گزارش آزمایش

آزمایشگاه مدار منطقی

آزمایش اول

آشنایی با محیط‌های شبیه‌سازی

اعضای گروه:

نام و نام خانوادگی	شماره دانشجویی
روناک ابوطالبی	۴۰۴۱۰۵۳۹۴
حسین مرادی	۴۰۴۱۰۶۳۶۶

تاریخ تحویل: ۱۴۰۵/۰۵/۱۴

تابستان ۱۴۰۵

فهرست مطالب

۱ مقدمه ۲

۲ رسم مدار با **Fritzing** ۳

۱.۲ آشنایی با بردبورد ۳

۲.۲ مدار ساده LED-مقاومت-باتری ۳

۳.۲ مدار تراشه ۷۴۰۴ . . . . . ۵

۳ ساخت مدار با **Logisim** ۷

۱.۳ مدار جمع‌کننده کامل(Full Adder) ۷

۲.۳ مدار جمع‌کننده/تفریق‌کننده چهاربیتی ۸

۳.۳ تست مدارها ۱۰

۴ ساخت مدار با **Proteus** ۲۱

۱.۴ طراحی مدار ۲۱

۲.۴ تست مدار ۲۳

۵ نتیجه‌گیری ۲۸

۱ مقدمه

هدف در این آزمایش این است که با سه نرم‌افزار طراحی و شبیه‌سازی مدارهای منطقی آشنا شویم. نرم‌افزارهای مذکور عبارت‌اند از:

- Fritzing
- Logisim
- Proteus

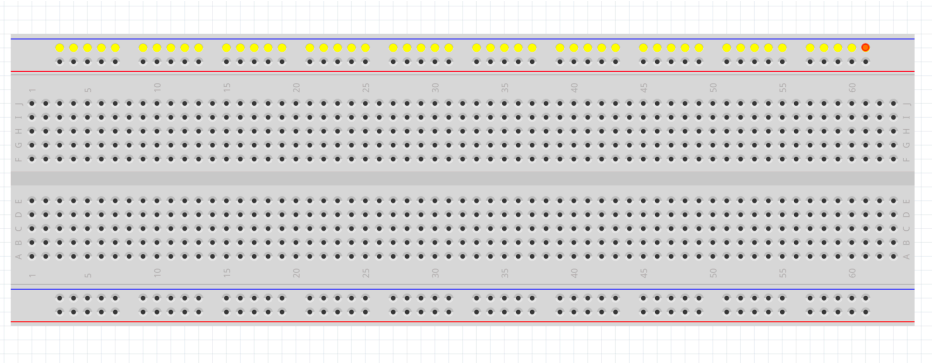
در قسمت اول، از نرم‌افزار Fritzing استفاده می‌کنیم تا با بردبورد[1]، اتصالات و نحوه عمکرد آن آشنا شویم. در قسمت دوم، از نرم‌افزار Logisim برای رسم کردن و تست یک مدار ترکیبی استفاده می‌کنیم و در نهایت، در بخش آخر، مدار دیگری را با استفاده از نرم‌افزار Proteus می‌سازیم.

[1] Breadboard

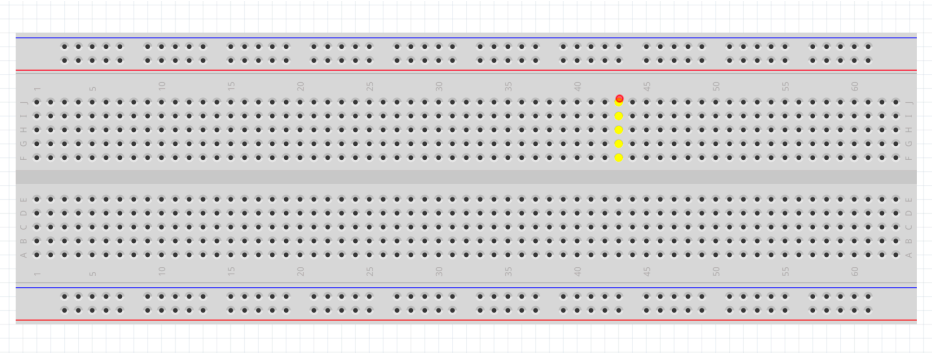
۲ رسم مدار با Fritzing

۱.۲ آشنایی با بردبورد

ابتدا می‌خواهیم با اتصالات بردبورد و نحوه کار با آن آشنا شویم.



شکل ۱.۱.۲: جفت‌ردیف‌های افقی بالا و پایین

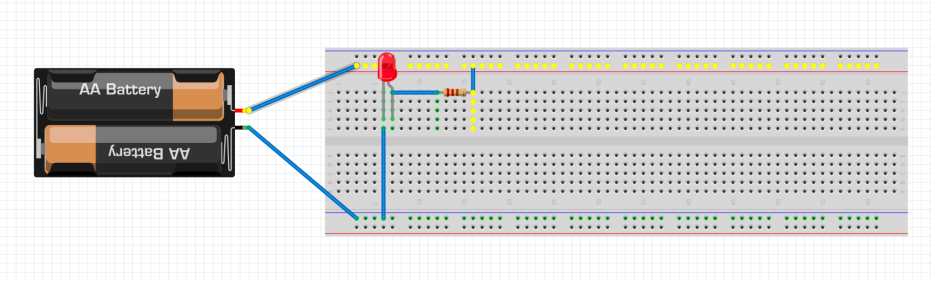


شکل ۲.۱.۲: ستون‌های عمودی پنج‌تایی

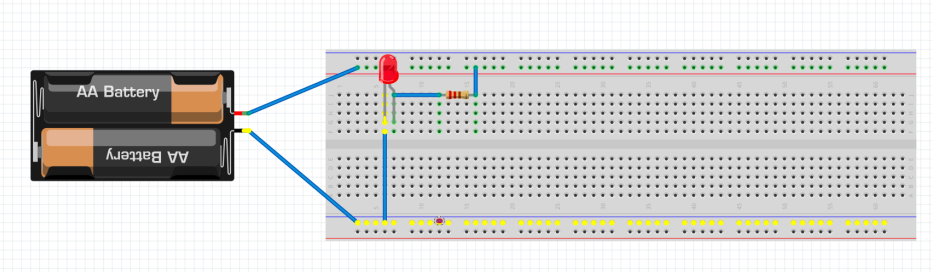
همانطور که در شکل ۱.۱.۲ مشاهده می‌شود، چهار ردیف طولانی از ورودی‌ها، که به صورت دو جفت موازی در بالا و پایین بردبورد قرار دارند به هم متصل هستند. در شکل ۲.۱.۲ نیز مشاهده می‌شود سایر وروری‌های بردبورد، به صورت گروه‌های پنج‌تایی عمودی به هم متصل می‌شوند.

۲.۲ مدار ساده LED-مقاومت-باتری

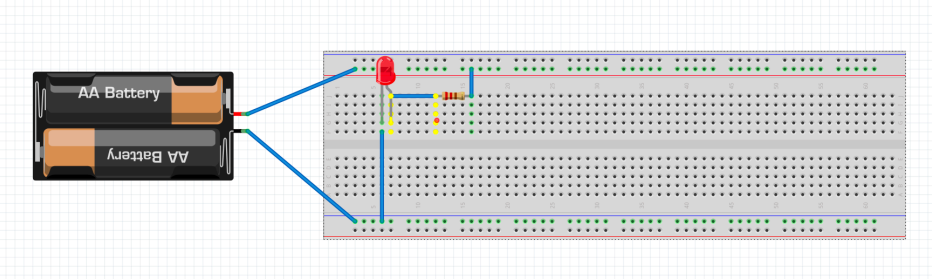
پس از آشنایی با نحوه کار بردبورد، یک مدار ساده متشکل از یک عدد LED، یک مقاومت و باتری را با استفاده از آن می‌بندیم.



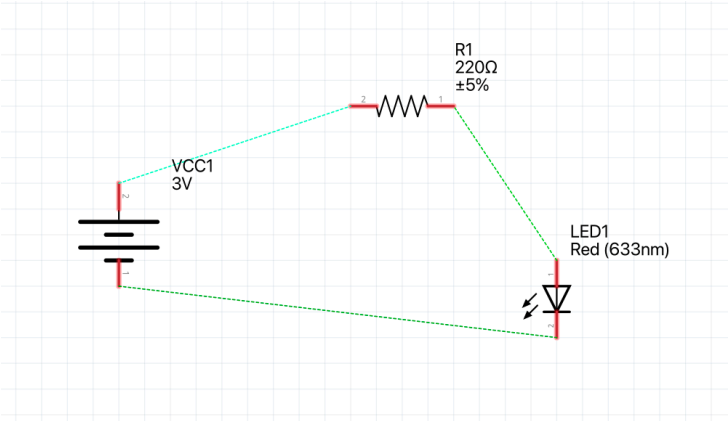
شکل ۱.۲.۲



شکل ۲.۲.۲



شکل ۳.۲.۲

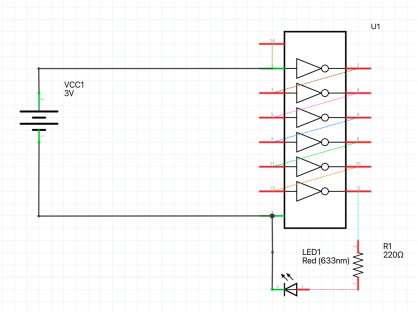


شکل ۴.۲.۲

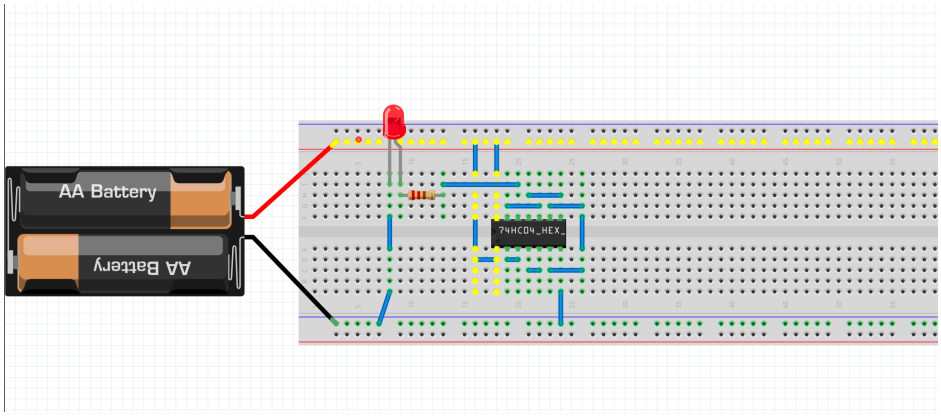
در شکل های ۱.۲.۲ تا ۳.۲.۲ اتصالات قطعات نمایش داده شده‌است. معمولا از چهار ردیف افقی طولانی ورودی‌های متصل به هم به عنوان محل اتصال به منبع تغذیه(در اینجا باتری) استفاده می‌شود. علت وجود خطوط آبی و قرمز در کنار این ردیف ها نیز همین است.

۳.۲ مدار تراشه ۷۴۰۴

خط افقی وسط بردبورد، که گروه های پنج‌تایی بالا و پایین را از هم جدا می‌کند، برای نصب IC ها روی بردبورد استفاده می‌شود. اگر IC را طور دیگری روی بردبورد ببندیم، بین پایه‌های آن اتصال کوتاه برقرار می‌شود.

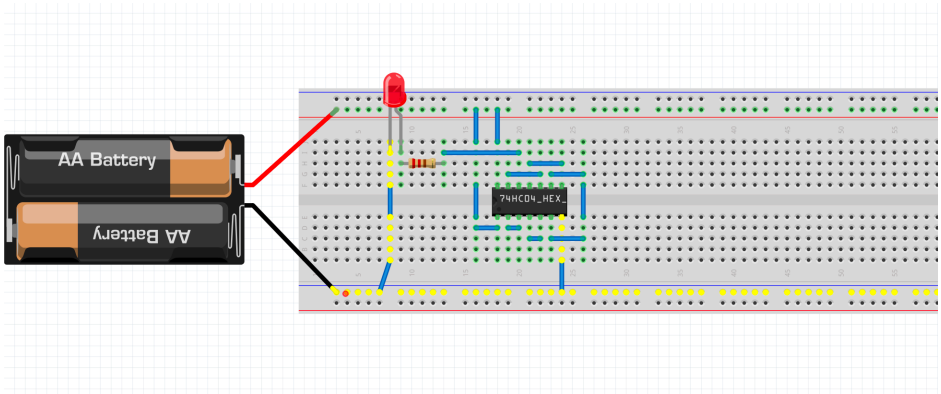


شکل ۱.۳.۲

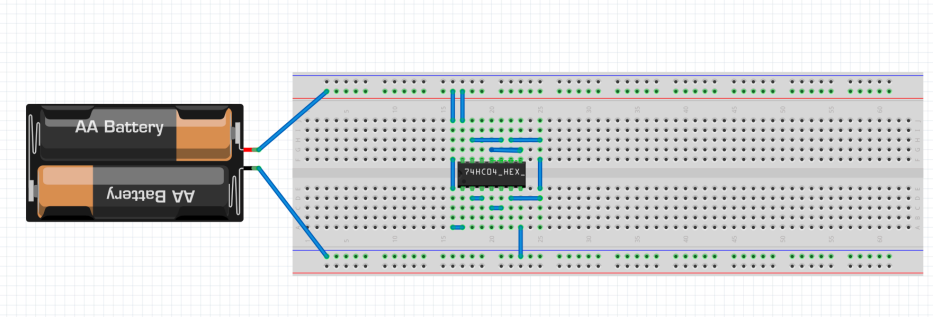


شکل ۲.۳.۲

تراشه ۷۴۰۴ در واقع ۶ گیت NOT درون خود دارد که به جز پایه ۷ و ۱۴ که به ترتیب GND و VCC آن هستند، هر دو جفت پایه مجاور دیگر ورودی و خروجی یک گیت می‌باشند.



شکل ۳.۳.۲



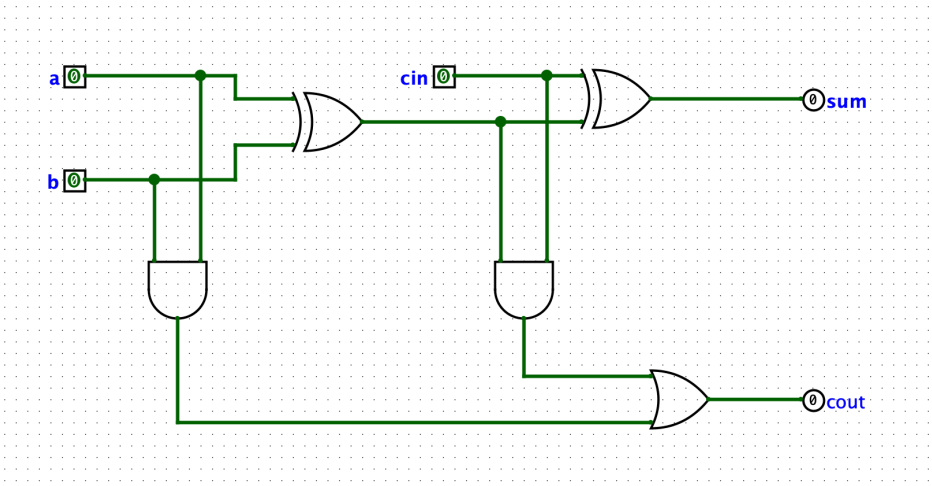
شکل ۴.۳.۲

۳ ساخت مدار با Logisim

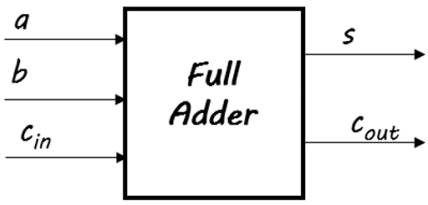
در این قسمت با استفاده از نرم‌افزار Logisim دو مدار ترکیبی می‌سازیم و تست می‌کنیم.

۱.۳ مدار جمع‌کننده کامل(Full Adder)

مدار جمع‌کننده کامل، سه ورودی a، b و c_{in} را دریافت می‌کند و حاصل جمع آنها(sum) و همچنین carry تولید شده(c_{out}) را خروجی می‌دهد. تصویر این مدار را در شکل ۱.۱.۳ مشاهده می‌کنید که با استفاده از گیت های AND، OR و XOR ساخته شده‌است.



شکل ۱.۱.۳: مدار Full Adder

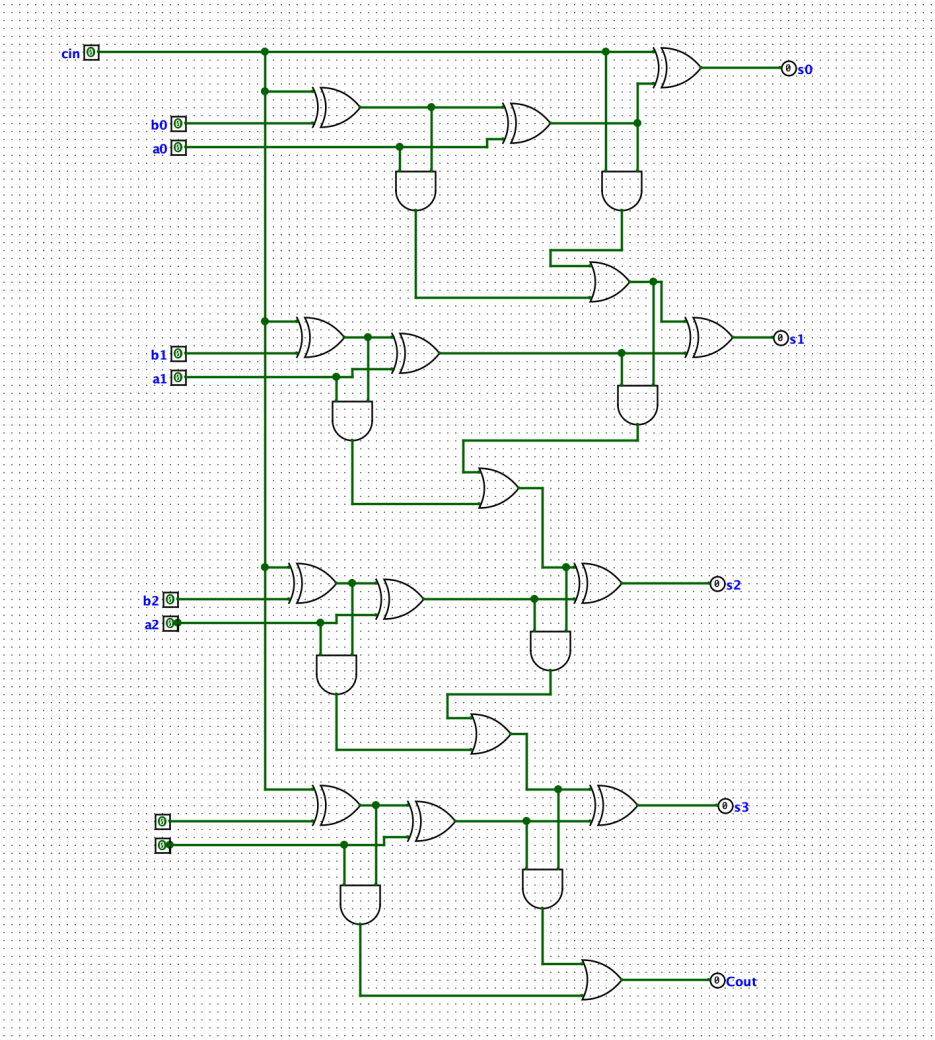


inputs			outputs	
a	b	c_{in}	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

شکل ۲.۱.۳

۲.۳ مدار جمع‌کننده/تفریق‌کننده چهاربیتی

حال با استفاده از چهار مدار جمع‌کننده کامل که در بخش ۱.۳ ساختیم، یک مدار جمع‌کننده/تفریق‌کننده[1] می‌سازیم که اگر مقدار ورودی c_{in} صفر باشد، حاصل جمع و در غیر این صورت، حاصل تفریق دو ورودی چهاربیتی را تولید کند.



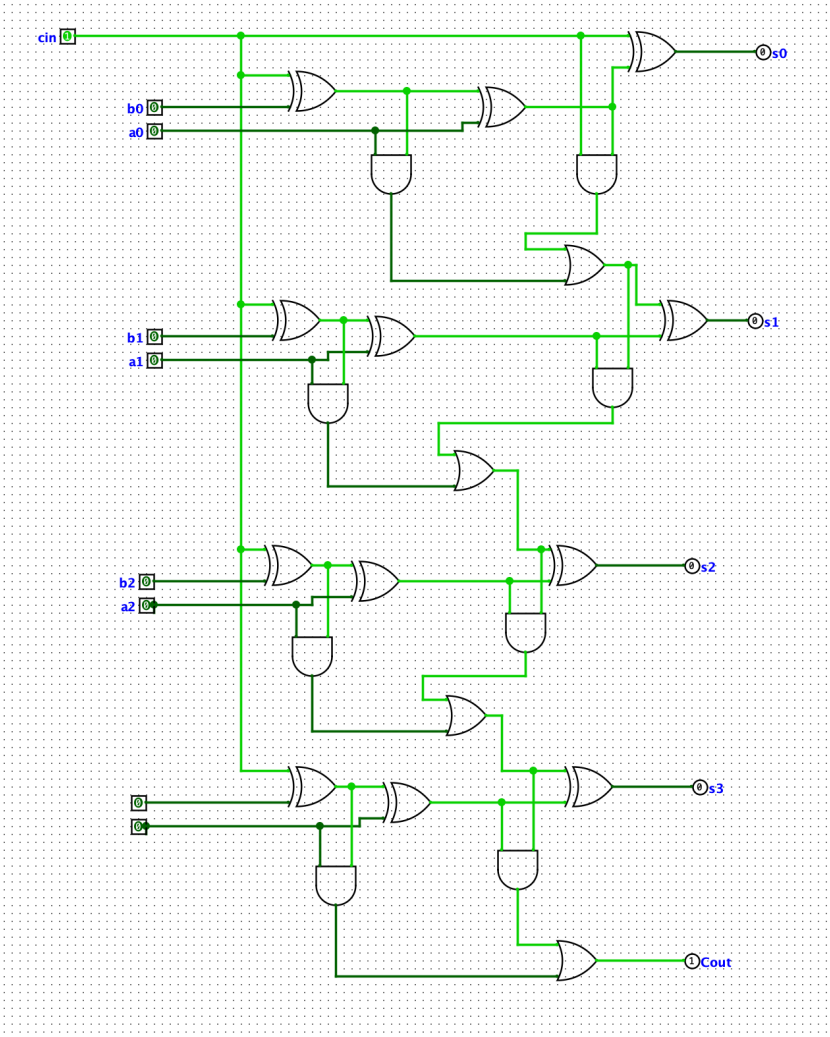
شکل ۱.۲.۳

عملکرد مدار شکل ۱.۲.۳ به این صورت است که اگر $c_{in} = 0$ گیت‌های XOR که یک ورودی آنها بیت های B است، آن بیت را بدون تغییر به Full Adder می‌فرستند تا با بیت متناظر از A جمع شود.

[1] Adder/Subtractor

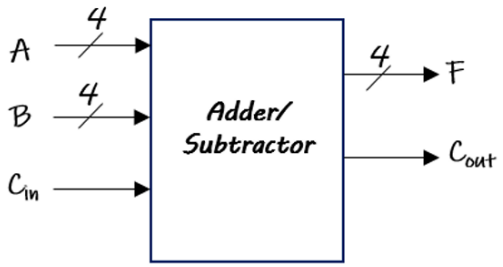
اما اگر $c_{in} = 0$، گیت‌های XOR تمام بیت‌های B را NOT می‌کنند و در نهایت عددی که با A جمع می‌شود، $\overline{B}$ خواهد بود. با در نظر گرفتن ورودی $c_{in} = 1$، در نهایت خروجی محاسبه شده توسط مدار، $A + \overline{B} + 1$ است که در آن $\overline{B} + 1$، مکمل دو[1] B می‌باشد. یعنی مدار در نهایت $A - B$ را محاسبه می‌کند.

$$A + \overline{B} + 1 = A + 2\text{'s-complement}(B) = A - B$$



شکل ۲.۲.۳

[1] 2's complement



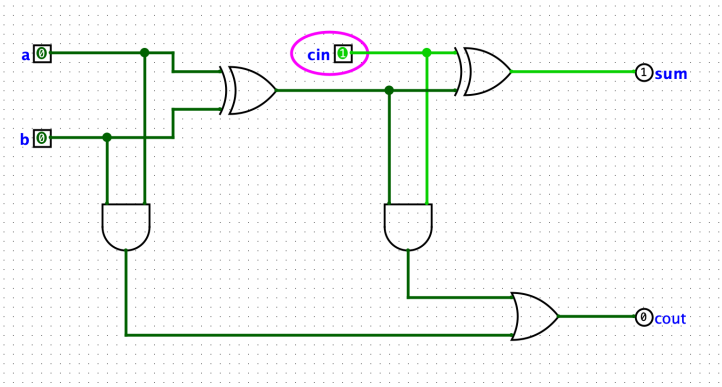
C_{in}	F
0	A+B
1	A-B

شکل ۳.۲.۳

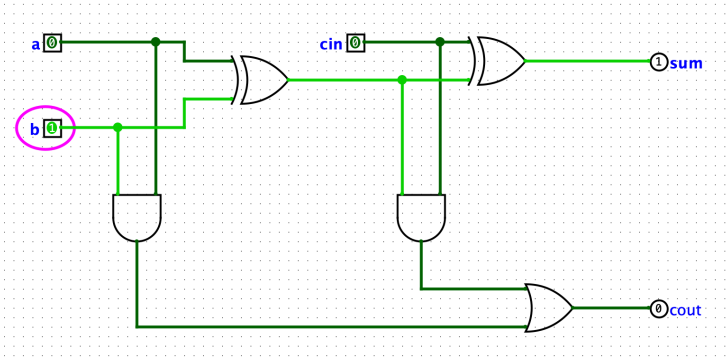
۳.۳ تست مدارها

تست مدار **Full Adder**

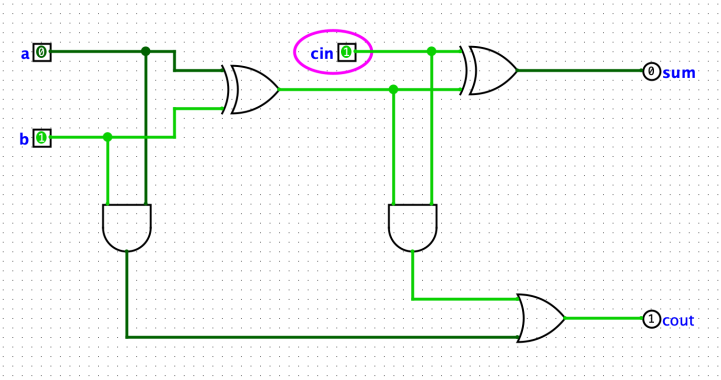
در تمام تست‌های ۱.۳.۳ تا ۷.۳.۳، خروجی مدار با جدول ۲.۱.۳ تطابق دارد.



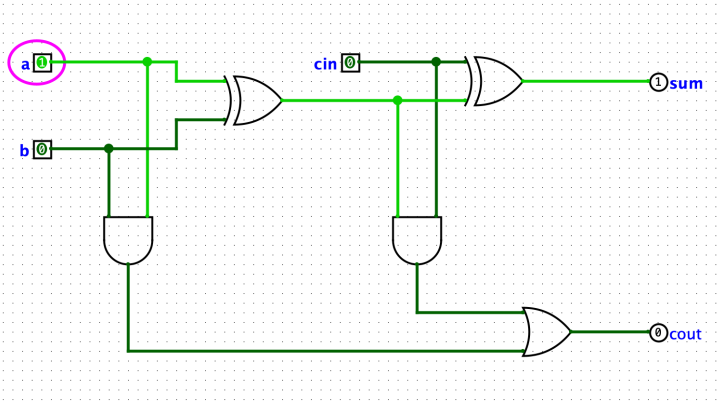
شکل ۱.۳.۳



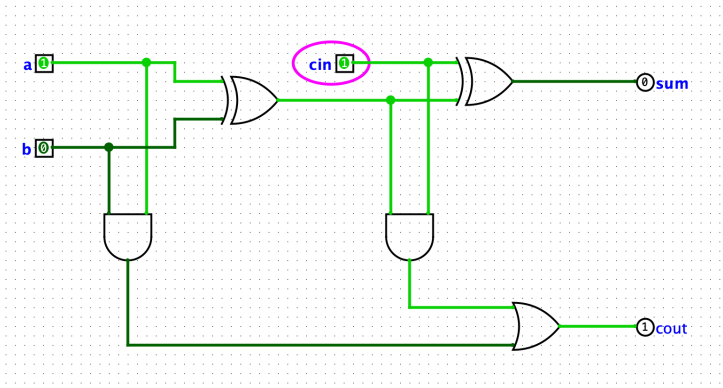
شکل ۲.۳.۳



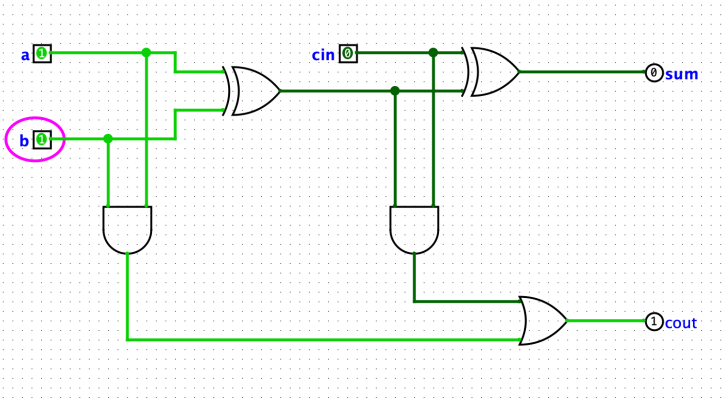
شکل ۳.۳.۳



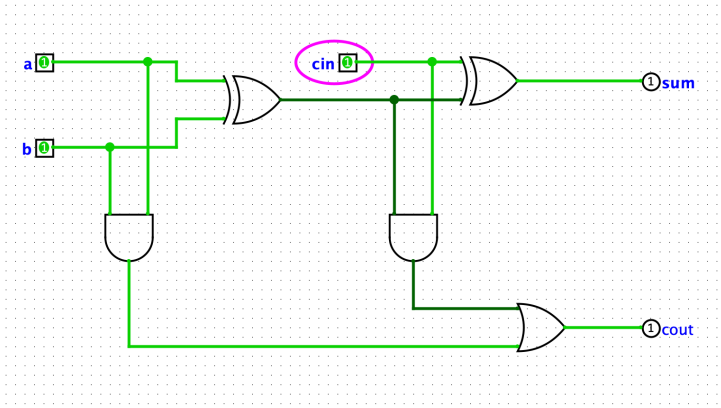
شکل ۴.۳.۳



شکل ۵.۳.۳



شکل ۶.۳.۳



شکل ۷.۳.۳

تست مدار جمع‌کننده/تفریق‌کننده

شکل ۸.۳.۳:

$$c_{in} = 0 \rightarrow \text{addition}$$
$$\text{binary: } (1111)_2 + (0100)_2 = (10011)_2$$
$$\text{decimal: } 15 + 4 = 19$$

شکل ۹.۳.۳:

$$c_{in} = 0 \rightarrow \text{addition}$$
$$\text{binary: } (0111)_2 + (0010)_2 = (01001)_2$$
$$\text{decimal: } 7 + 2 = 9$$

شکل ۱۰.۳.۳:

$$c_{in} = 0 \rightarrow \text{addition}$$
$$\text{binary: } (1111)_2 + (1111)_2 = (11110)_2$$
$$\text{decimal: } 15 + 15 = 30$$

شکل ۱۱.۳.۳:

$$c_{in} = 1 \rightarrow \text{subtraction}$$
$$\text{binary: } (1000)_2 - (0110)_2 = (0010)_2$$
$$\text{decimal: } 8 - 6 = 2$$

شکل ۱۲.۳.۳:

$$c_{in} = 1 \rightarrow \text{subtraction}$$
$$\text{binary: } (0000)_2 - (0101)_2 = (1011)_2$$
$$\text{decimal: } 0 - 5 = -5$$

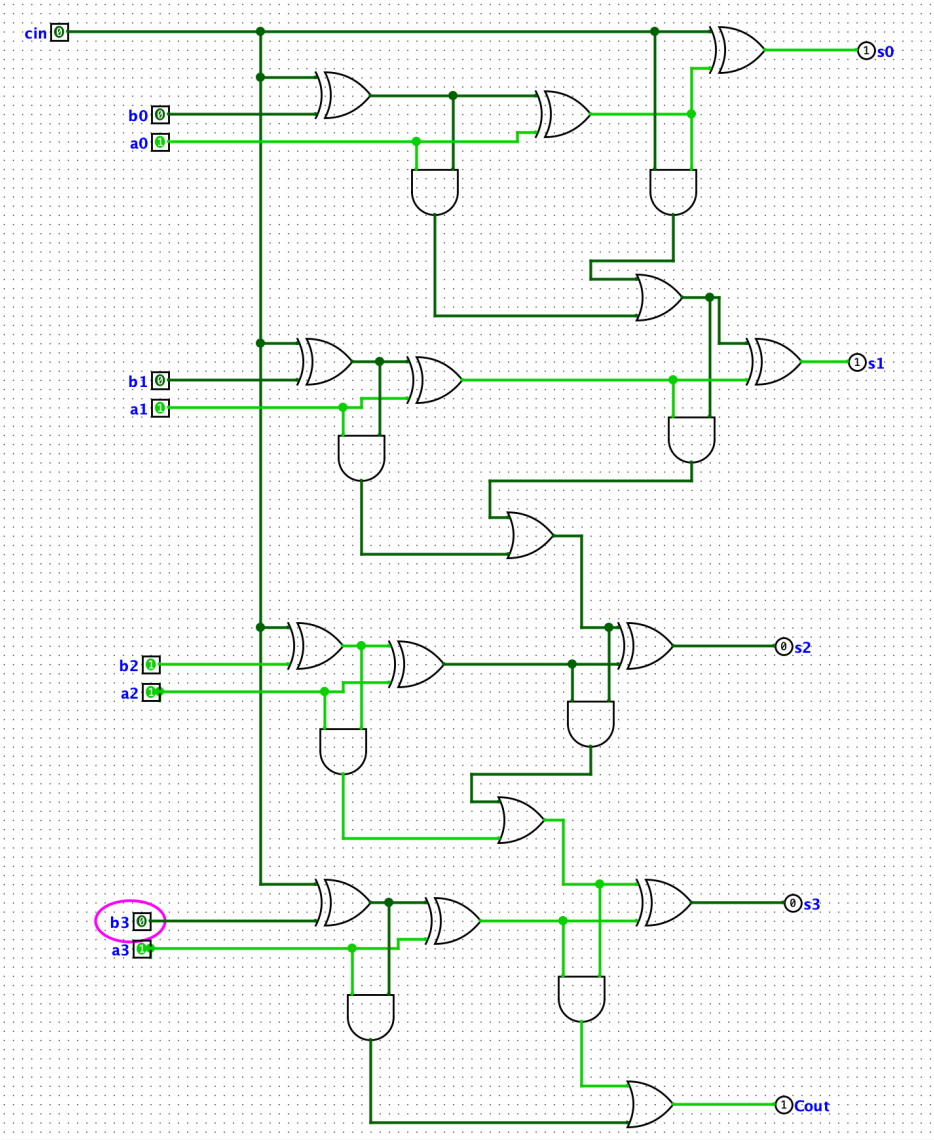
توجه کنید که در سیستم مکمل دو، MSB نمایانگر علامت است و در اینجا جواب ما عددی منفی است. برای بدست آوردن اندازه جواب از آن مکمل دو می‌گیریم.

شکل ۱۳.۳.۳:

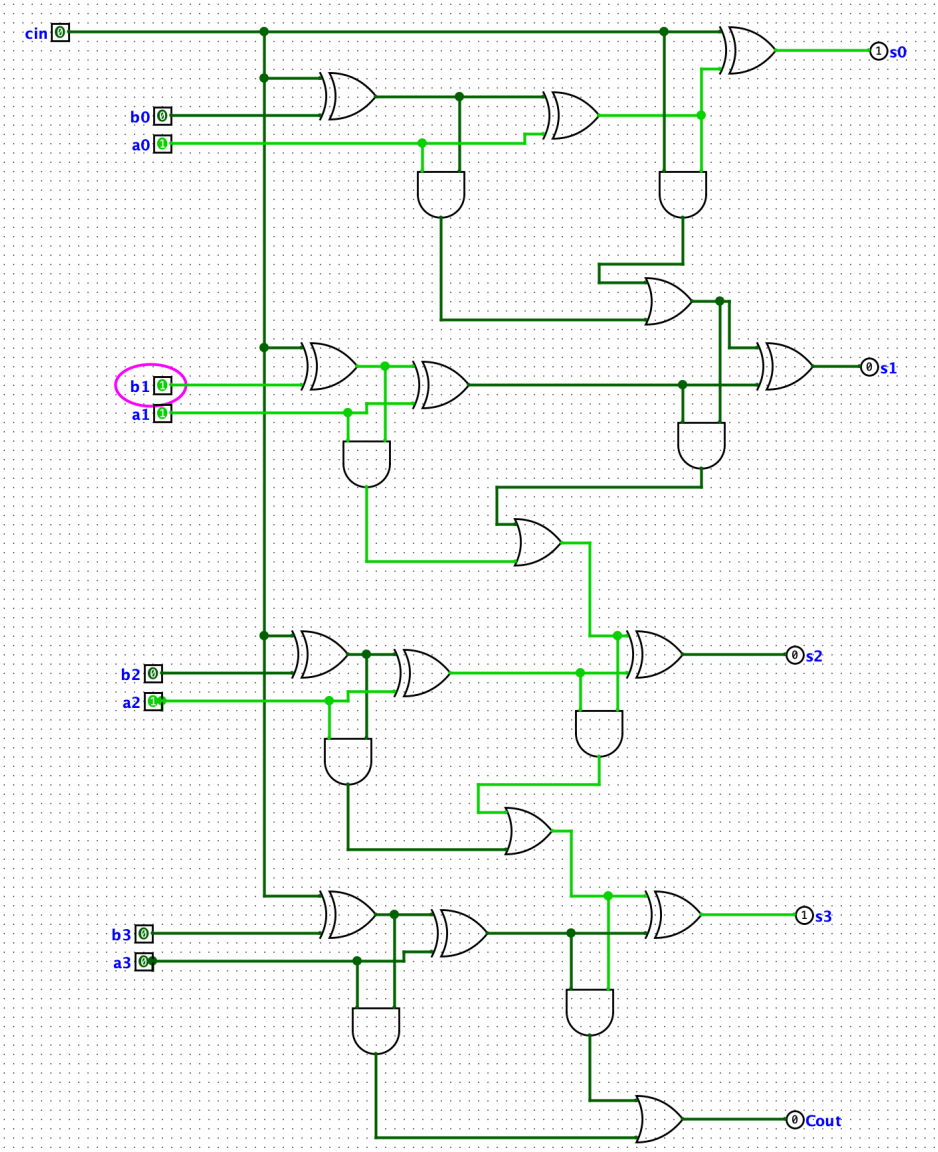
$$c_{in} = 1 \rightarrow \text{subtraction}$$
$$\text{binary: } (0111)_2 - (1000)_2 = (1111)_2$$
$$\text{decimal: } 7 - 8 = -1$$

شکل ۱۴.۳.۳:

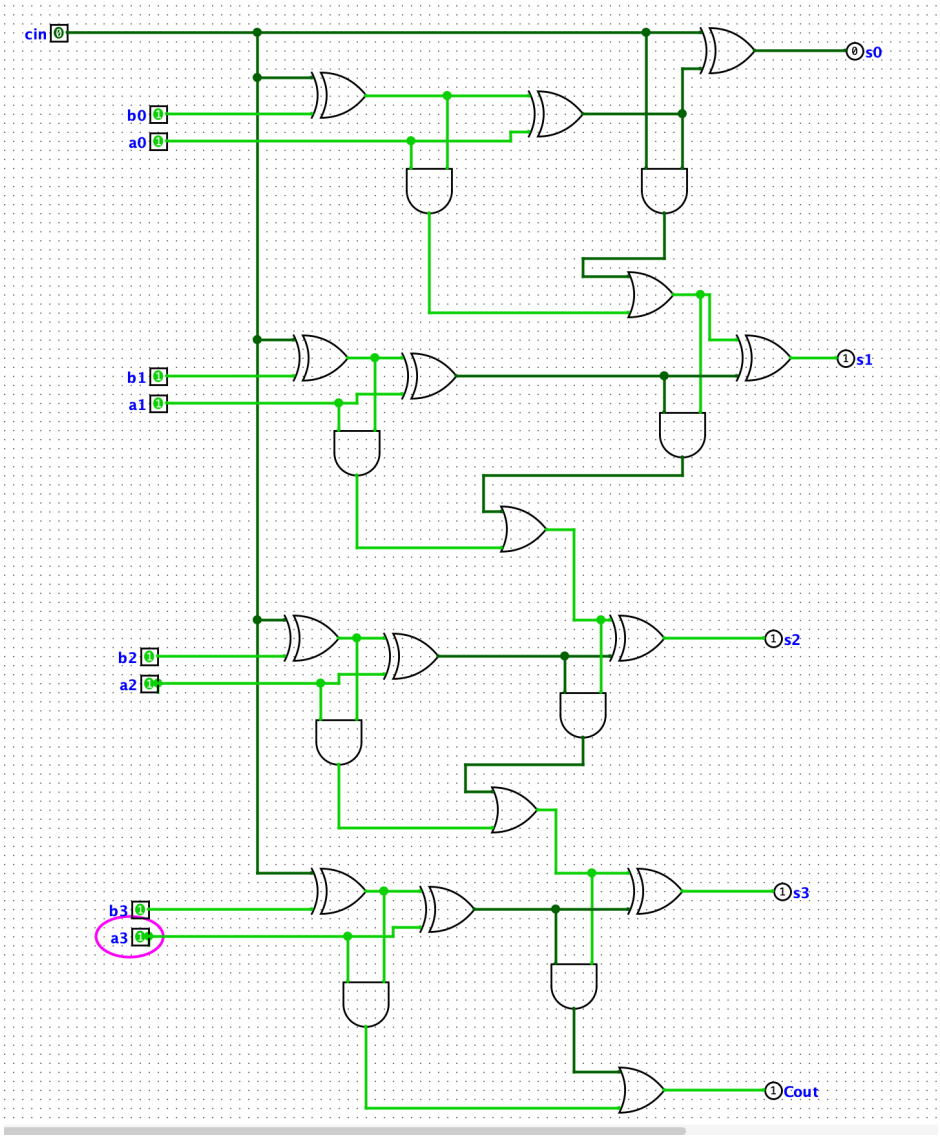
$$c_{in} = 1 \rightarrow \text{subtraction}$$
$$\text{binary: } (0100)_2 - (0010)_2 = (0010)_2$$
$$\text{decimal: } 4 - 2 = 2$$



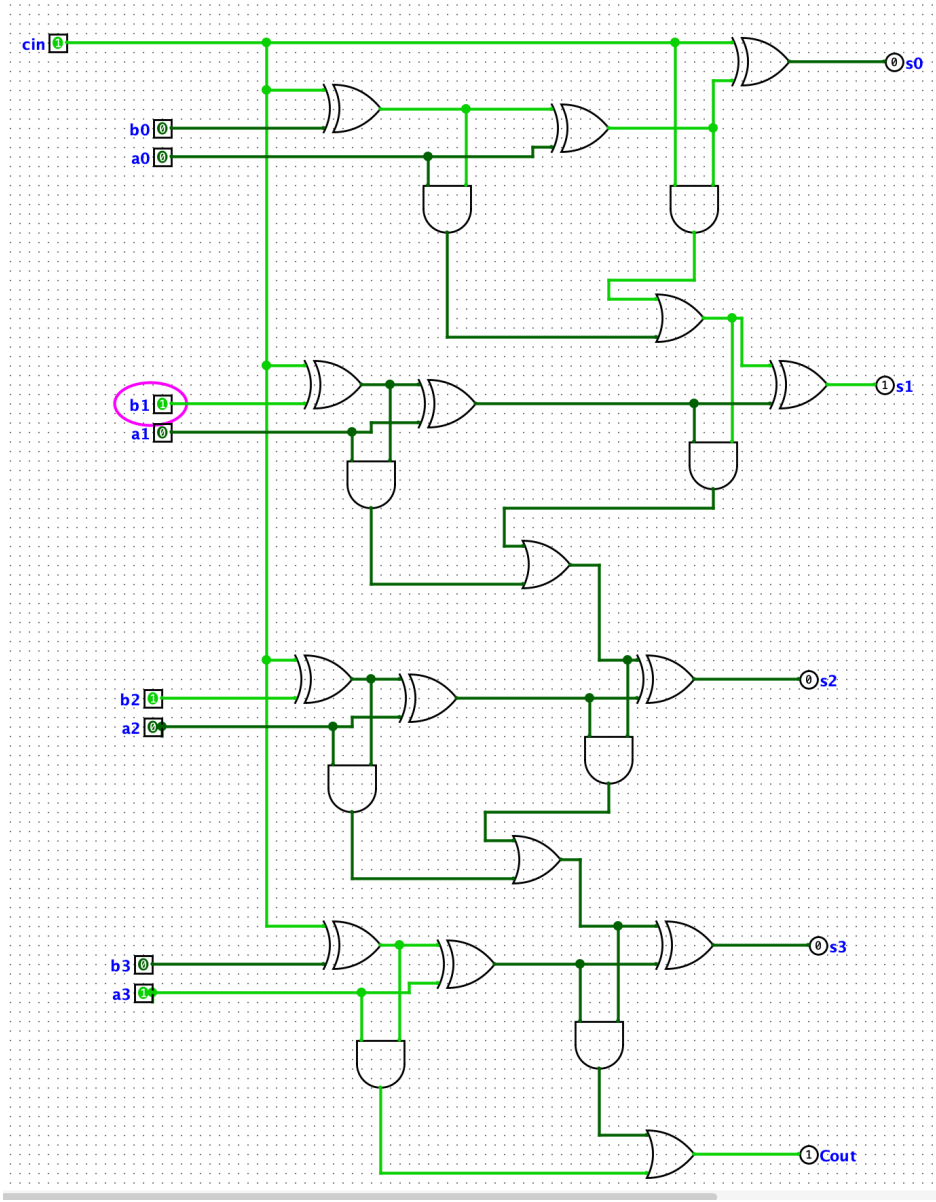
شکل ۸.۳.۳



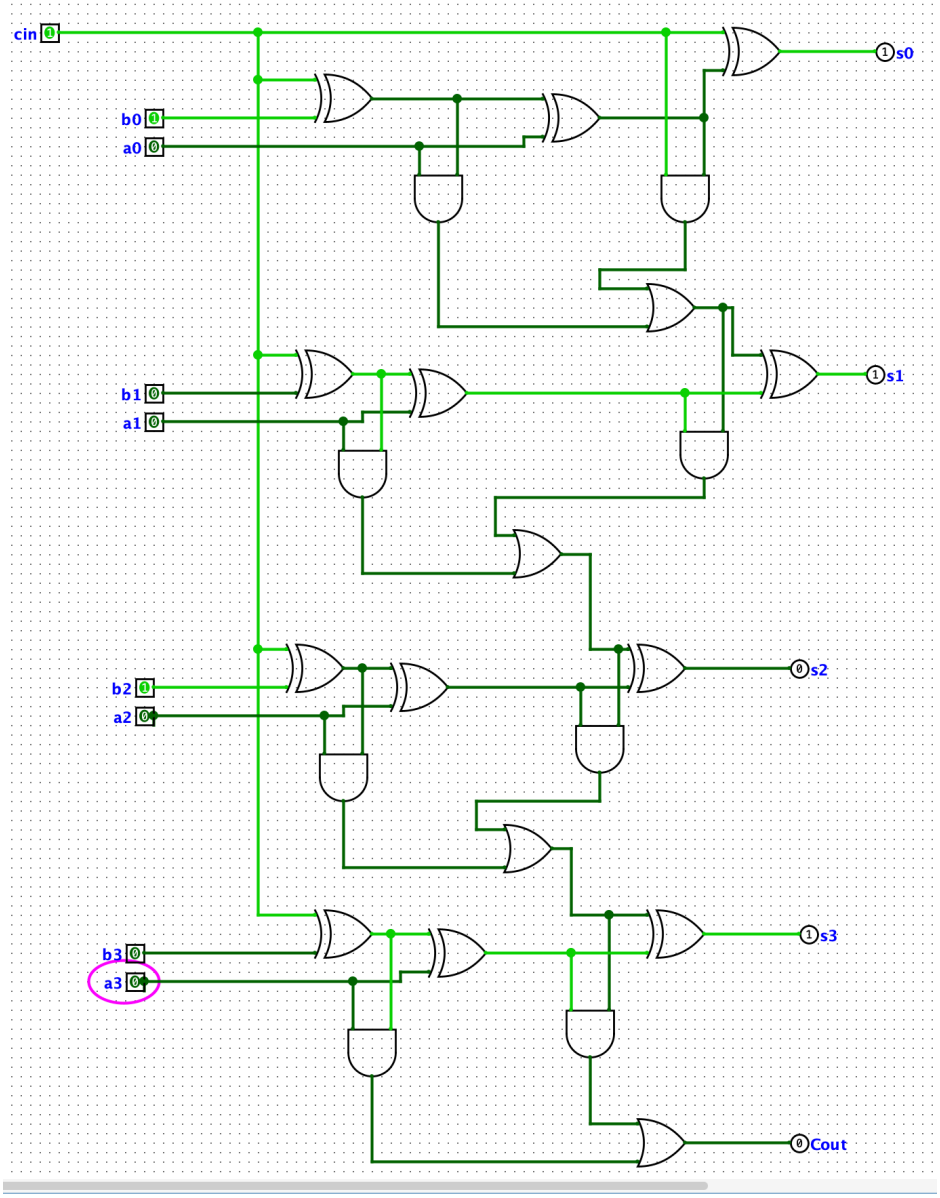
شکل ۹.۳.۳



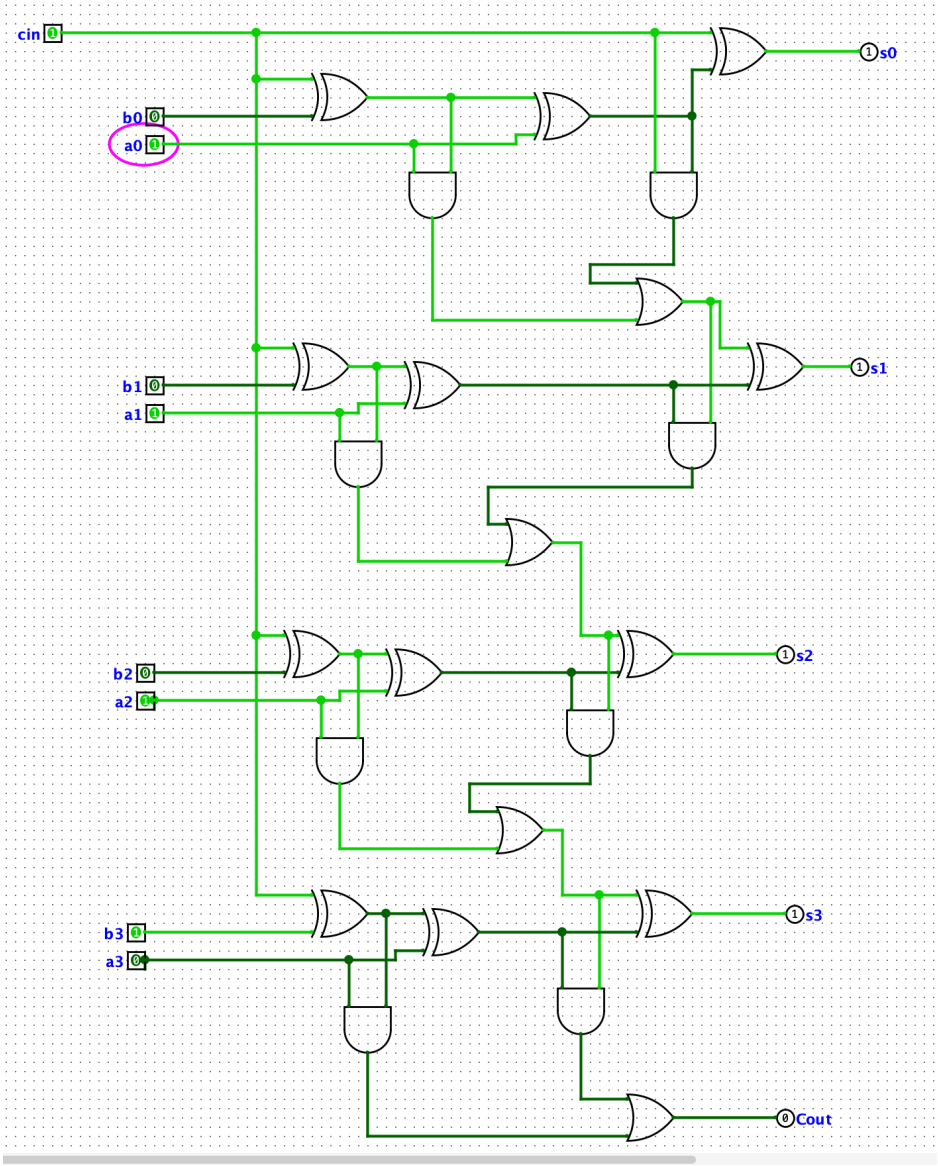
شکل ۱۰.۳.۳



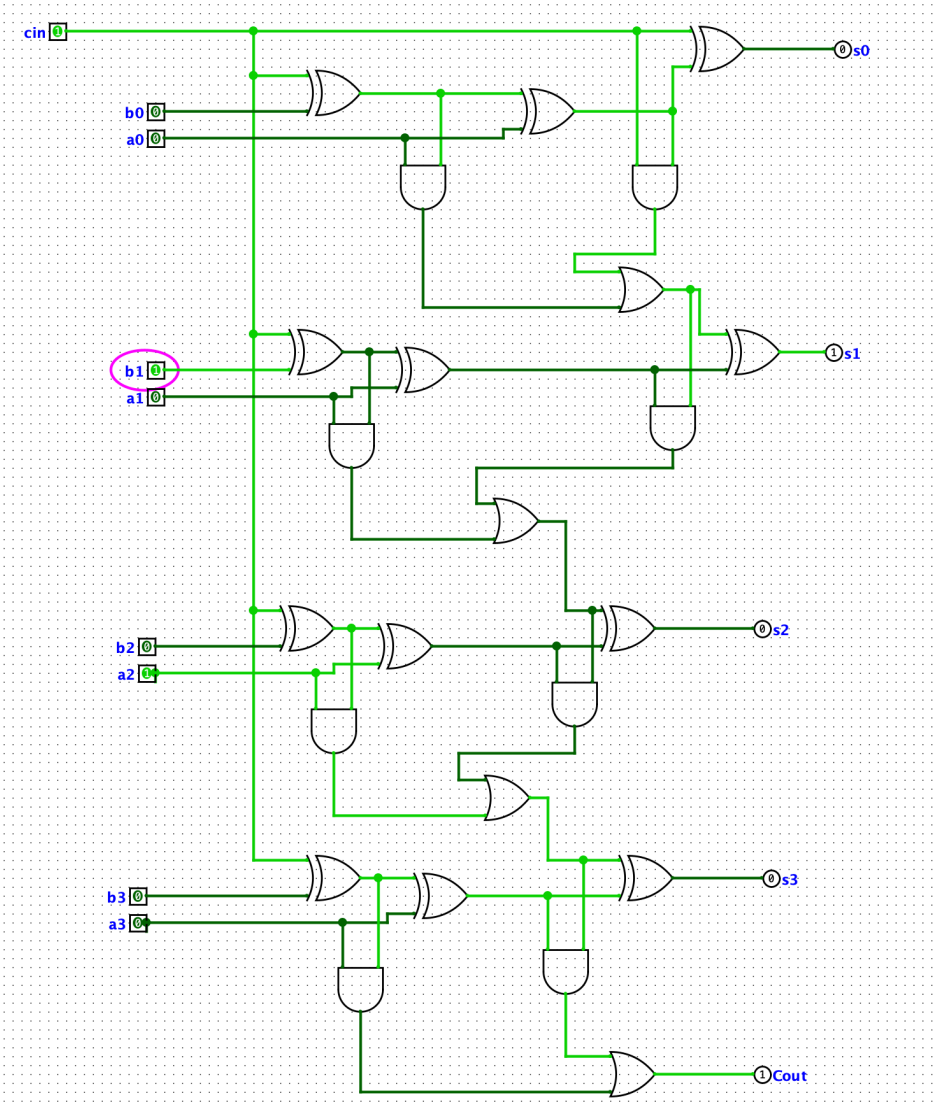
شکل ۱۱.۳.۳



شکل ۱۲.۳.۳



شکل ۱۳.۳.۳



شکل ۱۴.۳.۳

۴ ساخت مدار با Proteus

در این قسمت قصد داریم با استفاده از نرم‌افزار Proteus یک مدار جمع کننده چهاربیتی از نوع Carry-Look-Ahead(CLA) بسازیم.

۱.۴ طراحی مدار

مدار جمع‌کننده Carry-Look-Ahead یک مدار جمع‌کننده پرسرعت است که بر مشکل تاخیر انتشار [1] که در جمع‌کننده های Ripple Carry Adder وجود دارد غلبه می‌کند. زیرا در جمع‌کننده های RCA هر بیت برای محاسبه حاصل جمع نهایی باید منتظر دریافت Carry از بیت قبلی بماند. اما در مدار CLA تمام ارقام نقلی بیت ها به صورت همزمان و مستقیما از سیگنال های ورودی با استفاده از توابع منطقی تولید می‌شوند.

توابعی که CLA با استفاده از آنها ارقام نقلی را پیش‌بینی می‌کند، توابع Generate(G) و Propagate(P) هستند که به این صورت تعریف می‌شوند:

$$P_i = a_i \oplus b_i$$
$$G_i = a_i.b_i$$

که اگر شکل مدار جمع‌کننده کامل را با این توابع مقایسه کنیم، نتیجه می‌گیریم می‌توانیم S_i ، C_{i+1} را از این طریق بدست آوریم:

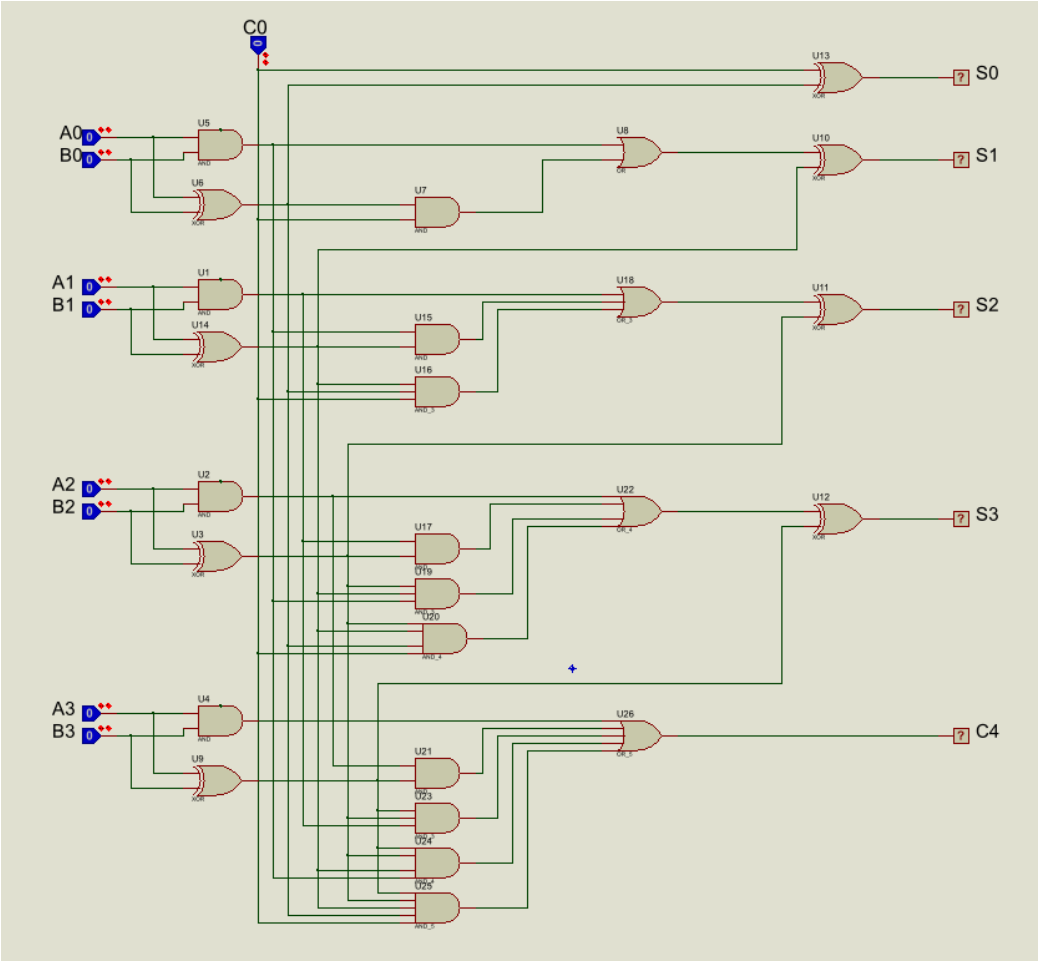
$$S_i = P_i \oplus C_i$$
$$C_{i+1} = G_i + P_i.C_i$$

برای مثال، C_1 و C_2 را با استفاده از رابطه بازگشتی بالا بدست می‌آوریم:

$$C_1 = G_0 + P_0.C_0$$
$$C_2 = G_1 + P_1.C_1 = G_1 + P_1(G_0 + P_0.C_0) = G_1 + G_0.P_1 + P_1.P_0.C_0$$

همانگونه که مشاهده می‌شود، به ازای تعداد بیت های بیشتر، تعداد گیت‌های استفاده شده و تعداد ورودی های آنها و در نتیجه پیچیدگی مدار بیشتر می‌شود. در واقع، بهای افزایش سرعت در CLA افزایش پیچیددگی(مساحت) مدار است.

[1] Propagation Delay



شکل ۱.۱.۴: مدار جمع‌کننده چهاربیتی Carry-Look-Ahead

۲.۴ تست مدار

شکل ۱.۲.۴:

$A = (1101)_2, B = (0010)_2, c_0 = 1$

binary: $(1101)_2 + (0010)_2 + 1 = (10000)_2$

decimal: $13 + 2 + 1 = 16$

شکل ۲.۲.۴:

$A = (0001)_2, B = (0001)_2$

binary: $(0001)_2 + (0001)_2 = (00010)_2$

decimal: $1 + 1 = 2$

شکل ۳.۲.۴:

$A = (0111)_2, B = (1000)_2$

binary: $(0111)_2 + (1000)_2 = (01111)_2$

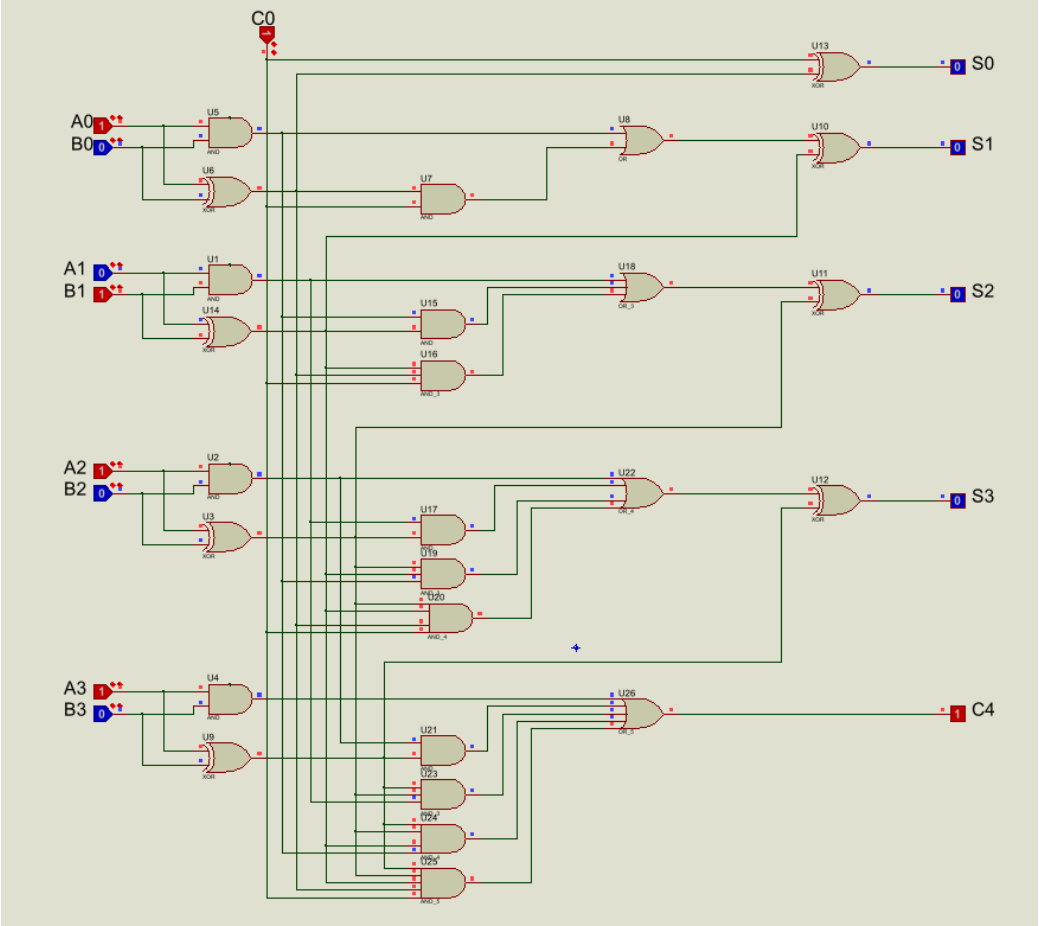
decimal: $7 + 8 = 15$

شکل ۴.۲.۴:

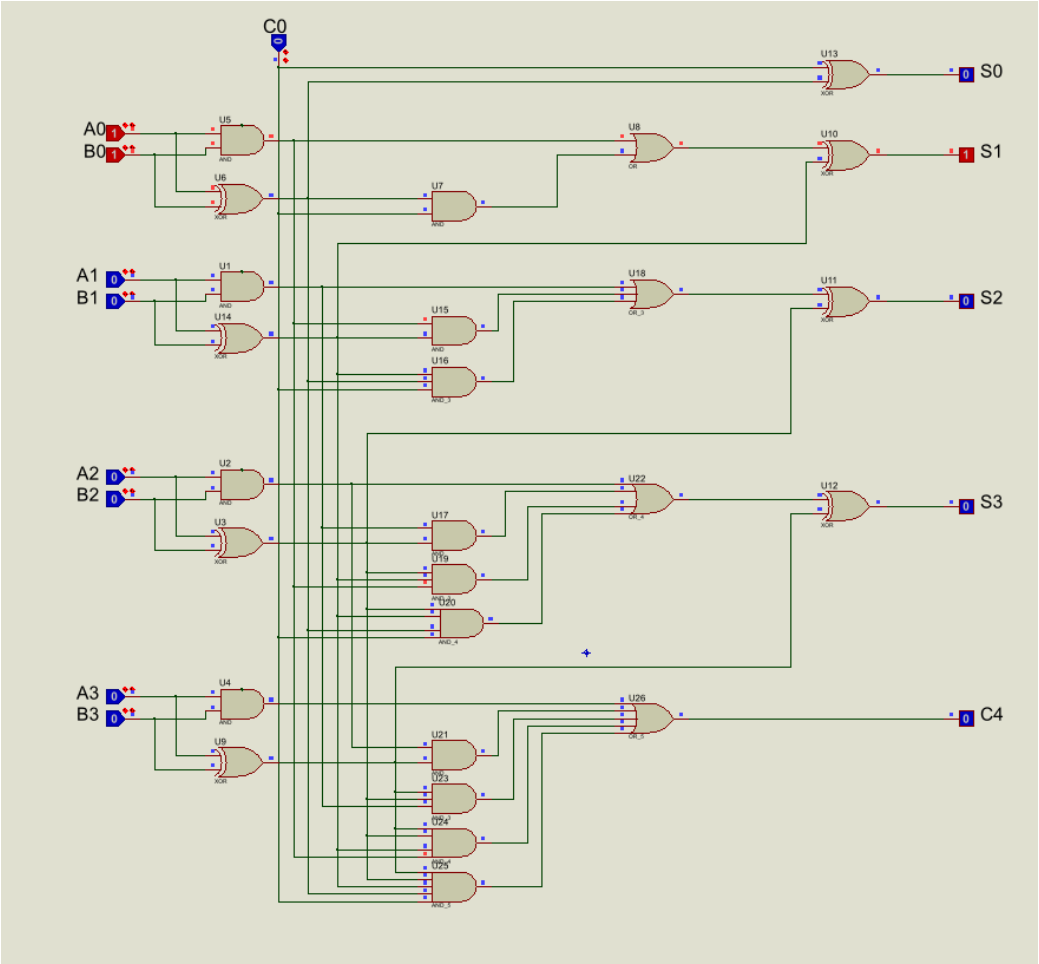
$A = (1111)_2, B = (1000)_2$

binary: $(1111)_2 + (1000)_2 = (10111)_2$

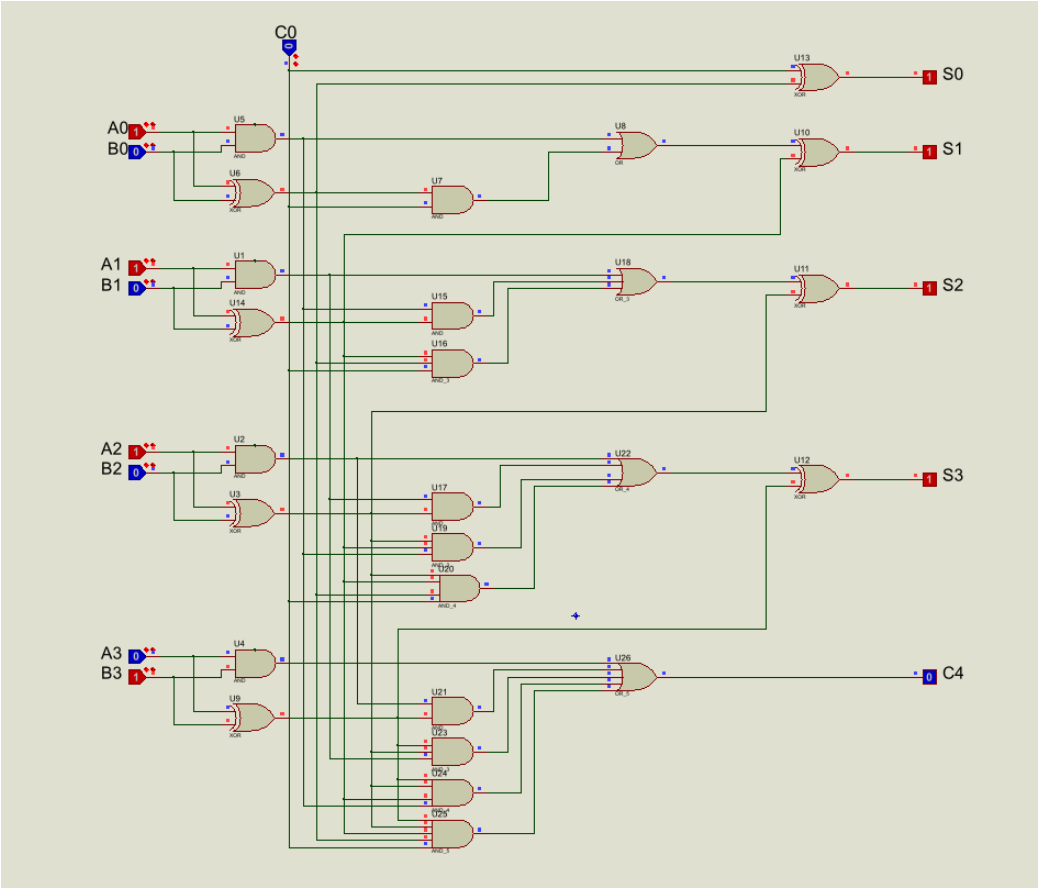
decimal: $15 + 8 = 23$



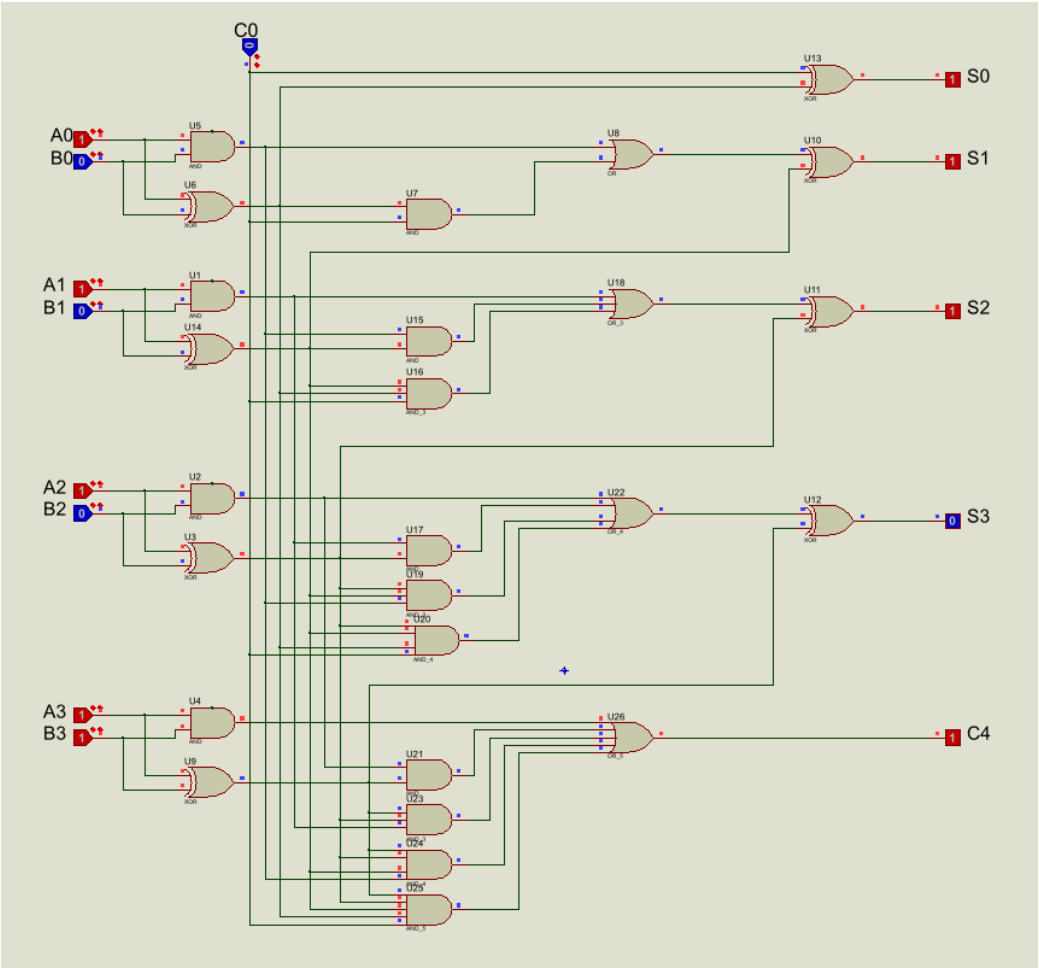
شکل ۱.۲.۴



شکل ۲.۲.۴



شکل ۳.۲.۴



شکل ۴.۲.۴

۵ نتیجه‌گیری

در این آزمایش، با محیط‌ها و ابزارهای مختلف شبیه‌سازی مدارهای منطقی از جمله Fritzing، Logisim، و Proteus آشنا شدیم. ابتدا با بررسی نحوه عملکرد بردبورد و پیاده‌سازی مدارهای ساده، نحوه قرار دادن قطعات مختلف و تراشه ها را آموختیم و درک درستی از اتصالات فیزیکی و پایه ها بدست آوردیم. سپس در نرم‌افزار Logisim، توانستیم مدار جمع‌کننده/تفریق‌کننده چهاربیتی را طراحی و تست کنیم و عمکرد آنرا با محاسبات تئوری تطبیق دهیم. در نهایت با پیاده‌سازی مدار جمع‌کننده از نوع Carry-Look-Ahead با محیط نرم‌افزار به نسبت قدرتمندتر Proteus آشنا شدیم. تمام اهداف این آزمایش به درستی محقق شد و نتایج شبیه‌سازی ها با محاسبات مطابقت داشت.