گزارش آزمایشگاه مدارهای منطقی

سال تحصیل01 -00

نویسنده

پارسا شریفی

99101762

آزمایش پنجم: واحد محاسبه و منطق (ALU)

مقدمه: هدف از این آزمایش، آشنایی با واحد محاسبات و منطقALU است. همه بخشهای این آزمایش را با نرمافزار Proteus انجام داده میشود.در بخش اول این آزمایش، با استفاده از دو رجیستر و یک ماژول آمادهی ALU، این قسمت از پردازنده مرکزی کامپیوترها را شبیهسازی میکنیم. مدار این بخش از چندین قسمت تشکیل شده است ، طبق جدول داده شده در دستور کار، تعیین میکند که مقادیر داخل رجیسترها چگونه باید تغییر کنند. در واقع مدار کنترل کننده با استفاده از سه سیگنال ورودی، 6 سیگنال را تعیین میکند: 4 بیت ورودی مدار ALU که تعیین کننده عملیت پردازشی داخل آن هستند، و 2 بیت که مشخص میکنند مقادیر بعدی که باید داخل رجیسترها ثبت شوند از کجا انتخاب میشوند (از ورودی، و یا از خروجی ALU).

در بخش دوم این آزمایش هم، مدار داخلی پکیج ALU را با استفاده از ماژولهای دیگر طوری پیاده سازی میکنیم که با دادن ورودیهای کنترلِ S_0 تا S_3 عملیات نظیر آن را که در جدولی در دستور کار مشخص شده است را انجام میدهد.

مراحل پیادهسازی و روند کار مدار:

چون این آزمایش از دو بخش اصلی تشکیل شده است، توضیحات را نیز در دو بخش ارائه میکنیم:

بخش اول: مداری طراحی کنید که طبق شکل در طرح آزمایش دارای دو ثبات داده A و B ،یک ALU و یک کنترل کننده باشد، به طوری که با دادن کدهای مختلف به ALU ،اعمال مختلف بر روی ورودیها انجام شود. این مدار شامل چند قسمت است .قسمت اول همان ماژول آمادهی ALU است که محاسبات را انجام میدهد. قسمت دوم که همان رجیسترها هستند همان عملوندهای ALU در هر پردازش هستند. بخش آخر یک مدار کنترل کننده است که با استفاده از سه سیگنال که در ورودی آن داده میشود.

سیگنالهای ورودی :

خطوط داده D3-D0 وخطوط دستور M2-M0 ویک کلید از نوع button-push برای بازگرداندن مدار به حالت اولیه (Reset.) یک کلید از نوع button-push 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.

برای پیاده سازی ALU، کافیست که تراشه 74181 را در مدار قرار داده شود. همچنین برای پیاده سازی بخش کنترل کننده به بررسی دیتاشیت آن احتیاج داریم.

برای پیاده سازی همان رجیسترها ابتدا به دو مالتی پلکسر نیاز است تا تعیین شود مقادیر داخل رجیسترها در هر حالت چگونه تامین شوند. در ادامه دو تراشه 74175 وارد مدار میشود و خروجیهای آنها را به عنوان ورودیهای ALU قرار میگیرد. این تراشهها از بین 4 بیت اول و یا 4

بیت دوم، یک مجموعه 4 بیت را با استفاده از سیگنال سلکت خروجی میدهند. بنابراین 4 بیت خروجی مالتی پلکسرها را به ورودی رجیسترها داه میشود.

4 بیت ورودی اول مالتی پلکسر A از خروجی ALU گرفته می‌شود و 4 بیت دوم آن هم به ورودی‌های D_0 تا D_3 متصل هستند. 4 بیت ورودی اول مالتی پلکسر B هم از همان ورودی‌ها تأمین می‌شوند و 4 بیت دوم نیز به خروجی خود رجیستر متصل هستند.

برای بخش کنترل ابتدا باید معادله 6 بیت Selector بر حسب ورودی‌های M_0 تا M_2 محاسبه شود.
جدول زیر اطلاعات ارتباط دهنده بین این سیگنال‌ها می‌باشد:

M_2	M_1	M_0	S_A	S_B	S_3	S_2	S_1	S_0
0	0	0	1	1	×	×	×	×
0	0	1	0	0	×	×	×	×
0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	0	1	0
1	0	0	0	1	0	0	1	1
1	0	1	0	1	0	0	0	0
1	1	0	0	1	1	0	1	1
1	1	1	0	1	1	1	1	0

S_A و S_B , Selector های مالتی پلکسرها می‌باشند و فقط وقتی مقدارشان عوض میشود که مقادیری وارد رجیستر شود. (حالت های 0 ام و 1 ام جدول)

ورودی‌های S_0 تا S_3 نیز با توجه به دیتاشیت ماژول 74181 بدست آمده‌اند. تصویری از قسمتی از این دیتاشیت به صورت زیر است:

در قسمت کنترل ورودی M را برابر مقدار ثابت 1 و ورودی C_n را خالی گذاشته‌ایم. بنابراین تراشه مطابق ستون اول این جدول عمل می‌کند.

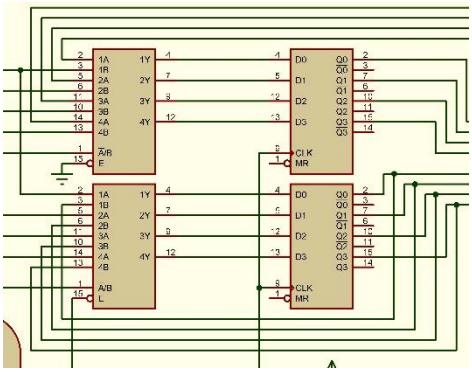
MODE SELECT INPUTS				ACTIVE HIGH INPUTS AND OUTPUTS	
S_3	S_2	S_1	S_0	LOGIC (M=H)	ARITHMETIC(2) (M=L; C_n=H)
L	L	L	L	$\overline{A}$	A
L	L	L	H	$\overline{A+B}$	A + B
L	L	H	L	$A\overline{B}$	A + $\overline{B}$
L	L	H	H	logical 0	minus 1
L	H	L	L	$\overline{AB}$	A plus $A\overline{B}$
L	H	L	H	$\overline{B}$	(A + B) plus $A\overline{B}$
L	H	H	L	$A \oplus B$	A minus B minus 1
L	H	H	H	$A\overline{B}$	$A\overline{B}$ minus 1
H	L	L	L	$\overline{A}+B$	A plus AB
H	L	L	H	$\overline{A \oplus B}$	A plus B
H	L	H	L	B	(A + $\overline{B}$) plus AB
H	L	H	H	AB	AB minus 1
H	H	L	L	logical 1	A plus A(1)
H	H	L	H	A + $\overline{B}$	(A + B) plus A
H	H	H	L	A + B	(A + $\overline{B}$) plus A
H	H	H	H	A	A minus 1

اکنون رابطه بین M_0 تا M_2 و دو بیت S_A و S_B را با جدول کارنو به دست می آید. اما برای چهار بیت بعدی از مالتی پلکسر 8 به 1 استفاده کنیم. معادله دو بیت اول به شرح زیر هستند:

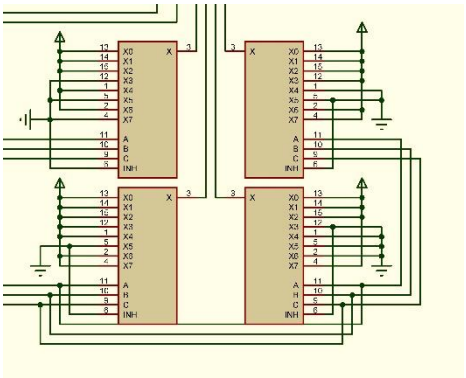
$$S_A = \overline{M_0}\,\overline{M_1}\,\overline{M_2} = \overline{(M_0 + M_1 + M_2)}$$

$$S_B = M_2 + M_1 + \overline{M_0}$$

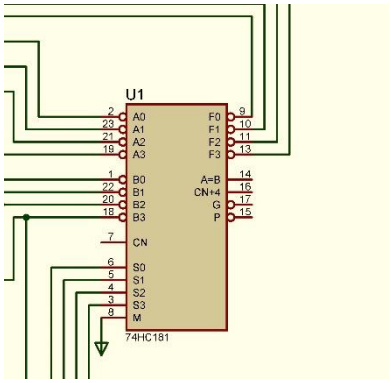
بنابرین بخش کنترل کننده با دریافت 3 ورودی 6 بیت انتخاب کننده را تولید



بخش رجیسترها و مالتی پلکسرها



بخش تولید کننده 4 سیگنال انتخاب کننده

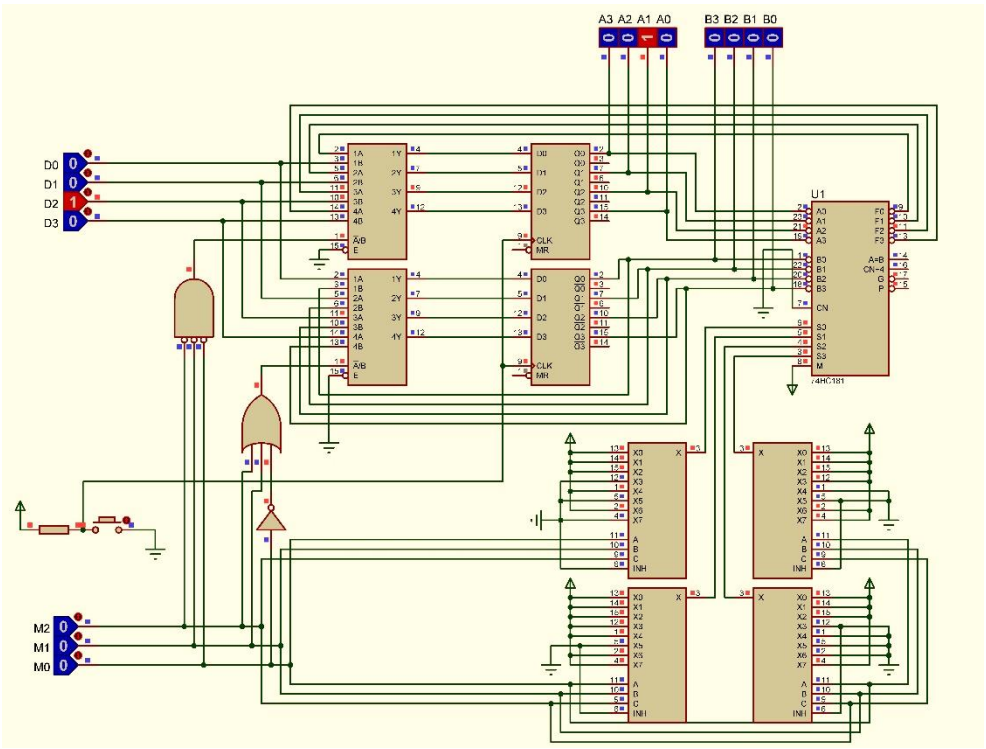


تراشه ALU

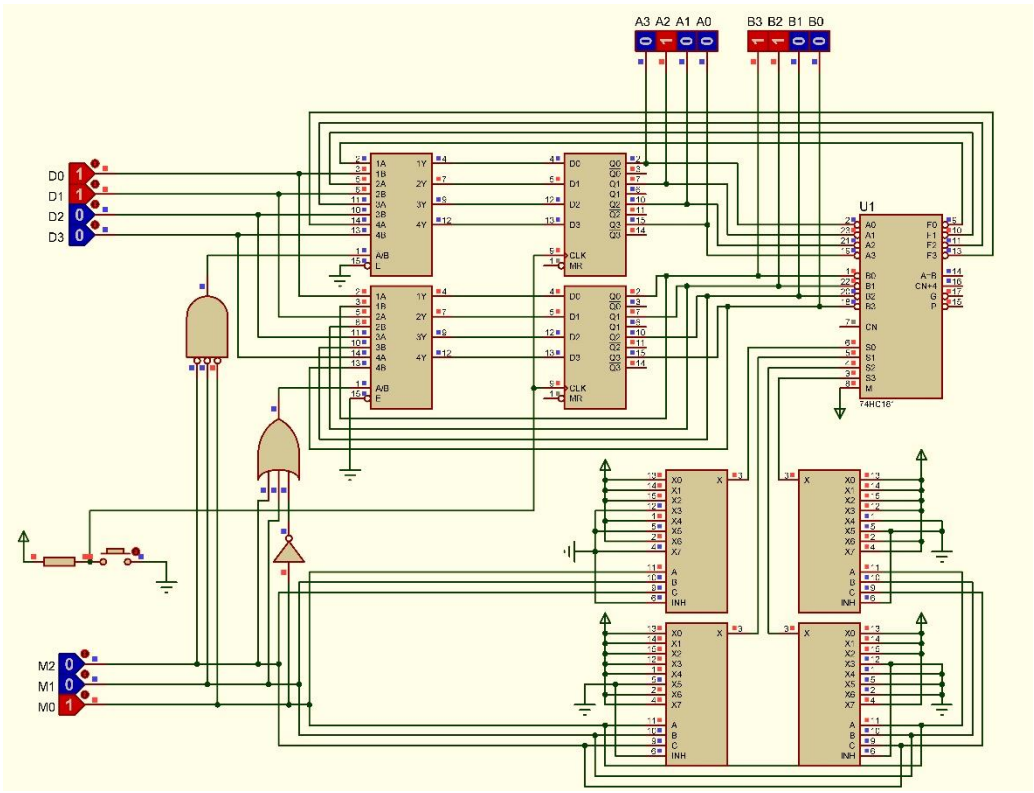
می‌کند. سه تصویر فوق، سه قسمت اصلی مدار را که در بالا توضیحات آن‌ها داده شد نشان می‌دهند. با تغییر ورودی، خروجی مالتی پلکسرها و خروجی بخش ALU تغییر کرده و با اعمال کلاک پالس، در رجیسترها ذخیره می‌شوند.

یک مثال در زیر توضیح داده میشود:

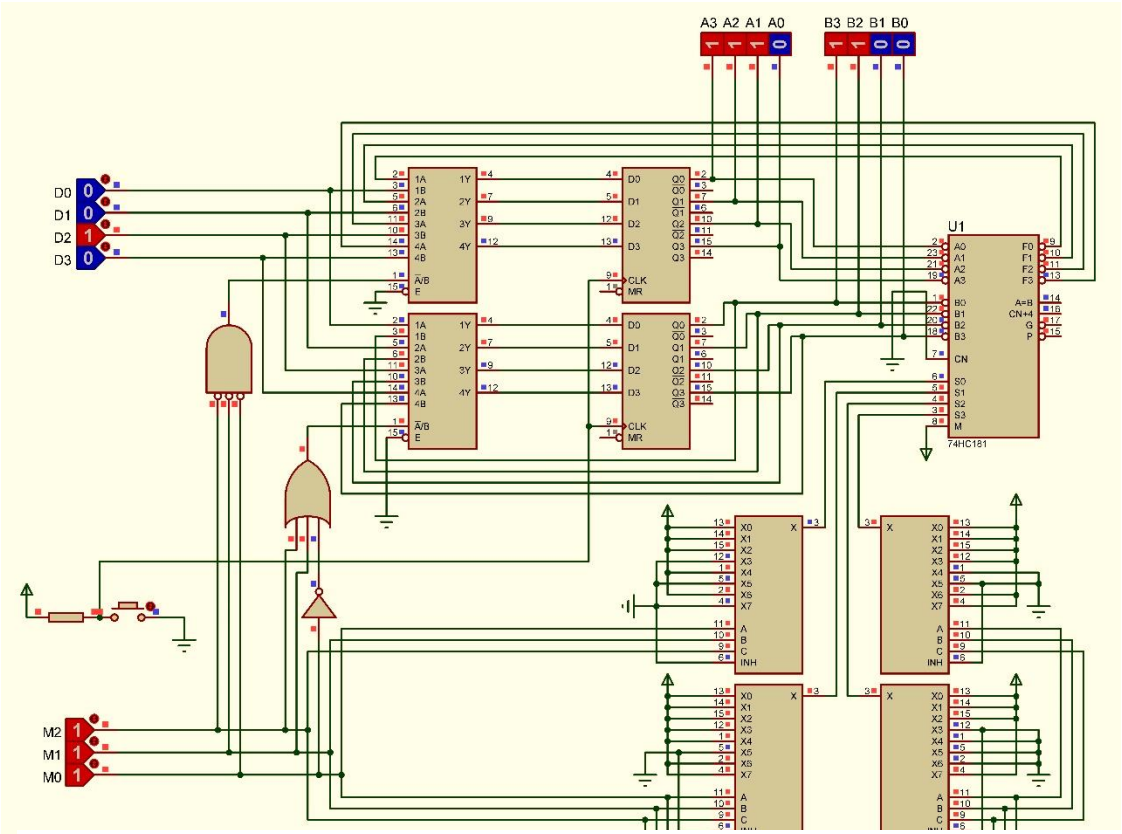
برای جمع دو عدد 4 و 3 ابتدا باید قرار دهیم: $M_2M_1M_0 = 000$ و $D_3D_2D_1D_0 = 0100$ و سپس کلاک را اعمال کنیم. عدد4 در رجیستر A ذخیره می‌شود. سپس قرار می‌دهیم $M_2M_1M_0 = 001$ و $D_3D_2D_1D_0 = 0011$ و کلاک را اعمال می‌کنیم. عدد 3 در رجیستر B ذخیره می‌شود. حالا برای اینکه دو عدد را جمع کنیم و در A ذخیره کنیم، باید قرار دهیم $M_2M_1M_0 = 111$ و کلاک را اعمال کنیم. می‌توانیم مشاهده کنیم که مقدار رجیستر A برابر 0111 خواهد شد:



ذخیره 4



ذخیره3

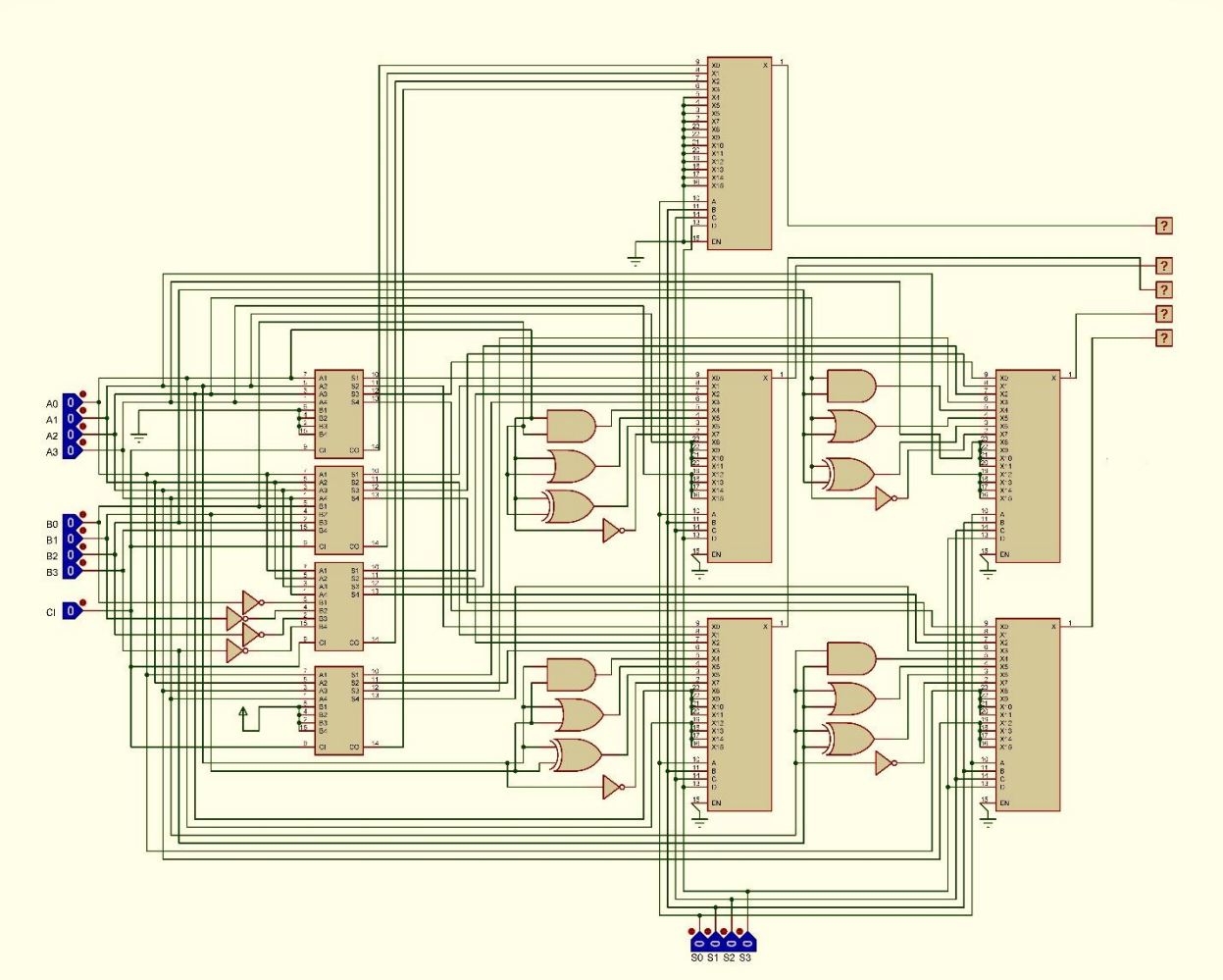


محاسبه 3 + 4

بخش دوم: یک واحد محاسبات و منطق چهاربیتی طبق شکل داده شده در طرح درس باید ساخته شود.

محاسبات ALU را چند Adder و گیت‌های And، Or و Xor انجام شده و با استفاده از چند مالتی پلکسر و بیت‌های سلکتور مشخص می شود حاصل کدام محاسبه باید در خروجی نمایش داده شود. تصویر زیر، تصویر مدار نهایی است:

در مدار بالا، تراشه Adder اول، دو عملیات اول مشخص شده در جدول، یعنی Transfer A و Increment A را انجام می‌دهد. به این صورت که A را با مقدار C_{in} جمع کرده و خروجی می‌دهد.



هر کدام از خروجی این Adder به ورودی X_0 مالتی پلکسرهای متناظر متصل شده‌اند. تراشه Adder دیگر دو عملیات بعدی که جمع A و B هستند را با ورودی C_{in} انجام می‌دهد. درست مانند جمع کننده قبلی خروجی‌های این تراشه نیز هر کدام به یک مالتی پلکسر متصل هستند. مدار جمع کننده بعدی، A را با مکمل B و ورودی C_{in} جمع می‌کند. به این صورت دو عملیات بعدی که تفریق با استفاده از Borrow هستند نیز پیاده سازی می‌شود. مدار جمع کننده آخری تا حد زیادی شبیه مدار اول عمل می‌کند با این

تفاوت که به جای 0، عدد A را با مقدار 1- و C_{in} جمع می‌کند. در نتیجه دو عملیات خواسته شده بعدی نیز به کمک این تراشه پیاده سازی می‌شوند. 4 عمل بعدی، یعنی And و Or و Xor و Not را هر کدام به کمک 4 گیت متناظر که هر کدام روی یک بیت از A و B کار می‌کنند، پیاده سازی کرده‌ایم. در نهایت، برای شیفت دادن A به چپ یا راست، به صورت دستی ورودی مالتی پلکسرها را ارقام متفاوت از A داده‌ایم. به عنوان مثال، ورودی مالتی پلکسر دوم در حالت شیفت به راست، رقم سوم A است؛ یعنی این مالتی پلکسر یک رقم شیفت داده شده A را خروجی می‌دهد. اکنون که همه‌ی توابع را پیاده کرده‌ایم، هرکدام را به ورودی نظیر خود در مالتی پلکسر متصل می‌کنیم. 4 مالتی پلکسر برای 4 خروجی F_0 تا F_3 داریم و یک مالتی پلکسر هم برای خروجی C_{out} داریم.

حال یک مثال از کارکرد مدار مطرح میکنیم:

فرض کنید می‌خواهیم مانند مثال قبل، دو عدد 4 و 3 را با هم جمع کنیم. ورودی عدد A را به صورت $A_3A_2A_1A_0 = 0100$ و ورودی عدد B را به صورت $B_3B_2B_1B_0 = 0011$ قرار می‌دهیم. چون می‌خواهیم عملیات جمع انجام شود، بیت‌های سلکتور را نیز برابر $S_3S_2S_1S_0 = 0001$ قرار می‌دهیم. بلافاصله خروجی مورد نظر، یعنی عدد 7 در خروجی نمایش داده می‌شود می‌خواهیم عملیات جمع انجام

شود، بیت‌های سلکتور را نیز برابر $S_3S_2S_1S_0 = 0001$ قرار می‌دهیم. بلافاصله خروجی مورد نظر، یعنی عدد 7 در خروجی نمایش داده می‌شود: