

## گزارش کار آزمایش ۳ قسمت ب

امین احسانی مهر / شماره دانشجویی: ۹۹۲۴۳۰۰۹

نگار هنرور صدیقان / شماره دانشجویی: ۹۹۲۴۳۰۷۶

---

### سوالات تحلیلی

۱. وقفه‌ها برای ایجاد توقفی در برنامه به منظور اجرای دستور دیگری خارج از روال عادی برنامه است. مثلاً زمانی که می‌خواهیم از ترمینال عددی دریافت کنیم، با دادن درخواست وقفه روال عادی برنامه را متوقف می‌کنیم و منتظر دریافت ورودی می‌شویم.

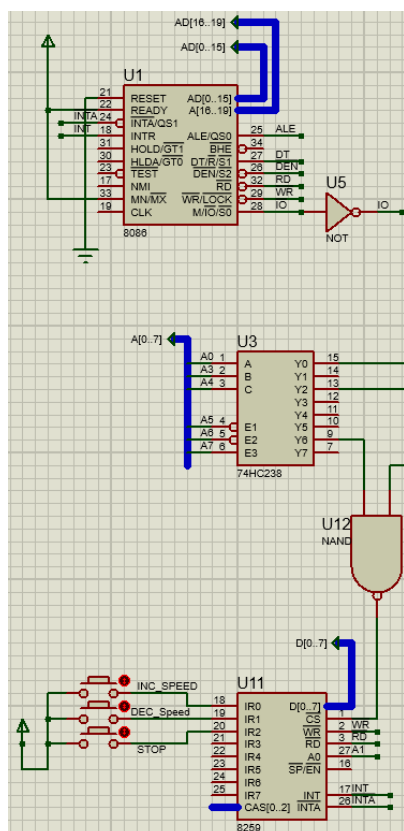
زمانی که درخواست وقفه‌ای داده می‌شود، از طریق سیگنالی به پردازنده اطلاع داده می‌شود و روند عادی برنامه متوقف می‌شود و درخواست داده شده توسط وقفه را اجرا می‌کند. سپس دوباره به روند عادی برنامه ادامه می‌دهد.

پس می‌توان گفت وقفه‌ها به دو صورت نرم افزاری و سخت افزاری هستند. وقفه‌های نرم افزاری می‌توانند برای اولویت دادن به بخشی از برنامه یا درون سیستم DOS به منظور استفاده از امکانات آن باشد. برای ایجاد وقفه‌های نرم افزاری باید درون کد برنامه نوشته شده دستورهای متناظر را پیاده سازی کنیم و با دستور INT و آدرس interrupt vector، ایجاد وقفه کنیم.

وقفه‌های سخت افزاری معمولاً برای استفاده از device‌های جانبی استفاده می‌شود. به طور مثال زمانی که می‌خواهیم از کیبورد استفاده کنیم، به ازای هر کلیدی که زده می‌شود، یک interrupt به پردازنده داده می‌شود و مقدار متناظر آن خوانده می‌شود.

۲. تراشه ۸۲۵۹ یک programmable interrupt handler است که به منظور افزایش تعداد اینترپت‌ها و تعریف آن‌ها استفاده می‌شود. به این صورت که اینترپت‌های جدید را مقداردهی می‌کنیم و زمانی که یکی از اینترپت‌ها زده شد، توسط سیگنال INTR به پردازنده اطلاع داده می‌شود که وقفه‌ای رخ داده است. سپس پردازنده یک سیگنال INTA به ۸۲۵۹ ارسال می‌شود تا شماره اینترپت زده شده را روی باس داده قرار دهد. سپس شماره اینترپت توسط پردازنده دریافت می‌شود و بخش مربوط به آن اینترپت اجرا می‌شود.

## گزارش کار

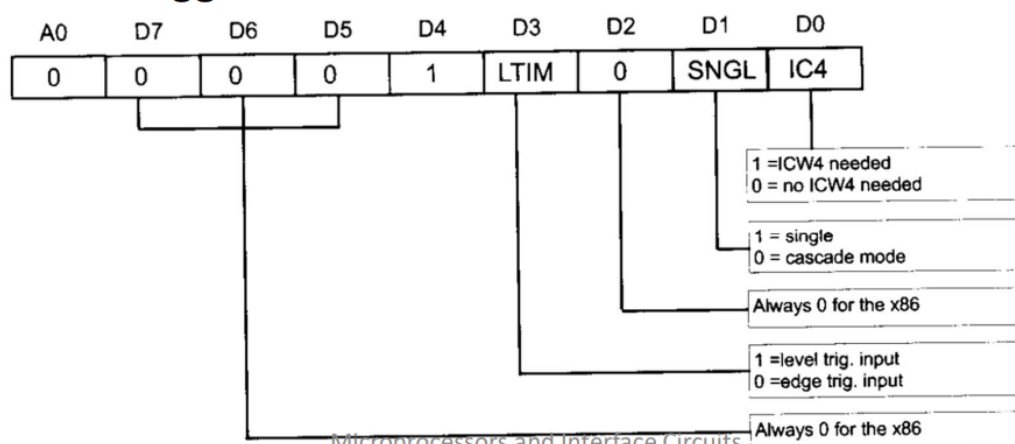


شکل ۱ - نحوه بستن تراشه ۸۲۵۹ در پروتئوس

این بخش ادامه بخش a همین دستور کار است؛ بدین صورت که یک LED داریم که زمانی که کلید متناظر با آن زده شد، عدد روی dip switch خوانده می‌شود و LED به مقدار عدد خوانده شده به صورت چشمک زن روشن و خاموش شود. حالا قرار است با استفاده از سه کلید که به تراشه ۸۲۵۹ وصل است، سرعت این چشمک زدن را با زیاد یا کم یا متوقف کنیم.

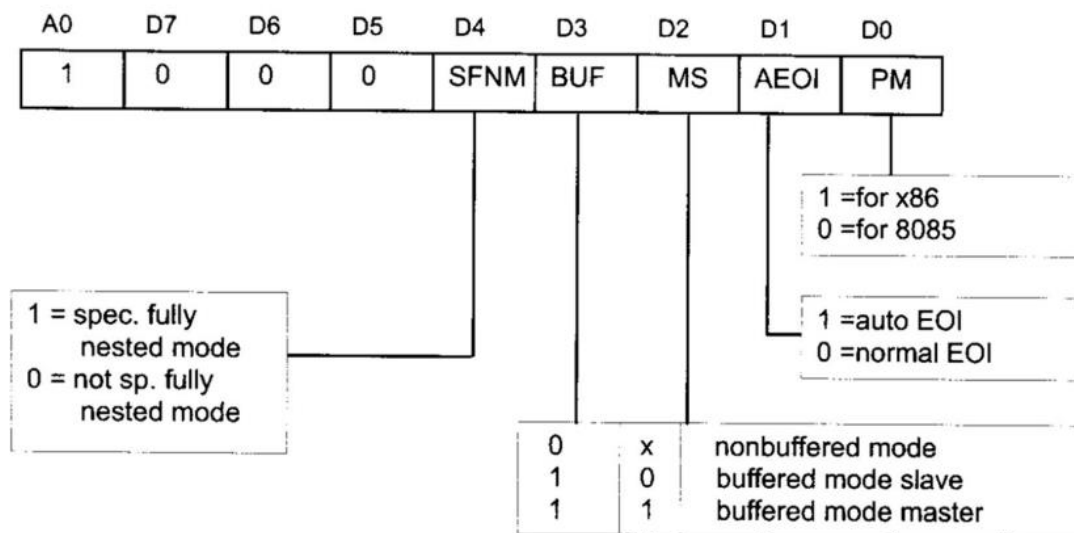
به این منظور، مشابه تمامی تراشه‌های استفاده شده، تراشه ۸۲۵۹ را روی یک پورت ست می‌کنیم. سیگنال‌های read و write و INT (به روی پورت INTR در پردازنده) و INTA (به پورت INTA در پردازنده) وصل می‌کنیم. باس داده را دیتای تراشه وصل می‌کنیم و کلیدهایمان را به پورت‌های اینترایت (اولویت تغییر نکرده و IRO بالاترین اولویت است)

حال وارد بخش کد می‌شویم. در ابتدا باید اینترپایت را مقداردهی اولیه کنیم. برای اینکار نیاز به ست کردن ICW1 و ICW2 و ICW4 داریم (ICW3 ست نمی‌شود چون در آن کانفیگ‌های مربوط به cascade کردن تراشه‌های ۸۲۵۹ انجام می‌شود). با توجه به مقادیر گفته شده در جدول زیر، مقدار ICW1 را برابر ۱۳ قرار می‌دهیم و روی پورت 98H ست می‌کنیم.



شکل ۲ - حدود اطلاعات موجود در ICW1

مقدار ICW2 با توجه به اینکه در حال program کردن تراشه ۸۲۵۹ هستیم، یک مقدار از 08H تا 0FH روی پورت 9AH ست می‌کنیم. مقدار ICW4 را نیز با توجه به جدول زیر برابر 03H قرار می‌دهیم و روی پورت 9AH ست می‌کنیم.



شکل ۳ - جدول اطلاعات موجود در ICW4

```
;CONFIG 8259
;CONFIG ICW1
MOV AL, 13H
OUT 98H, AL
```

```
;CONFIG ICW2
MOV AL, 0A0H
OUT 9AH, AL
```

```
;CONFIG ICW4
MOV AL, 03H
OUT 9AH, AL
```

شکل ۳ - دستورات توضیح داده شده

حال باید مشخص کنیم که هر کدام از کلیدهای متصل شده مربوط به کدام اینترپت است. پس آدرس هر کلید را به بخشی که قرار است کارکرد هر کلید را مشخص کند assign می‌کنیم.

```

MOV ES:[280H], OFFSET[IN_SPEED]
MOV ES:[282H], CS

MOV ES:[284H], OFFSET[DE_SPEED]
MOV ES:[286H], CS

MOV ES:[288H], OFFSET[STOP]
MOV ES:[28AH], CS

STI

```

شکل ۴ - آدرس‌های assign شده

حال به بخشی که قرار است پس از زده شدن کلید، اینترپت متناظر به آن را اجرا کند می‌رویم. در یکی از این labelها، قرار است سرعت را افزایش دهیم. به همین منظور، مقدار عددی که در ابتدا به تایمر دادیم را کم میکنیم و دوباره به تایمر می‌دهیم.

```

IN_SPEED:
MOV AL, 35
OUT 88H, AL
LOOP2:
JMP LOOP2
IRET

```

شکل ۵ - قطعه کد مربوط به بخش افزایش سرعت

در یکی دیگر از این labelها، قرار است سرعت را کم کنیم. مشابه بخش قبلی، این بار به مقدار ورودی تایمر اضافه می‌کنیم و به تایمر می‌دهیم تا با سرعت کمتری LED روشن و خاموش شود.

```

DE_SPEED:
MOV AL, 45
OUT 88H, AL
LOOP3:
JMP LOOP3
IRET

```

شکل ۶ - قطعه کد مربوط به بخش کاهش سرعت

در آخر، قرار است عملیات چشمک زدن متوقف شود. پس طول دوره تناوب تایمر را روی ۰ می‌گذاریم و به تایمر خروجی می‌دهیم.

```

STOP:
MOV AL, 0
OUT 88H, AL
LOOP1:
JMP LOOP1
IRET

```

شکل ۷ - قطعه کد مربوط به بخش stop