



دانشکده مهندسی کامپیوتر

آزمایشگاه طراحی سیستم‌های دیجیتال

## آزمایش اول - طراحی مدارهای ترکیبی

دکتر اجلالی، مهندس اثنی عشری

امیرمهدی کوششی — ۹۸۱۷۱۰۵۳

ایمان محمدی — ۹۹۱۰۲۲۰۷

شایان صالحی — ۹۹۱۰۵۵۶۱

۲۵ تیر ۱۴۰۲

## فهرست مطالب

۳	۱	مقدمه
۳	۲	بخش‌پذیری بر ۳
۳	۱.۲	ورودی‌ها و خروجی‌ها
۳	۲.۲	حل مسئله
۴	۳.۲	پیاده‌سازی مدار
۶	۳	بخش‌پذیری بر ۱۱
۶	۱.۳	ورودی و خروجی‌ها
۶	۲.۳	حل مسئله
۷	۳.۳	پیاده‌سازی مدار

## ۱ مقدمه

هدف از این آزمایش، طراحی دو مدار ترکیبی است که در مدار اول، یک ورودی ۱۶ بیتی که متشکل از عددی چهار رقمی در نمایش BCD است به مدار داده شده و خروجی یک بیتی آن نمایانگر آن است که آیا عدد دهدهی داده شده بر ۳ بخش پذیر است یا خیر. مدار دومی نیز مشابه اولی دارای یک دسته سیگنال ورودی ۱۶ بیتی است با این تفاوت که در تنها خروجی مدار که تک بیتی است، بخش پذیری عدد ورودی بر ۱۱ مشخص می‌شود.

## ۲ بخش پذیری بر ۳

### ۱.۲ ورودی‌ها و خروجی‌ها

ورودی این مدار یک سیگنال ۱۶ بیتی است که در واقع نمایش BCD یک عدد دهدهی چهار رقمی بوده و هر ۴ بیت کنار هم از راست، نشان‌دهنده‌ی یک رقم از عدد دهدهی هستند. خروجی این مدار سیگنال ۱ بیتی است که نشان می‌دهد آیا آن ۱۶ بیت ورودی یا به عبارتی آن عدد چهار رقمی بر عدد ۳ بخش پذیر است یا نه. در صورتی که بخش پذیر باشد مدار خروجی ۱ شده و در صورتی که نباشد مقدار خروجی ۰ می‌شود. البته در سایر حالاتی که غیر قابل قبول هستند نیز خروجی ۰ خواهد بود. حالات غیر قابل قبول مانند آن است که یکی از ارقام چهار رقم عددی بزرگ‌تر از ۹ باشد. (زیرا گفتیم هر ۴ بیت نشان‌دهنده‌ی ۱ رقم دهدهی است و با ۴ بیت می‌توان اعداد بزرگ‌تر از ۹ را نیز ساخت.)

### ۲.۲ حل مسئله

فرض کنید که عدد ورودی دهدهی چهار رقمی ما به صورت  $(a_3a_2a_1a_0)$  در مبنای ۱۰ است. ابتدا باقی‌مانده دویستی هر رقم عدد در تقسیم بر ۳ محاسبه می‌کنیم. سپس دو بیت باقی‌مانده رقم اول بر ۳ و دو بیت باقی‌مانده‌ی رقم دوم بر ۳ را در کنار یکدیگر می‌گذاریم تا یک عدد ۴ بیتی تشکیل شود با توجه به اینکه باقی‌مانده به ۳ یکی از حال‌های ۰ یا ۱ یا ۲ یا به عبارتی ۰۰ ی ۰۱ یا ۱۰ می‌باشد، عدد ۴ بیتی تشکیل شده در اثر concat شدن این دو عدد دو بیتی حداقل ۰۰۰۰ یا همان ۰ و حداکثر ۱۰۱۰ یا همان ۱۰ می‌باشد. پس باید این عمل را دوباره تکرار کنیم. سپس باقی‌مانده‌ی دو عدد جدید را به همان روش قبلی به دست آورده و این روند باقی‌مانده گرفتن را آنقدر ادامه می‌دهیم تا نهایتاً یک عدد دویستی که باقی‌مانده عدد اصلی ورودی مدار بر ۳ است را به دست می‌آوریم و از روی بخش پذیری عدد ورودی بر ۳ تعیین می‌کنیم.

در ادامه درستی این روند را با نمایش ریاضی نشان می‌دهیم.

$$a_3 = 3k_3 + r_3 \quad 0 \leq r_3 \leq 2$$

$$a_2 = 3k_2 + r_2 \quad 0 \leq r_2 \leq 2$$

$$a_1 = 3k_1 + r_1 \quad 0 \leq r_1 \leq 2$$

$$a_0 = 3k_0 + r_0 \quad 0 \leq r_0 \leq 2$$

$$\begin{aligned} a_3a_2a_1a_0 &\equiv 1000a_3 + 100a_2 + 10a_1 + a_0 \equiv 999a_3 + a_3 + 99a_2 + a_2 + 9a_1 + a_1 + a_0 \equiv a_3 + a_2 + a_1 + a_0 \\ &\equiv 3k_3 + r_3 + 3k_2 + r_2 + 3k_1 + r_1 + 3k_0 + r_0 \equiv r_3 + r_2 + r_1 + r_0 \end{aligned}$$

لازم به ذکر است که تمام هم‌نهشتی‌های نوشته شده به نهشت ۳ هستند. حال توجه داریم که  $r_i$  ها همان باقی‌مانده‌های دو بیتی هستند که می‌خواهیم کنار هم قرار دهیم. در واقع عدد  $(r_i r_j)$  یک عدد باینری ۴ بیتی است که دو بیت کم ارزش آن مربوط به  $r_j$  و دو بیت با ارزش آن مربوط به  $r_i$  است. بنابراین در مقداردهی ارزش  $r_i$  برابر  $r_j$  است و معادل دهی این عدد ۴ بیتی به صورت  $4r_i + r_j$  است. با توجه به این بحث و نحوه کار الگوریتم که بالاتر شرح داد شده است،  $r_{10} = 4r_1 + 4_0$  باقی‌مانده حاصل از کنار هم گذاشتن باقی‌مانده‌های رقم اول و دوم،  $r_{32} = 4r_3 + r_2$  باقی‌مانده حاصل از کنار هم گذاشتن رقم سوم و چهارم و در نهایت  $r_{3210} = 4r_{32} + r_{10}$  باقی‌مانده حاصل از کنار هم گذاشتن این دو عدد به دست آمده است. حال در ادامه‌ی هم نهشتی داریم:

$$r_3 + r_2 + r_1 + r_0 \equiv 3r_3 + r_3 + r_2 + 3r_1 + r_1 + r_0 \equiv (4r_3 + r_2) + (4r_1 + r_0) \equiv$$

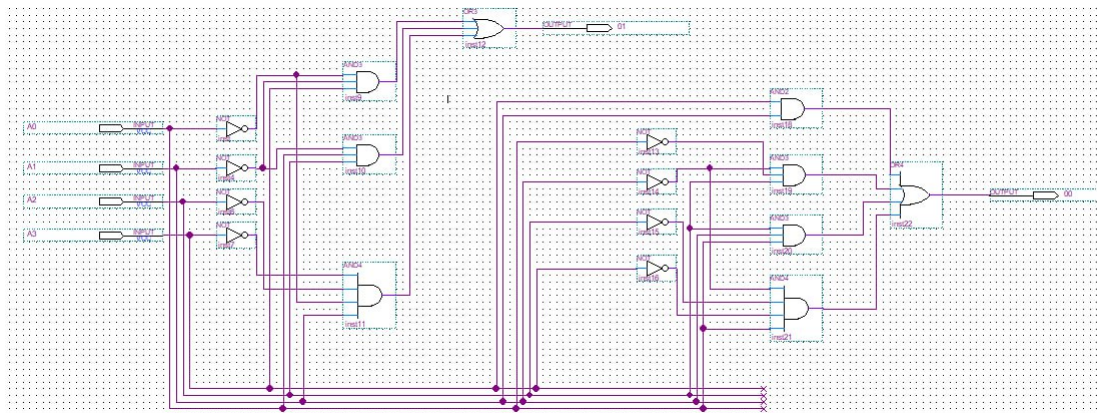
$$r_{32} + r_{10} \equiv 3r_{32} + r_{32} + r_{10} \equiv 4r_{32} + r_{10} \equiv r_{3210}$$

بنابراین باقی‌مانده‌ی عدد ورودی بر ۳ دقیقاً برابر با باقی‌مانده‌ی دوبیتی نهایی محاسبه شده طبق الگوریتم ارائه شده است و درستی الگوریتم ثابت شد. نهایتاً برای باقی‌مانده دوبیتی به دست آمده برای حالات ۰۱ و ۱۰ نادرست، حالت ۰۰ درست و حالت ۱۱ بی اهمیت است و می‌توان آن را نادرست در نظر گرفت. پس با NOR کردن بیت‌های عدد دوبیتی به دست آمده می‌توان بخش‌پذیری بر ۳ را تشخیص داد.

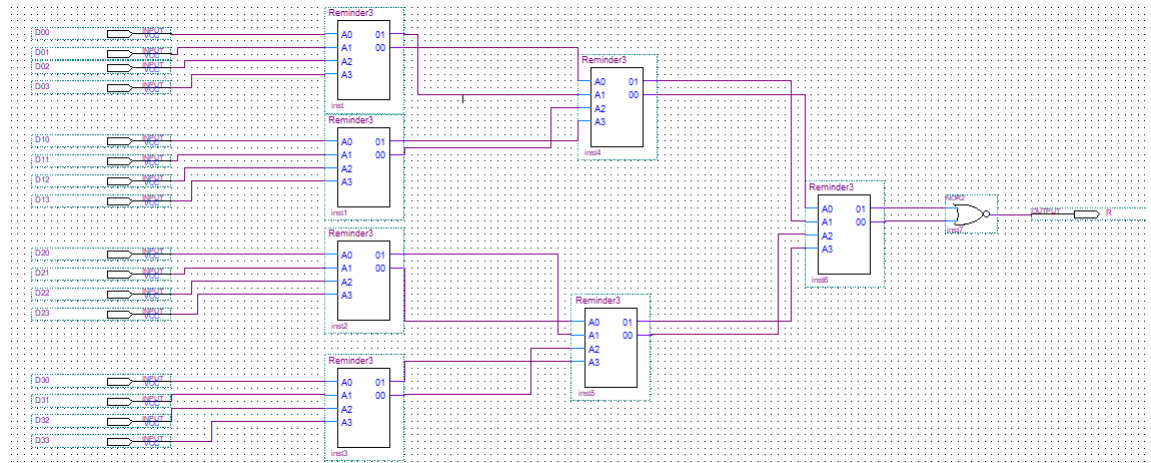
### ۳.۲ پیاده‌سازی مدار

ابتدا نیاز داریم یک مدار طراحی کنیم که برای یک عدد باینری ۴ بیتی که مقدار دهدهی آن بین ۰ تا ۱۰ است، مقدار باقی‌مانده بر ۳ را محاسبه کنیم (چون خود ارقام بین ۰ تا ۹ و اعداد حاصل از کنار هم قرار گرفتن باقی‌مانده‌های مراحل قبل بین ۰ تا ۱۰ هستند، اعداد ۴ بیتی بین ۱۱ تا ۱۵ اهمیتی در مدار ما ندارند چون حالاتی غیر ممکن هستند).

این مدار دارای یک ورودی چهار بیتی و یک خروجی دو بیتی است که در واقع باقی‌مانده عدد ورودی بر ۳ است. با توجه به توصیفات داده شده یک جدول کارنو برای هر بیت خروجی رسم و پس از ساده سازی کارنو مدار را به صورت زیر پیاده‌سازی می‌کنیم:



برای سادگی کار، یک Symbol File از روی مدار شماتیک طراحی شده بالا ساخته و آن را ذخیره می‌کنیم تا برای مدار نهایی از آن استفاده کنیم. درواقع این مداری که ما ساختیم برای ۴ بیت یا ۱ رقم ددهی است و باید حالا برای ۴ رقم یا ۱۶ بیت آن را طراحی کنیم و از آن استفاده کنیم.



در واقع برای همان ۴ رقم هم باید مانند همان ۱ رقم استفاده کنیم، باید برای تک رقم از آن ۴ رقم این پروسه را رفته و سپس خروجی دو رقم دو رقم را در کنار هم قرار داده و در نهایت باز خروجی ۲ رقم نهایی را در کنار هم قرار داده تا نتیجه نهایی به دست بیاید، به عبارتی اول برای  $r_0$  و  $R_1$  و  $R_2$  و  $r_3$  جدا جدا حساب کرده و در نهایت برای  $r_{10}$  و  $r_{32}$  حساب کرده و در نهایت از تجمیع آن‌ها یعنی  $r_{3210}$  حساب می‌کنیم. در مدار زیر این مورد مشخص است.

حال با سیگنال‌های متفاوت این مدار را تست می‌کنیم. ارقام ددهی در این تست ۱۲۳۴ می‌باشد.

2	D02	A 1	
3	D03	A 0	
4	D10	A 1	
5	D11	A 1	
6	D12	A 0	
7	D13	A 0	
8	D20	A 0	
9	D21	A 1	
10	D22	A 0	
11	D23	A 0	
12	D30	A 1	
13	D31	A 0	
14	D32	A 0	
15	D33	A 0	
16	R	A 0	

ارقام دهدهی در این تست ۳۳۳۳ است.

2	D02	A0	
3	D03	A0	
4	D10	A1	
5	D11	A1	
6	D12	A0	
7	D13	A0	
8	D20	A1	
9	D21	A1	
10	D22	A0	
11	D23	A0	
12	D30	A1	
13	D31	A1	
14	D32	A0	
15	D33	A0	
16	R	A1	

ارقام دهدهی در این تست ۹۵۶۱ است.

2	D02	A0	
3	D03	A0	
4	D10	A0	
5	D11	A1	
6	D12	A1	
7	D13	A0	
8	D20	A1	
9	D21	A0	
10	D22	A1	
11	D23	A0	
12	D30	A1	
13	D31	A0	
14	D32	A0	
15	D33	A1	
16	R	A1	

### ۳ بخش‌پذیری بر ۱۱

#### ۱.۳ ورودی و خروجی‌ها

مشابه مدار اول این مدار هم دارای ورودی ۱۶ بیتی است که نمایش BCD یک عدد دهدهی چهار رقمی است می‌باشد. خروجی این مدار هم یک سیگنال تک بیتی است که در صورتی که عدد دهدهی ورودی بر ۱۱ بخش‌پذیر باشد ۱ می‌شود و در غیر این صورت ۰ می‌شود.

#### ۲.۳ حل مسئله

فرض کنیم عدد ورودی دهدهی چهار رقمی ما به صورت  $(a_3a_2a_1a_0)$  در مبنای ۱۰ می‌باشد. طبق قاعده‌ی بخش‌پذیری بر ۱۱، لازم است ارقام را یکی در میان در ۱- ضرب کنیم و با هم جمع کنیم و سپس بررسی کنیم که حاصل مجموع بر ۱۱ بخش‌پذیر است یا نه. برای مثال برای یک عدد چهار رقمی، از سمت راست، مجموع رقم اول و قرینه‌ی رقم دوم رقم سوم و قرینه‌ی رقم چهارم را محاسبه می‌کنیم و سپس بررسی می‌کنیم که عدد به دست آمده بر ۱۱ بخش‌پذیر است یا نه.

درستی الگوریتم به صورت زیر قابل تحقیق است:

$$a_3a_2a_1a_0 \equiv 1000a_3 + 100a_2 + 10a_1 + a_0 \equiv 1001a_3 - a_3 + 99a_2 + a_2 + 11a_1 - a_1 + a_0$$

$$\equiv (a_2 + a_0) - (a_3 + a_1)$$

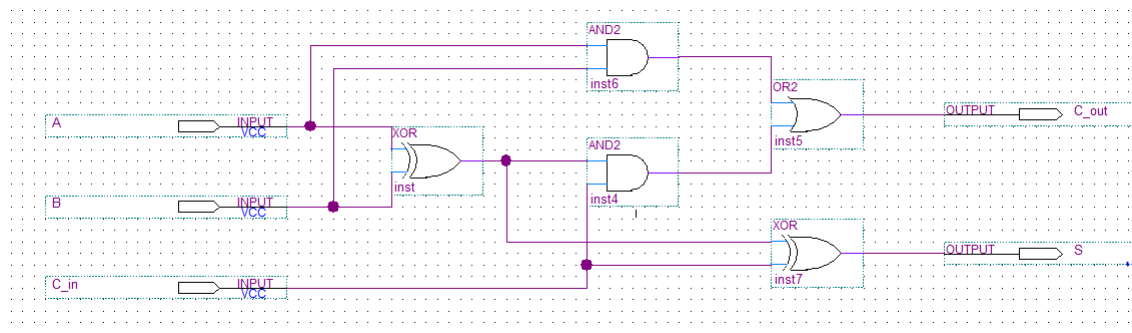
که این روابط هم نهشتی به ۱۱ هستند.

همانطور که مشخص است کافیسست مجموع ارقام در جایگاه فرد (با شروع جایگاه از صفر) را از مجموع ارقام در جایگاه زوج کم کرده و بخش پذیری حاصل را بر ۱۱ بررسی می‌کنیم. با توجه به اینکه مجموع دو رقم بین ۰ تا ۹ حداکثر ۱۸ و حداقل ۰ می‌باشد پس حاصل تفریق نیز حداکثر ۱۸ و حداقل برابر ۱۸- می‌باشد. بنابراین در شرایط مسئله ما کافی است این حاصل یکی از ۳ عدد دهدهی ۱۱، ۰ یا ۱۱- باشد تا عدد ورودی بر ۱۱ بخش پذیر باشد و خروجی ۱ شود. در سایر حالات عدد بخش تاپذیر و خروجی ۰ است.

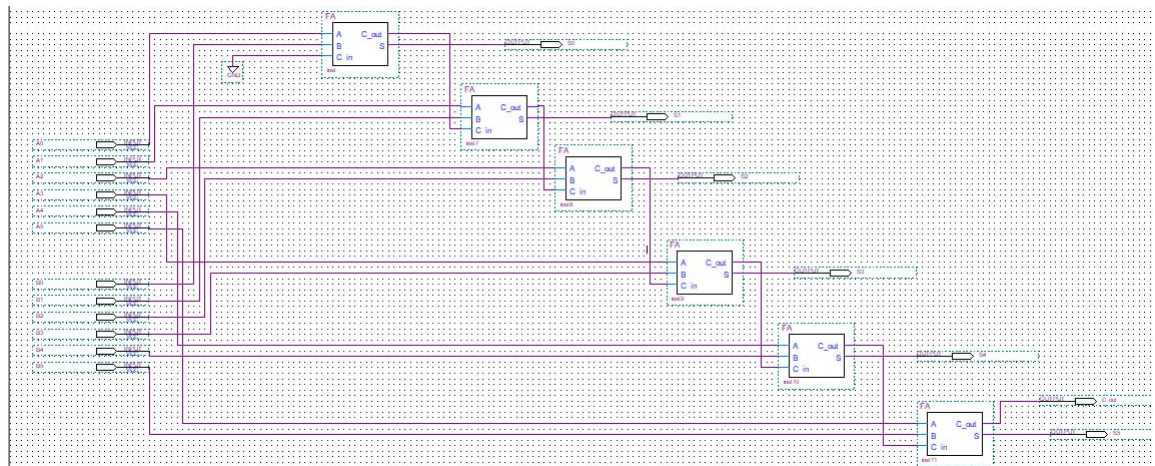
### ۳.۳ پیاده‌سازی مدار

با توجه به الگوریتم ارائه شده، به تعدادی مدار جمع‌کننده نیاز داریم، به منظور استفاده مومی در تمام قسمت‌هایی که به جمع‌کننده نیاز داریم، جمع‌کننده را به صورت ۶بیتی طراحی می‌کنیم. یک جمع‌کننده برای محاسبه مجموع ارقام جایگاه فرد و یک جمع‌کننده برای محاسبه مجموع اعداد با جایگاه ارقام زوج و همچنین یک جمع‌کننده به منظور عملیات تفریق نیاز داریم. همچنین برای محاسبه ی قرینه (منفی شده) مجموع ارقام جایگاه فرد نیاز به یک مدار محاسبه ی مکمل دو داریم. حال به طراحی مدارهای گفته شده می‌پردازیم. برای ساخت یک جمع کننده ی ۶ بیتی ابتدا نیاز به طراحی یک ماژول Full Adder داریم. این مدار را بدون استفاده از Half Adder طراحی می‌کنیم.

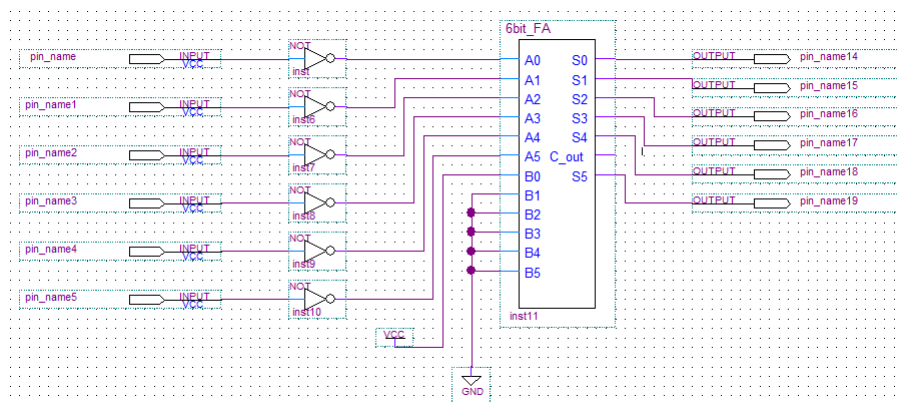
مدار نهایی به شکل زیر است:



حال مدار جمع‌کننده ۶بیتی را با استفاده از ماژول Full Adder طراحی شده به صورت Ripple Carry طراحی می‌کنیم. یعنی به ازای هر بیت یک Full Adder در نظر گرفته و مقدار ۰ را به عنوان carry ورودی جمع‌کننده بیت کم ارزش داده و carry خروجی هر جمع‌کننده را به عنوان carry ورودی جمع‌کننده بعدی با ارزش بالاتر می‌دهیم و نهایتاً خروجی carry جمع‌کننده آخر را به عنوان carry کل مدار جمع‌کننده ۶بیتی خروجی داده می‌شود. باتوجه به این توصیف تصویر جمع‌کننده ۶بیتی را در زیر مشاهده می‌کنید.



با توجه به اینکه مکمل ۲ عددی باینری از نقیض کردن تک‌تک بیت‌های آن و سپس جمع حاصل با ۱ به دست می‌آید، برای طراحی مکمل ۲ شش بیتی از جمع‌کننده‌ی ۶ بیتی که تا اینجا طراحی کرده‌ایم استفاده می‌کنیم و در یک ورودی آن عربیتی‌ای با اعمال NOT روی بیت‌های آن‌ها و در ورودی دیگر اعداد ۰۰۰۰۰۱ را می‌دهیم. با توجه به این توضیحات، خروجی این جمع‌کننده در واقع مکمل دوی عدد ورودی است. مدار طراحی شده را در زیر مشاهده می‌کنیم.

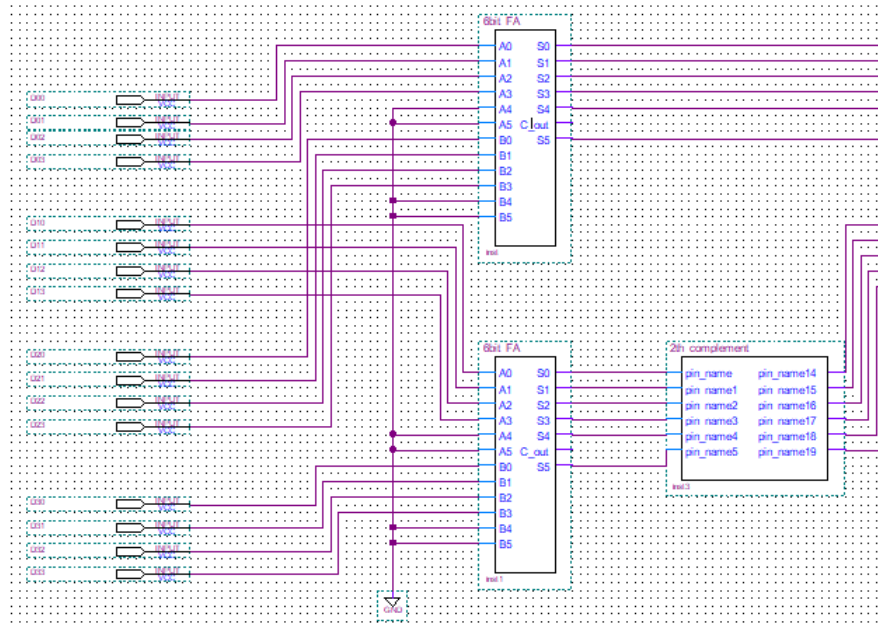
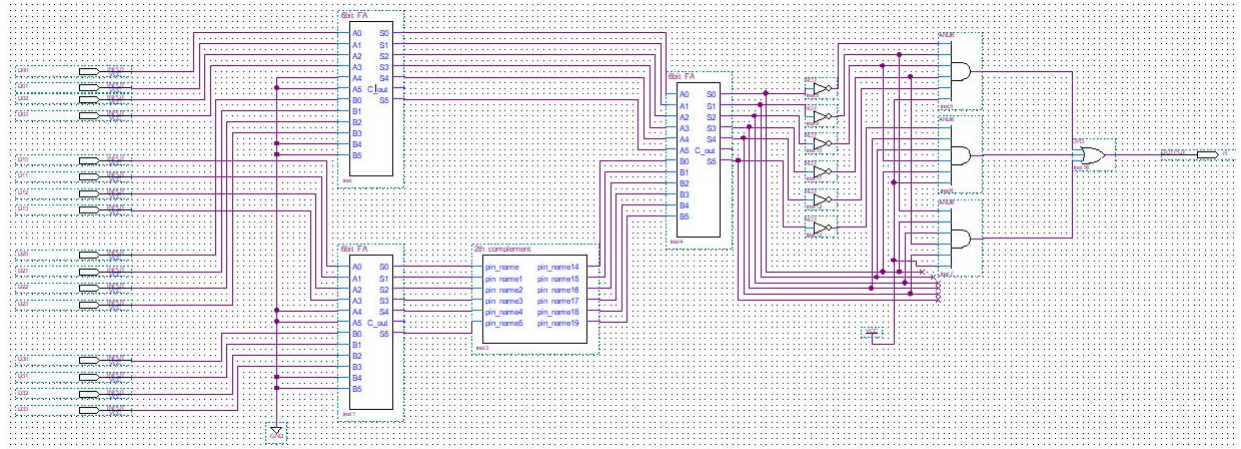


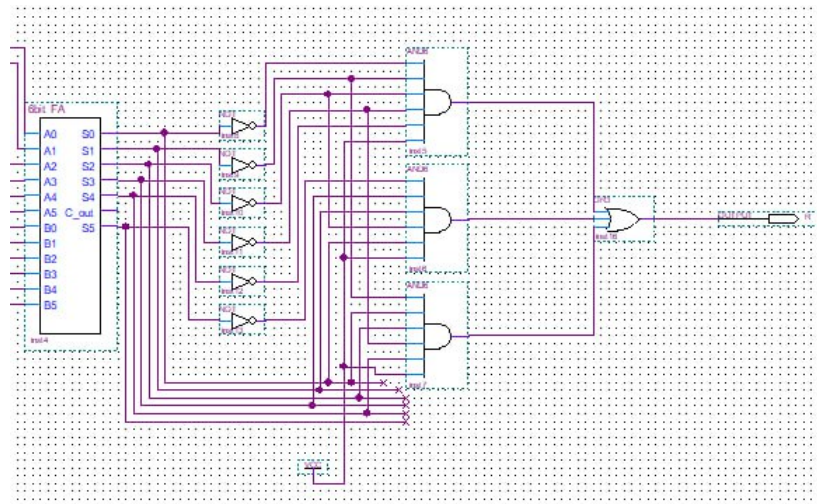
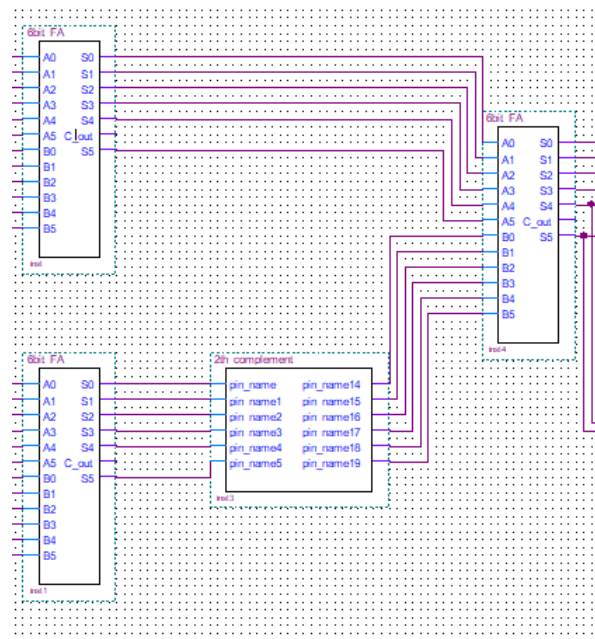
نهایتاً مطابق الگوریتم، یک جمع‌کننده‌ی ۶ بیتی برای رقم صفرم و دوم و یک جمع‌کننده ۶ بیتی برای رقم یکم و سوم در نظر می‌گیریم. از آنجا که ارقام اعداد باینری مثبت ۴ بیتی هستند، آن‌ها را با استفاده از دو بیت ۰ sign extend می‌کنیم و به ورودی جمع‌کننده‌ها می‌دهیم. با توجه به اینکه جمع دو عدد ۴ بیتی حداکثر ۵ بیت است، بنابراین بیت ششم هر دو جمع‌کننده حتماً برابر صفر است. خروجی جمع‌کننده دوم را مستقیماً به مدار محاسبه‌ی مکمل ۲ می‌فرستیم و عددی نامثبت در قالب ۶ بیت نتیجه می‌شود. نهایتاً حاصل جمع اول و عدد مکمل شده را به عنوان ورودی به جمع‌کننده ۶ بیتی می‌دهیم.

نهایتاً با بررسی خروجی ۶ بیتی جمع‌کننده آخر، که در سیستم مکمل ۲ مقداری بین ۱۸- تا ۱۸ دارد، بخش پذیری بر ۱۱ را محاسبه می‌کنیم. کافی است در جدول کارنو میترم‌های معادل ۱۱ و ۰ و ۱۱ را برابر ۱ و



خانه‌های درون بازدهی ۱۸ تا ۱۸- را ۰ و سایر خانه ها را dont care می‌گذاریم. مدار نهایی در شکل زیر آمده است.





حال مدار را با یک سری سیگنال تست می‌کنیم:  
در این تست مقدار دهنده‌ی عدد ورودی ۱۱۱۱ می‌باشد.

2	D02	A 0	
3	D03	A 0	
4	D10	A 1	
5	D11	A 0	
6	D12	A 0	
7	D13	A 0	
8	D20	A 1	
9	D21	A 0	
10	D22	A 0	
11	D23	A 0	
12	D30	A 1	
13	D31	A 0	
14	D32	A 0	
15	D33	A 0	
16	R	A 1	

در این تست مقدار دهنده‌ی عدد ورودی ۴۰۱۵ می‌باشد.

2	D02	A 1	
3	D03	A 0	
4	D10	A 1	
5	D11	A 0	
6	D12	A 0	
7	D13	A 0	
8	D20	A 0	
9	D21	A 0	
10	D22	A 0	
11	D23	A 0	
12	D30	A 0	
13	D31	A 0	
14	D32	A 1	
15	D33	A 0	
16	R	A 1	

در این تست مقدار دهنده‌ی عدد ورودی ۴۳۱۴ می‌باشد.

2	D02	A 1	
3	D03	A 0	
4	D10	A 1	
5	D11	A 0	
6	D12	A 0	
7	D13	A 0	
8	D20	A 1	
9	D21	A 1	
10	D22	A 0	
11	D23	A 0	
12	D30	A 0	
13	D31	A 0	
14	D32	A 1	
15	D33	A 0	
16	R	A 0	