

بسمه تعالی



گزارش کار پنجم آزمایشگاه مدارهای منطقی

واحد محاسبات و منطق (ALU)

استاد:

دکتر شاهین حسابی

نویسنده:

امیررضا آذری

۹۹۱۰۱۰۸۷

دانشگاه صنعتی شریف

تابستان ۱۴۰۱

## فهرست

هدف .....	۳
بخش ۶.۱ - آشنایی با تراشه ۷۴۱۸۱ .....	۳
تئوری آزمایش: .....	۴
گزارش کار آزمایش: .....	۵
بخش ۶.۲ - ساخت مدار داخلی ALU .....	۱۷
تئوری آزمایش: .....	۱۷
گزارش کار آزمایش: .....	۱۸
نتیجه گیری: .....	۲۸
منابع و مراجع: .....	۲۹

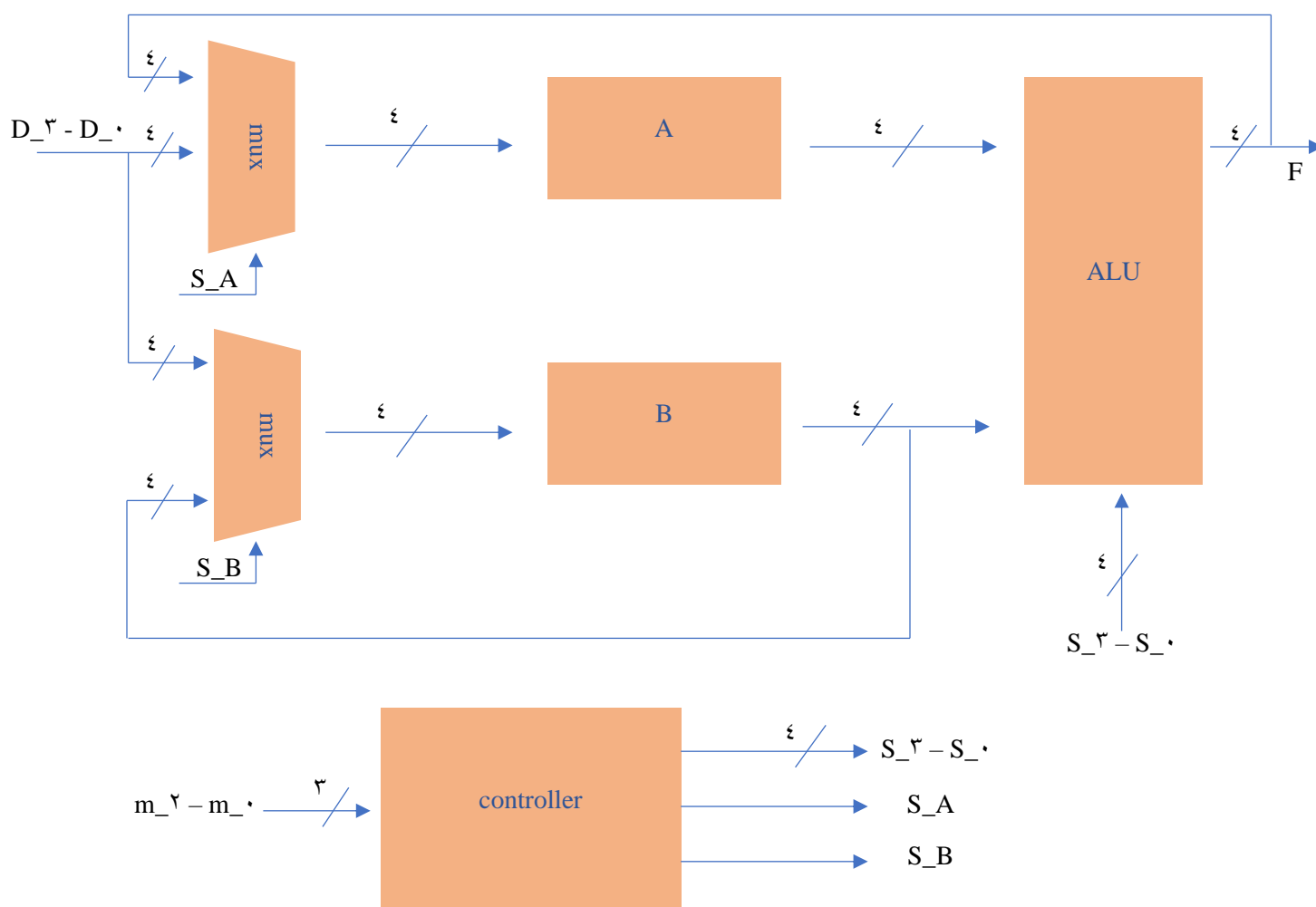
## هدف

هدف از این آزمایش، آشنایی با واحد محاسبات و منطق (ALU) است. در بخش اول با تراشه ۷۴۱۸۱ آشنا می‌شویم و در بخش دوم مدار داخلی ALU را می‌سازیم.

تمام بخش‌های این آزمایش را با نرم‌افزار Proteus انجام می‌دهیم.

## بخش ۶.۱ - آشنایی با تراشه ۷۴۱۸۱

مداری طراحی کنید که طبق شکل ۱، دارای دو ثبات داده A و B، یک ALU و یک کنترل‌کننده باشد، به طوری که با دادن کدهای مختلف به ALU، اعمال مختلف بر روی ورودی‌ها انجام شود.



شکل ۱. مدار آزمایش

### سیگنال‌های ورودی

خطوط داده  $D^0 - D^3$

خطوط دستور  $M^0 - M^2$

یک کلید از نوع push-button برای بازگرداندن مدار به حالت اولیه (RESET)

یک کلید از نوع push-button برای ورودی clock.

### سیگنال‌های خروجی

این مدار سیگنال خروجی خاصی ندارد. برای بررسی کارکرد درست مدار باید محتویات ثبات‌های A و B و خروجی ALU قابل مشاهده باشد.

### تئوری آزمایش:

مدار باید طبق جدول ۱، با توجه ورودی‌های  $M^0 - M^2$  عملیات خاصی را انجام دهد. برای ساخت این مدار از تراشه‌های ۷۴۱۸۱ (ALU)، ۷۴۱۷۵ (ثبات‌ها) و ۷۴۱۵۷ (MUX) و تعداد کافی گیت‌های پایه استفاده کنیم.

$M^2$	$M^1$	$M^0$	operation
۰	۰	۰	$A \leftarrow D^0 - D^3$
۰	۰	۱	$B \leftarrow D^0 - D^3$
۰	۱	۰	$A \leftarrow A$
۰	۱	۱	$A \leftarrow B$
۱	۰	۰	clear (A)
۱	۰	۱	$A \leftarrow \text{not}(A)$
۱	۱	۰	$A \leftarrow \text{and}(A, B)$
۱	۱	۱	$A \leftarrow \text{add}(A, B)$

جدول ۱. عملیات صورت گرفته در مدار برحسب ورودی‌های  $M^0 - M^2$

## گزارش کار آزمایش:

با استفاده از دیتاشیت تراشه و با توجه به جدولی که در دستور کار آورده شده است، جدول درستی را تشکیل می‌دهیم:

$M^2$	$M^1$	$M^0$	$S_B$	$S_A$	$S^3$	$S^2$	$S^1$	$S^0$	$M$	$C_n$
۰	۰	۰	۰	۱	x	x	x	x	x	x
۰	۰	۱	۱	۰	۱	۱	۱	۱	۱	x
۰	۱	۰	۰	۰	۱	۱	۱	۱	۱	x
۰	۱	۱	۰	۰	۱	۰	۱	۰	۱	x
۱	۰	۰	۰	۰	۰	۰	۱	۱	۱	x
۱	۰	۱	۰	۰	۰	۰	۰	۰	۱	x
۱	۱	۰	۰	۰	۱	۰	۱	۱	۱	x
۱	۱	۱	۰	۰	۱	۰	۰	۱	۰	۱

جدول ۲. جدول درستی

حال، بر اساس جدول فوق، معادله ورودی‌های کنترلی تراشه و mux ها را تعیین می‌کنیم:

$$S^3 = M^1 + M^2$$

$$S^2 = M^2 M^0 + M^2 M^1$$

$$S^1 = M^2 + M^0$$

$$S^0 = M^0 + M^2 M^1 + M^2 M^1$$

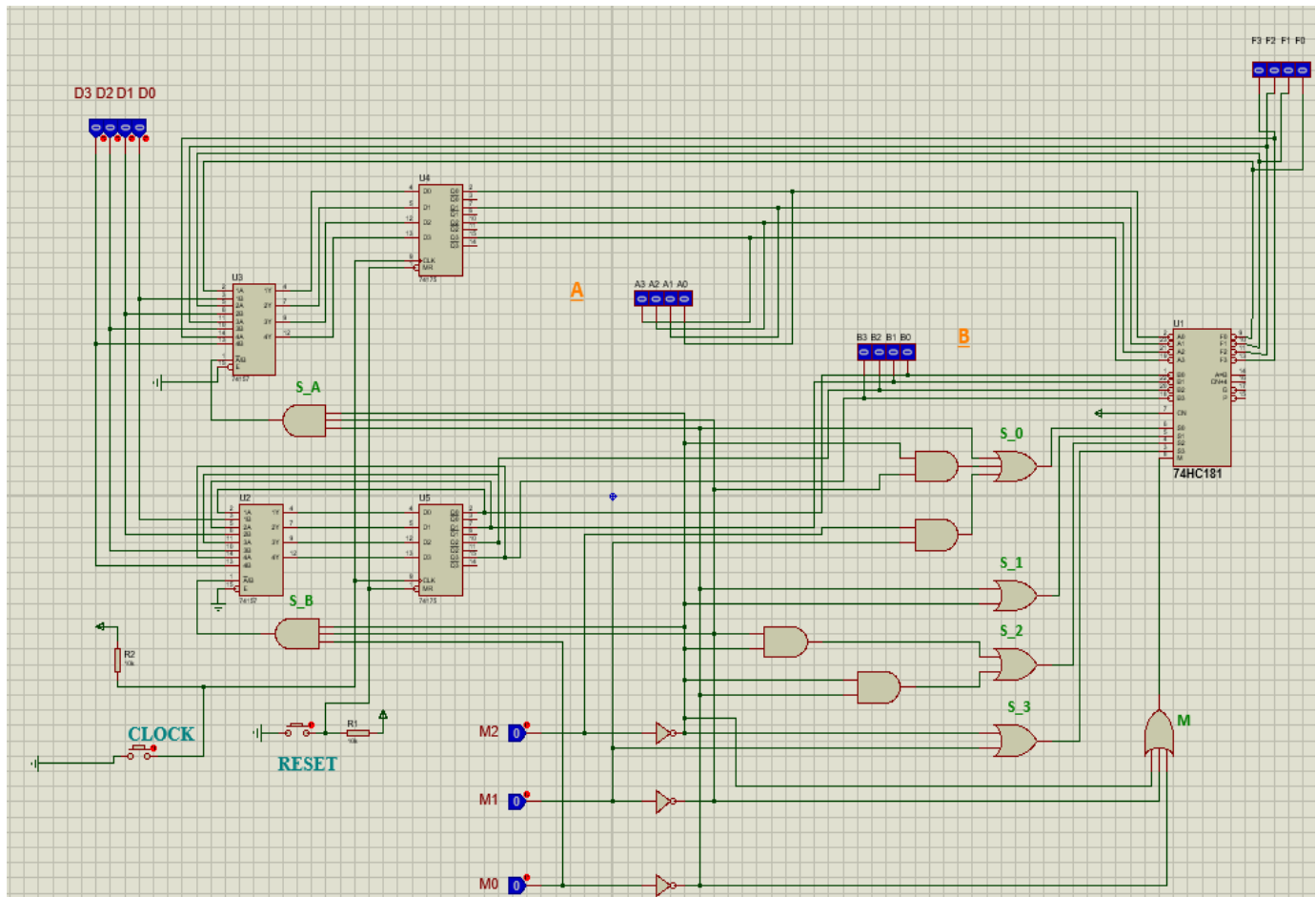
$$C_n = 1$$

$$M = M^2 + M^1 + M^0$$

$$SA = M^2 M^1 M^0$$

$$SB = M^2 M^1 M^0$$

اکنون بر اساس شکل ۱ و جداول ۱ و ۲، مدار را طراحی می‌کنیم:



شکل ۲. مدار طراحی شده

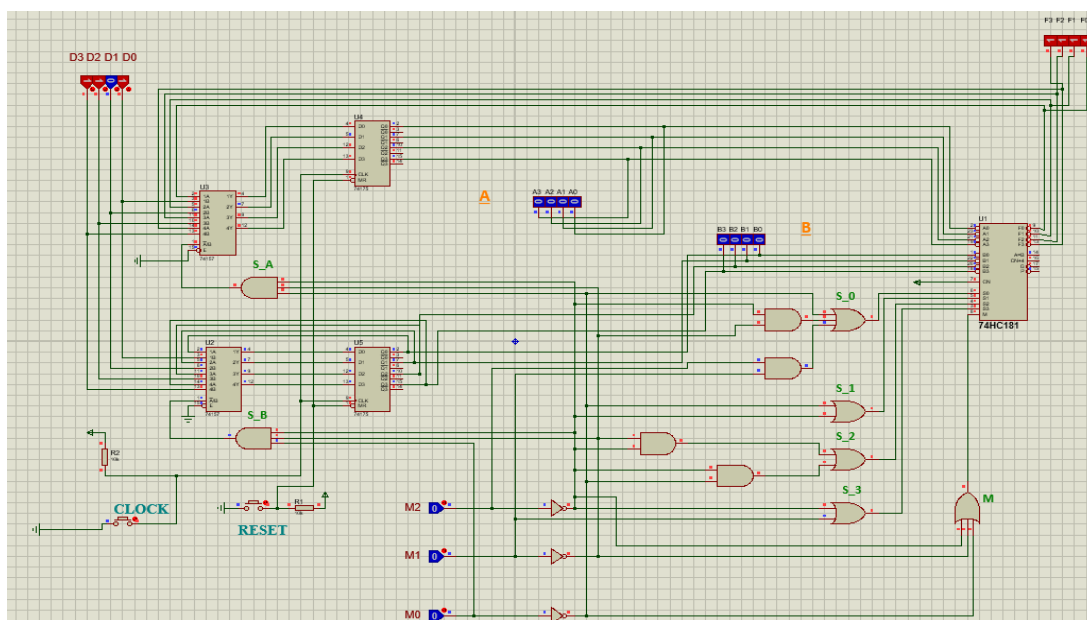
گیت‌های پایه استفاده شده در مدار طراحی شده، بر اساس روابط بدست آمده در صفحه قبل می‌باشد.

روابط میان ورودی خروجی‌های مولتی‌پلکسر‌ها و فلیپ‌فلاپ‌ها نیز به کمک شکل ۱ می‌باشد.

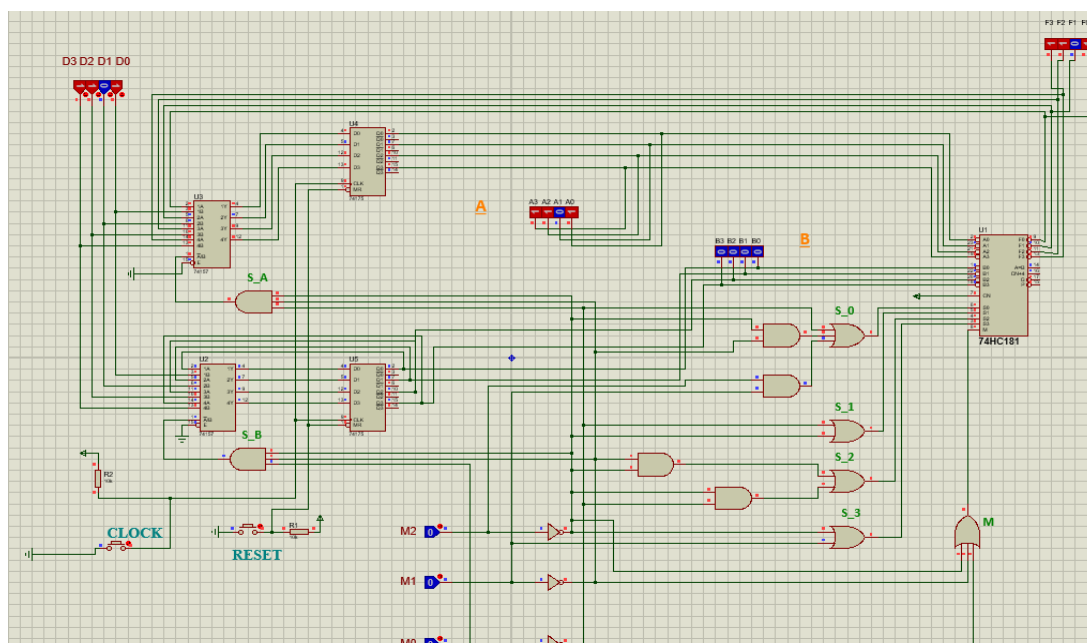
محتویات ثبات‌های A و B و خروجی ALU نیز طبق خواسته آزمایش قابل مشاهده است و درستی کارکرد مدار را در ادامه با کمک آن‌ها نشان می‌دهیم.

دستور :+++

این دستور برای load کردن A استفاده می‌شود.



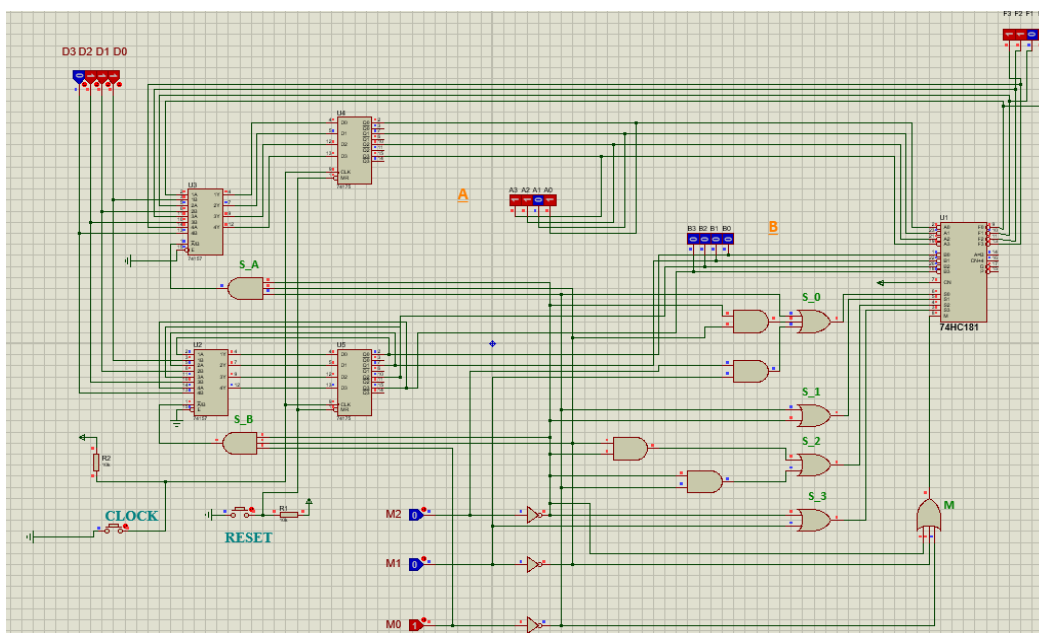
شکل ۳. ورودی‌های  $D_3-D_0$  داده شده‌اند (عدد ۱۳)، اما کلاک هنوز زده نشده است.



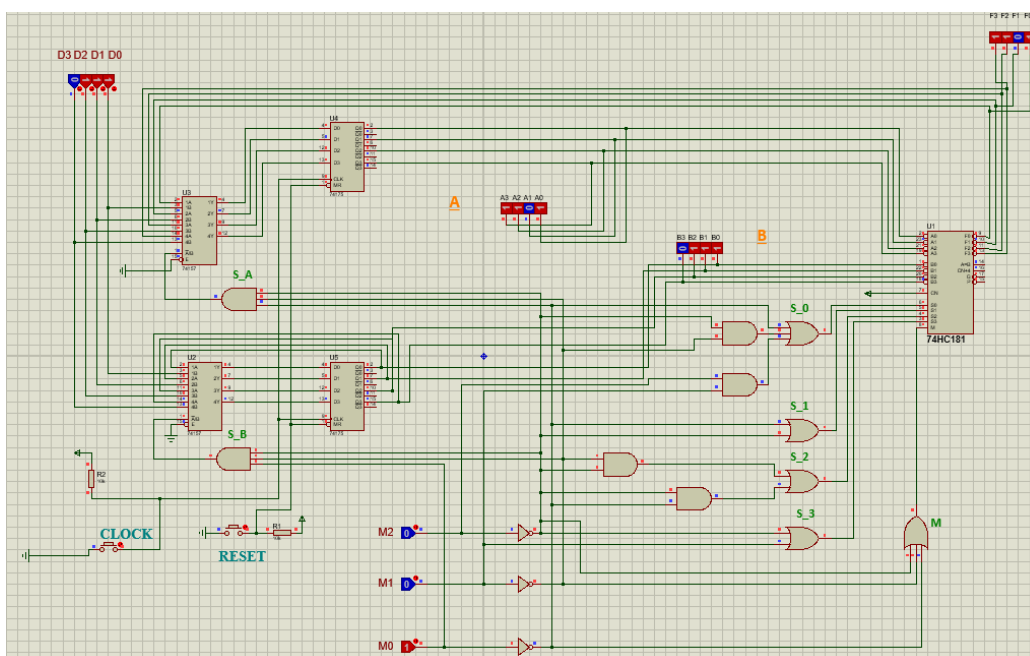
شکل ۴. بعد از کلاک، مقدار A بارگذاری می‌شود (خروجی ALU نیز قابل مشاهده است).

## دستور ۰۰۱ :

این دستور برای load کردن B استفاده می‌شود.



شکل ۵. ورودی‌های  $D_3-D_0$  داده شده‌اند (عدد ۷)، اما کلاک هنوز زده نشده است.

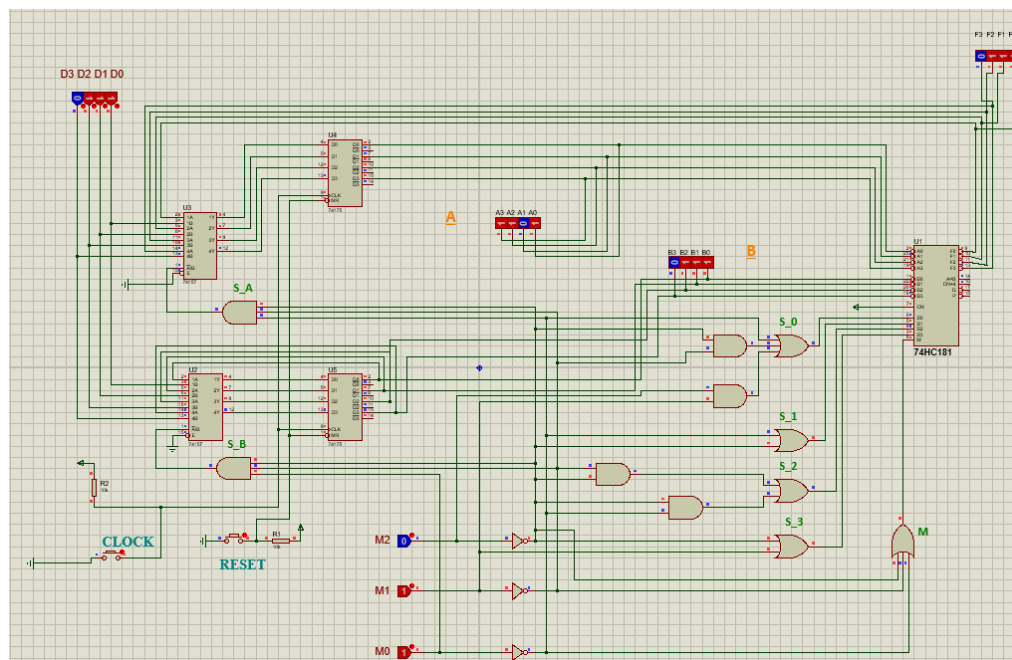


شکل ۶. بعد از کلاک، مقدار B بارگذاری می‌شود (خروجی ALU نیز قابل مشاهده است).

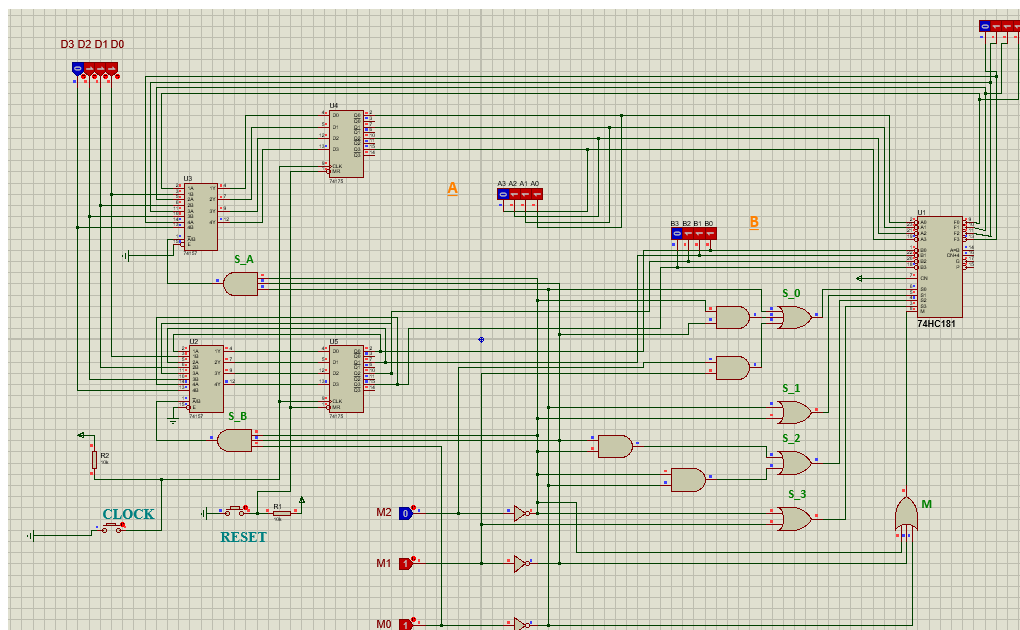


## دستور ۰۱۱ :

این دستور برای انتقال B به A استفاده می‌شود.



شکل ۷. کلاک هنوز زده نشده است.

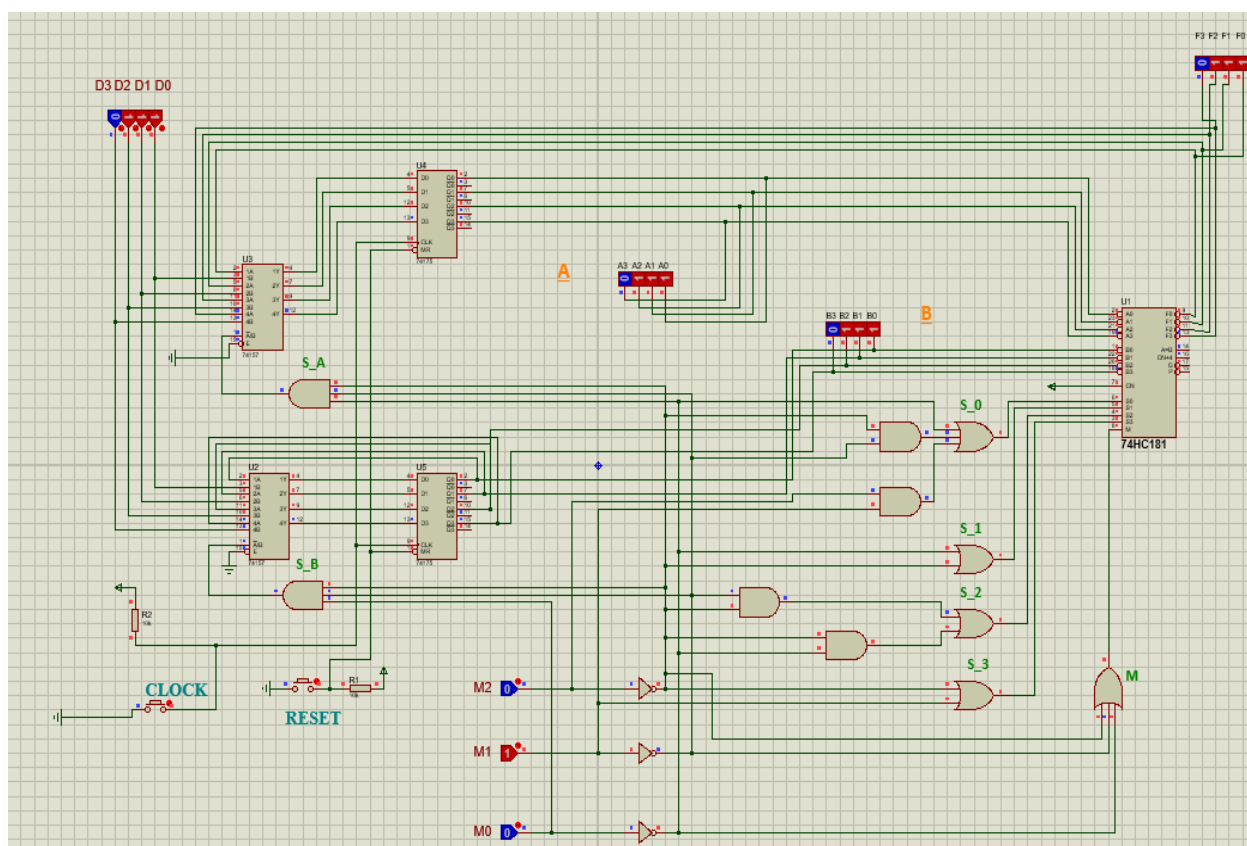


شکل ۸. بعد از کلاک، مقدار B در A بارگذاری می‌شود (خروجی ALU نیز قابل مشاهده است).

## دستور ۰۱۰:

این دستور برای انتقال A به A استفاده می‌شود.

مقدار کنونی A برابر ۷ است، و بعد از زدن کلاک، طبیعتاً تغییری نمی‌کند.

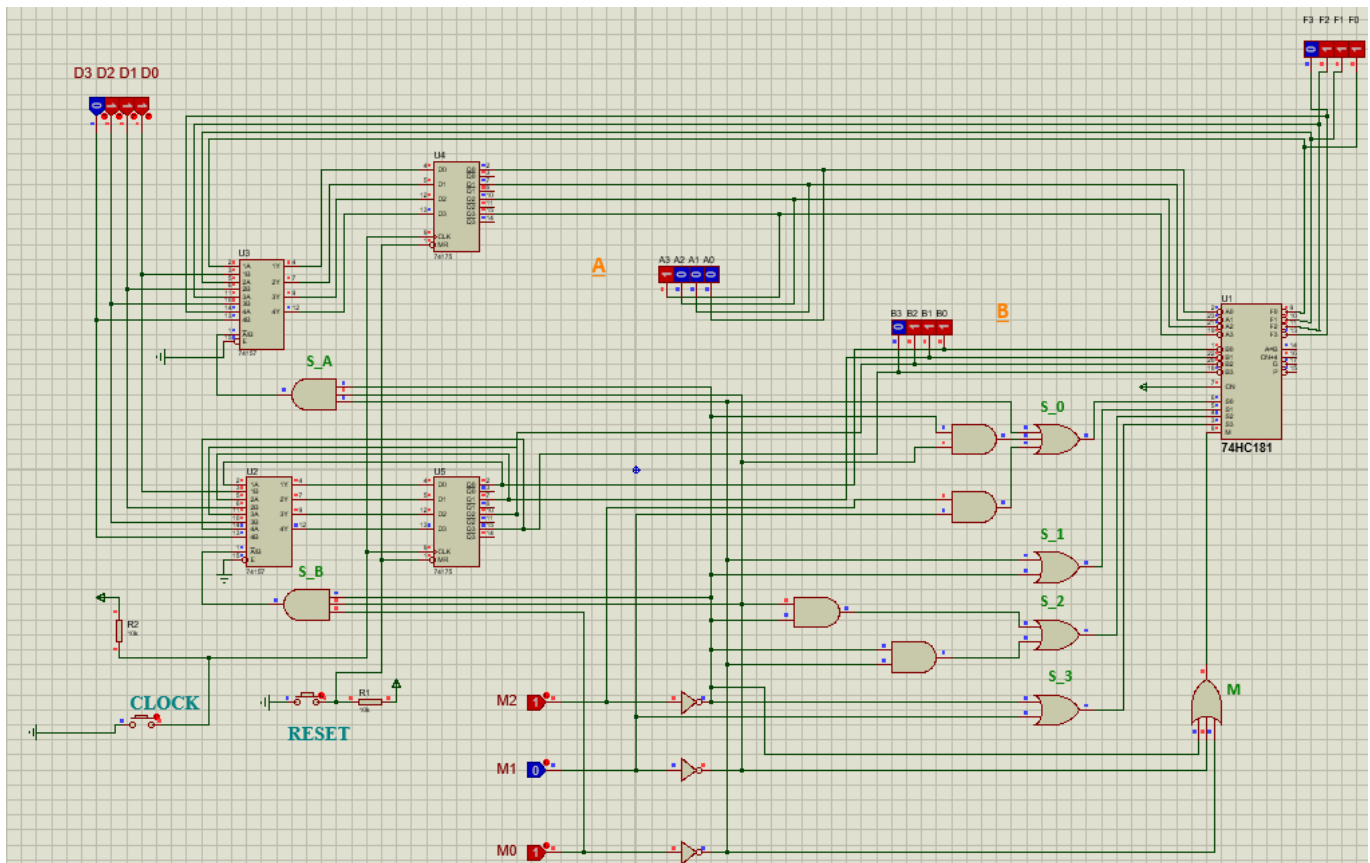


شکل ۹. بعد از کلاک، مقدار A در A بارگذاری می‌شود (خروجی ALU نیز قابل مشاهده است).

## دستور ۱۰۱:

این دستور برای NOT کردن A استفاده می‌شود.

A هم‌اکنون برابر ۰۱۱۱ می‌باشد و بعد از کلاک، باید به ۱۰۰۰ تغییر پیدا کند.

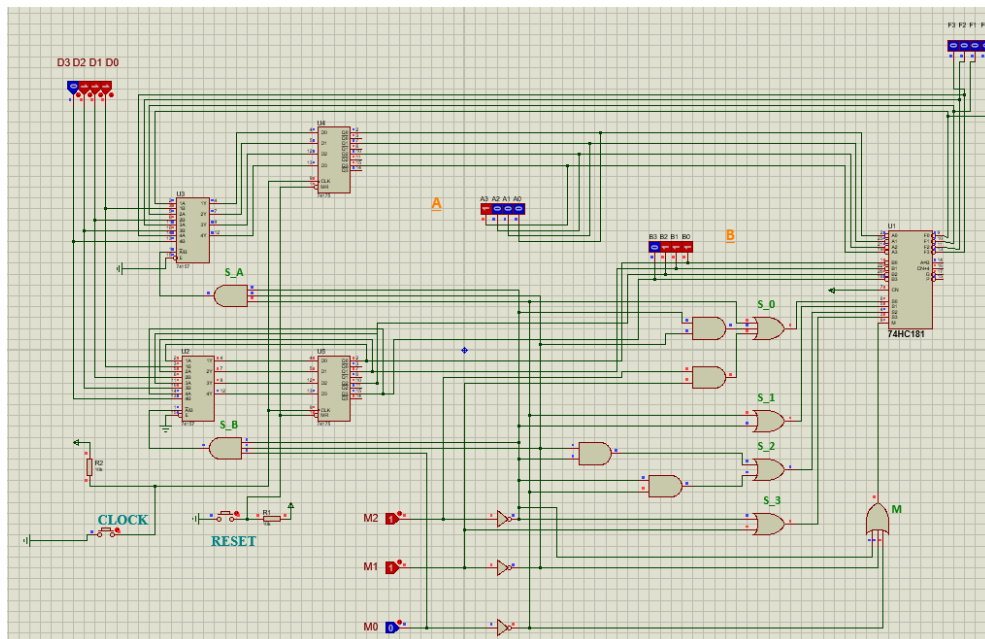


شکل ۱۰. بعد از کلاک، مقدار  $NOT(A)$  در A بارگذاری می‌شود (خروجی ALU نیز قابل مشاهده است).

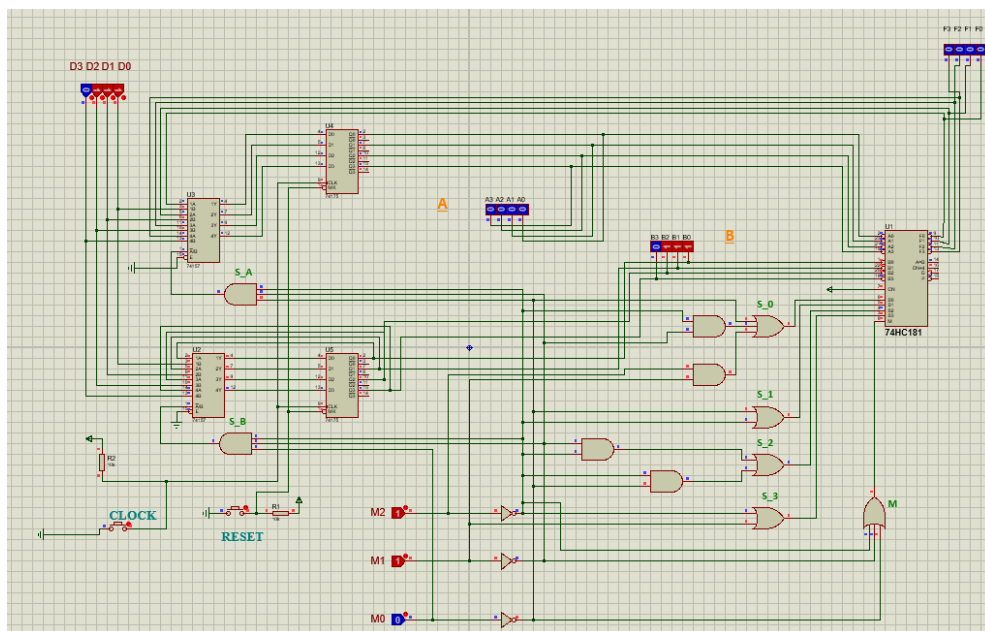
## دستور ۱۱۰:

این دستور  $AND(A, B)$  را در A قرار می‌دهد.

A برابر ۱۰۰۰ و B برابر ۰۱۱۱ می‌باشد، و AND آن‌ها برابر ۰۰۰۰ می‌شود.



شکل ۱۱. قبل از کلاک

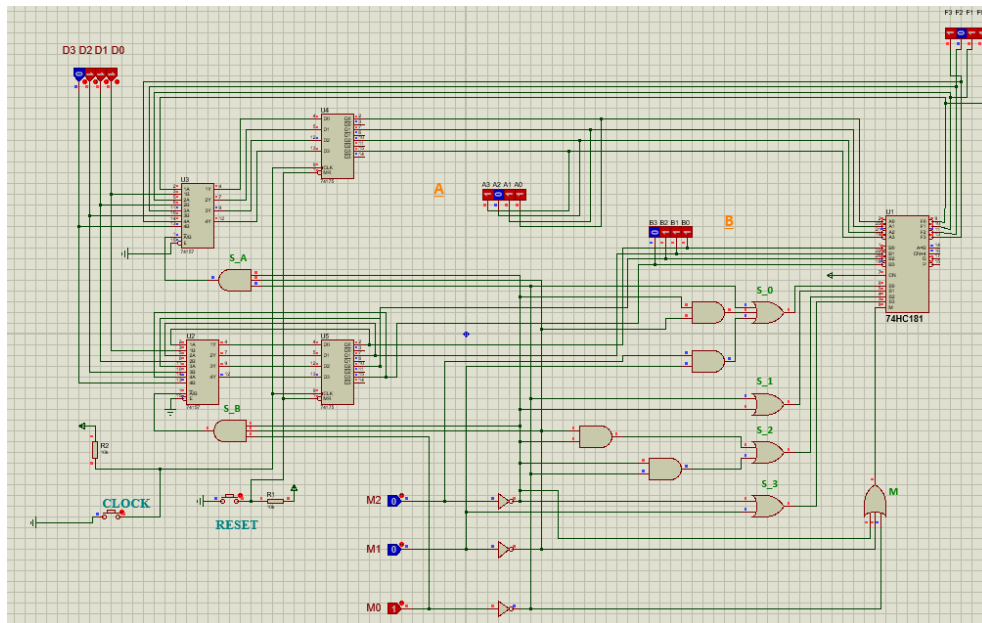


شکل ۱۲. بعد از کلاک، مقدار ۰۰۰۰ در A قرار می‌گیرد.

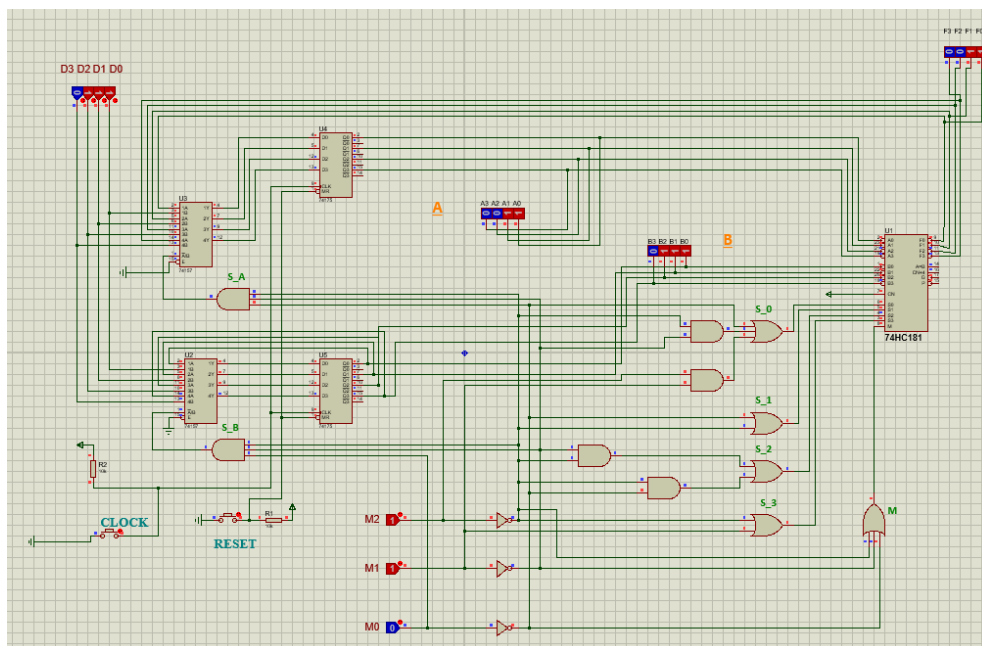
## مثال دیگری برای دستور ۱۱۰:

مقدار ۱۰۱۱ را در A و ۰۱۱۱ را در B بارگذاری می‌کنیم.

AND (A, B) برابر با ۰۰۱۱ خواهد بود.



شکل ۱۳. مقادیر ذکر شده را در A و B بارگذاری می‌کنیم.

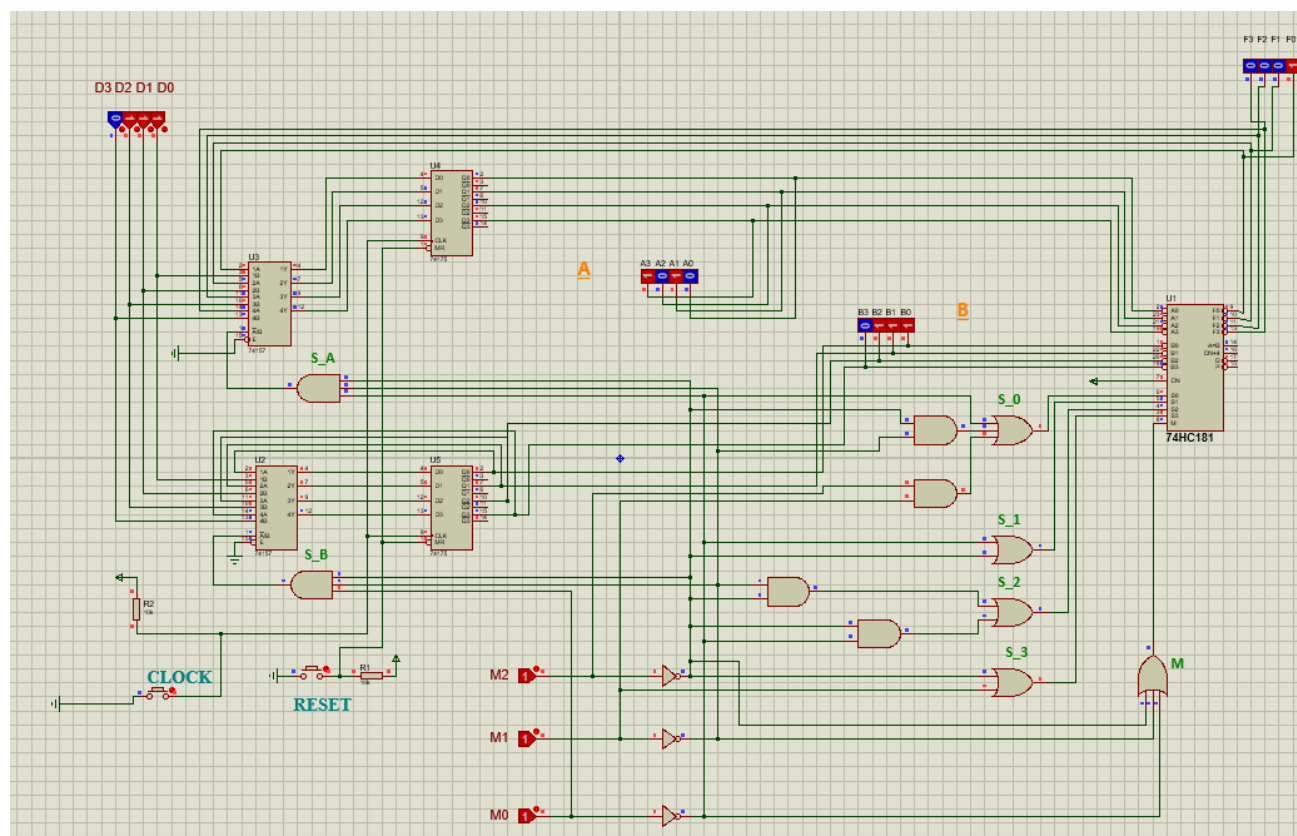


شکل ۱۴. بعد از زدن کلاک، مقدار ۰۰۱۱ در A قرار می‌گیرد.

## دستور ۱۱۱:

این دستور ADD (A, B) را در A بارگذاری می‌کند.

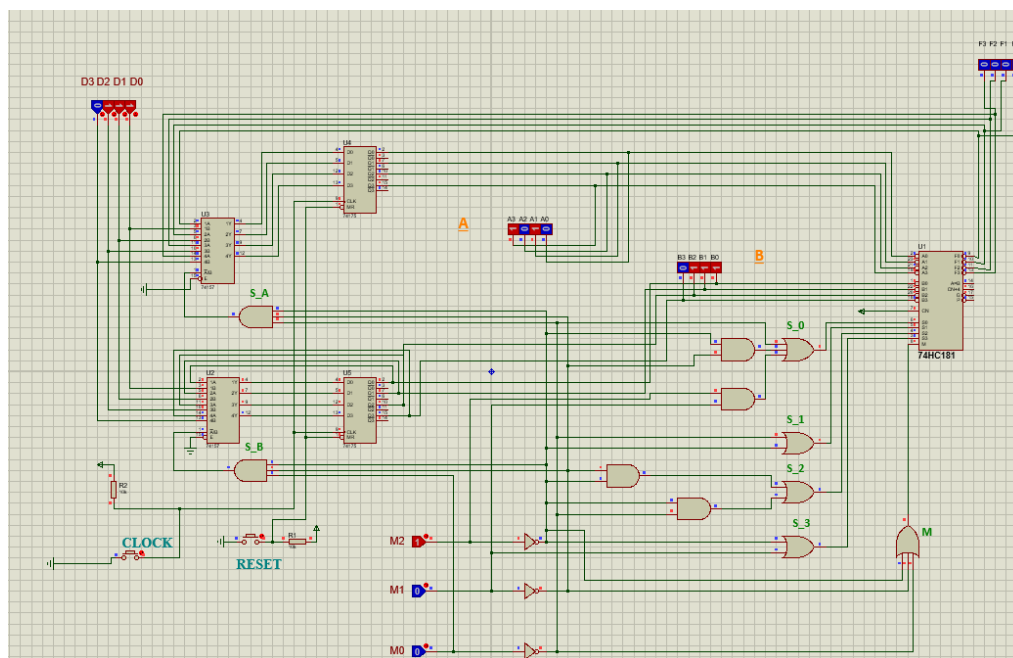
A برابر ۰۰۱۱ و B برابر ۰۱۱۱ است و مقدار  $ADD(A, B) = ۱۰۱۰$  در A قرار می‌گیرد.



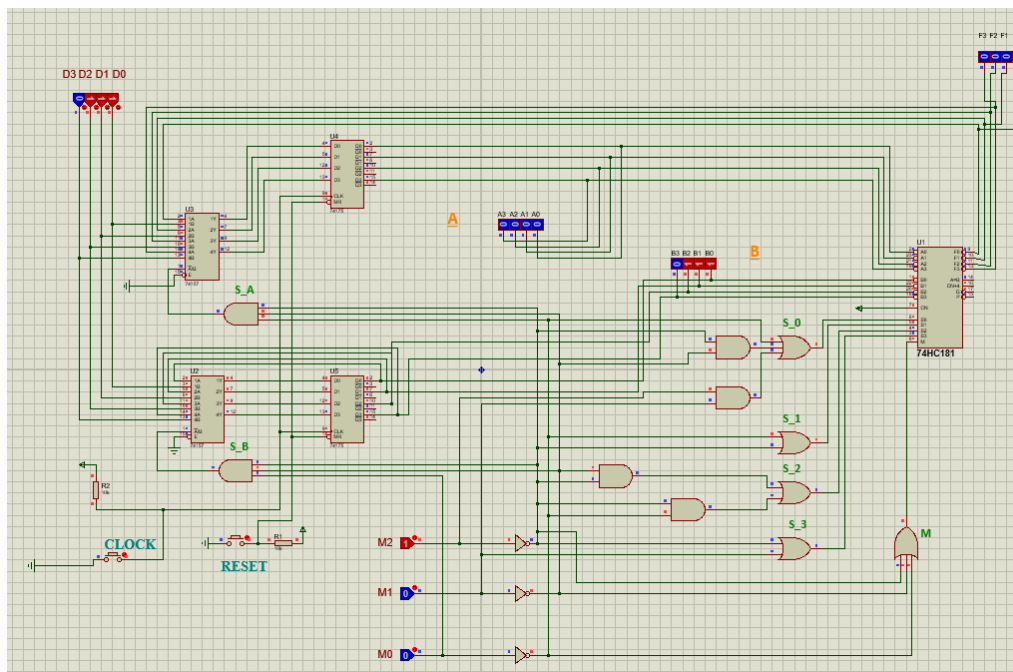
شکل ۱۵. بعد از زدن کلاک، مقدار ۱۰۱۰ در A قرار می‌گیرد.

## دستور ۱۰۰:

این دستور، مقدار A را clear کرده و A برابر ۰۰۰۰ می‌شود.



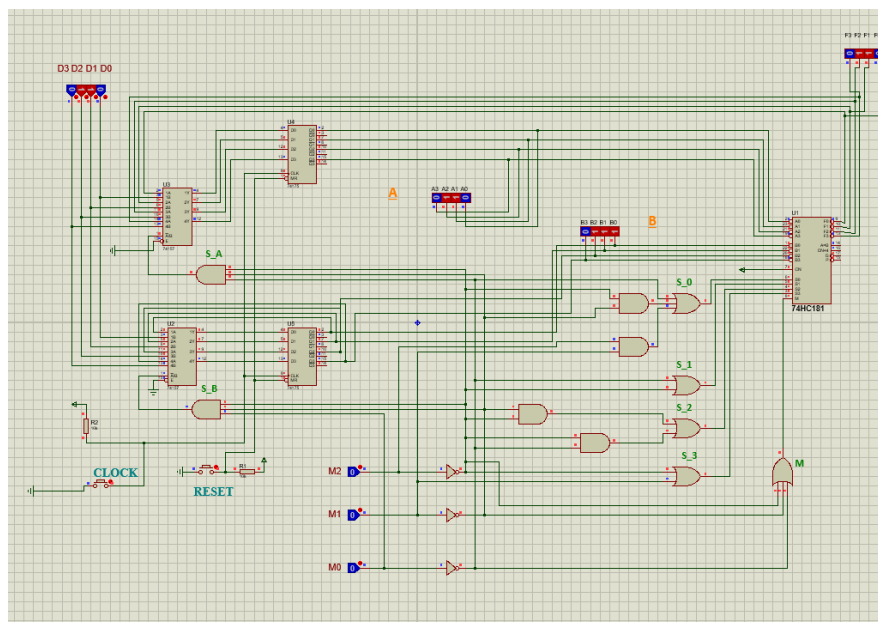
شکل ۱۶. مقدار A را بارگذاری کرده و آماده زدن کلاک می‌شویم.



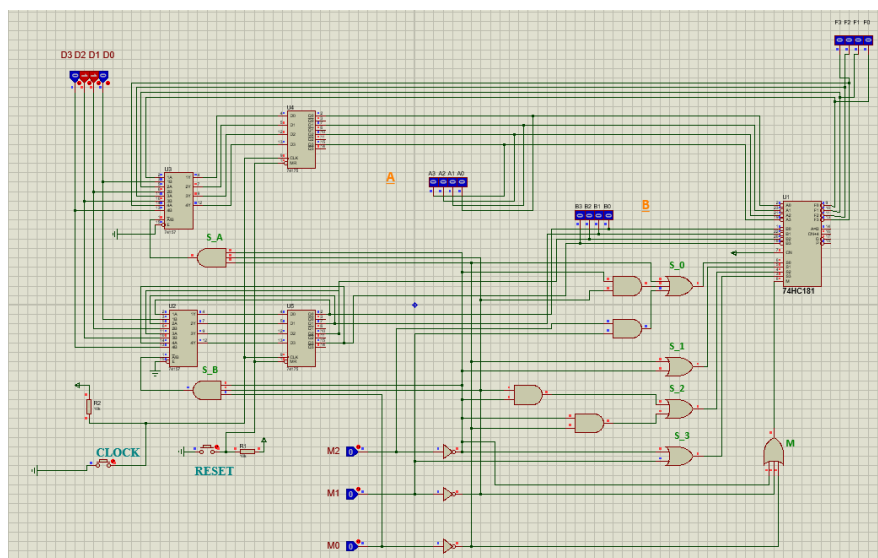
شکل ۱۷. بعد از زدن کلاک، A در clear(A) قرار می‌گیرد.

## کلید RESET :

با زدن این کلید، تمامی مقادیر A، B و خروجی ALU برابر ۰ خواهد شد.



شکل ۱۸. قبل از زدن کلاک، clear مقادیر قابل مشاهده می‌باشند.



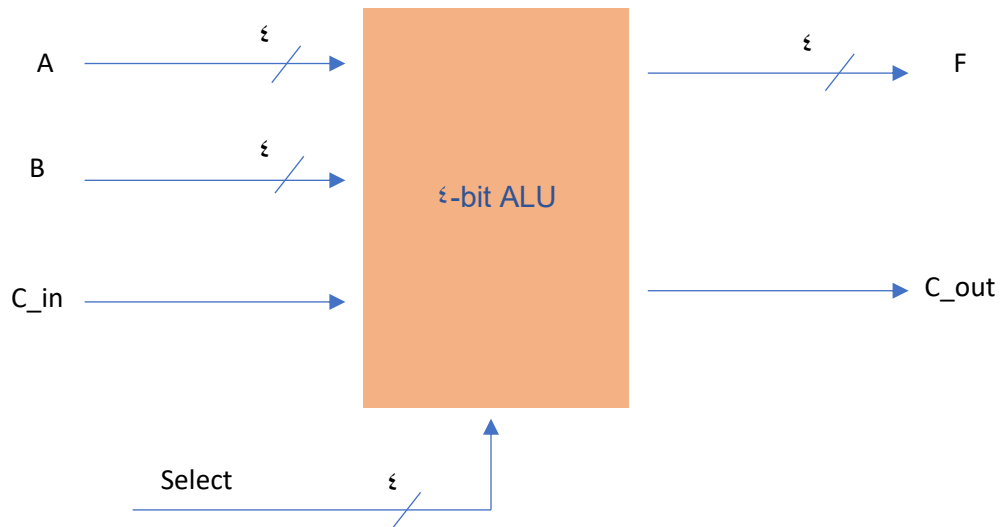
شکل ۱۹. بعد از زدن کلاک، تمامی مقادیر ۰ می‌شوند.



## بخش ۶.۲ \_ ساخت مدار داخلی ALU

یک واحد محاسبات و منطق چهاربیتی ( ۴-bit ALU ) بسازید.

تئوری آزمایش:



شکل ۲۰

$S^3$	$S^2$	$S^1$	$S^0$	$C_{in}$	Operation
۰	۰	۰	۰	۰	$F = A$
۰	۰	۰	۰	۱	$F = A + 1$
۰	۰	۰	۱	۰	$F = A + B$
۰	۰	۰	۱	۱	$F = A + B + 1$
۰	۰	۱	۰	۰	$F = A + B'$
۰	۰	۱	۰	۱	$F = A + B' + 1$
۰	۰	۱	۱	۰	$F = A - 1$
۰	۰	۱	۱	۱	$F = A$
۰	۱	۰	۰	X	$F = A \wedge B$
۰	۱	۰	۱	X	$F = A \vee B$
۰	۱	۱	۰	X	$F = A \oplus B$
۰	۱	۱	۱	X	$F = A'$
۱	۰	X	X	X	$F = SHR A$
۱	۱	X	X	X	$F = SHL A$

جدول ۳

طبق شکل ۲۰ و جدول ۳، یک واحد محاسبات و منطق ۴ بیتی می‌سازیم.

می‌دانیم دستورات به شکل  $S^3S^2S^1S^0$  می‌باشند. همانطور که مشخص می‌باشد، نیاز به حالت بندی دستورات از ۰۰۰۰ تا ۰۱۱۱ داریم و برای  $S^3 = ۱$ ، حالات شیفت را جداگانه در نظر می‌گیریم.

همچنین در مدار از تراشه‌های ۷۴۲۸۳ (Full adders with fast carry)، ۴۵۱۴ (Decoder with input latches)، ۷۴۱۹۵ (۴-bit parallel access register) استفاده می‌کنیم که در ادامه به توضیح می‌پردازیم.

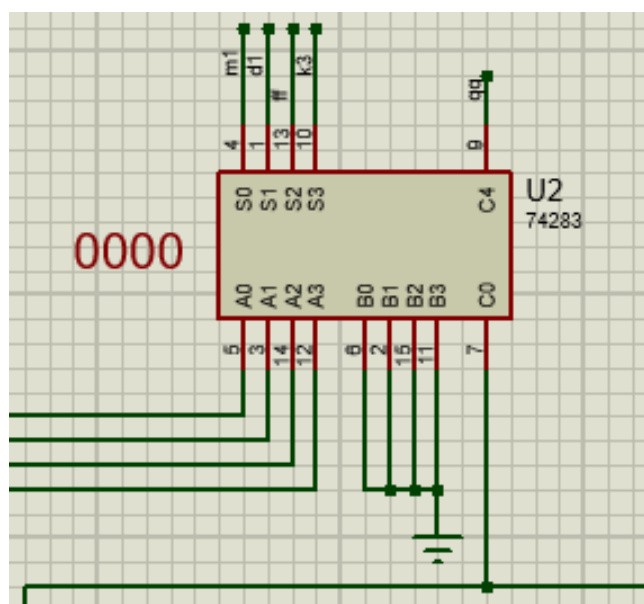
## گزارش کار آزمایش:

ابتدا نحوه انجام عملیات ها را توضیح می‌دهیم و سپس نحوه تشخیص و تعیین خروجی را بیان می‌کنیم:

نکته: به دلیل بزرگ شدن مدار و پرهیز از پیچیدگی های فراوان، بخشی از سیم ها وصل نشده اند و از قابلیت نام گذاری ورودی خروجی سیم ها استفاده کرده ایم. البته در آخر کار، شکل کامل مدار با سیم کشی کامل آورده شده است.

### حالت ۰۰۰۰:

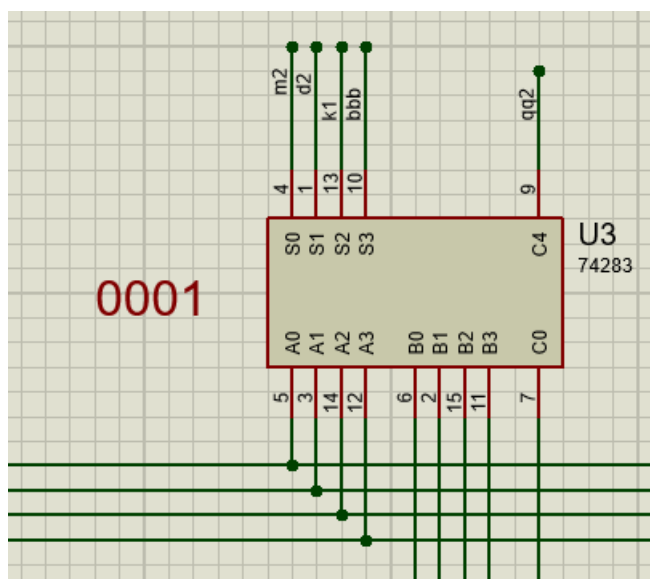
برای حالت ۰۰۰۰، با استفاده از یک adder (تراشه ۷۴۲۸۳)، مقدار  $A + ۰ + C_{in}$  را محاسبه می‌کنیم (ورودی های  $A^3-A^0$ ،  $C_{in}$  و ۰. این بخش از مدار را در شکل ۲۱ می‌توانید مشاهده کنید).



شکل ۲۱. تراشه ۷۴۲۸۳ برای حالت ۰۰۰۰

### حالت ۰۰۰۱:

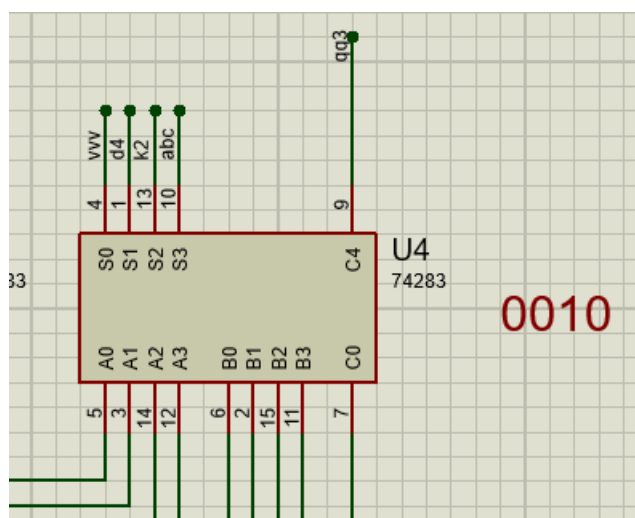
برای حالت ۰۰۰۱، با استفاده از یک adder (تراشه ۷۴۲۸۳)، مقدار  $A + B + C_{in}$  را محاسبه می‌کنیم (ورودی‌های  $A^3-A^0$ ،  $B^3-B^0$  و  $C_{in}$ ). این بخش از مدار را در شکل ۲۲ می‌توانید مشاهده کنید.



شکل ۲۲. تراشه ۷۴۲۸۳ برای حالت ۰۰۰۱

### حالت ۰۰۱۰:

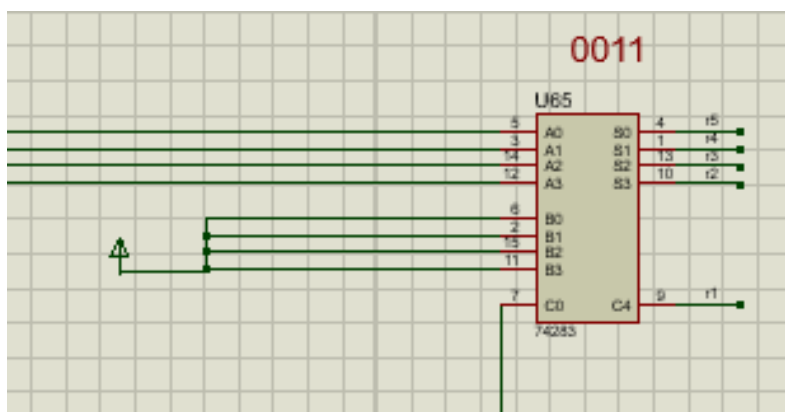
برای حالت ۰۰۱۰، با استفاده از یک adder (تراشه ۷۴۲۸۳)، مقدار  $A + B' + C_{in}$  را محاسبه می‌کنیم (ورودی‌های  $A^3-A^0$ ،  $B^3-B^0$  و  $C_{in}$ ). این بخش از مدار را در شکل ۲۳ می‌توانید مشاهده کنید.



شکل ۲۳. تراشه ۷۴۲۸۳ برای حالت ۰۰۱۰

### حالت ۰۰۱۱:

برای حالت ۰۰۱۱، با استفاده از یک adder (تراشه ۷۴۲۸۳)، مقدار  $A + 1 + C_{in}$  را محاسبه می‌کنیم (ورودی‌های  $A_3-A_0$ ،  $C_{in}$  و  $A - 1$  برابر  $A + (-1)$  است و منفی ۱ برابر ۱۱۱۱ در مکمل ۲ می‌باشد). این بخش از مدار را در شکل ۲۴ می‌توانید مشاهده کنید.

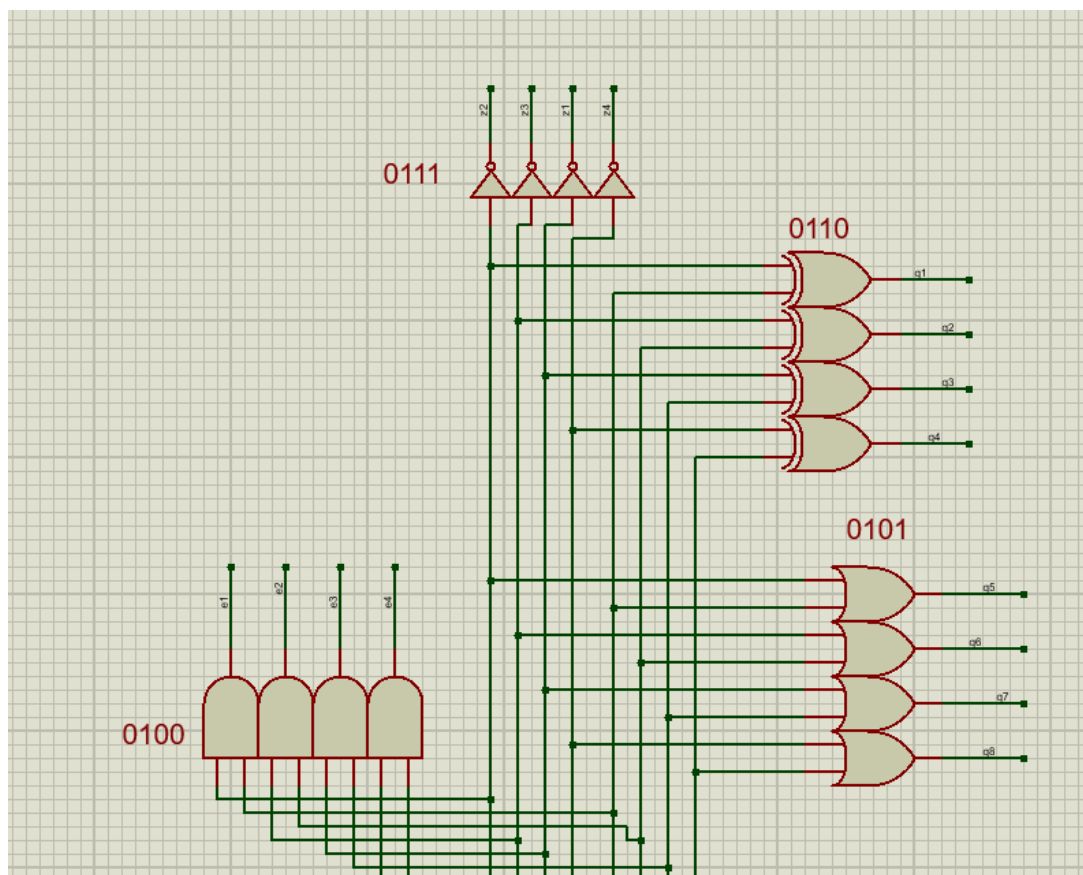


شکل ۲۴. تراشه ۷۴۲۸۳ برای حالت ۰۰۱۱

خروجی  $C_4$  تمامی این تراشه‌ها را برای  $C_{OUT}$  استفاده خواهیم کرد که بعد از معرفی بقیه حالات به آن خواهیم پرداخت.

### حالات ۰۱۰۰، ۰۱۰۱، ۰۱۱۰، ۰۱۱۱:

برای حالت ۰۱۰۰، ۰۱۰۱، ۰۱۱۰، ۰۱۱۱ نیز با استفاده از گیت‌های منطقی خروجی خواسته شده را به سادگی محاسبه می‌کنیم. تصویر صفحه بعد، قسمت مربوط به این بخش را نمایش می‌دهد.

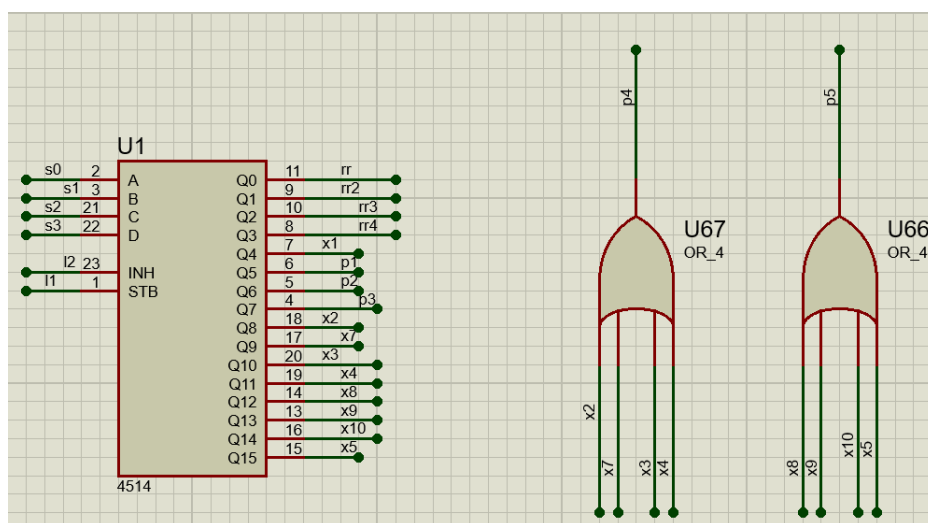


شکل ۲۵. حالات ۰۱۱۱، ۰۱۱۰، ۰۱۰۱، ۰۱۰۰

برای حالت مربوط به شیفت نیز، اگر شیفت چپ باشد، خروجی را به  $A_2A_1A_0$  وصل می‌کنیم و اگر شیفت به راست باشد آن را به  $A_3A_2A_1$  متصل می‌کنیم.

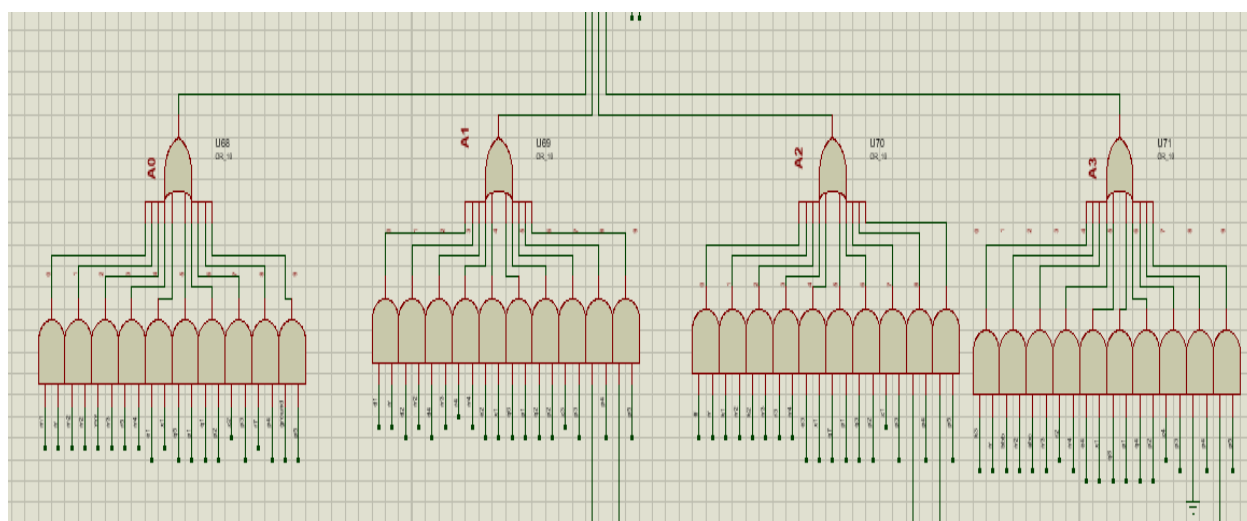
در توضیحات بالا نحوه محاسبه خواسته‌های مختلف آورده شد، حال برای آنکه خروجی متناسب با انتخاب کاربر داده شود، از یک دیکودر استفاده می‌کنیم و هر خروجی محاسبات بالا برای هر بیت را، با حالت متناظرش And می‌کنیم تا اگر حالت متناظر فعال بود خروجی مذکور از گیت And خارج شود و در غیر این صورت آن خروجی غیرفعال شود. سپس همه حالت را با هم OR می‌کنیم تا خروجی اصلی مشخص شود. در ادامه بیشتر به این بحث خواهیم پرداخت.

تصویر صفحه بعد دیکودر تعیین حالت را نشان می‌دهد:



شکل ۲۶

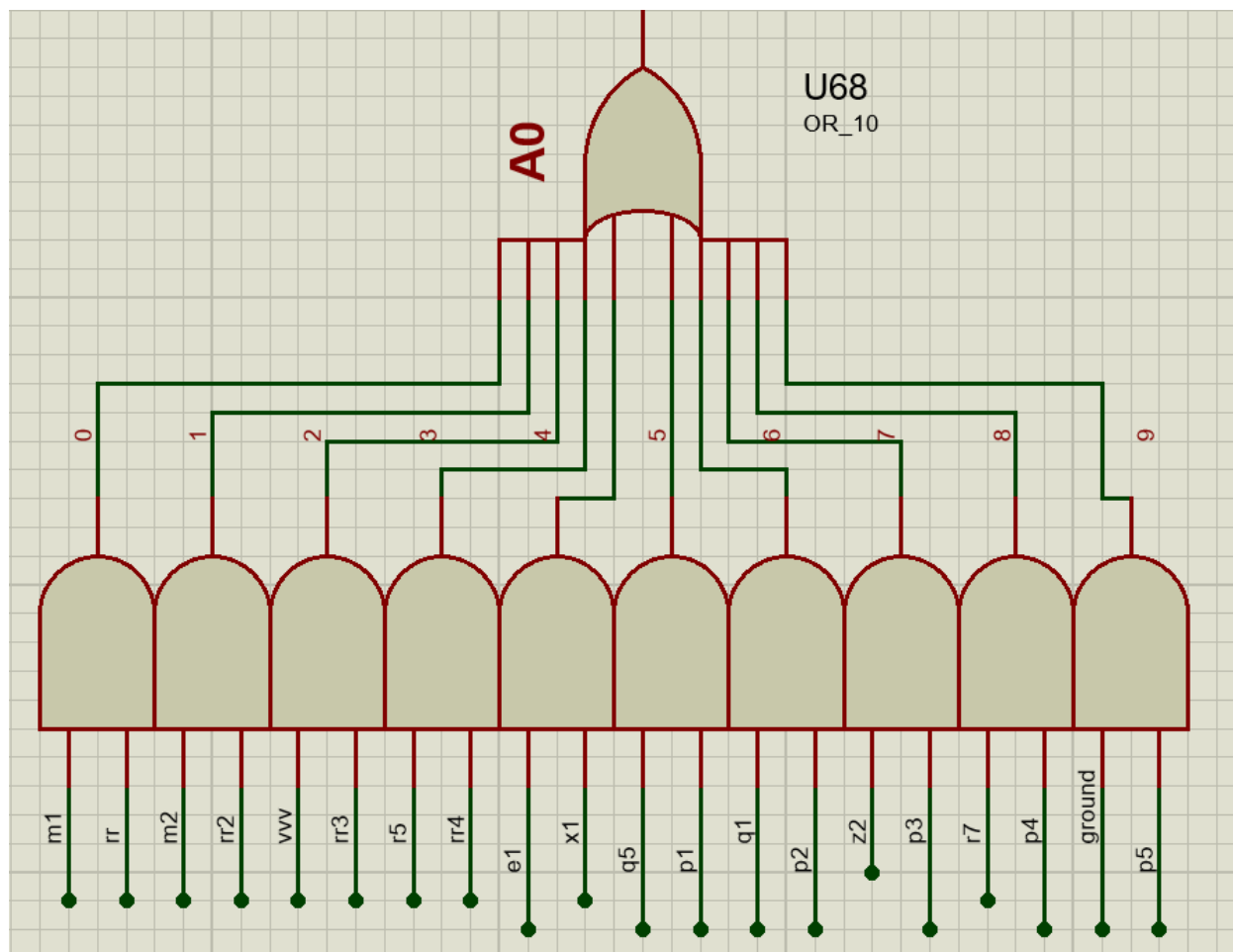
خروجی های این دیکودر برای تعیین حالات خروجی  $F_0-F_3$  به کار می رود که به توضیح آن خواهیم پرداخت.



شکل ۲۷

به دلیل عدم واضح بودن تصویر، حالت  $A_0$  را بررسی می کنیم.

حالات  $A_1-A_3$  نیز به همان ترتیب خواهند بود اما صرفاً برای بیت های بعدی به کار می روند.



شکل ۲۸. حالت  $A_0$

همانطور که در شکل ۲۸ واضح است، هر حالت ( $A_0 - A_3$ )، از ۱۰ گیت AND تشکیل شده است.

اگر گیت ها را از سمت چپ نام گذاری کنیم، خواهیم داشت:

۱ AND تا ۴ AND: ورودی ها برابر  $m_1, m_2, vv, r_5$  و  $rr, rr_2, rr_3, rr_4$  می باشند.

$m_1, m_2, vv, r_5$  خروجی های تراشه ۷۴۲۸۳ در حالات ۰۰۱۱، ۰۰۱۰، ۰۰۰۱، ۰۰۰۰ می باشند (به شکل های ۲۱ و ۲۲ و ۲۳ و ۲۴ مراجعه کنید).

$rr, rr_2, rr_3, rr_4$  خروجی های دیکودر ما می باشند (به شکل ۲۶ مراجعه کنید).

۵ AND تا ۸ AND: ورودی ها برابر  $e_1, q_5, q_1, z_2$  و  $x_1, p_1, p_2, p_3$  می باشند.

$e_1, q_5, q_1, z_2$  خروجی های گیت های ما در شکل ۲۵ می باشند (چون حالت  $A_0$  را بررسی می کنیم، خروجی اول گیت های شکل ۲۵ را ورودی داده ایم و طبیعتاً در حالات  $A_2 - A_3 - A_4$ ، به سراغ خروجی های دیگر رفته ایم).

$p_3, p_2, p_1, x_1$  خروجی‌های دیکودر ما می‌باشند (به شکل ۲۶ مراجعه کنید).

مشخص است که هر خروجی دیکودر، مربوط به دستور خاصی می‌باشد و ما طبق همین، آن‌ها را جفت جفت (به شکل متناظر) به گیت‌های AND داده‌ایم.

AND ۹ و AND ۱۰:

این دو برای حالات شیفت استفاده شده‌اند.

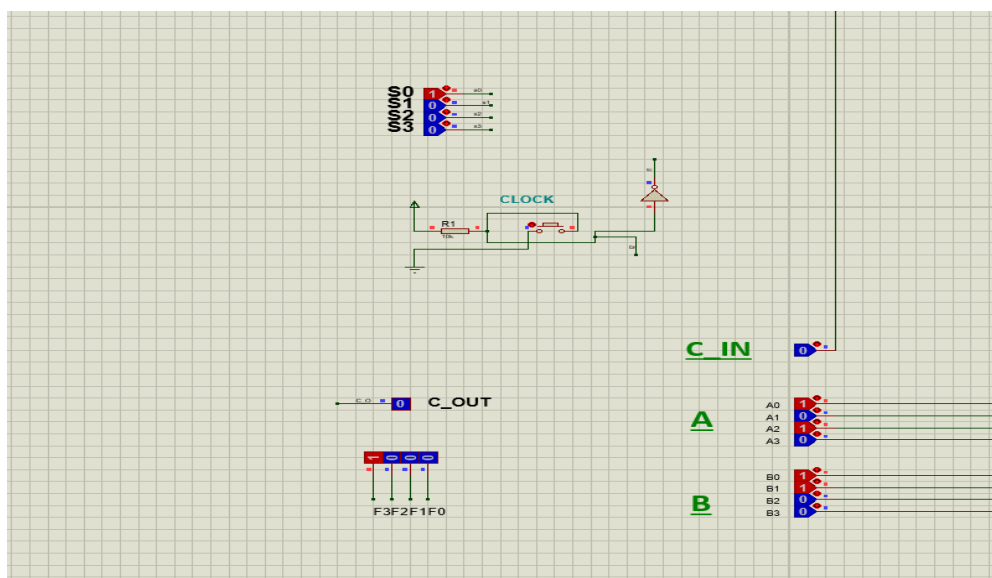
همانطور که در شکل ۲۶ مشخص است ۸ خروجی آخر، ۴ به ۴ باید OR شوند زیرا حالات ۱۵۲XX هست و با توجه به جدول ۳ این عمل صورت گرفته است.

حال به بررسی عملکرد در چند دستور می‌پردازیم:

### دستور ۰۰۰۱:

این دستور،  $F = A + B$  را خروجی می‌دهد.

برای مثال، A را ۵ (۰۱۰۱) و B را ۳ (۰۰۱۱) در نظر می‌گیریم.

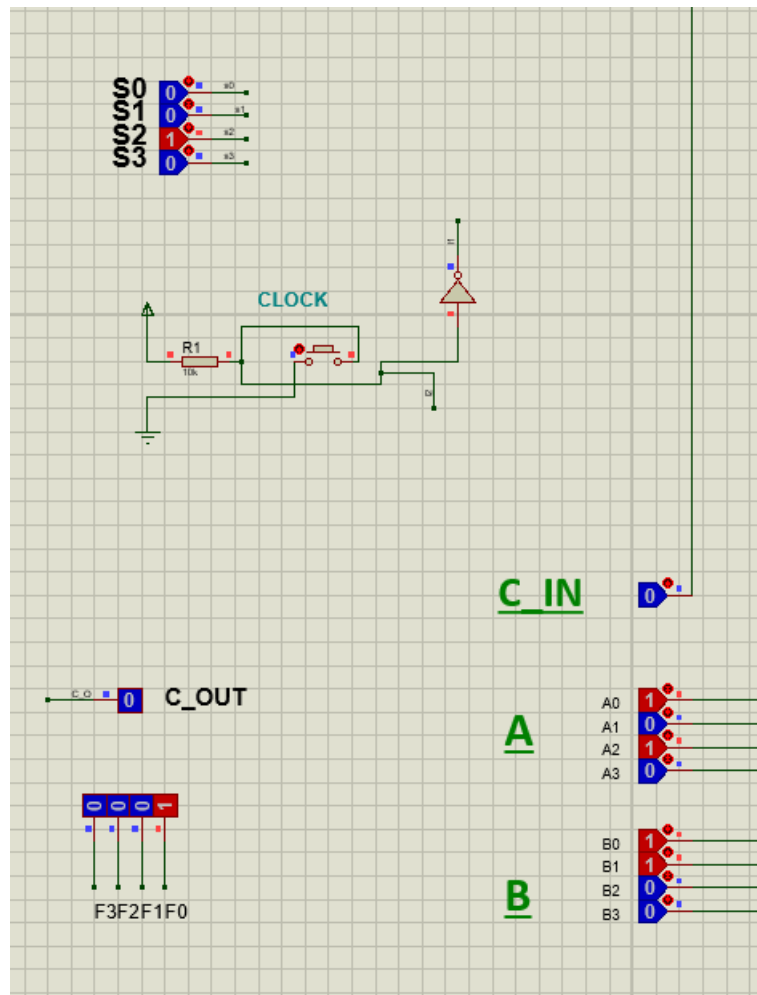


شکل ۲۹. دستور ۰۰۰۱



دستور ۱۰۰:

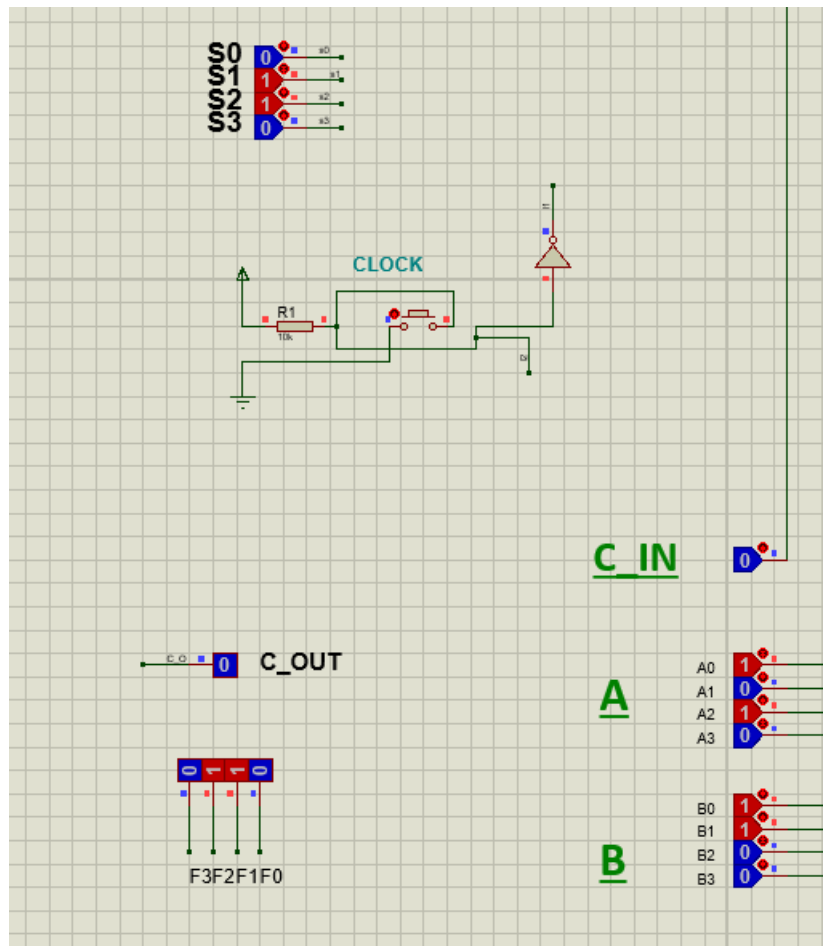
این دستور، A و B را AND می‌کند.



شکل ۳۰. دستور ۱۰۰.

## دستور ۰۱۱۰:

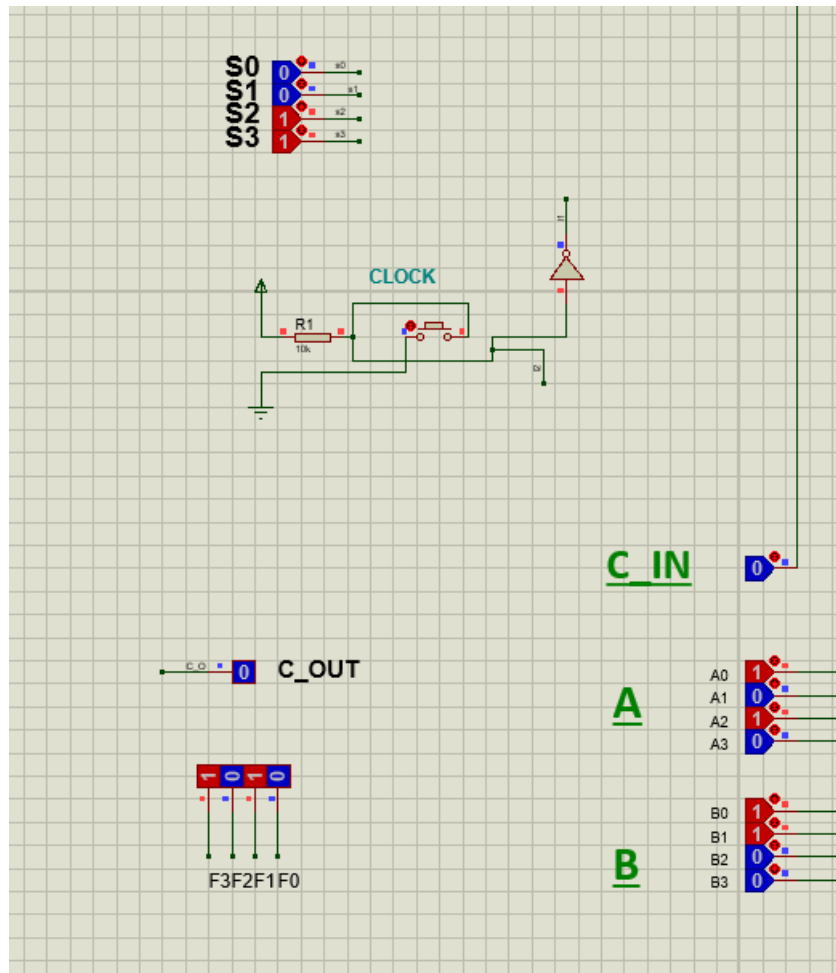
این دستور A و B را XOR می‌کند.



شکل ۳۱. دستور ۰۱۱۰

دستور ۱۱۰۰:

دستور شیفت به چپ برای A:



شکل ۳۲. دستور ۱۱۰۰

## نتیجه گیری:

در آزمایش پنجم درس آزمایشگاه مدارهای منطقی، به آشنایی با واحد محاسبات و منطق پرداختیم.

در بخش اول این آزمایش با تراشه ۷۴۱۸۱ آشنا شدیم و مدار خواسته شده را طراحی کردیم.

در بخش دوم که کار سخت تری پیش رو داشتیم، ابتدا برای حالات مختلف تراشه مناسب به کار بردیم و سپس برای دستورات خواسته شده مداری را طراحی کردیم تا به درستی کار کند.

## منابع و مراجع:

Mano, M. Morris. Computer system architecture. Prentice-Hall of India, ۲۰۰۳