

آزمایش اول طراحی سیستم‌های دیجیتال

احمد سلیمی

همیلا میلی

درنا دهقانی

شرح آزمایش

میخواهیم به کمک گیت‌های پایه، شمای مداری را طراحی کنیم که بخشپذیری یک عدد چهار رقمی بر ۳ و ۱۱ را بررسی می‌کند.

راه حل کلی

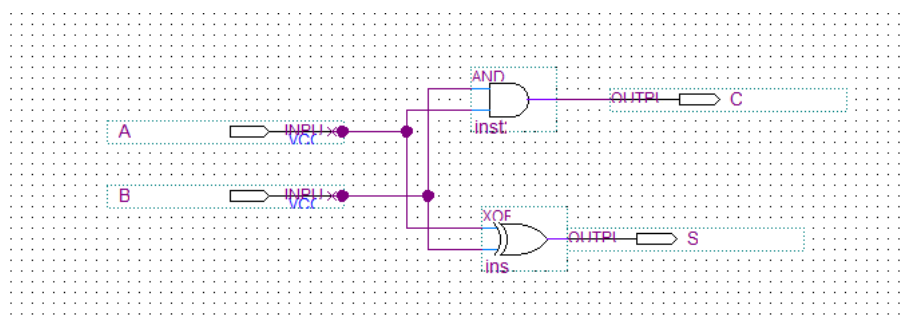
بخشپذیری بر ۳: ارقام عدد را با هم جمع می‌کنیم و اگر بزرگتر از ۱۰ شد، ارقام حاصل را با هم جمع می‌کنیم و آنقدر این کار را ادامه می‌دهیم تا حاصل یک رقمی شود. باقیمانده عدد حاصل را بر ۳ حساب می‌کنیم. اگر ۰ بود، عدد ابتدایی بر ۳ بخشپذیر است.

بخشپذیری بر ۱۱: ارقام هزارگان و دهگان را با هم جمع می‌کنیم و ارقام یکان و صدگان را از آن کم می‌کنیم. اگر حاصل ۱۱ یا ۰ یا ۱۱- شد، عدد ابتدایی بر ۱۱ بخشپذیر است.

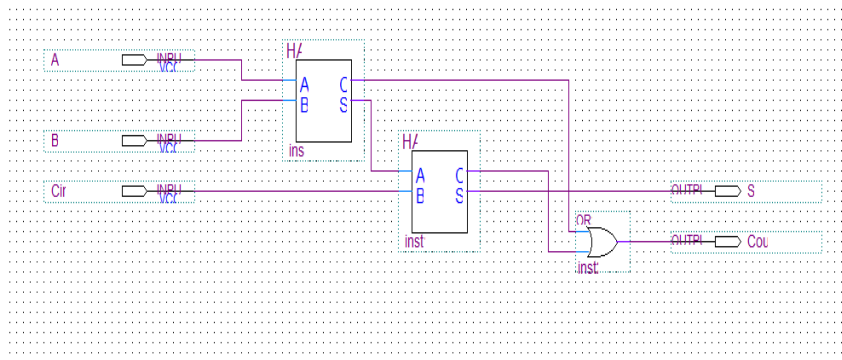
ماژول‌ها

HA: مدار نیم‌جمع‌کننده است که دو بیت را با هم جمع می‌کند. این مدار فقط از گیت‌های پایه تشکیل شده‌است.

Inputs		Outputs	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



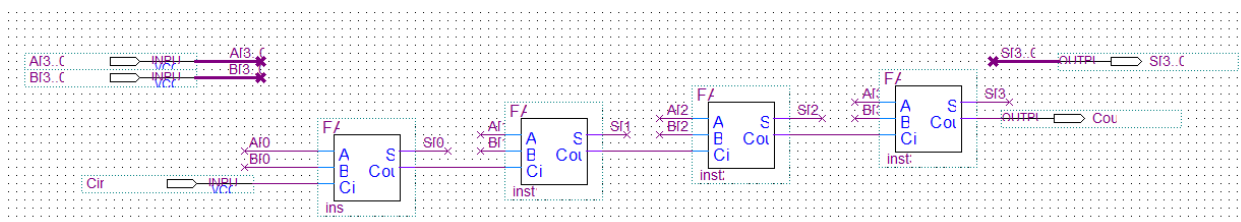
FA: مدار تمام جمع کننده که دو بیت و C_{in} را با هم جمع می کند. این مدار به کمک گیت های پایه و دو HA تشکیل



شده است.

Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

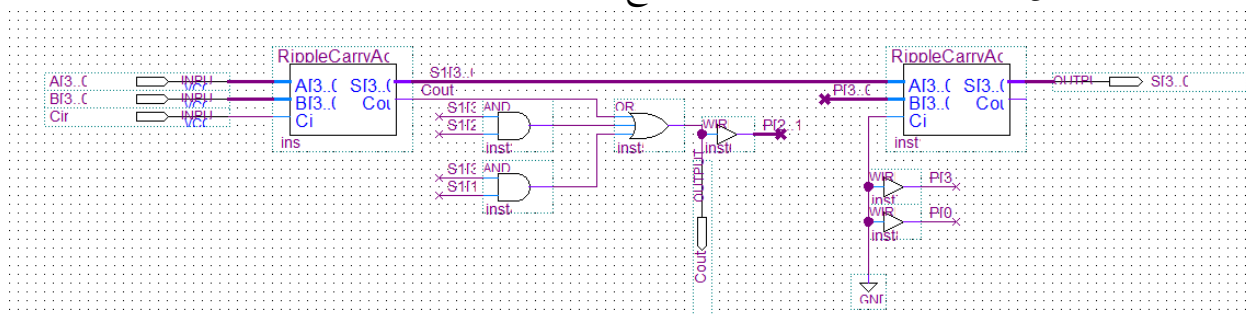
RippleCarryAdder: در این مدار به کمک چهار FA، مداری برای جمع دو رقم ساخته شده است. هر رقم متشکل از چهار بیت است و هر دو بیت متناظر این ارقام با هم جمع میشوند. ورودی C_{in} هر FA نیز C_{out} از FA قبلیست.



BCDAdder: در این مدار می خواهیم دو رقم را به شکل دهدهی جمع کنیم. ابتدا توسط RippleCarryAdder آنها را با هم جمع می زنیم. اگر جواب بیش از ۹ باشد باید Cout مازول BCDAdder یک شود. پس اگر Cout مازول RippleCarryAdder یک بود (یعنی حاصل جمع دو رقم بیش از ۱۵ شده بود) یا اگر بیت دوم و چهارم حاصل RippleCarryAdder یک بودند (یعنی حاصل جمع دو رقم ۱۰ یا ۱۱ یا ۱۴ یا ۱۵ شده بود) یا اگر بیت

سوم و چهارم حاصل RippleCarryAdder یک بودند (یعنی حاصل جمع دو رقم ۱۲ یا ۱۳ یا ۱۴ یا ۱۵ شده بود)، مقدار Cout ماژول BCDAdder یک می‌شود.

نهایتاً از جدول زیر برای تبدیل عدد حاصل به شکل دهدهی کمک می‌گیریم. یعنی کافیه در صورت نداشتن Cout، مقدار حاصل با صفر و در غیر اینصورت با ۶ جمع شود.



a + b + c _{in}		C _{out}	s
0 0000	→	0	0000
0 0001	→	0	0001
0 0010	→	0	0010
.....	→
0 1001	→	0	1001
0 1010	+6 →	1	0000
0 1011	+6 →	1	0001
0 1100	+6 →	1	0010
0 1101	+6 →	1	0011
0 1110	+6 →	1	0100
0 1111	+6 →	1	0101
1 0000	+6 →	1	0110
1 0001	+6 →	1	0111
1 0010	+6 →	1	1000
1 0011	+6 →	1	1001

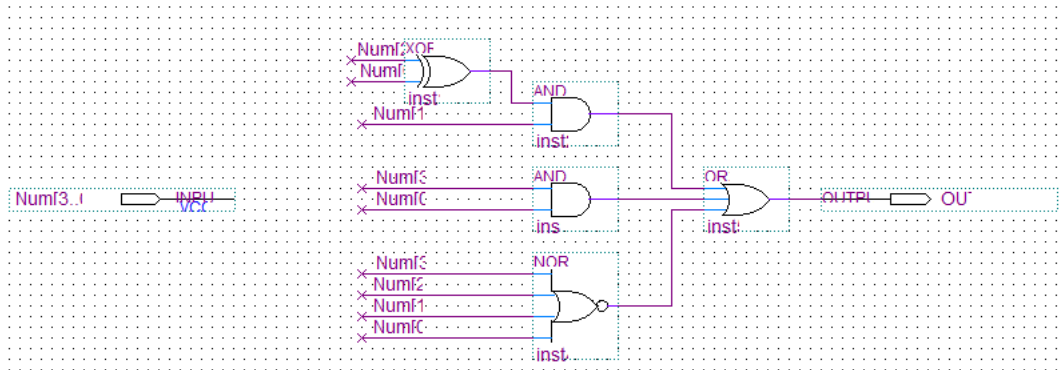
Mod3: در این ماژول بررسی می‌کنیم که یک عدد یک رقمی بر ۳ بخشپذیر است یا خیر.

بخشپذیر	Bit 0	Bit 1	Bit 2	Bit 3
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
1	1	1	0	0
0	0	0	1	0
0	1	0	1	0
1	0	1	1	0
0	1	1	1	0

1	0	0	0	0
1	0	0	1	1

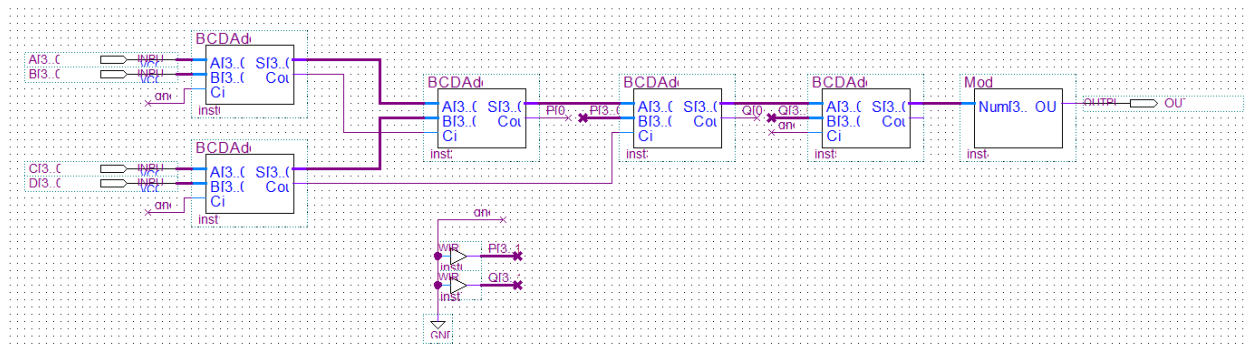
که نهایتاً توسط جدول کارنو میتوان نتیجه گرفت که بخشپذیری برابر است با:

$$\bar{A}\bar{B}\bar{C}\bar{D} + AD + B\bar{C}\bar{D} + \bar{B}CD = \overline{(A + B + C + D)} + AD + C(B \oplus D)$$

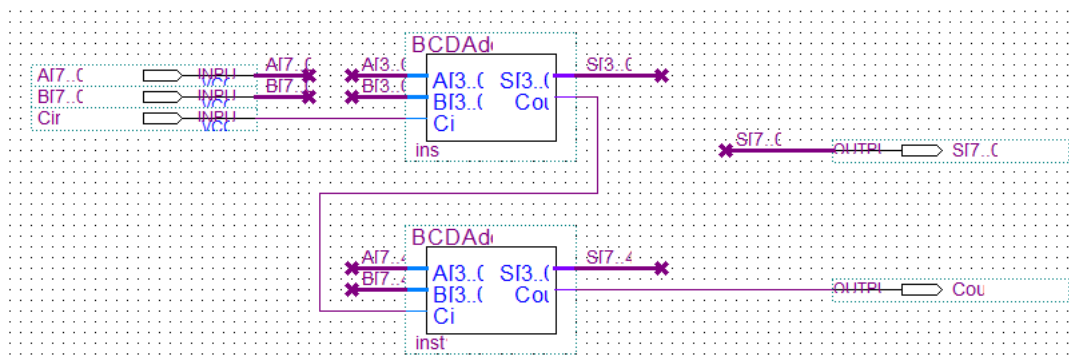


FourDigitMod3: طبق آنچه در راه حل کلی گفته شد، در این ماژول ارقام عدد را دو به دو با هم جمع می‌کنیم. سپس دو حاصل به دست آمده را با هم جمع می‌کنیم و Cin را برابر با یکی از Cout‌های پیشین قرار می‌دهیم. آنقدر ادامه می‌دهیم تا دیگر رقمی برای جمع کردن نداشته باشیم. (می‌توان دید برای بزرگترین عدد ممکن یعنی ۹۹۹۹ این تعداد جمع کافیست.) حاصل آخرین BCDAdder حتماً یک عدد یک رقمیست. توسط این رقم و ماژول Mod3 بررسی می‌کنیم که عدد بر ۳ بخشپذیر است یا نه.

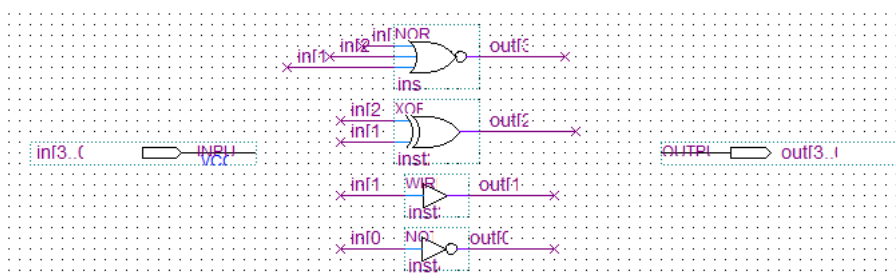
ضمناً از آنجایی که Cout‌ها یک بیتی هستند و ورودی‌های BCDAdder‌ها ۴ بیتی هستند، سه بیت صفر به ابتدای آنها اضافه می‌کنیم.



TwoDigitBCDAdder: در این ماژول دو عدد دو رقمی را می‌توان با هم جمع کرد. ارقام متناظر با هم جمع می‌گردند. از این ماژول در بخشپذیری بر ۱۱ استفاده می‌شود و به دلیل نیاز به جمع اعداد دورقمی با هم طراحی شده است، پس ورودی‌های A و B ۸ بیتی دارد.

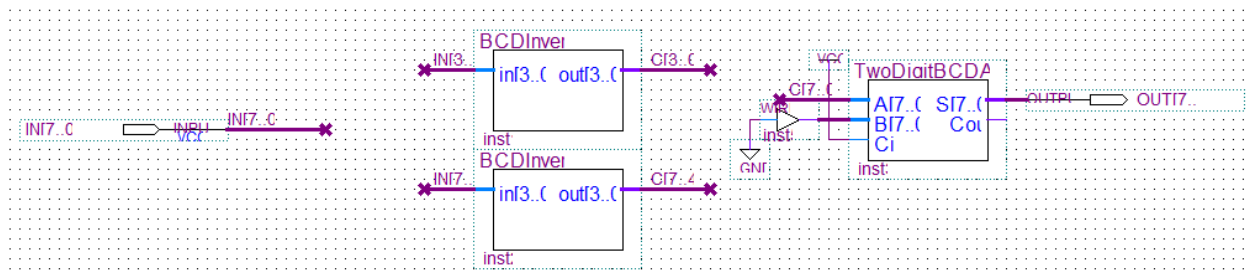


BCDInverter: در این ماژول 9's complement برای یک رقم را محاسبه می‌کنیم.

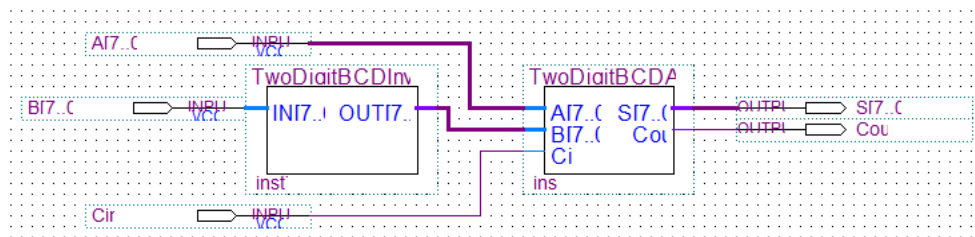


Digit	Nines' complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

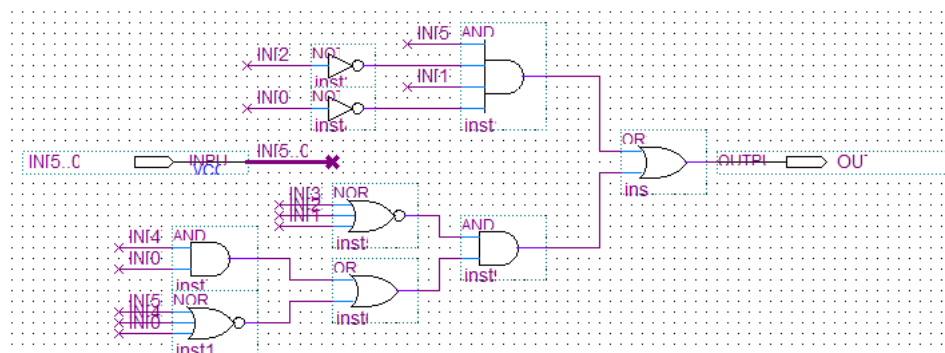
TwoDigitBCDInverter: در این ماژول میخواهیم 9's complement یک عدد دو رقمی را حساب کنیم. ابتدا 9's complement هر رقم را به دست می‌آوریم. سپس حاصل را توسط TwoDigitBCDAdder با یک جمع می‌کنیم که در Cin قرار دارد. (مشابه 1's complement در مبنای ۲)



TwoDigitBCDSubtractor: از این ماژول در بخشپذیری بر ۱۱ استفاده می‌شود. همانطور که در راه کلی ذکر شد، باید ارقام یکی در میان با هم جمع شوند. پس ممکن است حاصل این جمع دو رقمی شود (از ۱۰ تا ۱۸)، چون فقط ۴ رقم داریم و جمع دو رقم حداکثر ۱۸ می‌شود، پس لازم است تفریق‌کننده‌ای داشته باشیم که بتواند تفریق دو رقم از دو رقم را انجام دهد. به همین دلیل ورودی‌های A و B، ۸ بیتی هستند. در این ماژول A را با B- جمع می‌کنیم.



Mod11: در این ماژول به کمک جداول زیر نشان می‌دهیم عدد نهایی به ۱۱ بخشپذیر است یا نه. حاصل برای اعداد ۰، ۱۱ و ۲۲ یک می‌شود.



Map

	$\overline{D.E.F}$	$\overline{D.E.F}$	$\overline{D.E.F}$	$\overline{D.E.F}$	$D.E.F$	$D.E.F$	$D.E.F$	$D.E.F$
$\overline{A.B.C}$	1	0	0	0	0	0	0	0
$\overline{A.B.C}$	0	0	x	x	x	x	x	x
$\overline{A.B.C}$	0	0	x	x	x	x	x	x
$\overline{A.B.C}$	0	1	0	0	0	0	0	0
$A.B.C$	0	0	0	1	0	0	0	0
$A.B.C$	0	0	x	x	x	x	x	x
$A.B.C$	x	x	x	x	x	x	x	x
$A.B.C$	x	x	x	x	x	x	x	x

Map Layout

	$\overline{D.E.F}$	$\overline{D.E.F}$	$\overline{D.E.F}$	$\overline{D.E.F}$	$D.E.F$	$D.E.F$	$D.E.F$	$D.E.F$
$\overline{A.B.C}$	0	1	3	2	4	5	7	6
$\overline{A.B.C}$	8	9	11	10	12	13	15	14
$\overline{A.B.C}$	24	25	27	26	28	29	31	30
$\overline{A.B.C}$	16	17	19	18	20	21	23	22
$A.B.C$	32	33	35	34	36	37	39	38
$A.B.C$	40	41	43	42	44	45	47	46
$A.B.C$	56	57	59	58	60	61	63	62
$A.B.C$	48	49	51	50	52	53	55	54

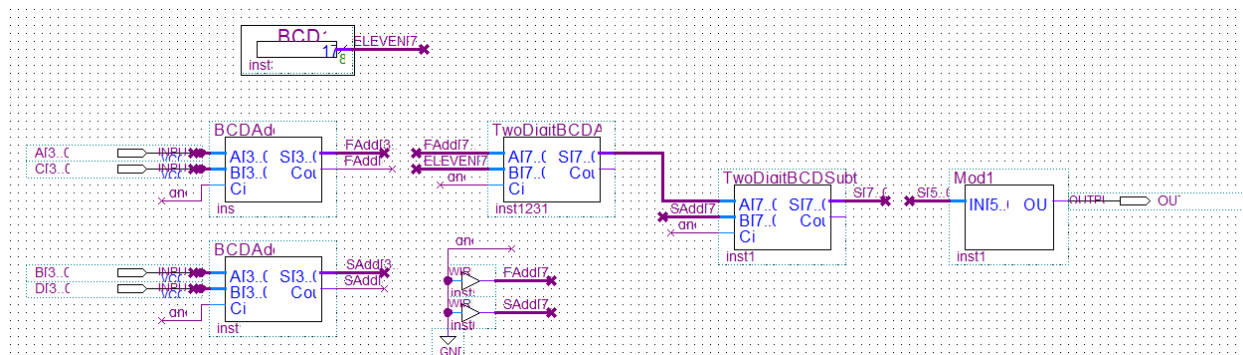
Groups

(34,42,50,58)	$A.D.E.F$
(17,49)	$B.C.D.E.F$
(0)	$A.B.C.D.E.F$

$$y = AD'EF' + BC'D'E'F' + A'B'C'D'E'F'$$

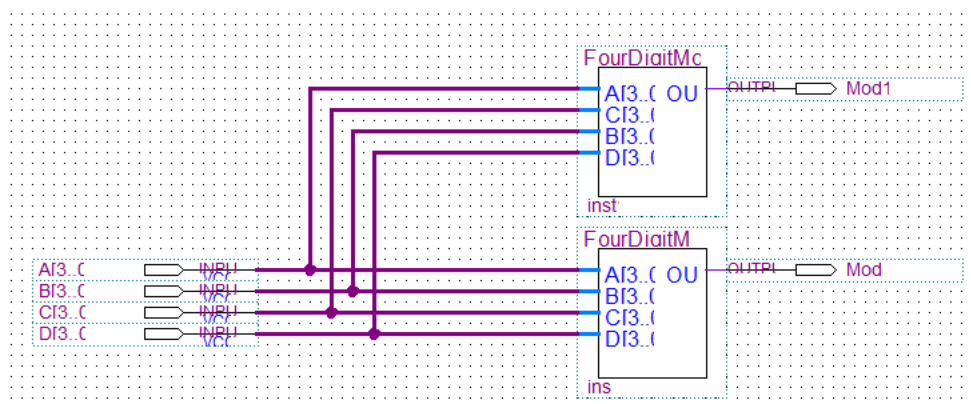
FourDigitMod11: این ماژول اصلی بررسی بخشپذیری بر ۱۱ است. ابتدا دو رقم یکان و صدگان با هم و دو رقم دهگان و هزارگان با هم جمع می‌گردند. برای تبدیل حاصل این جمع‌ها به اعداد ۸ بیتی، چهار بیت کم‌ارزش برابر حاصل جمع، بیت پنجم برابر Cout جمع و بقیه بیت‌ها صفر می‌شوند. سپس باید حاصل جمع یکان و صدگان را از حاصل جمع دهگان و هزارگان کم کنیم. نهایتاً می‌خواهیم بررسی کنیم که حاصل اعداد ۱۱ یا ۰ یا ۱۱- باشد. چون می‌خواهیم از اعداد منفی اجتناب کنیم، حاصل جمع دهگان و هزارگان را با ۱۱ دهدهی جمع می‌کنیم تا مقادیر فوق به ۲۲ یا ۱۱ یا ۰ تغییر کند و در ماژول Mod11 قابل محاسبه باشد. نمایش عدد ۱۱ به شکل BCD برابر با ۰۰۰۱ است که اگر مقدار باینری آن را حساب کنیم برابر با ۱۷ می‌شود. سپس تفریق مذکور را انجام می‌دهیم.

چون حداکثر حاصل این تفریق برابر با ۲۹ است $(9+9+11-0)$ که با 0010 1001 نشان داده می‌شود، می‌توان نتیجه گرفت دو بیت سمت چپ حتماً برابر با صفر هستند و در محاسبه نهایی بی‌تأثیرند. پس ۶ بیت دیگر را با ماژول Mod11 بررسی می‌کنیم.

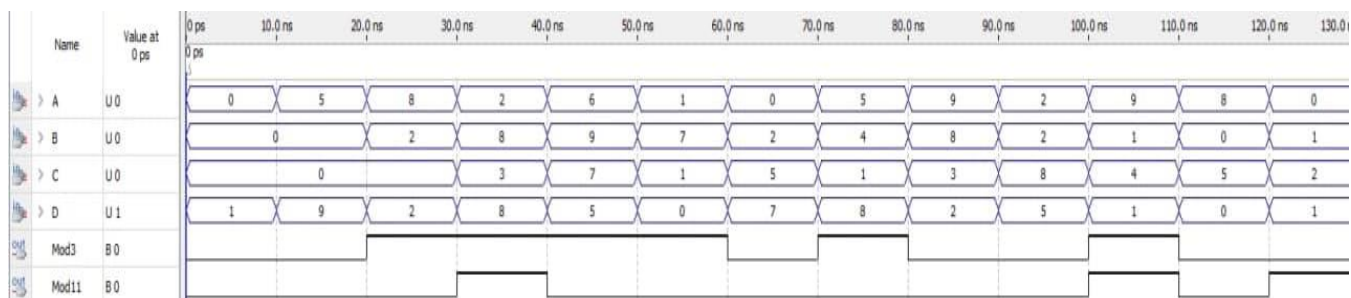


ماژول اصلی

در این ماژول (Mod) یک عدد چهاررقمی به شکل ABCD به عنوان ورودی به دو ماژول FourDigitMod3 و FourDigitMod11 داده می‌شود و خروجی‌های این دو ماژول به ترتیب Mod3 و Mod11 می‌باشند که در صورت بخشپذیری بر هر کدام از این دو عدد، ۱ می‌شوند.



Waveform



مشاهده می شود که مثلاً عدد ۸۲۰۲ بر ۳ بخشپذیر بوده و بر ۱۱ بخشپذیر نیست. یا عدد ۲۸۳۸ بر هر دو بخشپذیر است.