

گزارش کار بخش A آزمایش سوم – آزمایشگاه ریزپردازنده

نگار هنرور صدیقیان – ۹۹۲۴۳۰۷۶

امین احسانی مهر – ۹۹۲۴۳۰۰۹

سوالات تحلیلی:

سوال یکم:

مهم‌ترین کاربرد تراشه‌های تایمر، ایجاد شکل موج‌های مختلف است که با ۰ و ۱ 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 ۸۲۵۳ شش مد مختلف دارد که این مدها به شرح فوق هستند:

۱. (mode 0) interrupt on terminal count: در این حالت، سیگنالی با طول پالس مشخص تولید می‌شود. در این مود، بعد از وارد کردن مود ورودی که برابر ۰ 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 صرفاً برابر ۱ باشد، به صورت خودکار سیگنال out برابر ۰ می‌شود). این حالت توسط نرم افزار کنترل می‌شود. سیگنال gate در این مود نقش pause کردن عملیات را دارد؛ یعنی زمانی که gate برابر ۰ می‌شود، کلاک زدن باعث شمردن counter نمی‌شود و counter در مقدار خود باقی می‌ماند (مثلاً اگر counter ۴ بود و ۲ کلاک گذشته بود، زمانی که سیگنال gate فعال شود، این عدد همان ۲ خواهد ماند). این مود به صورت خودکار تکرار نمی‌شود.

۲. (mode 1) Programmable one-shot: 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.

۳. (mode 2) Rate generator: در این حالت، پس از وارد کردن مود ورودی که برابر ۲ است، سیگنال out برابر ۱ می‌شود و شروع به کاهش counter می‌کند. زمانی که counter به ۱ 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.

۴. (mode 3) Square wave rate generator: این حالت تا حدودی مشابه حالت قبلی است؛ با این تفاوت که سیگنال خروجی به صورت مداوم و با طول یکسان ۰ و ۱ می‌شود (یعنی مثلاً ۲ کلاک ۰ است و دو کلاک ۱). زمانی که counter ورودی زوج باشد، در هر کلاک مقدار counter دوتا دوتا کم می‌شود و زمانی که counter به ۲ 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 ۱ است. در این حال زمانی که counter برابر ۰ شد، در کلاک بعدی مقدار اولیه ریست می‌شود و سیگنال خروجی از High به Low تغییر می‌کند. این سیگنال مرتباً ۰ و ۱ می‌شود؛ پس در نتیجه به صورت خودکار تکرار می‌شود.

۵. Software triggered strobe (mode 4): این مود تا حدودی مشابه مود ۰ سیگنال one shot درست می‌کند؛ با این تفاوت که برخلاف مود ۰ تا زمانی که counter برابر ۰ نشده است، سیگنال out برابر ۱ است و زمانی که counter برابر ۰ شد، سیگنال خروجی از ۰ به ۱ تغییر می‌کند و در کلاک بعدی، counter روی عدد FF ست می‌شود و خروجی سیگنال دوباره ۱ می‌شود. این عملیات مشابه حالت ۰ توسط نرم افزار کنترل می‌شود و به صورت خودکار تکرار نمی‌شود.

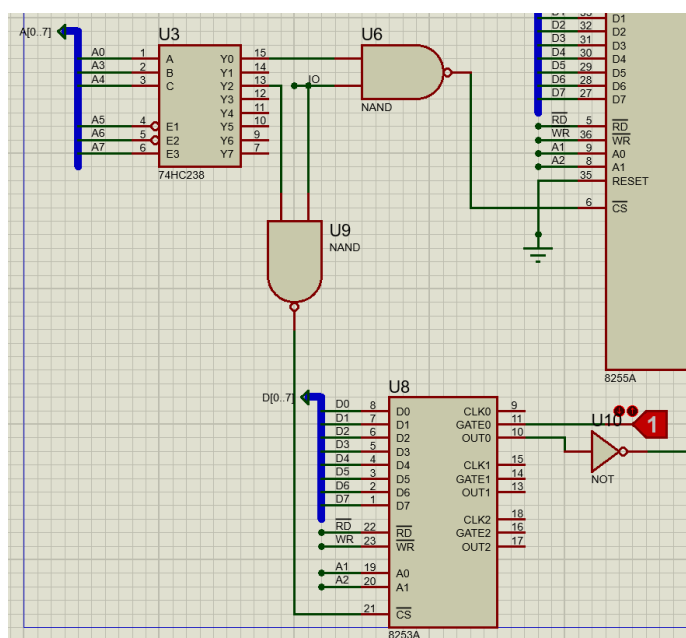
۶. Hardware triggered strobe (mode 5): این مود از جهت کارکرد مشابه حالت قبلی است؛ با این تفاوت که در مود قبلی به صورت خودکار شروع به شمردن counter می‌کند اما در این مود همانند مود ۱، زمانی که سیگنال gate یک کلاک بخورد (از Low به High تبدیل شود و دوباره Low گردد) شروع به شمردن counter می‌کند.

تمامی حالات بالا زمانی سیگنال خروجیشان تغییر می‌کند که سیگنال gate برابر ۱ باشد. در تمامی مودها بجز ۱ و ۵ زمانی که سیگنال gate برابر ۰ می‌شود، روند خروجی دادن متوقف می‌شود. حال در برخی از مودها مانند مود ۰ یا ۳ سیگنال خروجی با صفر شدن سیگنال gate تغییر نمی‌کند و در برخی مودها مانند مود ۲ با صفر شدن سیگنال gate مقدار سیگنال خروجی برابر ۱ می‌شود. در مودهای ۱ و ۵، صفر شدن سیگنال gate باعث می‌شود تا زمانی که دوباره مقدار آن برابر ۱ 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 ۸۲۵۳ استفاده می‌کنیم.

همانند تراشه ۸۲۵۵، تراشه ۸۲۵۳ را روی یک پورت ست می‌کنیم (در این کد روی پورت 8E ست شده است).



شکل ۱ - تراشه ۸۲۵۳ و آدرس دیکود شده آن (8Eh)

[illegible]

مطلوب است مدار ما به صورت زیر عمل کند:

- حال وارد بخش source code می شویم.

در ابتدا، setup تراشه ۸۲۵۵ را ست می‌کنیم (که هر سه پورت آن ورودی هستند و روی مود I/O هستیم و

mode selection ہا رانیز برابر • قرار مے دھیم و وارد بخش push_button مے شویم۔

در این بخش، منتظر می‌مانیم تا کاربر یکی از دکمه‌ها را فشار بدهد. اگر بعد از فشردن دکمه، عدد پورت برابر ۲ بود، بدان معنی است که کلید **blink** زده شده و باید وارد بخش **blink** شویم. اگر عدد پورت برابر ۱ 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 بالاخره کلیدی زده شود. حال هر دو حالت را بررسی می‌کنیم:

شکل ۳ - کانفیگ ۸۲۵۵
push_button و بخش

۱. 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 ۰ بود، به همان push_button برمیگردیم (چون انگار گفتیم ۰ ثانیه چشمک بزند؛ پس چراغ نباید روشن شود). حال تایمر اول این تراشه را روی مود ۳ قرار می‌دهیم و وارد حلقه B_TURN_OFF می‌شویم.

```
BLINK:
;INPUTING THE NUMBER OF THE DIP SWITCH
IN AL, 80H
NOT AL
MOV CL, AL
CMP CL, 0 ;CHECK IF THE DIP SWITCH NUMBER IS 0 OR NOT
JZ PUSH_BUTTON

;CONFIG 8253 TIMER0 TO SQUARE WAVE
MOV AL, 00010110B
OUT 8EH, AL
MOV AL, 10B
OUT 88H, AL
```

شکل ۴ - بخش خواندن عدد dip switch و کانفیگ کردن تایمر اول تراشه ۸۲۵۳ بر روی حالت blink

در این حلقه، بیت خروجی از تایمر را چک می‌کنیم و مادامی که بیت ورودی به لامپ برابر ۱ 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.

```
;CHECKS WHEN THE LED TURNS ON
BLINK_TURN_OFF:
IN AL, 84H
CMP AL, 0
JZ BLINK_TURN_OFF
```

همانند حلقه قبلی، بیت خروجی تایمر را چک می‌کنیم و این بار تا زمانی که بیت ورودی به LED برابر ۰ نشده بود، درون این حلقه میمانیم. در حقیقت زمانی از این حلقه بیرون می‌آییم که LED خاموش شده باشد.

```
;CHECKS WHEN THE LED TURNS OFF
BLINK_TURN_ON:
IN AL, 84H
CMP AL, 0
JNZ BLINK_TURN_ON
```

شکل ۵ - 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 N مان یکی کم می‌کنیم (به فرض قرار بوده ۴ بار روشن خاموش شود و حالا یکبار انجام شده؛ در نتیجه باید ۳ بار دیگر این عمل را تکرار کنیم) و تا زمانی که مقدار N (که در رجیستر CL نگهداری می‌شود) برابر ۰ نشده است، درون این بخش میمانیم. زمانی که مقدار CL برابر ۰ شد، طول دوره تناوب چشمک زدن را برابر ۰ می‌کنیم (که LED را کاملاً خاموش نگه دارد) و دوباره به بخش push_button می‌رویم و منتظر کلیدزنی مجدد میمانیم.

```
;REDUCING THE NUMBER OF THE DIP SWITCH FOR ITERATING
SUB CL, 1
CMP CL, 0
JNZ BLINK_TURN_OFF ;JUMP IF THE ITERATING HAS NOT FINISHED YET
MOV AL, 0
OUT 88H, AL
JMP PUSH_BUTTON
```

شکل ۶ - بخش کم کردن مقدار CL و بررسی صفر بودن و یا نبودن آن

نکته: می‌توانستیم ورودی لامپ را به تراشه ۸۲۵۵ 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 ۸۲۵۳ تا بینهایت این عملیات روشن و خاموش کردن را ادامه

می‌دهد و نمی‌توانیم بگوییم که مثلاً بعد ۴ بار تکرار لامپ را کلاً 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.

ONE_SHOT:

```
;INPUTING THE NUMBER OF THE DIP SWITCH
IN AL, 80H
NOT AL
MOV CL, AL
CMP CL, 0 ;CHECK IF THE DIP SWITCH NUMBER IS 0 OR NOT
JZ PUSH_BUTTON
```

```
;CONFIG 8253 TO ONE SHOT WAVE
```

```
MOV AL, 00010000B
OUT 8EH, AL
MOV AL, CL
OUT 88H, AL ;SET THE LENGTH OF BEING TURNED ON
```

```
JMP PUSH_BUTTON
```

شکل 4 – قطعه کد مربوط به بخش one shot

shot برویم و مقدار روشن بودن LED به مدت M (عدد جدید وارد شده) تمدید شود. به عنوان مثال، اگر عدد ورودی اول ۵ ثانیه بود و پس از گذشت ۲ ثانیه کلید one shot دوباره فشرده شد و عدد جدید برابر ۴ بود، میزان روشن بودن آن برابر $2 + 4 = 6$ ثانیه می‌شود.

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.

۲. مود One Shot: همانند مود قبلی، زمانی که وارد

این بخش می‌شویم، ابتدا عدد dip switch را می‌

خوانیم و در صورتی که عدد dip switch برابر ۰

بود، به بخش push_button برمی‌گردیم. حال

تایمر اول را روی مود ۰ 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 برمی‌

گردیم تا اگر کاربر در حین روشن بودن LED کلید

one shot را دوباره فشرده، دوباره به بخش one