

# دانشگاه صنعتی خواجه نصیرالدین طوسی

پروژه درس سیستم های دیجیتال 2

مدرس: دکتر رسول دلیر روی فرد

مصطفی لطیفیان - 40122193

پروژه شماره 5

بازی حافظه دو نفره

برای محاسبه راحت تر زمان 200 میلی ثانیه از فرکانس 1.024 MHz استفاده می کنیم.

$$f_{timer} = \frac{f_{cpu}}{prescaler} = \frac{1.024 \text{ Mhz}}{1024} = 1 \text{ KHz} , T_{timer} = 1 \text{ ms} , counter = \frac{200 \text{ ms}}{1 \text{ ms}} = 200 , 256 - 200 = 56$$

ابتدا آدرس های مربوطه و تایمر و وقفه ها را ست می کنیم.

```
.include "m64def.inc"

.org 0x0000
jmp main

.ORG 0X0002
JMP EXT_INT0_ISR

.ORG 0X0004
JMP EXT_INT1_ISR

.ORG 0X0014
JMP T2_OV_ISR

.org 0x0050
```

در مرحله بعدی در برنامه اصلی آدرس پشته هارا ست می کنیم.

```
main :
LDI R20 , LOW(RAMEND)
OUT SPL , R20
LDI R20 , HIGH(RAMEND)
OUT SPH , R20 ; set stack pointer
```

حال پذیرش وقفه های 0 و 1 را فعال و همچنین برای آن ها لبه پایین رونده را ست می کنیم.

```
LDI R20 , 0X0A
STS EICRA , R20 ; falling edge for intrupt 0,1
LDI R20 , 0X03
OUT EIMSK , R20 ; Set intrupts
;=====
```

حال برای تایمر 2 مقادیر مورد نیاز را قرار می دهیم.

```
LDI R20 , 56
OUT TCNT2 , R20 ; Start timer from 56
LDI R20 , 0X05
OUT TCCR2 , R20 ; Set the prescaler
LDI R20 , 0X40
OUT TIMSK , R20 ; Turn on Timer2
;=====
```

در قسمت بعدی پورت های A و C را به عنوان ورودی و پورت E را به عنوان خروجی تعریف می کنیم، همچنین برای خاموش ماندن ال ای دی های سفید در ابتدای بازی که به پورت E متصل هستند تمام پورت E را با توجه به طراحی انجام شده در پروتئوس برابر یک قرار می دهیم.

```
LDI R20 , 0X00
OUT DDRA , R20 ;Set input Port A
OUT DDRC , R20 ;Set input Port C
LDI R20 , 0X0FF
OUT DDRE , R20 ;Set output Port E
OUT PORTE , R20 ;Write 0xFF in Port E
```

بیت های 0 و 1 پورت D را برابر یک قرار می دهیم تا ورودی شوند که این کار به علت این است که وقفه های صفر و یک پذیرفته شوند، همچنین بیت های 2 و 3 و 4 پورت D را خروجی می کنیم و به علت اتصال ال ای دی های وضعیت در آن یک را قرار می دهیم.

```
CBI DDRD , 0
CBI DDRD , 1 ;Set input bit 0,1 Port D
SBI DDRD , 2
SBI DDRD , 3
SBI DDRD , 4 ;Set output bit 2,3,4 Port D
SBI PORTD , 0
SBI PORTD , 1
SBI PORTD , 2
SBI PORTD , 3
SBI PORTD , 4 ;Write 1 in Port D
LDI R20 , 0X0FF
OUT PORTC , R20 ;Write 1 in Port C
;=====
```

در R16 عدد 5 را قرار می دهیم و این رجیستر برای کاهش زمان برای پس از پیروزی است. رجیستر R17 و R27 شرط شروع و پایان وقفه ها را انجام می دهند. R18 نیز در وقفه یک قرار دارد، R21 برای شمارش هر 200ms است. R22 برای شمارش تعداد پیروزی های بازیکن B است و R23 برای تعداد حرکت های بازیکن B است و R30 برای ذخیره محتوای پورت C در نظر گرفته شده است و در بیت 1 را فعال می کنیم.

```
LDI R16 , 5 ;Set for wining time
LDI R17 , 0X0FF ;Set for intruupt condition
CLR R27 ;Set for intruupt condition
LDI R18 , 1 ;Set for intruupt1
CLR R21 ;Timer counter 200ms
CLR R22 ;win counter
CLR R23
LDI R30 , 0X0FF
SEI
;=====
```

حال به سراغ حلقه اصلی بازی می رویم، در ابتدا شرط آمدن وقفه صفر را بررسی می کنیم و تا هنگامی که وقفه صفر فشرده نشده باشد ورودی پذیرفته نمی شود بعد از آمدن وقفه صفر به سراغ دریافت شماره ال ای دی های صفر تا چهار توسط بازیکن A می رویم و پس از آمدن وقفه صفر برای بار دوم نوبت بازیکن A تمام می شود و پردازنده منتظر وقفه 0 می شود تا بازیکن B حدس زدن خود را همانند بازیکن A شروع کند. همچنین R19 به عنوان نگه دارنده الگوی بازیکن A و R24 به عنوان نگه دارنده تعداد الگوی بازیکن A است.

GAME :

```
CALL SHART ;Int0 check for player B start
CALL DARYAFT
CALL DARYAFT
CALL DARYAFT
CALL DARYAFT ;Get input from Player A
CALL SHART ;Int0 check for player A end

MOV R19 , R30 ;Hold Player A inputs
LDI R30 , 0X0FF
MOV R24 , R23 ;Counting Player A inputs
CLR R23
```

برنامه شرط به صورت زیر است و بعد از هر بار پذیرش وقفه صفرم R17 مکمل می‌شود و شرط را اجرا می‌کند.

```
EXT_INT0_ISR:
    COM R17
    RETI

SHART:
    CP    R17 , R27
    BRNE SHART
    COM   R27
    RET
```

برنامه دریافت نیز به این صورت است که در R18 عدد یک ذخیره شده است و پس از آمدن هربار وقفه صفر می‌شود و به برنامه دریافت می‌رود و تا زدن بار چهارم وقفه یک اعداد را از پورت A دریافت می‌کند و پس از دریافت هر عدد آن را با عدد ال ای دی متناظر چک می‌کند و در صورت برابر بودن ال ای دی مدنظر را روشن می‌کند و در صورت نابرابری یک عدد از تعداد الگوها کم می‌کنیم.

نحوه روشن کردن ال ای دی ها به این صورت است که بیت متناظر را برابر صفر قرار می‌دهیم و پس از 1 ثانیه آن را یک می‌کنیم تا ال ای دی مورد نظر خاموش شود و همچنین عدد را در R30 ذخیره می‌کنیم.

```
EXT_INT1_ISR:
    CLR R18
    RETI
```

```
DARYAFT:
    SHART1:
        CPI    R21 , 51      ;Counting for 10s
        BRCC   END
        CPI    R18 , 0
        BRNE   SHART1
        INC    R23
        CPI    R23 , 5
        BRCC   BAZGASHT
        IN     R20 , PINA    ;Get number for turn on LED
        CPI    R20 , 1
        BREQ   LED1
        CPI    R20 , 2
        BREQ   LED2
        CPI    R20 , 3
        BREQ   LED3
        CPI    R20 , 4
        BREQ   LED4
        DEC    R23
        JMP    BAZGASHT

BAZGASHT:    LDI    R18 , 1
             RETI

;=====

LED1:
    CBI    PORTE , 0
    IN     R30 , PINC
    ANDI   R30 , 0X0FF
    CALL   DELAY1s
    SBI    PORTE , 0
    JMP    BAZGASHT

LED2:
    CBI    PORTE , 1
    IN     R30 , PINC
    ANDI   R30 , 0X0FF
    CALL   DELAY1s
```

```

SBI PORTE , 1
JMP BAZGASHT

LED3:    CBI PORTE , 2
          IN  R30 , PINC
          ANDI R30 , 0X0FF
          CALL DELAY1s
          SBI PORTE , 2
          JMP BAZGASHT

LED4:    CBI PORTE , 3
          IN  R30 , PINC
          ANDI R30 , 0X0FF
          CALL DELAY1s
          SBI PORTE , 3

;=====

```

پس از اتمام نوبت بازیکن A بعد از آمدن وقفه صفرم نوبت بازیکن B می شود و چراغ زرد روشن می شود. نحوه وارد کردن الگو توسط بازیکن B درست همانند بازیکن A می باشد. R29 برای نگه داری ورودی بازیکن B و R25 برای شمارش تعداد الگوی وارد شده توسط بازیکن B است.

```

CALL SHART
CBI PORTD ,4           ;Turn on Yellow LED
CLR R21
LDI R20 ,56           ;Set counter again
OUT TCNT2 ,R20

CALL DARYAFT
CALL DARYAFT
CALL DARYAFT
CALL DARYAFT           ;Get input from player B
CPI R21 , 51           ;Counting 10s
BRCC END               ;End game after 10s
CALL SHART
SBI PORTD , 4           ;Turn off Yellow LED B
MOV R29 , R30           ;Hold player B inputs
LDI R30 , 0X0FF         ;Reset R30 for next round
MOV R25 , R23           ;Counting Player B inputs
CLR R23

```

پس از اتمام نوبت بازیکن B به سراغ چک کردن درستی جواب می رویم، ابتدا تعداد الگو ها را محاسبه می کنیم و در صورت برابر بودن الگو ها با یکدیگر مقایسه می کنیم.

در صورت اشتباه بودن پاسخ بازیکن B چراغ قرمز به مدت ثانیه و در صورت درست بودن پاسخ چراغ سبز به مدت 5 ثانیه روشن می شود و R16 را یک عدد کاهش می دهیم تا از زمان نشان دادن چراغ سفید 200ms کم شود و پس از اتمام هریک از آن ها بازی به حالت اول بازمی گردد.

```

CP R25 , R24           ;First check
BRNE GHERMEZ
CP R19 , R29           ;Second check
BREQ SABZ

;=====
GHERMEZ:
CBI PORTD , 2           ;Torn on RED LED
CALL DELAY3s
SBI PORTD , 2           ;Turn off RED LED
LDI R16 , 5
LDI R18 , 1             ;Ready for intrupt1
CLR R23
LDI R17 , 0X0FF
CLR R27                 ;Ready for next round
JMP GAME

```

```

SABZ :
    CBI   PORTD    ,3           ;Turn on GREEN LED
    CALL  DELAY5s
    SBI   PORTD    ,3           ;Turn off GREEN LED
    INC   R22        ;Counting number of player B wins
    STS   0X6400 ,R22
    DEC   R16        ;For decrement 1s
    CPI   R16        ,1
    BREQ  END
    LDI   R17        ,0X0FF
    CLR   R27
    LDI   R18        ,1
    CLR   R23
    SBI   PORTD     ,4
    JMP   GAME
;=====

```

برنامه دیلی یا تاخیر برای چراغ سبز و قرمز نیز به صورتی نوشته شده است که تا هنگامی که میزان زمان خواسته شده سپری نشده باشد از حلقه خواسته شده خارج نشود.

```

DELAY1s:  SEI
          CLR    R21
          LDI    R20    , 56
          OUT    TCNT2  , R20
CPAGAIN1: CP     R21    , R16
          BRNE   CPAGAIN1
          CLI
          RET

DELAY3s:
          CLR    R21
          LDI    R20    , 56
          OUT    TCNT2  , R20
CPAGAIN3: CPI    R21    , 15
          BRNE   CPAGAIN3
          RET

DELAY5s:
          CLR    R21
          LDI    R20    , 56
          OUT    TCNT2  , R20
CPAGAIN5: CPI    R21    , 25
          BRNE   CPAGAIN5
          RET

```

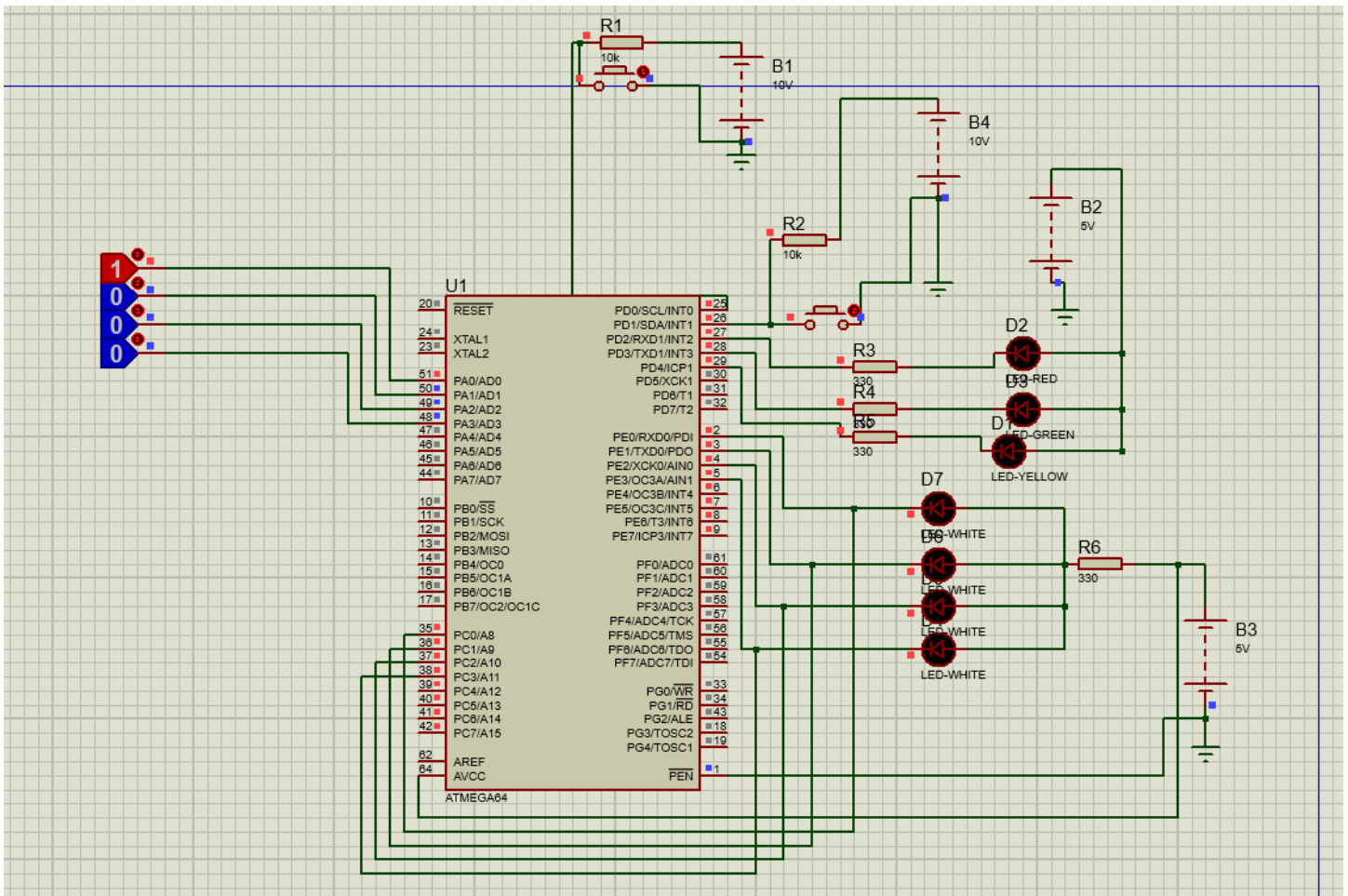
در صورت اتمام زمان نیز بازی به پایان خواهد رسید و تمام مقادیر اصلی دوباره مقدار دهی می‌شوند.

```

END:
    LDI   R16        ,5
    LDI   R18        ,1
    CLR   R23
    LDI   R17        ,0X0FF
    CLR   R27
    SBI   PORTD      , 4
    JMP   GAME;
;=====

```

همچنین پروژه پیاده سازی شده در پروتئوس نیز به شکل زیر است.



توضیحات بیشتر درمورد نحوه کارکرد این پروژه در ویدیو ذکر شده است.