گزارش کار بخش A آزمایش سوم – آزمایشگاه ریزپردازنده

نگار هنرور صدیقیان – 99243076

امین احسانی­مهر – 99243009

سوالات تحلیلی:

**سوال یکم:**

مهم­ترین کاربرد تراشه­های تایمر، ایجاد شکل موج­های مختلف است که با 0 و 1 کردن سیگنال خروجی تایمر، می­توان این موج­های متفاوت را ساخت. برای ساخت این تراشه­ها، از اتصال پشت سر هم عناصری همانند flip flopها و تولید دوره تناوب­های مختلف با استفاده از یک دوره تناوب ثابت اولیه می­توان بهره برد. از این سیگنال­ها می­توان در warning lightها که چراغی روشن و خاموش می­شود استفاده کرد. همچنین از این سیگنال­ها می­توان برای تولید نوت­های موسیقی در یک فرکانس خاص و یا کنترل servo deviceها استفاده کرد.

**سوال دوم:**

به طور کلی 8253 شش مد مختلف دارد که این مدها به شرح فوق هستند:

1. interrupt on terminal count (mode 0): در این حالت، سیگنالی با طول پالس مشخص تولید می­شود. در این مود، بعد از وارد کردن مود ورودی که برابر 0 است، سیگنال خروجی از High به Low تغییر پیدا می­کند و با دادن counter مدنظر N، این سیگنال به مدت N کلاک در حالت Low می­ماند و بعد از مدت N، سیگنال خروجی از Low به High تغییر پیدا می­کند. در این مود برای تغییر سیگنال، نیاز به تغییر سیگنال gate نداریم (gate صرفا برابر 1 باشد، به صورت خودکار سیگنال out برابر 0 می­شود). این حالت توسط نرم افزار کنترل می­شود. سیگنال gate در این مود نقش pause کردن عملیات را دارد؛ یعنی زمانی که gate برابر 0 می­شود، کلاک زدن باعث شمردن counter نمی­شود و counter در مقدار خود باقی می­ماند (مثلا اگر counter 4 بود و 2 کلاک گذشته بود، زمانی که سیگنال gate فعال شود، این عدد همان 2 خواهد ماند). این مود به صورت خودکار تکرار نمی­شود.

2. Programmable one-shot (mode 1): این حالت هم مشابه حالت قبل، سیگنالی با طول پالس مشخص تولید می­کند؛ با این تفاوت که زمانی شروع به تغییر سیگنال خروجی می­کند که سیگنال gate یک کلاک بخورد (از Low به High تبدیل شود و دوباره Low گردد) پس به صورت خودکار سیگنال خروجی تغییر نخواهد کرد. این مود توسط سخت افزار کنترل می­شود و همچنین به صورت خودکار تکرار نمی­شود.

3. Rate generator (mode 2): در این حالت، پس از وارد کردن مود ورودی که برابر 2 است، سیگنال out برابر 1 می­شود و شروع به کاهش counter می­کند. زمانی که counter به 1 رسید، در کلاک بعد سیگنال خروجی از High به Low تغییر می­کند و پس از این کلاک، counter دوباره مقدار آن برابر مقدار counter ورودی داده شده می­شود. پس در نتیجه در این حالت، سیگنال خروجی تکرار می­شود.

4. Square wave rate generator (mode 3): این حالت تا حدودی مشابه حالت قبلی است؛ با این تفاوت که سیگنال خروجی به صورت مداوم و با طول یکسان 0 و 1 می­شود (یعنی مثلا 2 کلاک 0 است و دو کلاک 1). زمانی که counter ورودی زوج باشد، در هر کلاک مقدار counter دوتا دوتا کم می­شود و زمانی که counter به 2 رسید، در کلاک بعد مقدار آن به مقدار اولیه ریست می­شود و سیگنال خروجی تغییر می­کند (از High به Low یا برعکس). اگر counter ورودی فرد باشد، از مقدار counter یکی کم می­کنیم و مراحل بالا را تکرار می­کند. تنها یک تفاوت دارد؛ آن هم زمانی است که مقدار سیگنال خروجی برابر 1 است. در این حال زمانی که counter برابر 0 شد، در کلاک بعدی مقدار اولیه ریست می­شود و سیگنال خروجی از High به Low تغییر می­کند. این سیگنال مرتباً 0 و 1 می­شود؛ پس در نتیجه به صورت خودکار تکرار می­شود.

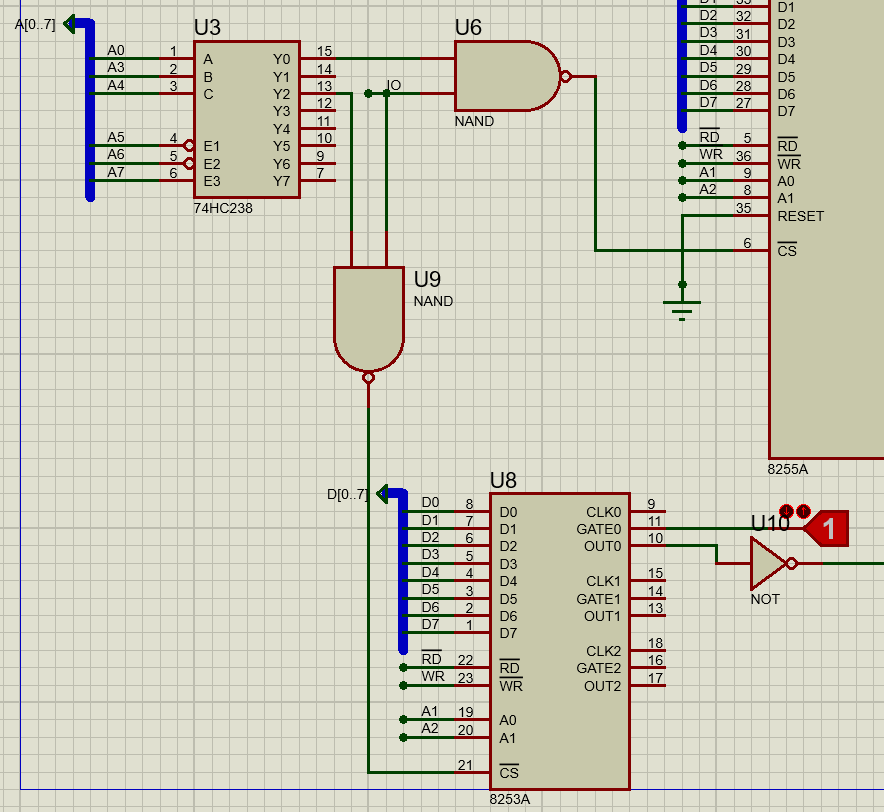
5. Software triggered strobe (mode 4): این مود تا حدودی مشابه مود 0 سیگنال one shot درست می­کند؛ با این تفاوت که برخلاف مود 0 تا زمانی که counter برابر 0 نشده است، سیگنال out برابر 1 است و زمانی که counter برابر 0 شد، سیگنال خروجی از 0 به 1 تغییر می­کند و در کلاک بعدی، counter روی عدد FF ست می­شود و خروجی سیگنال دوباره 1 می­شود. این عملیات مشابه حالت 0 توسط نرم افزار کنترل می­شود و به صورت خودکار تکرار نمی­شود.

6. Hardware triggered strobe (mode 5): این مود از جهت کارکرد مشابه حالت قبلی است؛ با این تفاوت که در مود قبلی به صورت خودکار شروع به شمردن counter می­کند اما در این مود همانند مود 1، زمانی که سیگنال gate یک کلاک بخورد (از Low به High تبدیل شود و دوباره Low گردد) شروع به شمردن counter می­کند.

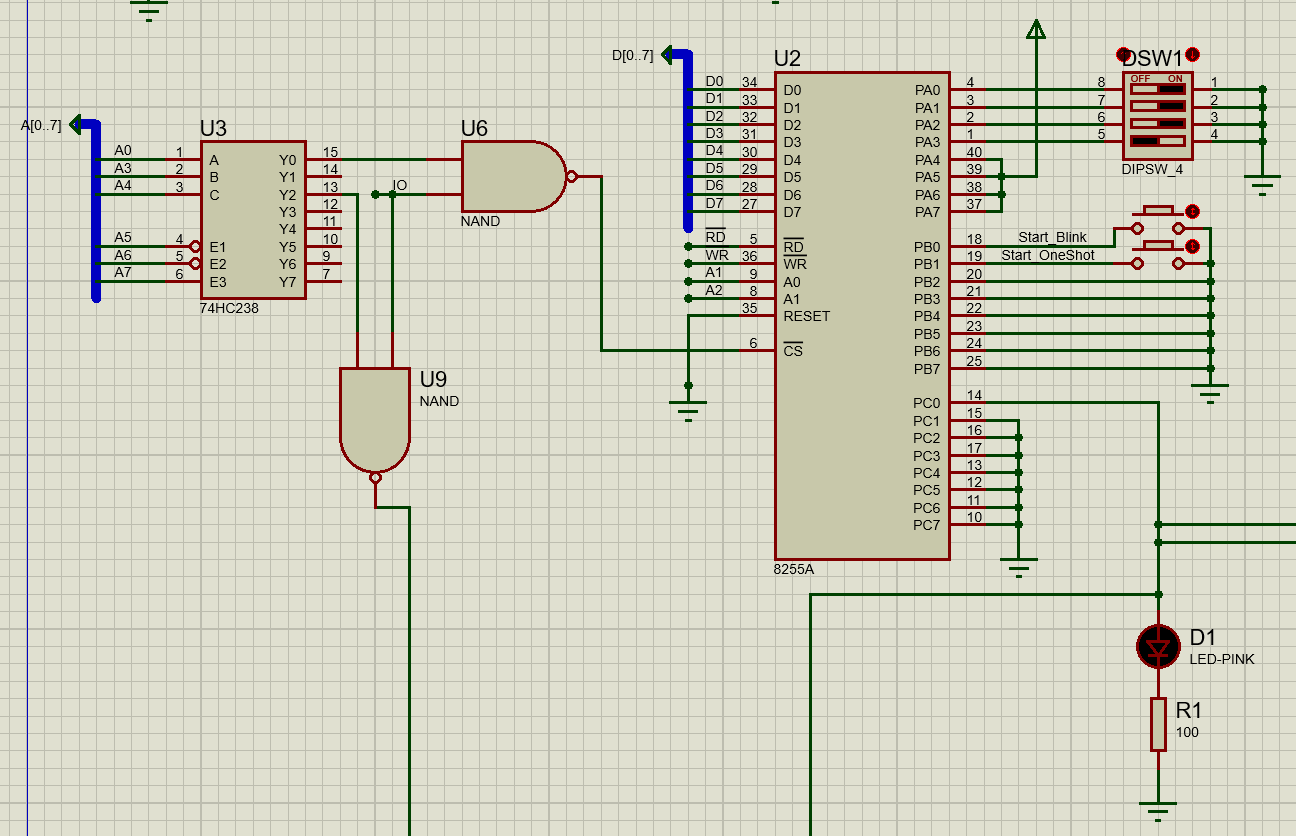
تمامی حالات بالا زمانی سیگنال خروجیشان تغییر می­کند که سیگنال gate برابر 1 باشد. در تمامی مودها بجز 1 و 5 زمانی که سیگنال gate برابر 0 می­شود، روند خروجی دادن متوقف می­شود. حال در برخی از مودها مانند مود 0 یا 3 سیگنال خروجی با صفر شدن سیگنال gate تغییر نمی­کند و در برخی مودها مانند مود 2 با صفر شدن سیگنال gate مقدار سیگنال خروجی برابر 1 می­شود. در مودهای 1 و 5، صفر شدن سیگنال gate باعث می­شود تا زمانی که دوباره مقدار آن برابر 1 شد، تایمر شروع به تولید سیگنال one shot مدنظر بکند.

گزارش دستور کار:

ماهیت این آزمایش، همانند بخش سوم قسمت B آزمایش پیشین است؛ با این تفاوت که بجای استفاده از یک حلقه درون کد که با استفاده از یک عدد بزرگ برنامه را متوقف می­کنیم، از تراشه 8253 استفاده می­کنیم.

همانند تراشه 8255، تراشه 8253 را روی یک پورت ست میکنیم (در این کد روی پورت 8E ست شده است).

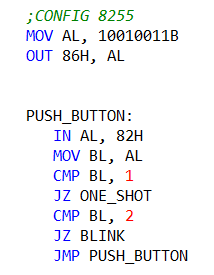
شکل 1 – تراشه 8253 و آدرس دیکود شده آن (8Eh)

تراشه 8255 نیز روی پورت 86 ست شده است. پورت A تراشه 8255 یک dip switch است که عدد مد نظر را آنجا وارد می­کنیم. پورت B تراشه دارای دو کلید است که یکی مربوط به سیگنال Blink و دیگری مربوط به سیگنال One shot می­شود و پورت C تراشه سیگنال خروجی از تایمر را نشان می­دهد که این سیگنال به لامپ ما متصل می­شود.

شکل 2 – تراشه 8255 به همراه پورت­های ورودی و خروجی آن و آدرس دیکود شده متناظر آن (86h)

مطلوب است مدار ما به صورت زیر عمل کند:

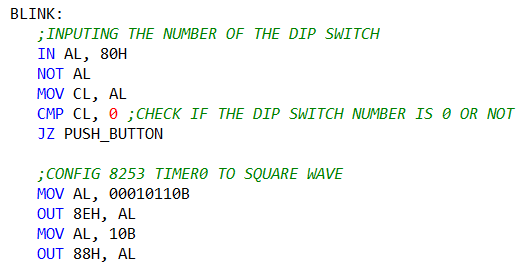
* کاربر یک عدد روی dip switch ست می­کند و یکی از دکمه­ها را فشار می­دهد.
* با فشار دادن یکی از دکمه­ها، عدد روی dip switch خوانده شود و به پردازنده داده شود.
* اگر کلید start\_blink زده شده بود، setup تایمر اول را روی حالت blinking (square wave rate generator) که عدد آن برابر 3 است، قرار می­دهیم و سپس با دادن عددی که طول تناوب خروجی را ست می­کند، چراغ شروع به روشن و خاموش شدن کند و زمانی که به تعداد N بار (عدد خوانده شده از dip switch) روشن شد، چراغ به طور کامل خاموش شود.
* اگر کلید start\_OneShot زده شد، setup تایمر اول را روی حالت OneShot (interrupt on terminal count) که عدد آن برابر 0 است، قرار می­دهیم و سپس با دادن عدد N به تایمر اول، چراغ به مدت N ثانیه روشن شده و سپس خاموش شود. همچنین اگر در حینی که چراغ روشن بود، کلید oneshot دوباره زده شد، مقدار روشن بودن چراغ N+M ثانیه شود (M: عدد جدید خوانده شده از dip switch)

حال وارد بخش source code می­شویم.

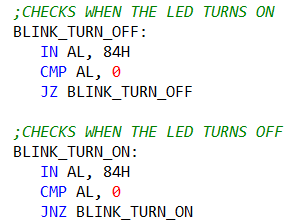
در ابتدا، setup تراشه 8255 را ست می­کنیم (که هر سه پورت آن ورودی هستند و روی مود I/O هستیم و mode selectionها را نیز برابر 0 قرار می­دهیم) و وارد بخش push\_button می­شویم.

در این بخش، منتظر می­مانیم تا کاربر یکی از دکمه­ها را فشار بدهد. اگر بعد از فشردن دکمه، عدد پورت برابر 2 بود، بدان معنی است که کلید blink زده شده و باید وارد بخش blink شویم. اگر عدد پورت برابر 1 شد، بدان معنی است که دکمه one shot زده شده و باید وارد بخش one shot شویم. در غیر این صورت درون این بخش می­مانیم تا بلاخره کلیدی زده شود. حال هر دو حالت را بررسی می­کنیم:

شکل 3 – کانفیگ 8255 و بخش push\_button

1. مود blink: زمانی که وارد این بخش می­شویم، ابتدا عدد روی dip switch را می­خوانیم و درون یک رجیستر قرار می­دهیم. در صورتی که عددمان برابر 0 بود، به همان push\_button برمیگردیم (چون انگار گفتیم 0 ثانیه چشمک بزند؛ پس چراغ نباید روشن شود). حال تایمر اول این تراشه را روی مود 3 قرار می­دهیم و وارد حلقه B\_TURN\_OFF می­شویم.

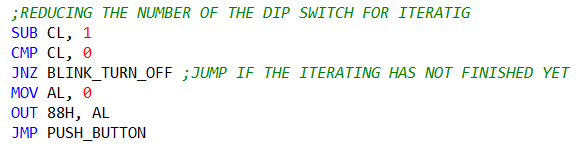
شکل 4 – بخش خواندن عدد dip switch و کانفیگ کردن تایمر اول تراشه 8253 بر روی حالت blink

در این حلقه، بیت خروجی از تایمر را چک میکنیم و مادامی که بیت ورودی به لامپ برابر 1 نشده بود، درون این حلقه میمانیم. در حقیقت زمانی از این حلقه بیرون می­آییم که LED روشن شده باشد. حال وارد حلقه B\_TURN\_ON می­شویم.

همانند حلقه قبلی، بیت خروجی تایمر را چک میکنیم و این بار تا زمانی که بیت ورودی به LED برابر 0 نشده بود، درون این حلقه میمانیم. در حقیقت زمانی از این حلقه بیرون می­آییم که LED خاموش شده باشد.

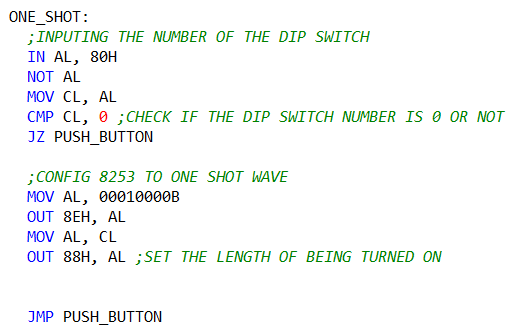
شکل 5 – حلقه­های turn on (برای چک کردن اینکه LED چه زمانی خاموش می­شود) و turn off (برای بررسی اینکه LED چه زمانی خاموش می­شود)

­

زمانی که این حلقه تمام می­شود، در حقیقت یکبار چراغ ما خاموش و روشن شده است. پس از مقدار Nمان یکی کم میکنیم (به فرض قرار بوده 4 بار روشن خاموش شود و حالا یکبار انجام شده؛ در نتیجه باید 3 بار دیگر این عمل را تکرار کنیم) و تا زمانی که مقدار N (که در رجیستر CL نگهداری می­شود) برابر 0 نشده است، درون این بخش میمانیم. زمانی که مقدار CL برابر 0 شد، طول دوره تناوب چشمک زدن را برابر 0 می­کنیم (که LED را کاملا خاموش نگه دارد) و دوباره به بخش push\_button می­رویم و منتظر کلیدزنی مجدد میمانیم.

شکل 6 – بخش کم کردن مقدار CL و بررسی صفر بودن و یا نبودن آن

نکته: می­توانستیم ورودی لامپ را به تراشه 8255 وصل نکنیم و در نتیجه دو حلقه B\_TURN\_OFF و B\_TURN\_ON را نداشته باشیم اما مشکلی که پدید می­آید، این است که مود square wave در تراشه 8253 تا بینهایت این عملیات روشن و خاموش کردن را ادامه می­دهد و نمی­توانیم بگوییم که مثلا بعد 4 بار تکرار لامپ را کلا خاموش نگه دارد. بنابراین باید دفعات تکرار روشن و خاموش شدن را با وصل کردن بیت ورودی LED و مدیریت آن به وسیله دو حلقه ذکر شده، انجام داد.

2. مود One Shot: همانند مود قبلی، زمانی که وارد این بخش می­شویم، ابتدا عدد dip switch را می­خوانیم و در صورتی که عدد dip switch برابر 0 بود، به بخش push\_button برمی­گردیم. حال تایمر اول را روی مود 0 قرار می­دهیم و سپس عدد خوانده شده را به تایمر می­دهیم تا به مدت N ثانیه چراغ روشن باشد و بعد از آن خاموش شود. این بخش حلقه روشن و خاموش ندارد و بعد از وارد کردن عدد ورودی در تایمر، به حلقه push\_button برمی­گردیم تا اگر کاربر در حین روشن بودن LED کلید one shot را دوباره فشرد، دوباره به بخش one shot برویم و مقدار روشن بودن LED به مدت M (عدد جدید وارد شده) تمدید شود. به عنوان مثال، اگر عدد ورودی اول 5 ثانیه بود و پس از گذشت 2 ثانیه کلید one shot دوباره فشرده شد و عدد جدید برابر 4 بود، میزان روشن بودن آن برابر 2 + 4 = 6 ثانیه می­شود.

شکل 4 – قطعه کد مربوط به بخش one shot

نکته: در این بخش برخلاف حالت Blink، نیازی به بررسی خروجی تایمر نداریم چرا که خود تایمر بعد مدت N ثانیه روشن بودن، خاموش می­شود. پس دیگر نیازی به زدن دو حلقه Turn on و turn off نداریم.