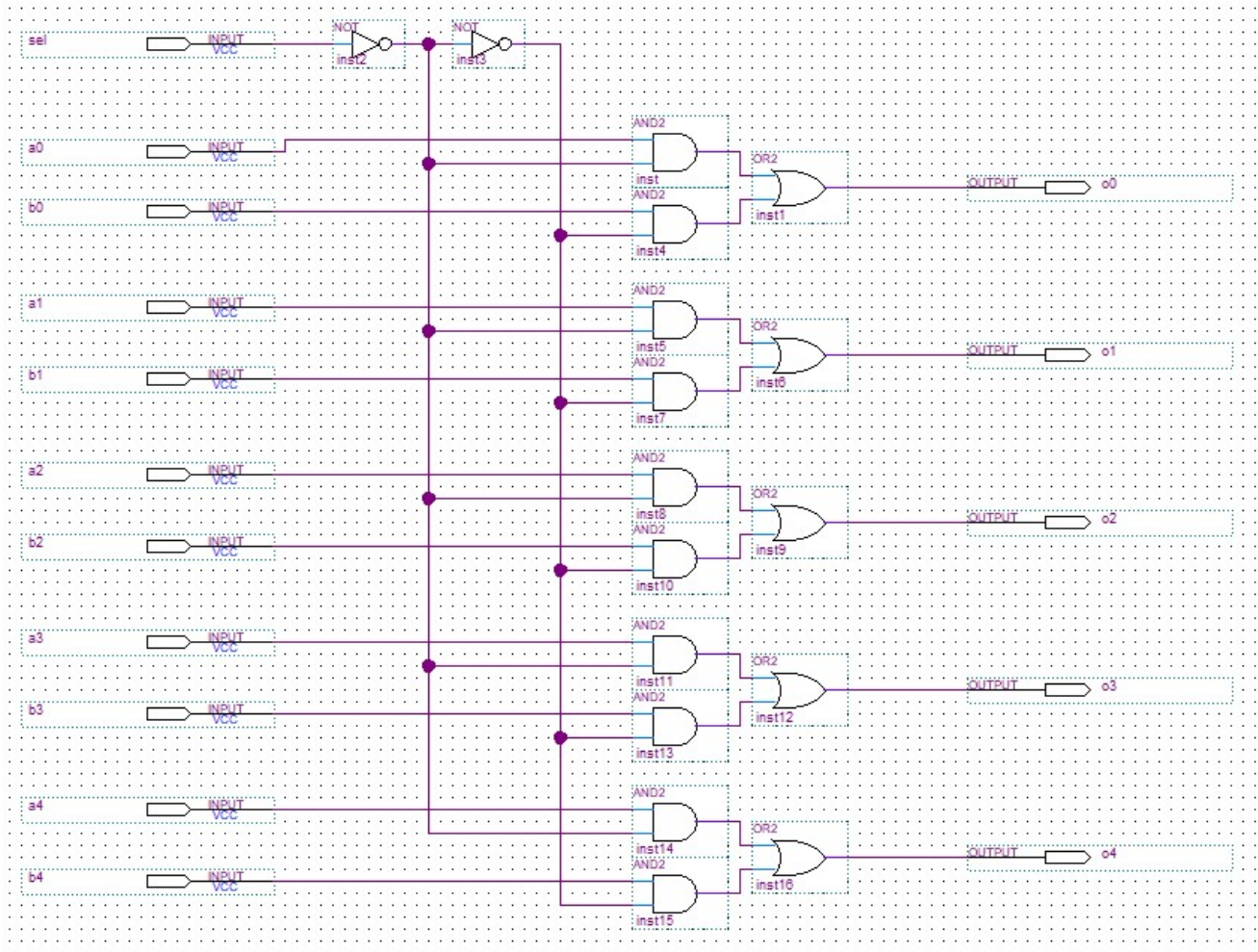
مدار HA:



مدار 1 to 2 Multiplexer:



مدار 5 to 10 Multiplexer:



آرشیو تصویر تمام مدارهای داخلی و خارجی تمام مازول ها در پوشه Report موجود است.

پایان