

تکلیف سری دوم ریزپردازنده  
دانشگاه صنعتی اصفهان  
ترم اول سال تحصیلی ۹۸ - ۹۹

مریم سعیدمهر  
ش.د.: ۹۶۲۹۳۷۳  
استاد مربوطه: دکتر شایق

**تاریخ تحویل: 16 آذر 1398**

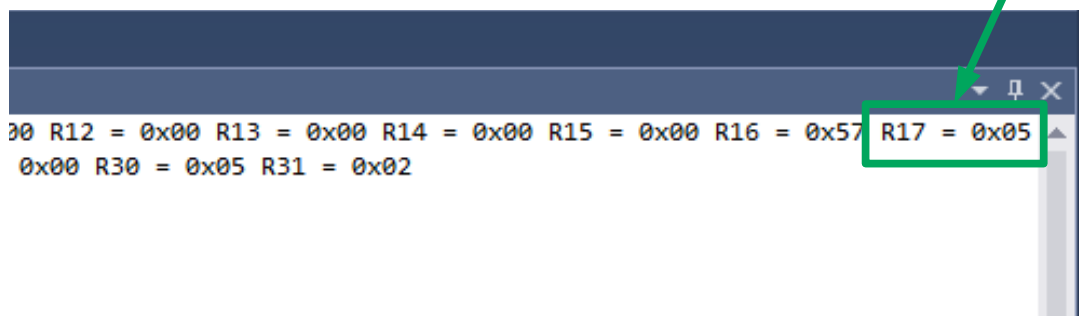
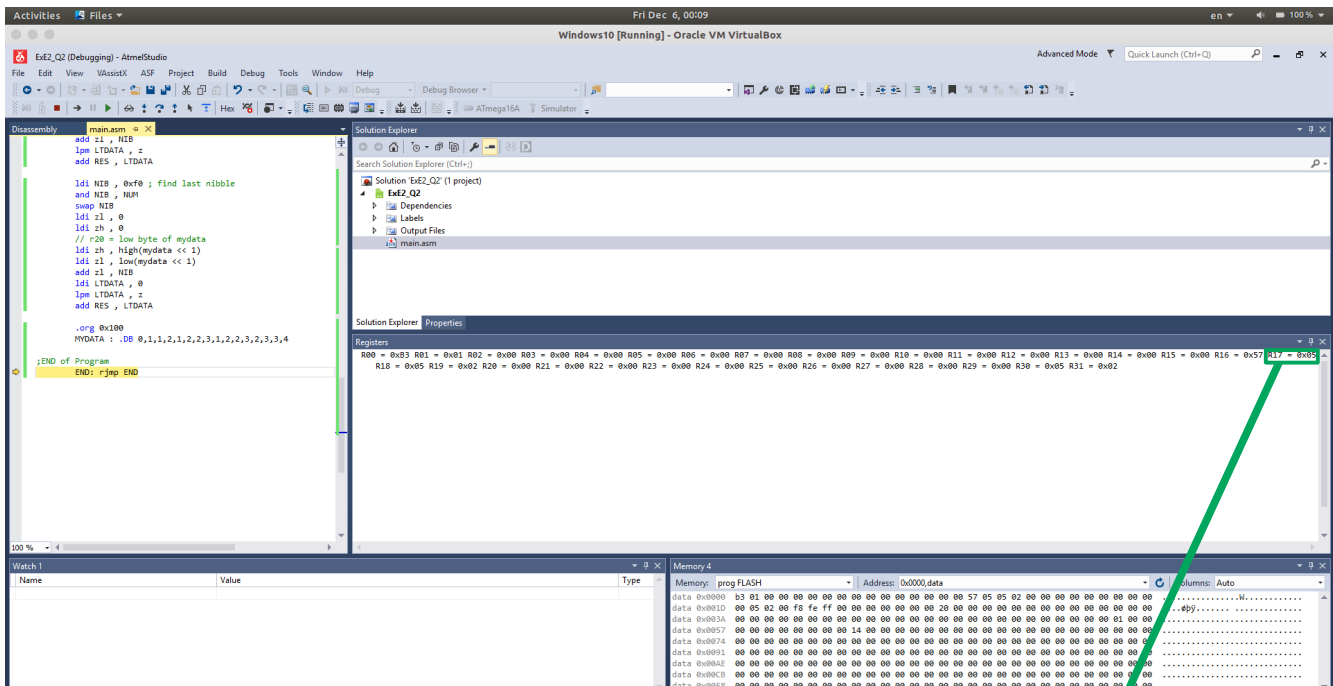
**برنامه های این تمرین را با اسمبلی و در گروه های دو نفره انجام دهید.**

**(همگروهی: ساجده نیک نداف)**

۱. بدون استفاده از امکانات سریال AVR، برنامه ای بنویسید که به صورت سریال یک بایت را از طریق PINC7 دریافت کند و آن را در R20 ذخیره کند و روی LEDهای پورت B نشان دهد. ابتدا کم ارزش ترین بیت وارد میشود. این برنامه را در پروتئوس با استفاده از یک سوئیچ متصل به پایه ی 7 پورت C اجرا کنید.

کد این سوال، در فایل زیپ پاسخنامه کلی ضمیمه شده است... همچنین از طریق این [لینک](#) نیز قابل دسترسی است

۲. به سه طریق مختلف برنامه ای بنویسید که تعداد یک های موجود در یک بایت را بشمارد. کفایت این برنامه را روی سیمولاتور اجرا کنید.



عدد مورد تست در سوال 0b 01010111 بوده که ۵ تا رقم یک دارد و همانطور که در صورت کدها هست ، پاسخ نهایی در رجیستر R17 ذخیره شده است ....

(\*) توجه ! برای تست کدها فقط متد سوم ، روی سیمولاتور تست شد! درستی دو روش اول واضح و قابل

تست است: )

(\*) توجه ! کدهای این سوال، در فایل زیپ پاسخنامه کلی ضمیمه شده است... همچنین از طریق این [لینک](#)

نیز قابل دسترسی است)

۳. چند خط برنامه معادل دستور SWAP بنویسید.

.ORG 100;

;consider that our number is in R16

LDI r16,0xc8;

MOV r17,r16;

ANDI r17,0xf0;left nibel

ANDI r16,0x0f:right nibel

LSR r17;

LSR r17;

LSR r17;

LSR r17;

LSL r16;

LSL r16;

LSL r16;

LSL r16;

ADD r16,r17;

Loop: RJMP loop

۴. برنامه ی Checksum را برای کنترل محتویات EEPROM بنویسید.

ldi r16,0;

\* این برنامه، checksum را از آدرسی که در X هست

ldi r18,0

تا R17 تا خانه بعد ، محاسبه کرده و در R18 میریزد.

ldi r17,8;this number can change

again:

sbic eecr,eewe;

rjmp again;

out eearh,xh

out eearl,xl

sbi eecr,eere;

in r16,eedr;

add r18,r16;

dec r17;

brne again;

neg r18;

۵. برنامه ی تقسیم دو عدد علامت دار را بنویسید.

```
.def    d8s    =r14        ;sign register
.def    drem8s    =r15        ;remainder
.def    dres8s =r16        ;result
.def    dd8s    =r16        ;dividend
.def    dv8s    =r17        ;divisor
.def    dcnt8s =r18        ;loop counter

;***** Code
clr drem8s;
clr dres8s;
LDI dd8s, 8;
LDI dv8s,4;

div8s: mov    d8s,dd8s      ;move dividend to sign register
        eor    d8s,dv8s     ;xor sign with divisor
        sbrc   dv8s,7       ;if MSB of divisor set
        neg    dv8s        ;  change sign of divisor
        sbrc   dd8s,7       ;if MSB of dividend set
        neg    dd8s        ;  change sign of divisor
        sub    drem8s,drem8s ;clear remainder and carry
        ldi    dcnt8s,9     ;init loop counter
d8s_1:  rol    dd8s          ;shift left dividend
        dec    dcnt8s       ;decrement counter
        brne   d8s_2        ;if done
        sbrc   d8s,7        ;  if MSB of sign register set
        neg    dres8s       ;    change sign of result
        ;ret                ;  return
        loop:rjmp loop

d8s_2:  rol    drem8s        ;shift dividend into remainder
        sub    drem8s,dv8s   ;remainder = remainder - divisor
        brcc   d8s_3        ;if result negative
        add    drem8s,dv8s   ;  restore remainder
        clc                ;  clear carry to be shifted into result
        rjmp   d8s_1        ;else
d8s_3:  sec                ;  set carry to be shifted into result
        rjmp   d8s_1
```

۶. این برنامه چه کار میکند ؟

```
***** Subroutine Register Variables

.def drem8u  =r15      ;remainder
.def dres8u  =r16      ;result
.def dd8u    =r16      ;dividend
.def dv8u    =r17      ;divisor
.def dcnt8u  =r18      ;loop counter
```

این کد ، دو عدد بدون

علامت ۸ بیتی را بر هم

تقسیم میکند .

```
***** Code
```

```
div8u:      sub      drem8u,drem8u ;clear remainder and carry
            ldi      dcnt8u,9      ;init loop counter
d8u_1:      rol      dd8u          ;shift left dividend
            dec      dcnt8u        ;decrement counter
            brne     d8u_2         ;if done
            ret
            ; return
d8u_2:      rol      drem8u        ;shift dividend into remainder
            sub      drem8u,dv8u    ;remainder = remainder - divisor
            brcc     d8u_3         ;if result negative
            add      drem8u,dv8u    ; restore remainder
            clc
            ; clear carry to be shifted into result
            rjmp     d8u_1         ;else
d8u_3:      sec
            ; set carry to be shifted into result
            rjmp     d8u_1
```

۷. برنامه زیر نوشته شده و بدون خطا اجرا میشود. چگونه ممکن است خط اول برنامه درست باشد؟(هدف

این خط تنظیم پشته در انتهای RAM است)

```
start:      setStack RAMEND,r16    ; initialize stack pointer
...
loop:       rjmp     loop          ; infinite loop
```

\* اگر منظور خط اول ، ریختن مقدار RAMEND در R16 باشد، به شرطی این خط به درستی اجرا

میشود که ۸ بیت بتوان به کل حافظه ، آدرس دهی کرد.

۸. با استفاده از یک Look Up Table برای اعداد 7seg ، مداری ببندید که تعداد دفعات فشردن شدن دو

سوئیچ را بشمارد و روی 7seg نمایش دهد. یکی از سوئیچ ها افزایش و دیگری کاهش عدد را سبب می شود. برنامه مربوط را نوشته و بعد از اطمینان از درستی، آن را در پروتئوس اجرا کنید و سپس روی میکروی خود پروگرام کنید.

I. فرض کنید مجموع تعداد دفعات فشار دادن کلیدها از ۹ بیشتر نمیشود.

II. درابتدا اجرای مدار ، عدد ۰ نمایش داده میشود.

III. 7seg مستقیماً و بدون آی سی راه انداز به مدار وصل میشود.

\* لطفاً به این [لینک](#) مراجعه کنید.