

به نام خدا

95101247

مهرسا پوریا

95101322

صبا چرمی مطلق

گزارش کار پروژه

عنوان پروژه:

محاسبه ب.م.م دو عدد داده شده

هدف پروژه:

شبیه سازی و ساختن مدار محاسبه کننده ب.م.م دو عدد هشت بیتی و نمایش حاصل روی سون سگمنت

توضیح منطق و الگوریتم های استفاده شده:

منطق کلی مدار بر اساس الگوریتم مشهور زیر است:

الگوریتم اقلیدس، روشی موسوم به روش نردبانی یا تقسیمات متوالی برای یافتن بزرگترین مقسوم علیه مشترک دو عدد است که در ادامه، با مثالی آن را شرح می دهیم.

عدد بزرگتر را بر عدد کوچکتر تقسیم می کنیم و سپس عدد کوچکتر را بر باقی مانده ی تقسیم مزبور تقسیم می کنیم و این عمل را تا جایی که باقی مانده صفر شود ادامه می دهیم، آخرین باقی مانده غیر صفر، بزرگترین مقسوم علیه مشترک دو عدد مزبور است.

البته در پیاده سازی عملی به علت برتری تفریق بر تقسیم، برای یافتن باقی مانده متوالیاً عدد کوچکتر را از عدد بزرگتر کم می کنیم. اولین عدد حاصل که از عدد کوچکتر، کوچکتر است باقی مانده ی تقسیم است.

خارج قسمت	۴	۶	۱	۴
۸۴۶	۲۰۴	۳۰	۲۴	۶
باقی مانده	۳۰	۲۴	۶	۰

$$۸۴۶ = ۲۰۴(۴) + ۳۰$$

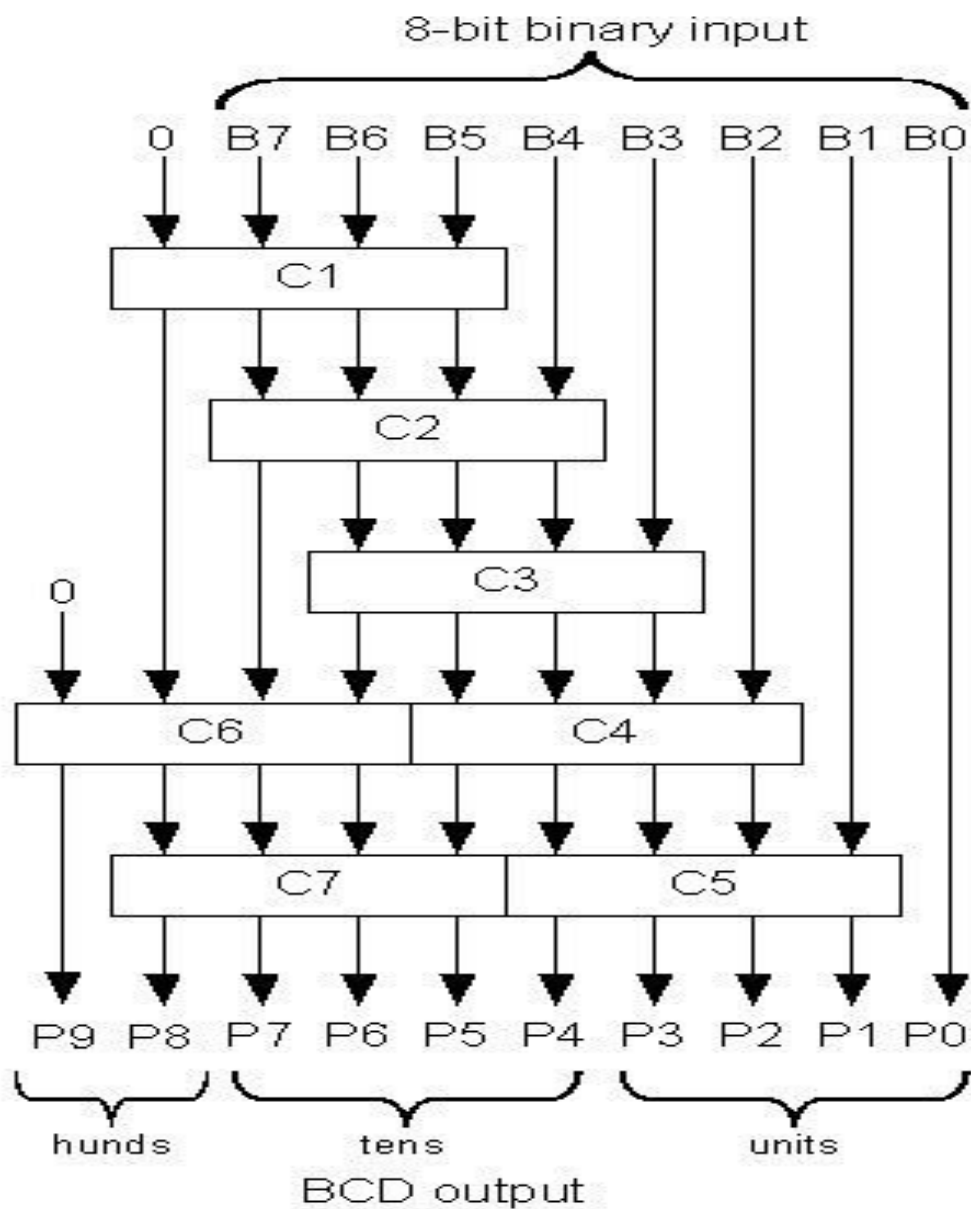
$$۲۰۴ = ۳۰(۶) + ۲۴$$

$$۳۰ = ۲۴(۱) + ۶$$

$$۲۴ = ۶(۴) + ۰$$

الگوریتم قسمت نمایش عدد هشت بیتی توسط سون سگمنت:

بدین منظور باید عدد هشت بیتی باینری را به معادل bcd آن تبدیل کنیم که بزرگترین حالت ممکن ۲۵۵ است که معادل سه رقم bcd است.



که هر کدام از بلوک‌ها عمل زیر را انجام می‌دهند :

اگر عدد ۴ و یا کوچکتر بود خروجی خود عدد است.

اگر عدد بزرگتر از ۴ باشد خروجی ورودی به علاوه ۳ است.

در پایان رقم‌های bcd را در اختیار داریم.

**مراحل عملی و پیاده‌سازی با ماژول‌های منطقی:**

با توجه به الگوریتم توضیح داده‌شده در بخش قبل متوجه می‌شویم که در مدار شاهد تفریق‌های متوالی هستیم؛ که به این منظور مدارمان را با کلاک هماهنگ می‌کنیم در پیاده‌سازی این عمل از رجیسترهای موازی-موازی استفاده می‌کنیم. دو عدد از

این حافظه‌ها در مدار داریم که در یکی حاصل عدد کوچکتر ورودی هر چرخه و در دیگری قدر مطلق اختلاف عدد بزرگتر و کوچکتر هر چرخه نگهداری می‌شود.

حال به بیان دقیق‌تر ورودی‌های هر حافظه می‌پردازیم :

ورودی‌های رجیستری که مسئول نگهداری عدد کوچکتر است خروجی‌های یه مالتی پلکسر ۲ به ۱ هشت‌بیتی است که سلکت آن به خروجی بزرگتر بودن A از B بلوک مقایسه‌کننده‌ی هشت‌بیتی است و ورودی‌های متناظر صفر آن به عدد A است و یک آن به عدد B است. تا همیشه عدد کوچکتر به ورودی این رجیستر وارد شوند و در هر کلاک به خروجی انتقال یابند تا در چرخه‌ی بعدی مورد استفاده قرار گیرند.

رجیستر دوم مسئول نگهداری قدر مطلق اختلاف دو عدد ورودی هر چرخه است.

به این منظور ورودی‌های آن خروجی یک مالتی پلکسر ۲ به ۱ هشت‌بیتی است که سلکت آن همان سلکت ماکس گفته شده در بالاست و ورودی متناظر صفر آن B-A و متناظر یک آن A-B است.

برای پیاده سازی عملی رجیسترهای بالا از ۴ عدد آی‌سی 19574s که یک شیفت‌رجیستر موازی-موازی چهاربیتی است استفاده کردیم که ورودی‌های آن مطابق دیتاشیت پیوست شده، متصل شده اند و مود مورد استفاده نیز به صورت زیر است :

همچنین برای ایجاد کلید ریست در مدار پایانه Clear هر کدام از شیفت رجیستر ها به هم متصل شده‌اند و به وسیله یک گیت نات و کلید به زمین متصل شده اند.

که با بستن کلید ب.م.م محاسبه و با باز شدن آن عدد صفر روی سون‌سگمنت نمایش داده می‌شود.

INPUTS					OUTPUTS				
CLEAR	SHIFT/ LOAD	CLOCK	SERIAL		PARALLEL				$\bar{Q}_D$
			J	$\bar{K}$	A	B	C	D	
L	X	X	X	X	X	X	X	X	L
H	L	↑	X	X	a	b	c	d	$\bar{d}$

برای هر کدام از مالتی پلکسر های مدار نیز از ای سی 74ls157 استفاده کردیم که دیتاشیت این ای سی نیز پیوست شده است.

خروجی های واحد های حافظه گفته شده به ابتدای مدار باز می گردند و توسط یک مالتی پلکسر در ابتدای مدار تعیین میشود که ورودی های هر چرخه ورودی های جدیدی که کاربر میخواهد ب.م.م آنها را محاسبه کند باشد یا در مراحل محاسبات قرار داشته باشیم و مدار از حافظه ی مدار اطلاعات بخواند. بدین منظور باید سلکت ماکس مناسبی استفاده کنیم. خروجی های رجیستر را به ورودی های متناظر یک ماکس و ورودی مدار که همان اعداد مورد نظر کار بر است را به سلکت صفر آن متصل می کنیم.

ما فقط پس از بستن کلید ریست تمایل داریم مدار از کاربر ورودی بگیرد یعنی زمانی که خروجی های هر دو واحد حافظه گفته شده صفر باشند سلکت ماکس ابتدایی صفر باشد و در بقیه حالات سلکت یک شود.

به این منظور بیت به بیت ۸ بیت حافظه اول را با هم nor می کنیم. و همین کار را با حافظه دوم انجام می دهیم و حاصل دو فرایند را با هم or می کنیم. حال این مقدار به دست آمده هنگامی که در حداقل یکی از واحد ها صفر باشد مقدار یک به خود میگیرد.

اما این هنوز برای سلکت ماکس اولیه مناسب نیست. پس از ریست در هر دو واحد حافظه صفر ذخیره شده و مقدار گفته شده در بالا یک میشود که مناسب است که با یک نات از این لحاظ سلکت مناسبی است اما در آخرین مرحله محاسبه ی ب.م.م نیز در یکی از واحد ها عدد صفر قرار میگیرد که باعث یک شدن مقدار بالا می شود و نامطلوب است. بنابراین باید این اشکال را رفع کنیم.

به خروجی های رجیستر ها باز می گردیم. میدانیم که اگر خروجی یکی از حافظه ها صفر شد عدد حافظه ی دیگر باید به عنوان ب م در خروجی ظاهر شود.

بدین منظور خروجی های حافظه ی مسئول تفاضل را به سلکت متناظر صفر و خروجی های واحد حافظه عدد کوچکتر هر چرخه را به سلکت یک یک ماکس نهایی می دهیم که خروجی آن ب.م.م است. حال باید برای این ماکس نیز سلکت مناسب انتخاب کنیم.

که به صورت زیر محاسبه می شود :

محاسبه ی Select ، mux نهایی :

or بیت بیت حافظه تفاضل :  $x$  ( 0 )  
 مدلولیت :  $y$  ( 1 )

$x$	$y$	$f$
0	0	x
0	1	1
1	0	0
1	1	x

$\Rightarrow f = \bar{x}y$

Select ، mux نهایی and Nor بیت بیت  
 حافظه ی مدلولیت و Or حافظه ی مدلولیت است.  
 اختلاف مدلولیت

وقتی رجیستر ها ریست می شوند محتویات هر دو صفر میشوند بنابراین به ناچار خروجی ماکس که همان خروجی مدار است صفر میشد و در غیر این صورت قطعا صفر نمی شود. پس or تک به تک ۸ بیت خروجی فقط در هنگام ریست است که صفر میشود. با توجه به این اشکال سلکت ماکس اولیه مدار به این صورت رفع می شود :

محاسبه ی Select ، mux اول :

$x$  : or ، Nor های bitwise دو واحد حافظه  
 $y$  : Nor خروجی mux نهایی ( bitwise )

$x$	$y$	$f$
0	0	1
0	1	1
1	0	1
1	1	0

$f = \bar{x}y$

Select ، mux ابتدای Nand  $x$  و  $y$  است  
 که به حافظه سه است

برای ساختن تفریق‌کننده از فول ادر (آی سی 8374 Is) استفاده کردیم. برای مثال برای ساختن A-B از دو فول ادر استفاده کرده Cin آی سی اول را به Vcc و Cout آن را به Cin آی سی بعد متصل کردیم سپس ۴ بیت کم ارزش A را به ورودی‌های متناظر با ورودی A آی سی اول و چهار بیت بعدی آن را به ورودی‌های متناظر با ورودی A آی سی دوم وصل می‌کنیم. سپس توسط آی سی 0474 Is بیت های عدد B را تک به تک نات کرده و سپس چهار بیت کم ارزش حاصل را به چهار بیت ورودی متناظر با B آی سی اول و چهار بیت بعدی آن را به ورودی‌های متناظر با B آی سی بعدی وصل می‌کنیم. حاصل A-B می‌باشد.

قابل ذکر است که تمام محاسبات به صورت متمم یک می‌باشند.

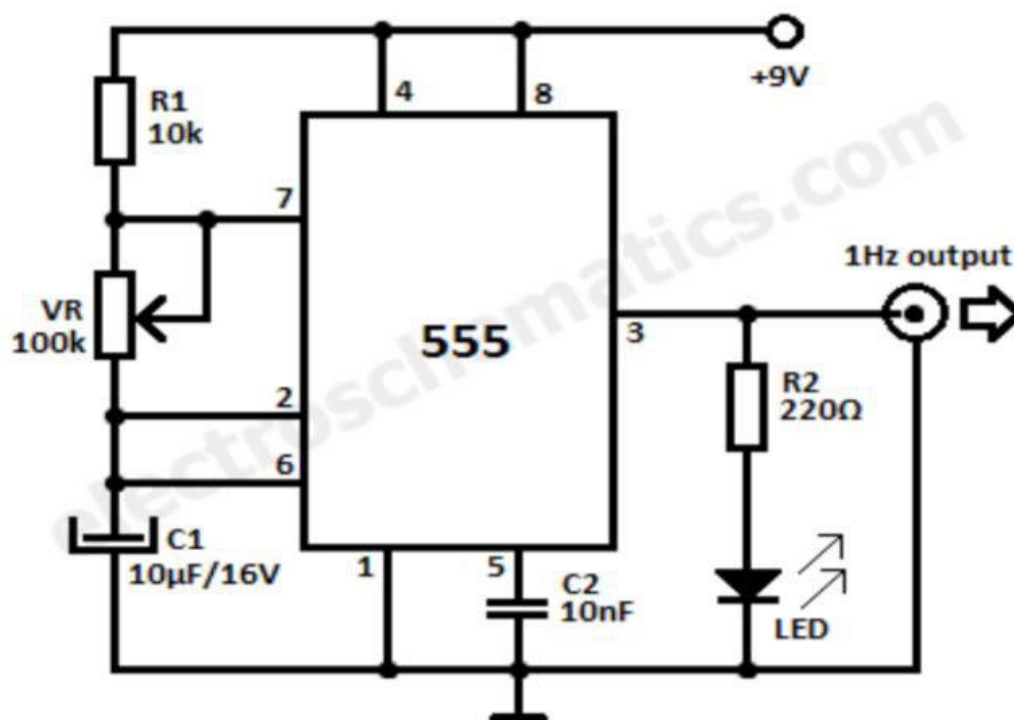
قابل ذکر است که برای مقایسه دو عدد A و B از آی سی 574 Is و cascade آنها استفاده کردیم.

مدار ب م م اکنون کامل است

حال نحوه نمایش خروجی عدد هشت بیتی بر روی سون سگمنت را شرح می‌دهیم.

الگوریتم استفاده شده در بخش های قبل گفته شده است. بلوک مورد بحث را با استفاده از یک ماکس که سلکت آن خروجی یک مقایسه گر است که عدد ۴ بیتی وارد شده به آن بلوک را با ۴ مقایسه می‌کند و متناسب با آن در خروجی یا خود عدد یا به وسیله ی یک فول ادر جمع یافته ان با سه در خروجی ظاهر می‌شود ، می‌سازیم. بدین ترتیب bcd عدد هشت بیتی را در اختیار داریم. به وسیله ی انکودر bcd به سون سگمنت و خود سون سگمنت خروجی خواسته شده را نمایش می‌دهیم.

برای تولید کلاک از آی سی 555 استفاده کردیم مدار تولید کلاک به صورت زیر است:

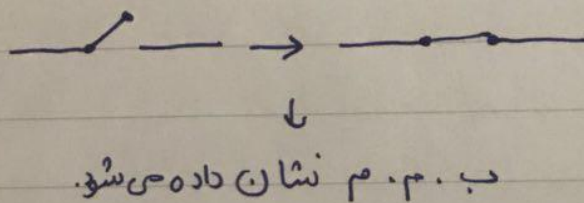
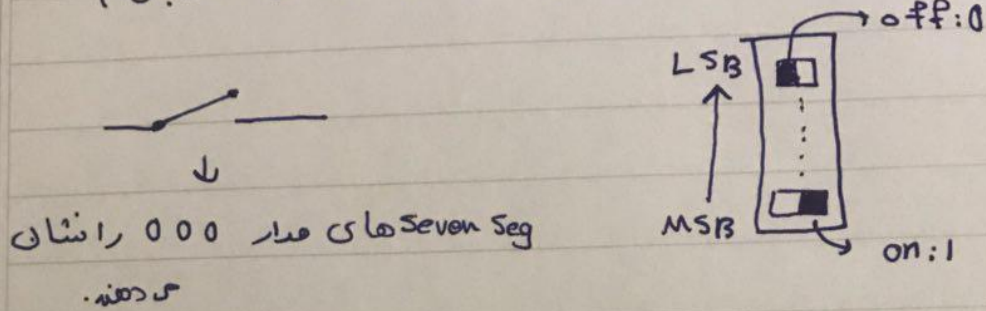


تایمینگ مدار :

مدت زمان لازم برای محاسبه ی ب.م.م به فرکانس کلاک استفاده شده و همچنین تعداد مراحل تا وقتی که حاصل تفاضل یک مرحله صفر شود بستگی دارد.

نحوه اجرای فایل شبیه سازی proteous :

ورودی ها توسط dip switch ها گرفته می شوند. (در مبنای ۲)



مثال : ب. م. م. ۲۷ و ۶۳

۲۷: ۰۰۰ ۱۱۰۱۰

۶۳: ۰۰ ۱۱۱۱۱۱

	پایین register 1	پایین register 2
reset ←	۰	۰
	۳۶	۲۷
	۹	۲۷
	۱۸	۹
	۹	۹
	۵	۹

فروچی شبیه سازی : عکس بعدی

ب. م. م. ۹

Scb6

DATE



