



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف اول آزمایشگاه ریزپردازنده

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۲/۱۴۰۱

فهرست مطالب

۱	۱
۲	۱.۱ الف
۲	۲.۱ ب
۲	۳.۱ ج
۲	۴.۱ د
۲	۲
۳	۱.۲ الف
۳	۲.۲ ب
۴	۳

۱

۱.۱ الف

برای جلوگیری از رخ دادن حالت float در پایه‌های میکروکنترلر و جلوگیری از اینکه در اثر تغییرات ولتاژ، تغییرات جریان نویز و... حالت آن پایه به صورت ناخواسته تغییر کند. با توجه به سرعت بالای ریزپردازنده این تغییر ممکن است به اشتباه برای ریزپردازنده به معنی یک یا صفر شدن پایه تلقی شود و در اجرای برنامه اخلال ایجاد کند. مقاومت‌های بالاکش بین تغذیه مدار و پایه میکروکنترلر وصل می‌شوند. استفاده از مقاومت‌های بالاکش در مدارات رایج‌تر است.

۲.۱ ب

□ کلیدهای کشویی: به منظور تولید صفر و یک دائمی در بلوکی تحت عنوان Data Latch Switch قرار داده شده است. این کلیدها وظیفه‌ی تولید صفر و یک منطقی را بر عهده دارند.

□ دکمه فشاری: به منظور تولید صفر و یک لحظه‌ای استفاده می‌شود. از آنها برای تولید صفر و یک منطقی استفاده می‌شود. با فشار دادن این کلیدها، یک سیگنال صفر یا یک در پایه متناظر ایجاد می‌کند و بلافاصله پس از رها کردن کلید، خروجی به سطح یک یا صفر باز می‌گردد.

۳.۱ ج

با کامپایل پروژه گزینه‌ی پروگرام تراشه فعال می‌شود و بدین وسیله روند برنامه‌ریزی ساده‌تر می‌گردد.

Project-> configure -> program the chip

روش دیگر با استفاده از این روش می‌توان هنگامی که کدهای ایجاد شده در Codevision موجود نیست و فقط کد Hex وجود دارد را نیز روی ریزپردازنده بارگذاری نمود. بدین منظور از منوی file می‌توان فایل hex مورد نظر را انتخاب نمود.

Tools-> chip programmer -> program all

۴.۱ د

□ حافظه‌ی Flash: قسمتی که برای نوشتن کد برنامه استفاده می‌شود. مثلاً برای ATmega32 داریم:

32Kbytes of In-System Self-programmable Flash program memory

□ حافظه‌ی RAM: حافظه‌ای موقت که به طور پیشفرض برای ذخیره‌سازی داده استفاده می‌شود. مثلاً برای ATmega32 مقدار آن 2 Kbytes هست.

□ حافظه‌ی EEPROM: حافظه‌ای دائمی که برای داده‌هایی استفاده می‌شود که پس از روشن و خاموش شدن میکرو باقی می‌ماند که مقدار آن برای ATmega32 1024 Bytes می‌باشد.

۲

□ PORTx: ثبات داده خروجی

□ DDRx: ثبات جهت داده

□ PINx: ثبات داده ورودی

۱. ثبات PINx تنها قابل خواندن است و ثبات‌های DDRx و PORTx هم قابل خواندن و هم قابل نوشتن هستند.
۲. اگر DDRxn یک پایه برابر صفر باشد پایه به صورت ورودی و اگر ۱ باشد پایه به صورت خروجی خواهد بود.
۳. برای خواندن وضعیت پایه‌هایی که به صورت ورودی تعریف شده است، ثبات PINx خوانده می‌شود و برای تعیین وضعیت پایه‌هایی که بصورت خروجی تعریف شده‌اند مقدار آن‌ها در ثبات PORTx نوشته می‌شود.

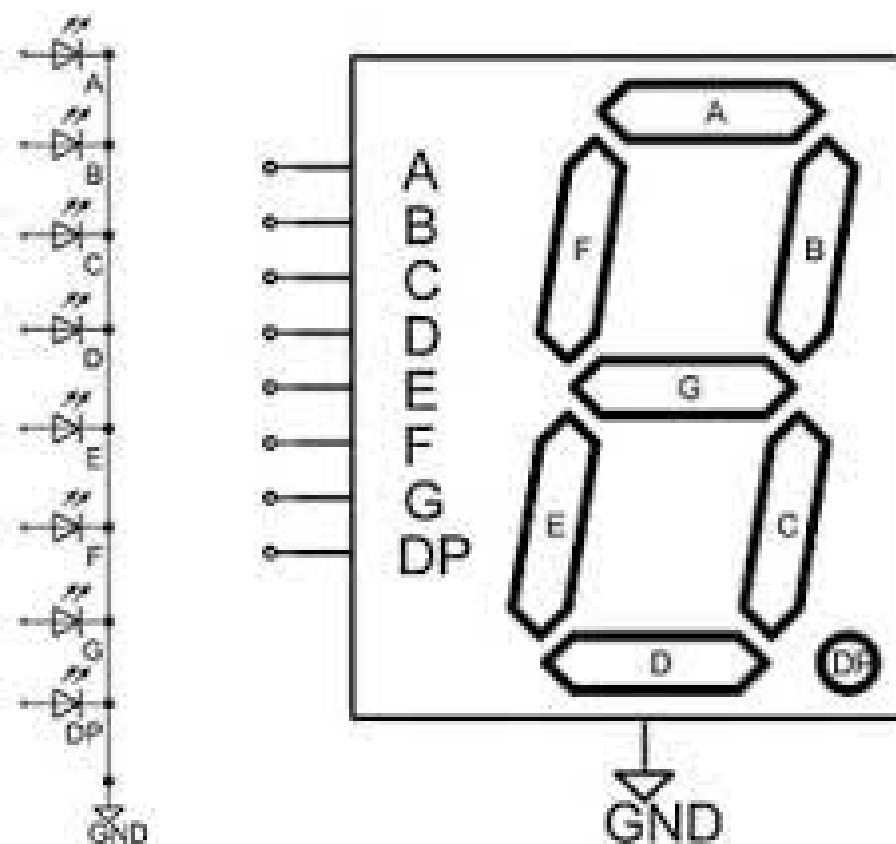
۱.۲ الف

```
DDRA = 0xAA;
```

۲.۲ ب

هنگام استفاده از محیط Wizard پس تعریف پورت به عنوان خروجی، مقاومت بالاکش را یک قرار می‌دهیم.

```
// Port B initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1)
| (1<<DDB0);
// State: Bit7=1 Bit6=1 Bit5=1 Bit4=1 Bit3=1 Bit2=1 Bit1=1 Bit0=1
PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) |
(1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);
```



شکل ۱:

فرض می‌کنیم پایه‌های سون سگمنت نظیر به نظیر به درگاه A متصل شده‌اند. باتوجه به ساختار پایه‌ها در سون سگمنت، برای تولید کاراکترهای مذکور باید معادل باینری سگمنت‌هایی که قرار است روشن شوند را محاسبه کنیم. پس به ترتیب داریم:

```
PORTA = 0x5E; //d = 01011110
```

```
PORTA = 0x77; //A = 01110111
```

```
PORTA = 0x76; //H = 01110110
```

```
PORTA = 0x79; //F = 01111001
```

منابع