

«دانشگاه علم و فرهنگ»

مجموعه گزارش کارهای
آزمایشگاه معماری کامپیوتر

استاد سلیمانی

تهیه و تنظیم:

حمیدرضا قاضی سعیدی
(۸۲۱۱۱۲۲۰)

خرداد ۱۳۸۵

عنوان آزمایش	فهرست گزارش‌ها	شماره صفحه
آزمایش شماره ۱:	ضرب‌کننده سریع	۲
آزمایش شماره ۲:	Carry Look-Ahead Adder	۷
آزمایش شماره ۳:	گذرگاه (Bus)	۱۱
آزمایش شماره ۴:	طراحی مولد سیگنال کنترل	۱۵
آزمایش شماره ۵:	آشنایی با حافظه‌های EPROM	۱۷
آزمایش شماره ۶:	آشنایی با حافظه‌های RAM	۲۱

مجموعه گزارش کارهای آزمایشگاه معماری کامپیوتر
 کارشناسی پیوسته کامپیوتر (استاد سلیمانی)

تهیه و تنظیم:

خرداد ماه ۱۳۸۵

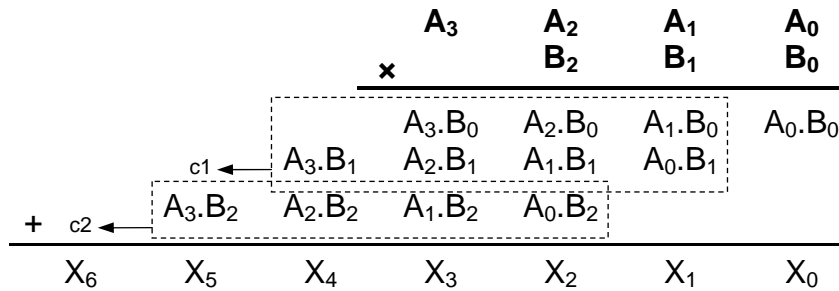
حمیدرضا قاضی سعیدی (۸۲۱۱۱۲۲۰)

آزمایش شماره یک:

« ضرب کننده سریع »

تئوری آزمایش:

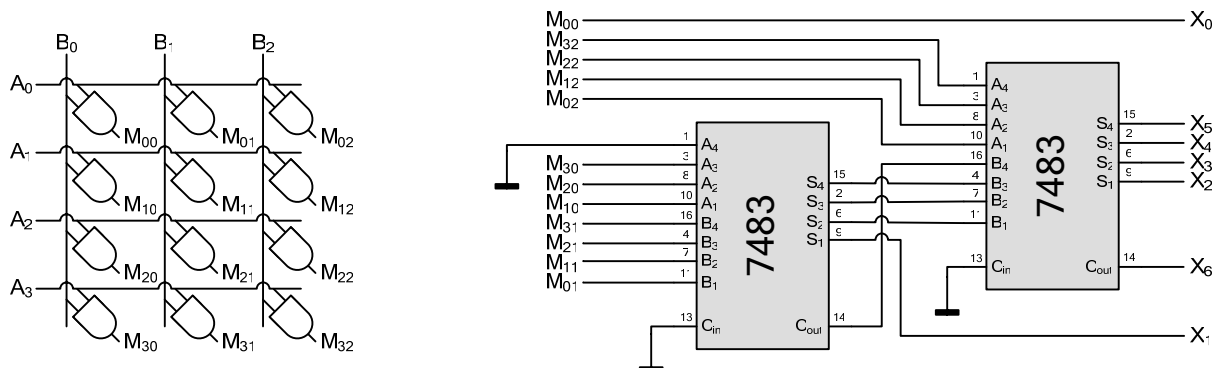
اساس کار ضرب کننده سریع با استفاده از تکنیک Array Multiplier، شبیه به عمل ضرب دستی می باشد. هر بیت مضروب در تک تک بیت های مضروب فیه ضرب شده و این حاصل ها با هم جمع می گردند. عمل ضرب بیت در بیت را بسادگی می توان با یک گیت AND پیاده سازی کرد. برای انجام عمل جمع نهایی می توان از Full Adder های چند بیتی بهره برد. ضرب چهار بیت در سه بیت به این صورت خواهد بود:



برای جمع از دو Full Adder چهار بیتی استفاده شده است. ستون سمت راست دارای تنها یک عملگر است و نیازی به جمع ندارد. مستطیل نقطه چین بالا نشان دهنده اولین Full Adder است که عمل جمع دو سطر اول را انجام می دهد. بیت X_1 جواب نهایی، بیت سمت راست خروجی این Full Adder است. Full Adder دوم که با مستطیل نقطه چین پایینی نمایش داده شده، سطر سوم را با سه بیت سمت چپ جواب Full Adder اول و carry خروجی آن جمع می کند که خروجی آن بیت های X_2 تا X_5 و carry تولید شده آن بیت X_6 حاصل ضرب می باشند.

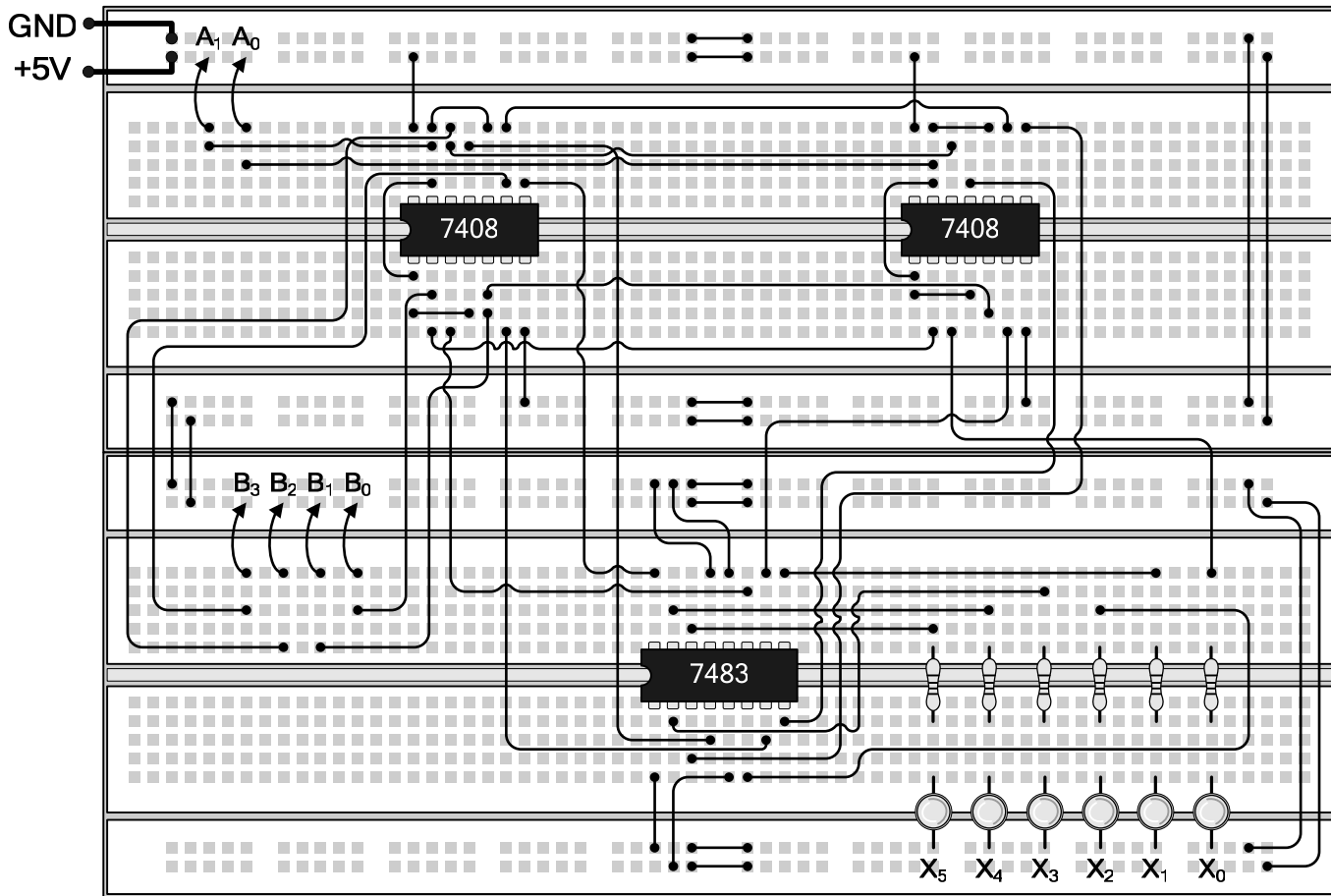
طراحی مدار:

شکل زیر مدار ضرب فوق را نمایش می دهد. برای ساده تر شدن مدار، بجای عبارت $(A_i.B_j)$ از متغیر (M_{ij}) استفاده شده است:



شکل مدار روی Bread-Board:

شکل زیر سیم‌بندی عملی یک مدار ضرب کننده چهار بیت در دو بیت را نمایش می‌دهد:



سؤالات:

۱) روش‌های مختلف ضرب را نام ببرید.

- ضرب کننده آرایه‌ای (Array Multiplier)
- الگوریتم ضرب مقدار - علامت به روش Shift & Add
- الگوریتم ضرب مکمل ۲- علامت به روش Shift & Add (الگوریتم ضرب بوت)
- Wallace Tree
- Modified (Carry Save)
- روش شیفت یکنواخت

۲) سرعت ضرب کننده فوق را با روش‌های دیگر ضرب مقایسه و علت آن را بیان کنید.

این روش از الگوریتم‌های ترتیبی سریعتر است. زیرا اساس این الگوریتم‌ها، حلقه‌هایی با مراحل مختلف است و همچنین باید به مسائلی مثل زمانبندی مراحل و انتقال بین رجیسترها نیز توجه شود. درحالی که روش آرایه‌ای یک روش ترکیبی است. همچنین الگوریتم‌های ترتیبی

شامل اعمالی مثل شیف و جمع/تفریق بازای تعداد بیت‌های ورودی هستند، درحالی که روش آرایه‌ای فقط شامل یک عمل جمع چند بیتی است که می‌توان آن را بصورت یک جمع کننده واحد پیاده‌سازی کرد.

۳) رابطه‌ای بین تعداد بیت‌های ورودی و تعداد IC های AND و Full Adder بکار رفته پیدا کنید.

اگر ضرب کننده مورد نظر m در n بیتی باشد، ما برای طراحی مدار به تمام ترکیب‌های AND بیت‌های ورودی ($A_i.B_j$) نیاز داریم، بنابراین به $m \times n$ گیت AND احتیاج داریم. از آنجاییکه هر آی‌سی 7408 چهار گیت AND دارد، تعداد آی‌سی‌های AND مورد نیاز برابر است با:

$$\left\lceil \frac{m.n}{4} \right\rceil = \text{تعداد آی‌سی‌های 7408 (AND)}$$

با توجه به حاصل جمع نوشته شده در صفحه اول پیش‌گزارش، تعداد سطرهایی که باید جمع شوند برابر با تعداد بیت‌های عملوند B (n بیت) است، بنابراین به n-1 جمع نیاز است. تعداد بیت‌های Full Adder هایی که با مستطیل‌های نقطه‌چین نمایش داده شده‌اند، نیز برابر با تعداد بیت‌های عملوند A (m بیت) است. از آنجاییکه آی‌سی 7483 یک جمع کننده ۴ بیتی است، برای جمع‌های بیش از ۴ بیت باید از چند آی‌سی متصل به هم استفاده کرد. بنابراین:

$$\left\lceil \frac{m}{4} \right\rceil \times (n-1) = \text{تعداد آی‌سی‌های 7483 (Full Adder)}$$

۴) روش Bit-Slice در طراحی ALU چیست؟ یک نمونه از کاربرد آن را در مورد ضرب کننده‌های سریع TTL بیان کنید.

در این روش بجای آنکه ALU هایی با تعداد بیت‌های زیاد بسازیم، ALU هایی با بیت‌های کمتر را طوری طراحی می‌کنیم که امکان توسعه داشته باشند، بدین ترتیب که بتوان با ترکیب آنها یک ALU بزرگتر ساخت. مثلاً می‌توان از دو ALU ۸ بیتی، با وصل کردن carry خروجی یکی به carry ورودی دیگری، یک ALU بزرگتر ساخت که می‌تواند عمل جمع را روی ۱۶ بیت انجام دهد.

۵) روش Wallace Tree در بستن واحدهای ضرب کننده چگونه است؟

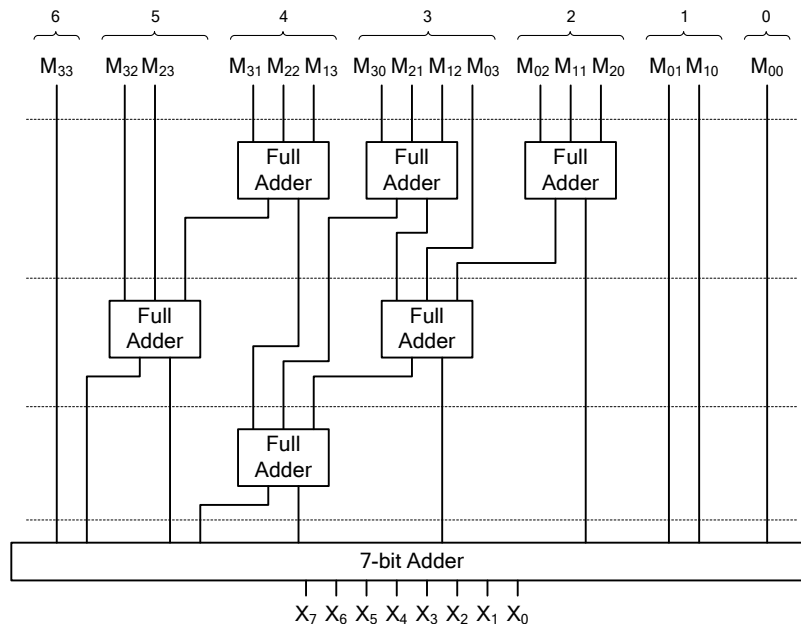
از آنجاییکه در روش Array Multiplier نیاز به یک جمع کننده داریم که تعداد ورودی‌های آن بازای بعضی از بیت‌ها زیاد است، طراحی مدار آن مشکل شده و زمان انتشار رقم‌های نقلی آن نیز باعث کند شدن مدار می‌شود. لذا برای کاهش دادن تعداد بیت‌های هم‌وزن از روش Wallace Tree استفاده می‌شود.

ابتدا تمامی ترکیب‌های AND بیت‌های دو ورودی (M_{ij}) را تولید می‌کنیم. به هر کدام از آنها بر این اساس که در کدام ستون جمع نهایی قرار می‌گیرند، یک وزن اختصاص می‌یابد. برای مثال در ضرب چهار بیت در سه بیت، M_{02} در بیت سوم از سمت راست (بیت با ارزش ۴) جمع قرار می‌گیرد، لذا به M_{02} وزن ۴ می‌دهیم. سپس تا هنگامی که بیش از دو ورودی با وزن‌های یکسان داشته باشیم، یک لایه کاهنده به ترتیب زیر ایجاد می‌کنیم:

- هر سه سیم با اوزان یکسان را به یک Full Adder می‌دهیم. خروجی تولید شده به ازای هر سه سیم، یک سیم به وزن مساوی اوزان ورودی و یک سیم به وزن دو برابر اوزان ورودی خواهد بود.
- اگر دو سیم با اوزان برابر باقی‌ماند، آنها را به یک Half Adder می‌دهیم.
- اگر فقط یک سیم باقی‌ماند، آن را به همین صورت به لایه بعد می‌فرستیم.

در پایان ورودی‌های باقیمانده را براساس وزن‌شان به یک جمع‌کننده دو ورودی معمولی می‌فرستیم. چون تعداد لایه‌ها حداکثر $\log n$ و تأخیر انتشار رقم نقلی هر لایه 1 است، تأخیر انتشار در روش Wallace Tree از مرتبه $\log n$ خواهد بود. در حالی که اگر از جمع معمولی استفاده می‌کردیم، تأخیر انتشار از مرتبه $(\log n)^2$ می‌شد.

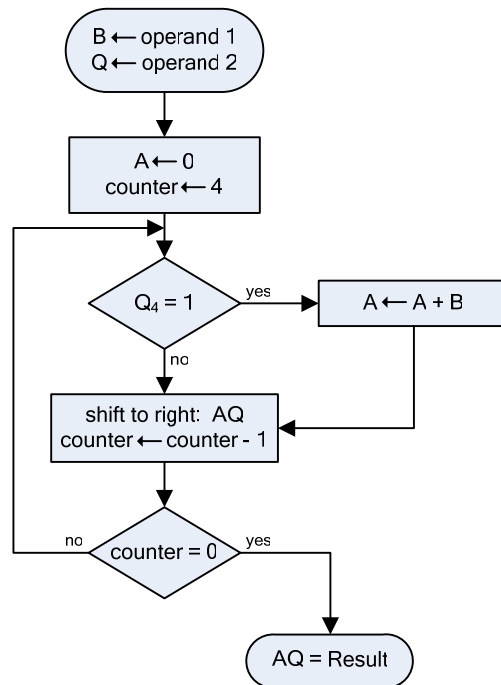
شکل زیر مثالی از روش Wallace Tree برای یک ضرب‌کننده چهار بیت در چهار بیت را نشان می‌دهد. ورودی‌ها دارای وزن‌های ۰ تا ۶ هستند و Full Adder ها، سه لایه کاهنده تشکیل داده اند:



۶) در مورد ضرب اعداد علامت‌دار و بدون علامت با روش Array Multiplier بحث کنید.

این روش در اصل برای اعداد بدون علامت طراحی شده است. برای استفاده از این روش در اعداد علامت‌دار، باید علامت اعداد جداگانه محاسبه شود و مقادیر با استفاده از این روش ضرب گردند. اگر اعداد به روش مقدار - علامت ذخیره شده باشند، این عمل بسادگی قابل اجرا است. اما اگر اعداد به روش مکمل - علامت ذخیره شده باشند، باید مکمل‌های ورودی‌ها به مقدار تبدیل شود و از جواب نهایی نیز در صورتی که علامت آن منفی باشد، مکمل گرفته شود که این کارها باعث پایین آمدن سرعت خواهد شد. لذا این روش ضرب برای اعداد مکمل - علامت مناسب به نظر نمی‌رسد.

(۷) بلوک دیاگرام یک ضرب کننده 4x4 bit را به روش Shift & Add رسم کنید.



(۸) ضرب دو عدد 53x72 را به روش Shift & Add مرحله به مرحله نشان دهید.

B = 0110101	A	Q	counter
Initialization	0000000	1001000	7
Shift Right: AQ	0000000	0100100	6
Shift Right: AQ	0000000	0010010	5
Shift Right: AQ	0000000	0001001	4
	+ 0110101		
Add B to A	0110101	0001001	
Shift Right: AQ	0011010	1000100	3
Shift Right: AQ	0001101	0100010	2
Shift Right: AQ	0000110	1010001	1
	+ 0110101		
Add B to A	0111011	1010001	
Shift Right: AQ	0011101	1101000	0
Result is in AQ	= 3816 (53*72)		

آزمایش شماره دو:

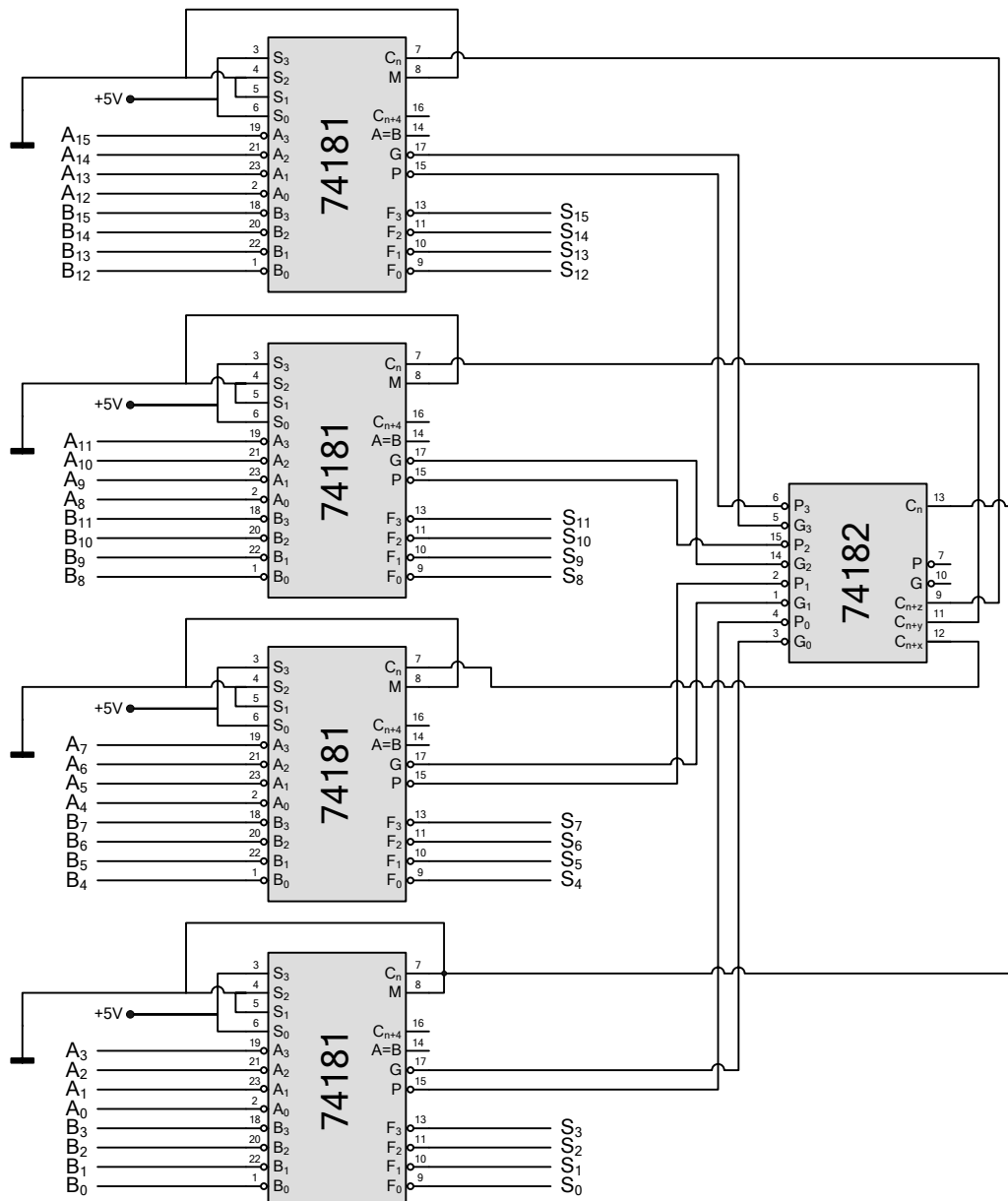
« Carry Look-Ahead Adder »

تئوری آزمایش:

اساس کار ضرب کننده سریع با استفاده از تکنیک Array Multiplier، شبیه به عمل ضرب دستی می باشد. هر بیت مضروب در تک تک بیت های مضروب فیه ضرب شده و این حاصل ها با هم جمع می گردند. عمل ضرب بیت در بیت را بسادگی می توان با یک گیت AND پیاده سازی کرد. برای انجام عمل جمع نهایی می توان از Full Adder های چند بیتی بهره برد. ضرب چهار بیت در سه بیت به این صورت خواهد بود:

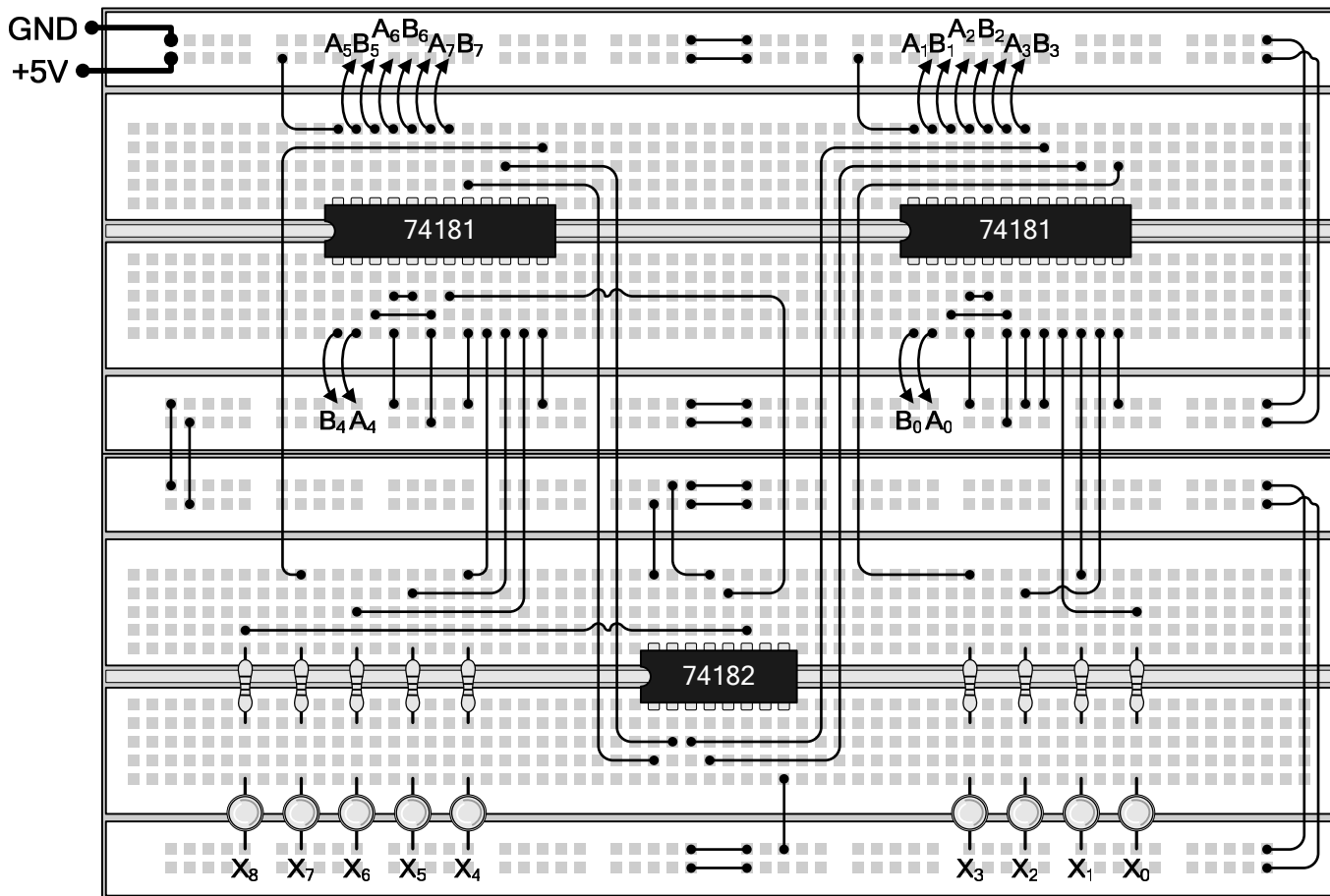
طراحی مدار:

شکل زیر مدار ضرب فوق را نمایش می دهد. برای ساده تر شدن مدار، بجای حاصل عبارت $(A_i \cdot B_j)$ از متغیر (M_{ij}) استفاده شده است:



شکل مدار روی Bread-Board:

شکل زیر سیم‌بندی عملی مدار Carry Look-Ahead Adder فوق را روی ۸ بیت نمایش می‌دهد:



سؤالات:

۱) مقدار تأخیر در روش CLA را از طریق محاسبه بدست آورید.

اگر زمان تأخیر گیت AND و OR دو ورودی را تقریباً برابر T در نظر بگیریم، زمان تأخیر گیت XOR تقریباً $3T$ خواهد بود. بر طبق پاسخ سؤال ۳، زمان مورد نیاز برای محاسبه بیت‌های G و P توسط IC های 74181 برابر است با تأخیر گیت XOR (برای محاسبه G_i و P_i) و مجموع تأخیر یک گیت AND چهار ورودی و یک گیت OR چهار ورودی (تقریباً $4T$). مدت زمان مورد نیاز برای تولید رقم‌های نقلی توسط آی‌سی 74182 نیز برابر با $4T$ می‌باشد. در نتیجه برای تولید رقم‌های نقلی جمع‌کننده فوق به $11T$ زمان نیاز است. زمان مورد نیاز یک آی‌سی 74181 برای اجرای عمل جمع (با استفاده از CLA) برابر $4T$ برای محاسبه C_3 و تأخیر یک گیت XOR دو ورودی برای محاسبه بیت چهارم حاصلجمع است. لذا کل تأخیر در روش CLA تقریباً $18T$ خواهد بود.

(۲) چرا روش CLA از روش جمع عادی سریعتر است؟ درصد این اختلاف را برای یک جمع کننده ۱۶ بیتی پیدا کنید.

زیرا در روش جمع عادی، برای محاسبه هر بیت حاصلجمع نیاز به رقم نقلی بیت قبل است و از آنجاییکه محاسبه هر بیت نقلی دارای یک تأخیر است، تأخیر محاسبه آخرین بیت، مضربی از تعداد بیت‌های عمل جمع خواهد بود. درحالی که در روش CLA، تأخیر محاسبه بیت‌های نقلی برابر با تأخیر محاسبه یک عبارت منطقی معلوم و عددی ثابت است.

زمان مورد نیاز برای محاسبه یک رقم نقلی از روی رقم نقلی بعدی برابر است با مجموع تأخیرهای یک گیت XOR، یک گیت AND و یک گیت OR دو ورودی (تقریباً 5τ). در یک جمع کننده ۱۶ بیتی، تأخیر رقم نقلی بیت آخر (C_{15})، ۱۵ برابر این زمان است. اگر تأخیر محاسبه جمع آخرین بیت را تأخیر گیت XOR دو ورودی در نظر بگیریم، مجموع تأخیرها برابر خواهد بود با:

$$\text{تأخیر کل مدار} = (15 \times 5\tau) + 3\tau = 78\tau$$

لذا تأخیر در روش CLA تقریباً 23% روش معمولی جمع ۱۶ بیتی است.

(۳) روش CLA را از طریق بدست آوردن روابط منطقی آن توضیح دهید.

در روش CLA بجای آنکه بیت نقلی ورودی یک واحد Full Adder از واحد قبلی آن انتقال یابد، بطور جداگانه از روی عملوندهای جمع محاسبه می‌گردد. بطور کلی اگر A_i و B_i دو بیت ورودی یک Full Adder و C_i رقم نقلی ورودی آن باشد، C_{i+1} رقم نقلی خروجی از رابطه زیر بدست می‌آید:

$$C_{i+1} = A_i.B_i + (A_i \oplus B_i).C_i$$

جمله $A_i.B_i$ را Generation Carry و جمله $(A_i \oplus B_i).C_i$ را Propagation Carry می‌نامند:

$$G_i = A_i.B_i$$

$$P_i = A_i \oplus B_i$$

فرض کنیم C_0 رقم نقلی ورودی باشد. اگر C_0 عبارت رقم نقلی دوم را با مقدارش جایگزین کنیم، سپس بجای C_1 در رقم نقلی سوم عبارت بدست آمده را قرار دهیم و این کار را به همین ترتیب ادامه دهیم، تمام بیت‌های نقلی را می‌توانیم مستقیماً از روی ورودی‌ها بدست آوریم:

$$C_1 = G_0 + P_0.C_0$$

$$C_2 = G_1 + P_1(G_0 + P_0.C_0) = G_1 + G_0P_1 + P_1P_0.C_0$$

$$C_3 = G_2 + P_2(G_1 + G_0P_1 + P_1P_0.C_0) = G_2 + G_1P_2 + G_0P_2P_1 + P_2P_1P_0.C_0$$

$$C_4 = G_3 + P_3(G_2 + G_1P_2 + G_0P_2P_1 + P_2P_1P_0.C_0) = G_3 + G_2P_3 + G_1P_3P_2 + G_0P_3P_2P_1 + P_3P_2P_1P_0.C_0$$

$$= (G_3 + G_2P_3 + G_1P_3P_2 + G_0P_3P_2P_1) + (P_3P_2P_1P_0).C_0$$

هر آی‌سی 74181 درحالت Carry Look-Ahead Adder نه ورودی دارد. رقم نقلی ورودی (C_0)، چهار ورودی $A_0...A_3$ و چهار ورودی $B_0...B_3$. این IC بصورت داخلی عمل جمع این چهار جفت بیت را انجام می‌دهد. رقم‌های نقلی C_1 ، C_2 و C_3 براساس معادلات بالا محاسبه می‌گردند. اما برای محاسبه رقم نقلی ورودی IC بعدی باید اطلاعاتی به آی‌سی Carry Generator (74182) ارسال شود.

ورودی‌های این IC، بصورت G و P است و ارقام نقلی را بر اساس جملات بالا محاسبه می‌کند. بنابراین بیت‌های P و G که به این IC ارسال می‌شوند باید طوری محاسبه شوند که رقم نقلی صحیح توسط Carry Generator ایجاد شود. رقم نقلی مورد نظر C_4 می‌باشد. همانطور که در معادله بالا مشاهده می‌شود، اگر C_4 به شکل $G + P.C_0$ نوشته شود، G و P بصورت زیر خواهند بود:

$$G = G_3 + G_2P_3 + G_1P_3P_2 + G_0P_3P_2P_1$$

$$P = P_3P_2P_1P_0$$

Carry Generator رقم‌های نقلی را بر این اساس محاسبه می‌کند:

$$C_{n+x} = G_0 + P_0.C_n$$

$$C_{n+y} = G_1 + P_1(G_0 + P_0.C_n) = G_1 + G_0P_1 + P_1P_0.C_n$$

$$C_{n+z} = G_2 + P_2(G_1 + G_0P_1 + P_1P_0.C_n) = G_2 + G_1P_2 + G_0P_2P_1 + P_2P_1P_0.C_n$$

$$G = G_3 + G_2P_3 + G_1P_3P_2 + G_0P_3P_2P_1$$

$$P = P_3P_2P_1P_0$$

۴) IC های 2901 و 2902 چه IC هایی هستند؟

IC های 2901 و 2902 نیز همانند 74181 و 74182، ALU و Carry Look-Ahead Generator هستند.

۵) یک جمع‌کننده ۶۴ بیتی را به کمک 74181 و 74182 رسم کنید. (رقم تمام پایه‌ها لازم نیست)

۶) آیا می‌توان فقط با استفاده از 74181 جمع‌کننده ۱۶ بیتی ساخت؟ چگونه؟

بله. در صورتی که خروجی C_{n+4} هر IC به ورودی C_n آ‌سی بعدی وصل شود، بیت نقلی بین IC ها مانند روش معمولی انتقال می‌یابد. اما در داخل IC جمع به روش CLA صورت می‌گیرد.

۷) با استفاده از این روش حداکثر چند بیت ورودی را می‌توان جمع نمود؟

با استفاده از این روش هر تعداد ورودی را می‌توان جمع نمود.



آزمایش شماره سه:

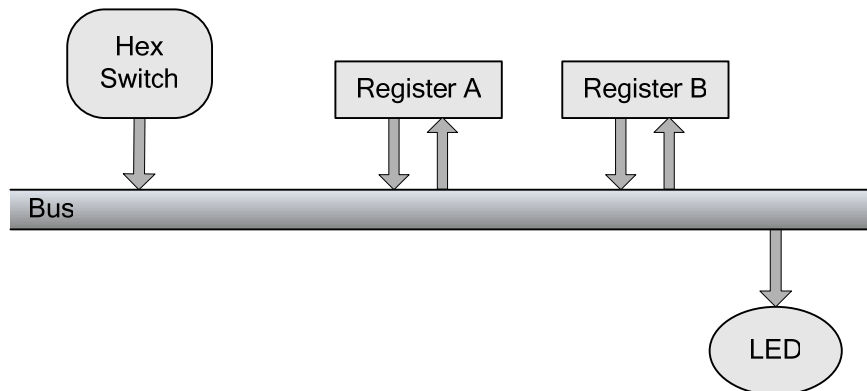
« گذرگاه (Bus) »

تئوری آزمایش:

یک سیستم دیجیتال شامل واحدهای مختلفی است. این واحدها نیازمند ایجاد ارتباط میان خود هستند. مثلاً لازم است که بتوان مقدار یک رجیستر را به رجیستر دیگری انتقال داد. ایجاد تمام ارتباطات بین واحدها، مدار را بسیار پیچیده می‌کند. روش کارآمدتر استفاده از یک خط گذرگاه مشترک است. این گذرگاه مشترک Bus نام دارد.

دو روش برای پیاده‌سازی Bus وجود دارد: استفاده از mux و استفاده از بافرهای سه‌حالت. از آنجاییکه خروجی‌های واحدها به یک خط مشترک متصل می‌شوند، خطر تأثیر بار شدن خروجی روی گیت‌ها بوجود می‌آید. می‌توان برای حل این مشکل از گیت‌های سه‌حالت استفاده کرد، بدین ترتیب که همواره خروجی تمام واحدها را به غیر از واحدی که خروجی آن مطلوب است، به حالت امپدانس بالا ببریم. بدین ترتیب عملاً تنها یک خروجی به خط انتقال وصل خواهد بود و بقیه خروجی‌ها مثل مدار باز عمل خواهند کرد.

در این آزمایش هدف طراحی یک Bus ۴ بیتی است که یک Hex-switch، چهار LED و دو رجیستر ۴ بیتی را به یکدیگر ارتباط می‌دهد. دو بافر ۴ بیتی سه‌حالت اتصال Hex-switch و LED ها را به Bus کنترل می‌کنند:



بافرهای، آی‌سی‌های 74126 هستند. سیگنال کنترل بافرهای Hex-switch و LED ها بترتیب C_L و C_S است که با ورودی 0 امپدانس بالا و با ورودی 1 انتقال معمولی انجام می‌دهند. از آی‌سی 74373 برای رجیستر استفاده شده است. دو رجیستر هر کدام دارای یک ورودی E برای کنترل بارگذاری ورودی هستند که با 1 بارگذاری و با 0 ذخیره می‌کند. رجیسترها همچنین دارای کنترل خروجی O هستند که با 0 مقدار ذخیره شده را روی خروجی‌ها می‌فرستد و با 1 خروجی‌ها را در حالت امپدانس بالا قرار می‌دهد.

ریز دستورالعمل‌هایی که این Bus باید قادر به اجرای آنها باشد، به همراه سیگنال کنترلی مربوطه عبارتند از:

- 1: MOV Reg A, Hex SW
- 2: MOV Reg B, Hex SW
- 3: MOV LED, Reg A
- 4: MOV LED, Reg B
- 5: MOV Reg A, Reg B

هر عمل انتقال معادل یک ورودی کنترلی 1 تا 5 است. بازای هر دستورالعمل باید یک ترکیب از سیگنال‌های کنترلی تولید شوند. اگر با ورودی صفر کلیه‌المان‌ها از Bus جدا شوند، جدول زیر سیگنال‌های تولید شده بر اساس کنترل را نمایش می‌دهد:

Ctrl	C _S	E _A	O _A	E _B	O _B	C _L
0	0	0	1	0	1	0
1	1	1	1	0	1	0
2	1	0	1	1	1	0
3	0	0	0	0	1	1
4	0	0	1	0	0	1
5	0	1	1	0	0	0

ورودی کنترلی به یک آی‌سی 74138 (Decoder) داده می‌شوند. سیگنال‌ها تابعی از خروجی‌های دیکدر (منطق صفر) هستند. روابط بدست آمده برای سیگنال‌ها بدین ترتیب می‌باشند:

$$C_S = D_1 + D_2 = \overline{D_1} \bullet \overline{D_2}$$

$$E_B = D_2 = \overline{D_2}$$

$$E_A = D_1 + D_5 = \overline{D_1} \bullet \overline{D_5}$$

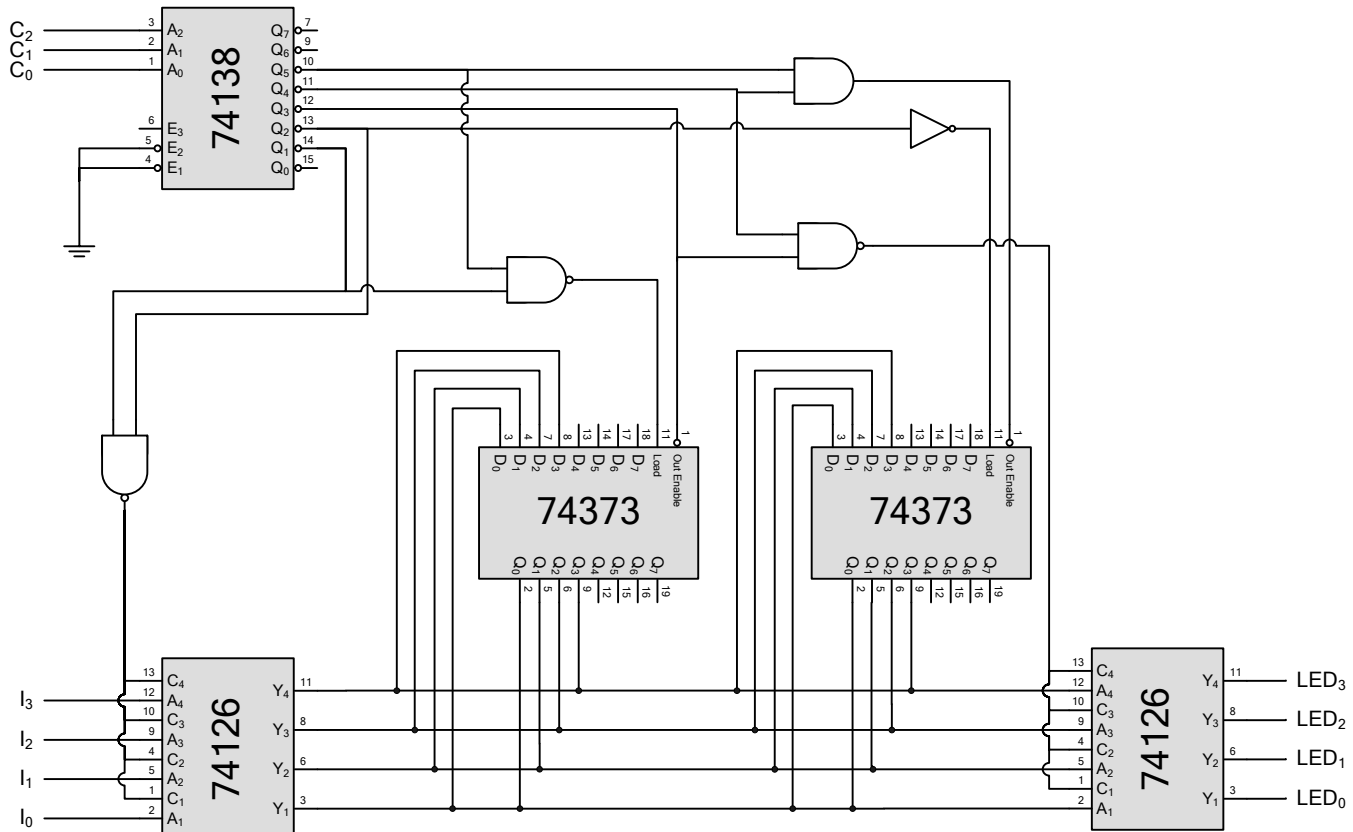
$$O_B = \overline{D_4} + \overline{D_5} = \overline{D_4} \bullet \overline{D_5}$$

$$O_A = \overline{D_3}$$

$$C_L = D_3 + D_4 = \overline{D_3} \bullet \overline{D_4}$$

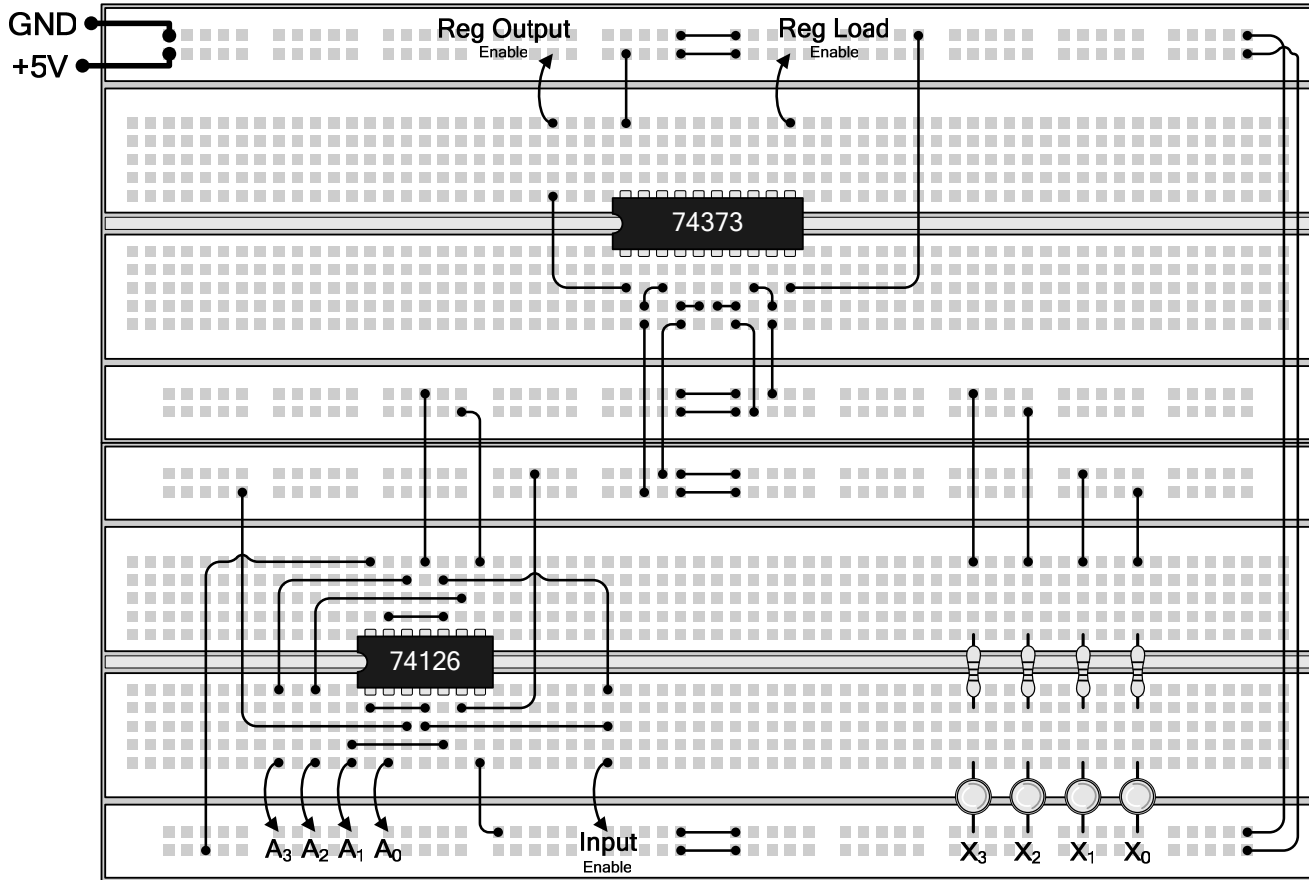
طراحی مدار:

شکل زیر مدار Bus مورد نظر را نشان می‌دهد:



شکل مدار روی Bread-Board:

شکل زیر سیم‌بندی عملی یک Bus ۴ بیتی را نشان می‌دهد که چهار ورودی با واسطه یک بافر ۴ بیتی، یک Register ۴ بیتی و چهار LED به عنوان خروجی (بصورت مستقیم) به آن متصل هستند. سیگنال‌های کنترل بصورت دستی به مدار داده می‌شوند:



سؤالات:

(۱) خاصیت آی‌سی‌های 3-state چیست؟ مدار خروجی آنها را آنالیز کنید.

خروجی آی‌سی‌های 3-state به غیر از دو منطق 0 و 1، یک حالت دیگر نیز دارد. اگر خروجی چند آی‌سی به هم وصل شوند، تاثیر بار باعث آسیب دیدن آی‌سی می‌شود. لذا در صورت نیاز باید روشی بکار برد تا همواره یکی از خروجی‌ها متصل باشند و بقیه را از مدار خارج کرد. آی‌سی‌های 3-state این قابلیت را دارند که با یک ورودی فعال‌ساز، خروجی‌های خود را به حالت سوم یعنی امپدانس بالا ببرند. بدین ترتیب می‌توان خروجی‌های این نوع آی‌سی‌ها را بدون نگرانی به هم متصل کرد و فعال‌ساز آنها را طوری کنترل کرد که در هر لحظه تنها یکی از آنها فعال و بقیه در حالت امپدانس بالا قرار بگیرند.

(۲) چنانچه در Bus طراحی شده توسط المان‌های 3-state، دو منبع، همزمان اطلاعات خود را بر روی Bus قرار دهند، چه اتفاقی می‌افتد؟

اگر خروجی‌های آنها باهم متفاوت باشد، تاثیر بار متفاوت ممکن است باعث اتصال در مدار و آسیب دیدن آی‌سی‌های مدار شود.

۳) کدام قسمت سخت‌افزار مسئول کنترل استفاده از Bus توسط منابع است؟

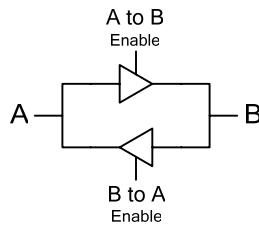
بخشی که کار تولید سیگنال‌های فعال‌ساز را از ورودی کنترل بر عهده دارد، کنترل‌کننده استفاده از Bus است که در اینجا شامل آی‌سی Decoder و گیت‌های تولید‌کننده توابع فعال‌ساز می‌باشد.

۴) شماره آی‌سی‌های TTL و CMOS از نوع 3-state را مشخص کنید.

74125 – 74126 – 74173 – 74240 – 74241 – 74242 – 74243 – 74244 – 74245 – 74251 –
74253 – 74257 – 74258 – 74353 – 74373 – 74374

۵) Bus یک‌طرفه و Bus دوطرفه چیست؟

Bus یک‌طرفه فقط می‌تواند اطلاعات منابع مبدأ را از یک طرف دریافت و به طرف مقصد انتقال دهد، اما یک Bus دوطرفه این قابلیت را دارد که مبدأ و مقصد داده‌ها بتوانند در هر دو طرف خط انتقال قرار داشته باشند. شکل زیر یک مسیر دوطرفه را نشان می‌دهد که توسط دو بافر سه‌حالتی ایجاد شده است. به چنین مدارهایی Transceiver (Transmitter/Receiver) نیز گفته می‌شود:

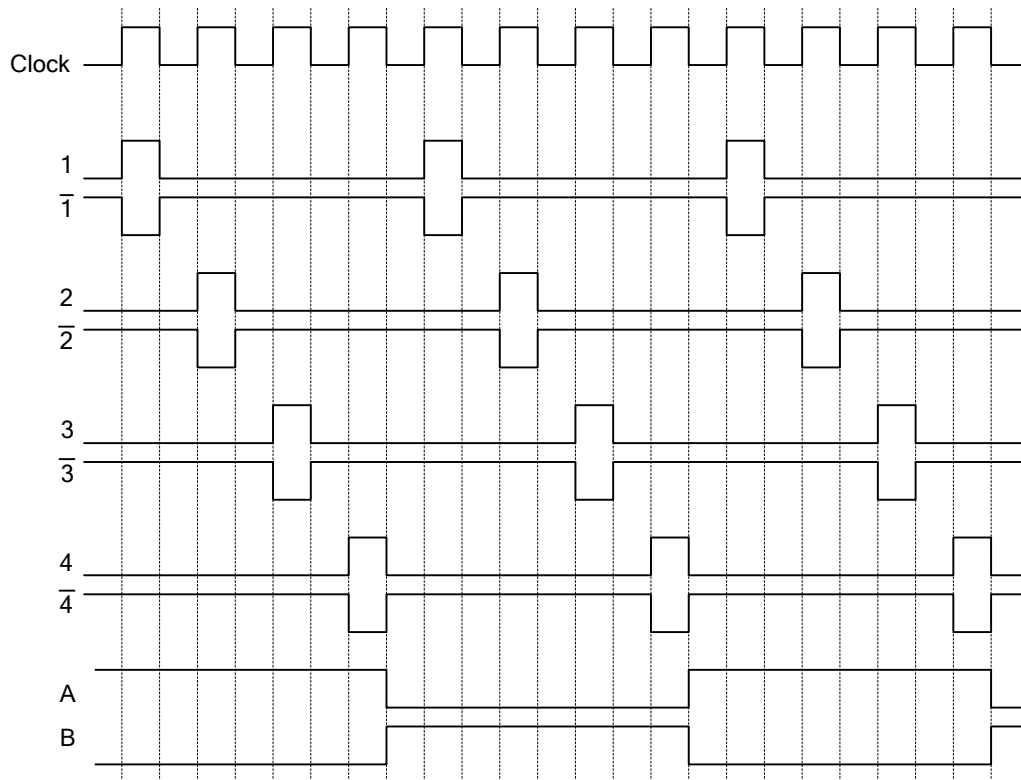


آزمایش شماره چهار:

« طراحی مولد سیگنال کنترل »

تئوری آزمایش:

می‌خواهیم مداری طراحی کنیم که یک پالس Clock را به عنوان ورودی بگیرد و از روی این پالس، سیگنال‌های خروجی 1 ، $\bar{1}$ ، 2 ، $\bar{2}$ ، 3 ، $\bar{3}$ ، 4 ، $\bar{4}$ و A و B را مطابق نمودار زیر تولید کند.

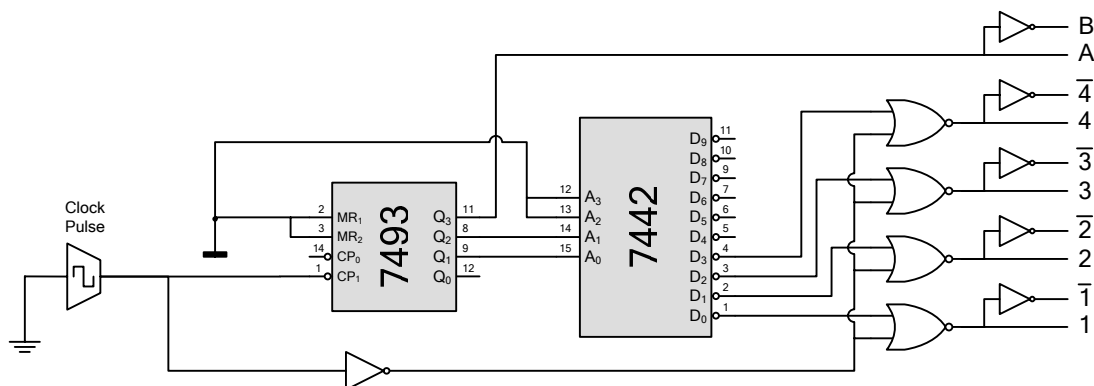


کوچکترین فرکانس خروجی تولید شده (سیگنال‌های A و B) یک هشتم فرکانس Clock است. لذا با استفاده از یک شمارنده ۸ تایی می‌توان این سیگنال را به راحتی تولید نمود (این سیگنال‌ها، بیت پر ارزش این شمارنده و معکوس آن هستند). خروجی‌های 1 ، 2 ، 3 و 4 به ترتیب در پالس اول، دوم، سوم و چهارم Clock، پالس می‌زنند و دوره تناوب آنها ۴ پالس است. در نتیجه این سیگنال‌ها حاصل دیکد کردن یک شمارنده ۴ تایی هستند که می‌توان بجای شمارنده جدید از دو بیت کم‌ارزش‌تر شمارنده ۸ تایی استفاده کرد. چون عرض پالس این چهار خروجی برابر عرض پالس Clock است، این چهار خروجی را باید با سیگنال Clock، AND نمود.

در این آزمایش از آی‌سی 7442 برای دیکدر استفاده شده که خروجی‌های آن معکوس هستند؛ لذا رابطه AND خروجی به رابطه NOR خروجی با معکوس Clock تبدیل می‌شود.

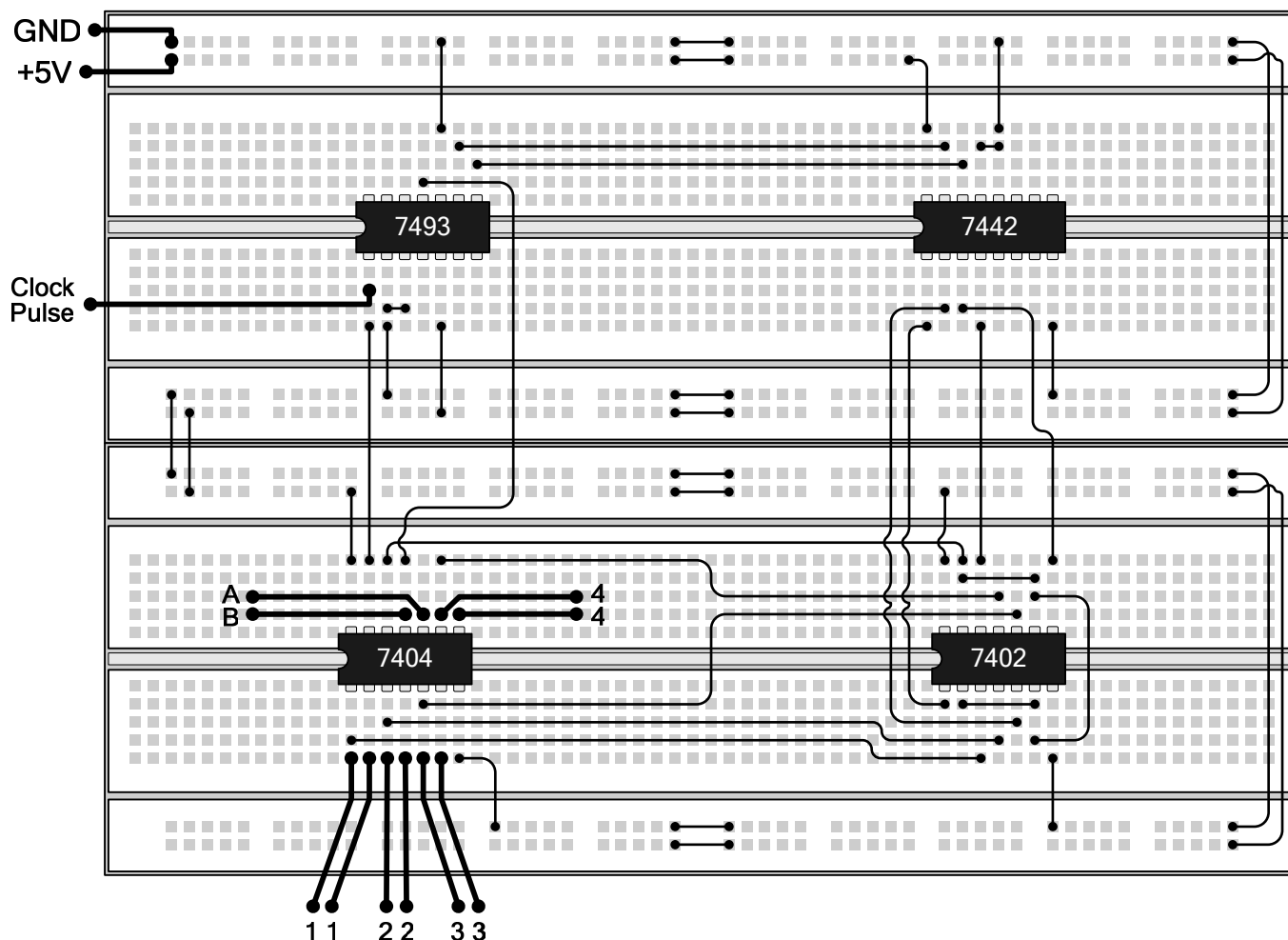
طراحی مدار:

شکل زیر مدار تولید کننده سیگنال های مورد نظر را نشان می دهد:



شکل مدار روی Bread-Board:

شکل زیر سیم بندی عملی یک مدار ضرب کننده چهار بیت در دو بیت را نمایش می دهد:



آزمایش شماره پنج:

« آشنایی با حافظه‌های EPROM »

(تولید سیگنال به روش نرم‌افزاری)

تئوری آزمایش:

می‌خواهیم با استفاده از EPROM مداری طراحی کنیم که شکل موج‌های آزمایش قبل را تولید کند. یک روش برای انجام این کار، بکارگیری یک شمارنده به تعداد دور تناوب سیگنال‌های خروجی است. اگر خروجی این شمارنده به ورودی آدرس EPROM وصل شود، آی‌سی مقادیر موجود در حافظه خود را به ترتیب روی پایه‌های خروجی قرار می‌دهد. با تنظیم درست مقادیر موجود در حافظه (program کردن) می‌توان کاری کرد که داده‌های قرار گرفته روی پایه‌های خروجی با تغییر پالس ساعت، شکل موج‌های مورد نظر را تولید نماید.

دوره تناوب شکل موج‌های آزمایش قبل ۸ پالس ساعت است، لذا می‌توانیم از یک شمارنده ۳ بیتی استفاده کنیم. حال باید بازای هر پالس ساعت تعیین نماییم که شکل موج‌های ما در آن پالس چه مقداری دارند. جدول زیر مقدار هر سیگنال را در ۸ پالس نشان می‌دهد:

CP	Counter	1	1	2	2	3	3	4	4	A	B
0	000	1	0	0	1	0	1	0	1	1	0
1	001	0	1	1	0	0	1	0	1	1	0
2	010	0	1	0	1	1	0	0	1	1	0
3	011	0	1	0	1	0	1	1	0	1	0
4	100	1	0	0	1	0	1	0	1	0	1
5	101	0	1	1	0	0	1	0	1	0	1
6	110	0	1	0	1	1	0	0	1	0	1
7	111	0	1	0	1	0	1	1	0	0	1

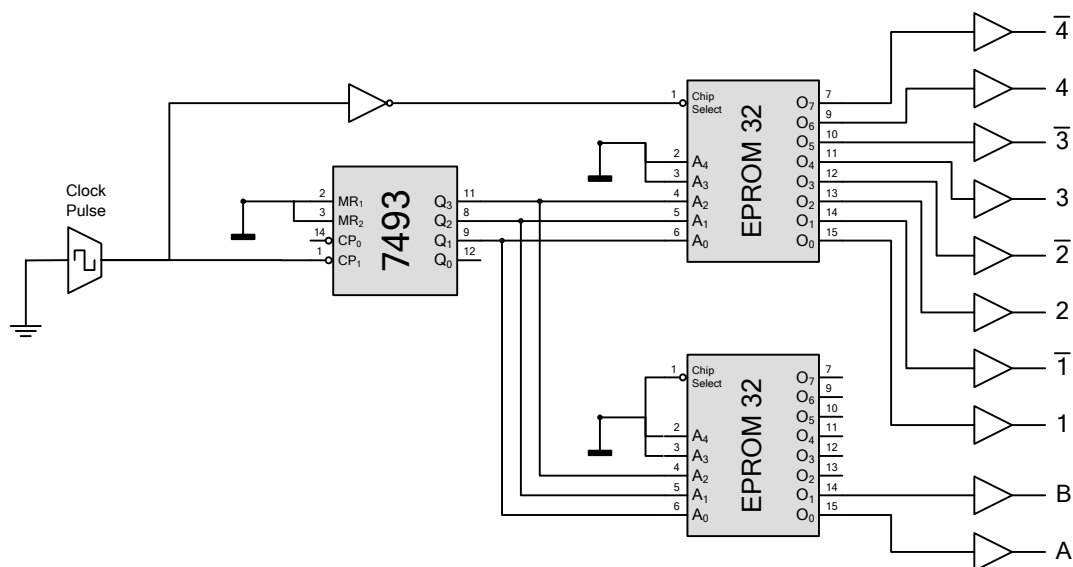
اگر بیت‌های سیگنال‌های 1، 1، 2، 2، 3، 3، 4 و 4 را به ترتیب بیت‌های با ارزش ۰ تا ۷ یک EPROM هشت بیتی و بیت‌های سیگنال‌های A و B را بیت‌های با ارزش ۰ و ۱ یک EPROM هشت بیتی دیگر قرار دهیم، داده‌هایی که باید در EPROM ها program شوند، بدین ترتیب خواهند بود:

Addr	Data	Addr	Data
000	10101001	000	00000001
001	10100110	001	00000001
010	10011010	010	00000001
011	01101010	011	00000001
100	10101001	100	00000010
101	10100110	101	00000010
110	10011010	110	00000010
111	01101010	111	00000010

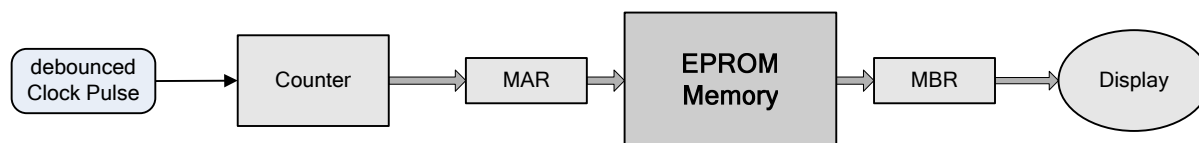
در این آزمایش EPROM های ۳۲ بیتی بکار رفته است که فقط از ۸ بایت آنها استفاده شده. ورودی فعال‌ساز این آی‌سی‌ها به پالس ساعت متصل شده، بدین ترتیب که هنگام ۱ بودن پالس ساعت، خروجی آنها فعال است. شمارنده روی لبه منفی کار می‌کند و درست هنگام ورود EPROM ها به حالت غیر فعال، آدرس را تغییر می‌دهد. فعال بودن خروجی با پالس ساعت باعث گسسته شدن شکل موج‌های معکوس‌ها می‌شود. از آنجاییکه EPROM آنقدر انرژی ندارد که LED به عنوان خروجی به پایه‌های آن وصل شود، باید بین LED ها و IC از بافر استفاده کرد.

طراحی مدار:

شکل زیر مدار تولید کننده سیگنال‌ها را نشان می‌دهد:



شکل زیر بلوک دیاگرام کلی یک مدار تولید کننده سیگنال به روش نرم‌افزاری را با استفاده از EPROM نمایش می‌دهد:



سؤالات:

۱) مدار فوق را با مدار قبل مقایسه و مزایا و معایب آن را شرح دهید.

مدار فوق نسبت به مدار آزمایش قبل از سخت‌افزارهای گران‌قیمت‌تری استفاده کرده است، اما با توجه به پایین آمدن قیمت سخت‌افزار این اختلاف قیمت در برابر افزایش انعطاف پذیری مدار، می‌تواند به صرفه باشد. مدار فوق این قابلیت را دارد که بسادگی برنامه ریزی مجدد گردد و کارکرد آن اصلاح و یا در کاربرد دیگری مورد استفاده قرار گیرد. اما مدار آزمایش قبل قابل تغییر نیست و باید بخش زیادی از آن تعویض گردد.

۲) در بلوک دیاگرام فوق واحدهای MAR و MDR چیست و نقش آنها چه می‌باشد؟

MAR (Memory Address Register) و MBR (Memory Buffer Register) دو رجیستر مرتبط با حافظه EPROM هستند. هنگام خواندن از حافظه ابتدا آدرس مورد نظر (که در اینجا توسط شمارنده تولید می‌شود) درون MAR قرار داده می‌شود. هنگامی که فرمان خواندن به EPROM ارسال می‌شود، EPROM آدرس را از MAR خوانده و محتویات حافظه آن آدرس را در MBR قرار می‌دهد. بدین ترتیب بخش نمایش می‌تواند همیشه از طریق MBR به داده‌ها دسترسی داشته باشد.

۳) حداکثر فرکانس مدار فوق توسط چه عاملی تعیین می‌شود و نیز تعداد شکل‌موج‌های خروجی به چه چیز بستگی دارد؟

حداکثر فرکانس خروجی بستگی مستقیمی به تأخیر خواندن EPROM (و همچنین MBR در صورت استفاده) دارد. حداقل دامنه پالس خروجی ممکن، برابر با زمان این تأخیر می‌باشد. همچنین از آنجایی که هر شکل خروجی در یک بیت حافظه ذخیره می‌شود، حداکثر تعداد سیگنال‌ها برابر تعداد بیت‌های داده است که EPROM قادر به ذخیره کردن آنها می‌باشد. در ضمن طول شکل موج ذخیره شده به ظرفیت EPROM بستگی خواهد داشت.

۴) جهت اضافه کردن امکان تغییر محتویات حافظه چه تغییری می‌بایست در نوع حافظه و نیز طرح مدار ایجاد کرد؟

قطعاً بدین منظور باید از حافظه RAM استفاده نمود و همچنین یک رجیستر دیگر باید به مدار اضافه گردد تا هر بار داده‌های جدید در آن قرار گرفته و با فرمان خواندن، حافظه اطلاعات را از آن رجیستر برداشته و در آدرسی که MAR نشان می‌دهد، ذخیره نماید. بدین ترتیب در صورت نیاز باید در MAR نیز تغییراتی اعمال شود تا غیر از توسط شمارنده نیز بتوان آدرس‌های مورد نظر را درون آن قرار داد.

۵) روش ریز برنامه‌سازی (Micro Programming) را توضیح دهید.

Micro Programming روشی است که در آن عملکرد بخشی از سیستم سخت‌افزاری بجای آنکه توسط گیت‌های منطقی پیاده سازی شود، بصورت یک برنامه ساده نرم‌افزاری پیاده‌سازی و در جایی ذخیره شده و سپس سخت‌افزاری برای اجرای ریز دستورالعمل‌ها (Micro Instruction) پیاده سازی می‌شود. این روش سرعت سخت‌افزار را کمی پایین می‌آورد اما در ازای آن باعث ساده شدن سیستم و همچنین ارزان تمام شدن آن می‌شود.

۶) موارد کاربرد حافظه ROM را در سیستم‌های کامپیوتری نام ببرید.

اصولاً از حافظه ROM در جاهایی استفاده می‌شود که نیاز به ذخیره داده‌هایی است که نیازی به تغییر آنها احساس نمی‌شود یا اصولاً کاربر عادی نباید مجاز به تغییر آنها باشد. از جمله کاربردهای ROM ذخیره‌سازی BIOS در کامپیوترها و برخی برنامه‌های سیستمی و همچنین ذخیره Micro Program ها در میکرو کنترلر یا ریزپردازنده‌ها می‌باشد.

۷) این مدار چه قسمتی از یک واحد سخت‌افزاری را تشکیل می‌دهد؟

این مدار تولیدکننده سیگنال‌های کنترلی یک میکروکنترلر یا یک ریزپردازنده را نشان می‌دهد.

۸) برای طراحی این مدار از چه روش‌هایی می‌توان استفاده کرد؟ مزایا و معایب آن را مشخص کنید.

غیر از روش سخت‌افزاری آزمایش قبل و روش ریزبرنامه‌نویسی فوق، از حافظه RAM نیز می‌توان برای ذخیره‌سازی ریزبرنامه استفاده کرد که نسبت به روش‌های قبل، گران‌تر تمام می‌شود، اما این قابلیت را دارد که شکل‌موج‌ها را بتوان هنگام کار دستگاه، تغییر داد.

۹) آیا هر مداری را می‌توان با اعمال Clock استاتیک امتحان کرد؟ چرا؟

خیر. زیرا ممکن است مدار با پالس Clock خاصی درست کار کند ولی با ایجاد تغییراتی در پالس دچار خطا شود. لذا برای حصول اطمینان از صحت کار مدار، امتحان کردن مدار با Clock استاتیک کافی نیست.

۱۰) چه پارامترهایی حداکثر فرکانس Clock ورودی را تعیین می‌کنند؟

تاخیر گیت‌های شمارنده و رجیستر MAR و همچنین سرعت خواندن داده‌های حافظه EPROM در آدرس تولید شده.

۱۱) مداری طراحی کنید که Clock پالس مناسب برای اعمال به واحدهای سخت‌افزاری بسازد.



آزمایش شماره شش:

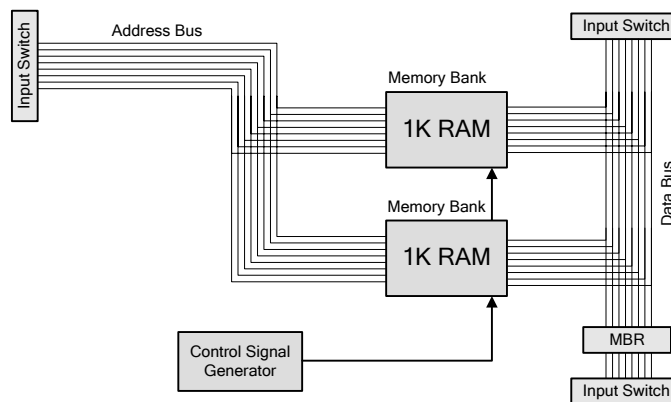
« آشنایی با حافظه‌های RAM »

تئوری آزمایش:

هدف این آزمایش طراحی مداری با چند آی‌سی RAM است که با استفاده از یک Bus آدرس RAM ها به آنها داده شده و ورودی، خروجی و پایه‌های داده RAM ها به یک Bus دیگر (Data Bus) متصل اند. از طریق Address Bus آدرس مورد نظر تنظیم می‌شود. داده‌های ورودی از طریق switch به Bus انتقال یافته و با استفاده از سیگنال‌های کنترلی در آی‌سی RAM مورد نظر ذخیره می‌شود. عمل انتقال و نمایش داده‌ها را می‌توان توسط تولید سیگنال‌های کنترلی مناسب انجام داد.

طراحی مدار:

شکل زیر مدار شماتیک دو آی‌سی 1KB RAM را به همراه Bus آدرس و داده نمایش می‌دهد:



سؤالات:

- ۱) در مدار فوق با خطوط آدرس ۱۶ بیتی، ماکزیمم چه حجمی از حافظه را می‌توان آدرس دهی کرد؟ چگونه؟
به چند عدد IC 6116 نیاز است؟

با خطوط ۱۶ بیتی، $2^{16} = 65536$ کلمه را می‌توان آدرس دهی کرد که اگر کلمه را یک بایت در نظر بگیریم، 64 KB خواهد شد. از آنجاییکه هر IC 6116، 1 KB حافظه دارد، قاعدتاً به ۶۴ آی‌سی نیاز خواهیم داشت، بدین ترتیب که مثلاً از ۱۶ بیت آدرس، ۱۰ بیت سمت راست (کم ارزش‌تر) آدرس درون آی‌سی و ۶ بیت پر ارزش‌تر، خود آی‌سی که آدرس در آن قرار دارد را مشخص کند.

۲) مفهوم Paging و Bank در حافظه را توضیح دهید.

برای افزایش سرعت آدرس دهی و سازمان دهی حافظه، آن را به بخش‌هایی مساوی تقسیم می‌کنند که این بخش‌ها Page نام دارند. طول Page های حافظه توانی از ۲ و مقداری ثابت است. شماره Page و افست آن از روی آدرس فیزیکی بسادگی قابل استخراج است. همچنین در پیاده‌سازی عملی، حافظه را به چند واحد ذخیره‌سازی سخت‌افزاری تقسیم می‌کنند. هر واحد ذخیره‌سازی یک Bank حافظه است. با روش‌های مختلفی می‌توان آدرس‌های حافظه را بین این بانک‌ها توزیع نمود.

۳) اگر بخواهیم محتوای یک بانک حافظه را به ترتیب در بانک دیگری کپی نماییم، به نظر شما چه می‌توان کرد؟

با استفاده از یک شمارنده، تمامی آدرس‌های بانک را تولید کرده و به Bus آدرس بانک‌های مبدأ و مقصد می‌فرستیم. همچنین با استفاده از سیگنال‌های کنترلی در Data Bus، خروجی بانک مبدأ را به ورودی بانک مقصد متصل و آنها را به ترتیب در حالت‌های خواندن و نوشتن قرار می‌دهیم. سرعت این عمل انتقال (فرکانس شمارنده) بستگی مستقیمی به فرکانس کاری حافظه‌ها دارد.

۴) در IC 6116 توسط خطوط آدرس چگونه یک خانه حافظه انتخاب می‌شود؟

پس از قرارگیری آدرس خانه مورد نظر روی خطوط آدرس IC، یک decoder ۱۰ به ۱۰۲۴ در درون آن، سلول‌های مربوط به آن حافظه را فعال می‌کند. بدین ترتیب فرامین خواندن و نوشتن وارد شده به IC روی آن سلول عمل خواهند کرد.

۵) اگر بخواهیم در مدار بالا خطوط data را ۱۶ بیتی کنیم، چه تغییری باید در مدار بدهیم؟

به جای هر IC 6116، دو عدد قرار داده و تمامی ورودی‌ها (فعال‌سازها و آدرس) را به هر دوی آنها می‌دهیم. حال از ۱۶ بیت خطوط Data Bus، هشت بیت را به یک IC و هشت بیت را به دیگری وصل می‌کنیم. بدین ترتیب کلیه عملیات بانک‌های حافظه روی دو آی‌سی اتفاق می‌افتند و بازای یک آدرس، دو کلمه ۸ بیتی داریم که ۱۶ بیت داده ما را تشکیل می‌دهند.

۶) مدار فوق را طوری تغییر دهید که آدرس نیز توسط Data Bus منتقل شود.

۷) Static RAM و Dynamic RAM چه تفاوت‌هایی دارند؟

ذخیره‌سازی اطلاعات روی RAM‌های دینامیک دائمی نیست و برای پایداری اطلاعات، این RAM‌ها باید بطور مداوم refresh شوند. اما در مقابل RAM‌های استاتیک پایدار بوده و نیازی به refresh شدن ندارند.

