

بسم الله الرحمن الرحيم

تمرین دهم ریزپردازنده

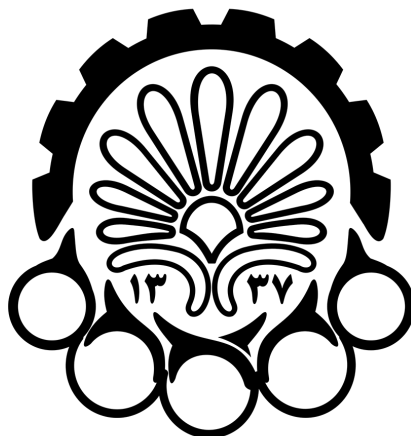
بهار ۹۷

امیرحسین عباسی

۹۳۳۱۰۷۰

دانشگاه امیرکبیر

دانشکده کامپیوتر و فناوری اطلاعات



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

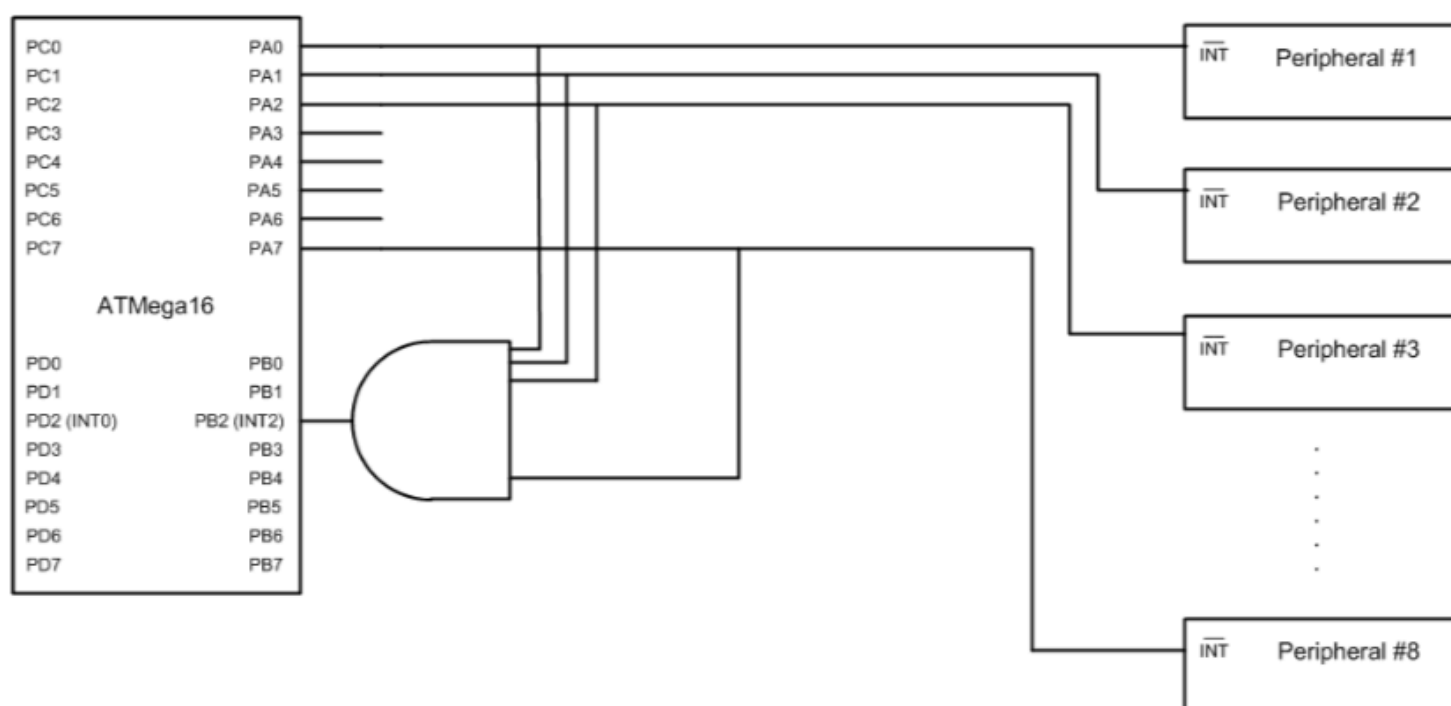
(۱) چرا نمی‌توان یک لچ را به عنوان پورت ورودی و یک بافر را به عنوان پورت خروجی استفاده کرد؟

ما همیشه لچ را برای پورت خروجی و بافر را برای پورت ورودی استفاده می‌کنیم. از اهداف اتصال بافر بر سر راه این دستگاه‌ها مراقبت از مدار و تنظیم سطح ولتاژ می‌باشد. لازم است که در محل اتصال دستگاه‌ها اتصال اشتباه ولتاژها صورت نگیرد و ضمناً نیازی به نگهداری مقادیر خروجی در حافظه موقت نیست به همین خاطر از بافر استفاده می‌کنیم.

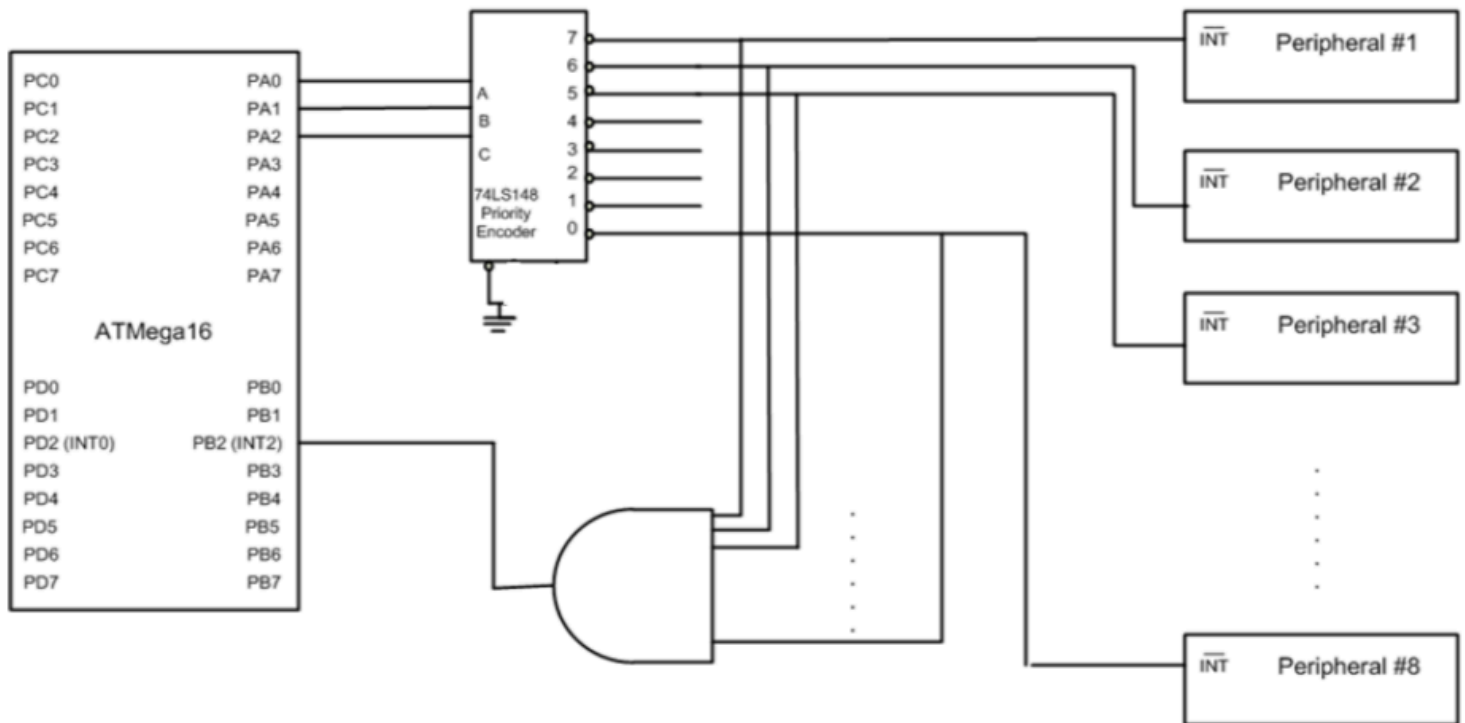
latch همانند یک حافظه موقت عمل می‌کند و دیتا را در خود نگه می‌دارد. هنگامی که یک میکروکنترلر داده‌ای را در یکی از پورت‌های خروجی خود قرار می‌دهد این داده به واسطه لچ به همه دستگاه‌ها متصل می‌شود و تا زمان متصل بودن لچ به پاور اگر داده‌ی جدیدی رو لچ از طرف میکروکنترلر نوشته نشود داده معتبر خواهد بود و دستگاه‌ها می‌توانند از آن استفاده کنند و می‌توان تا زمان معتبر شدن مقدار خروجی میکروکنترلر از نوشته شدن مقدار داده بر خروجی لچ جلوگیری کرد اما از بافر که تنها یک اتصال لحظه‌ای بین ورودی و خروجی خود برقرار می‌کند چنین قابلیتی را ندارد که داده معتبر را در خود نگه دارد.

۲) چه راه‌هایی را در عمل می‌توان برای رعایت اولویت‌گذاری برای وسایل متقاضی وقفه استفاده نمود؟

کار کلی ای که انجام می‌دهیم این است که بیت تقاضای وقفه همه وسایل را با هم ترکیب کنیم (OR یا AND) و آنها را تبدیل به یک خروجی کلی بکنیم و آن را به یک پورت میکرو وصل کنیم. اگر پورت ۱ شود یعنی وسیله ای تقاضای وقفه کرده و وارد روتین انجام وقفه می‌شویم. در روتین انجام وقفه بیت های پورتهی که بیت تقاضای وقفه وسایل به آن متصل است را چک می‌کنیم تا ببینیم کدام یک از وسایل تقاضای وقفه کرده است.



حال اگر دو وسیله هم زمان تقاضای وقفه کرده باشند میتوان آن ها را شناسایی کرد و یکی یکی به انجام کارهای آن ها پرداخت.



راه اول برای اولویت گذاری ترتیب همین چک کردن بیت های پورت **A** میباشد. (همچنین میتوانیم وسایل را به ترتیب اولویت به پورت **A** وصل کنیم) بیت تقاضای وقفه ی وسیله ای که برای ما اولویت بیشتری دارد را زودتر چک میکنیم و لذا روتین آن زودتر انجام میشود.

راه دوم استفاده از یک تراشه کدگذار اولویت گذار (priority encoder) مثل ۷۴۱۴۸ میباشد. مطابق با تصویر بالا وسایل را به ترتیب اولویت از بالا به پایین میگذاریم. چنانچه دو وسیله همزمان تقاضای وقفه بکنند انکودر وسیله با اولویت بیشتر را کدگذاری میکند و به سه بیت کم ارزش پورت **A** میدهد. در نتیجه وسیله با اولویت بیشتر زودتر سرویس داده میشود

۳) آیا اولویت‌دهی به تقاضای سرویس وسایل جانبی در روش سرکشی قابل انجام است؟

بله ، در روش سرکشی یک روتین سرکشی داریم که در آن از بالا به پایین بیت **busy** وسایل مختلف را چک میکنیم و در صورت صفر بودن آن روال وقفه آن وسیله را انجام میدهیم . میتوانیم بیت **busy** وسایل با اولویت بیشتر را زودتر چک کنیم تا در صورت درخواست همزمان در روتین زوتر به روتین آن وسیله برسد و لذا سرویس دهی آن زودتر انجام شود . اما بهتر است در صورت نیاز به اولویت دادن به وسایل جانبی از روش مبتنی بر وقفه استفاده کنیم . در این صورت با کمک یک **priority encoder** میتوانیم به وقفه با اولویت بالاتر زودتر رسیدگی کنیم .

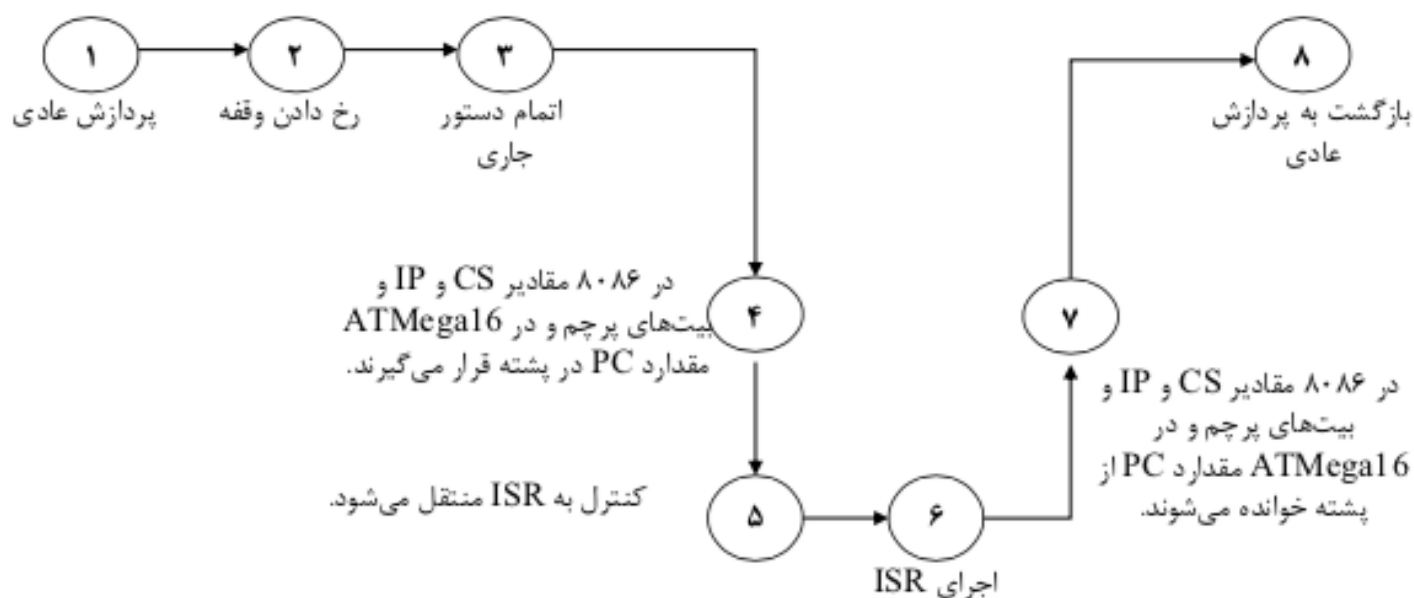
```
; Polling of 8 devices
; Busy Line of all 8 devices are connected to PB

LOOP1:      SBIS    PINB, 0      ; Skip next inst. if FD is not Ready      [1/3]
             RCALL   FD          ;                               [3]
             SBIS    PINB, 1      ; Skip next inst. if LP is not Ready      [1/3]
             RCALL   LP          ;                               [3]
             SBIS    PINB, 2      ; Skip next inst. if DWP is not Ready    [1/3]
             RCALL   DWP         ;                               [3]
             SBIS    PINB, 3      ; Skip next inst. if PLOT is not Ready  [1/3]
             RCALL   PLOT        ;                               [3]
             SBIS    PINB, 4      ; Skip next inst. if ADC is not Ready  [1/3]
             RCALL   ADC         ;                               [3]
             SBIS    PINB, 5      ; Skip next inst. if DAC is not Ready  [1/3]
             RCALL   DAC         ;                               [3]
             SBIS    PINB, 6      ; Skip next inst. if MOD is not Ready [1/3]
             RCALL   MOD         ;                               [3]
             SBIS    PINB, 7      ; Skip next inst. if TERM is not Ready [1/3]
             RCALL   TERM        ;                               [3]
             JMP     LOOP1        ;                               [3]
```

مثلا با استفاده از روتین بالا **FD** بیشترین اولویت و **TERM** کمترین اولویت را خواهد داشت .

۴) در صورت استفاده از میکروکنترلر ATmega16 چه اطلاعات مهمی از برنامه اصلی می‌بایست در بدو ورود به روتین وقفه حفظ و در هنگام خروج از آن بازیابی شود؟ ذخیره و بازیابی در کجا صورت می‌گیرد؟

قبل از ورود به وقفه مقدار PC در پشته حفظ می‌شود .
 سپس کنترل برنامه به ISR می‌رود ، روتین وقفه انجام می‌شود .
 هنگام خروج از وقفه مقدار PC از پشته بازیابی می‌شود .
 ذخیره و بازیابی در پشته انجام می‌شود.



شکل ۱۷- فرآیند انجام شونده بعد از رخ دادن وقفه

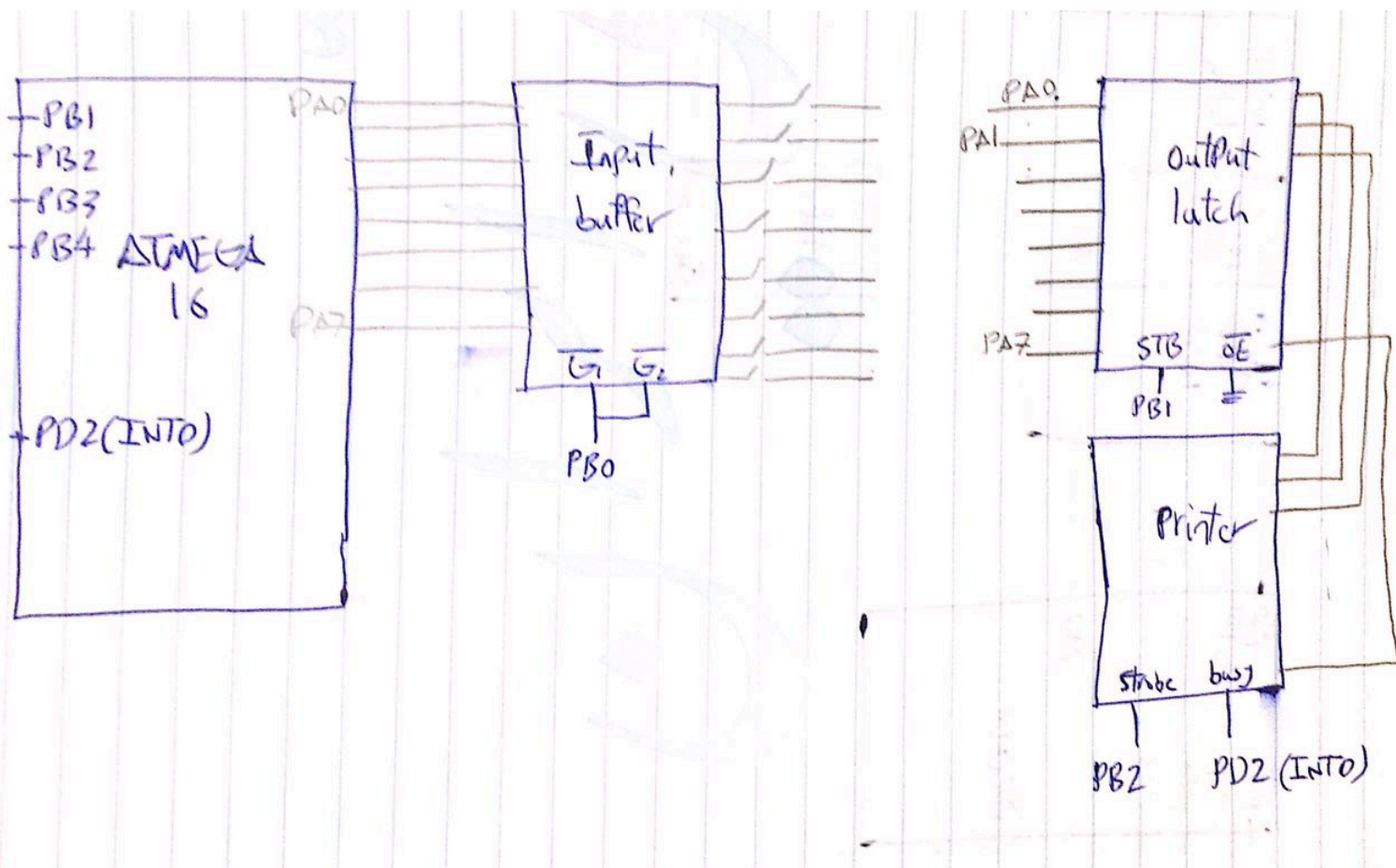
(۵) اقدامات زیر را انجام دهید:

الف- یک درگاه ورودی (بافر 74LS244) و یک خروجی هشت بیتی (لج 82C82) به نحو مناسب به میکروکنترلر ATmega16 متصل نمایید و دیکودینگ مورد نیاز برای فعال سازی هر دو را ارائه نمایید. از پورت A برای نقل و انتقال داده بین میکروکنترلر و درگاه های ورودی و خروجی و از بیت های پورت B برای فعال سازی این درگاه ها استفاده نمایید.

ب- درگاه ورودی را به ۸ کلید که هر کلید می تواند در وضعیت باز یا بسته قرار گیرد و درگاه خروجی را به یک پرینتر متصل نمایید. برنامه دریافت داده از درگاه متصل به کلیدها و نیز برنامه نوشتن داده در درگاه متصل به پرینتر را بنویسید. برنامه را به گونه ای بنویسید که زمانبندی مناسب در خواندن درگاه ورودی و نوشتن در درگاه خروجی مراعات گردد.

ج- چنانچه با بسته یا باز شدن یکی از کلیدها یک وقفه به میکروکنترلر اعمال شود، و پرینتر هم هر وقت آمادگی چاپ داده جدید دارد یک وقفه بدهد (خروجی وقفه active low)، مدار لازم برای اعمال وقفه ها به ورودی وقفه INT0 و تشخیص اینکه کدام وسیله (مجموعه کلیدها یا پرینتر) وقفه داده است را ارائه کنید.

د- برنامه روتین وقفه INT0 برای مدیریت وقفه ها و سرویس دادن به مجموعه کلیدها (دریافت وضعیت کلیدها) و پرینتر (دادن وضعیت کلیدها برای چاپ) را بنویسید. فرض کنید وقفه مجموعه کلیدها دارای اولویت بیشتری است.



```

reset:
    cli
    ldi r16,0x00
    out DDRA ,r16 ;define PORT A as Input
    ldi r16,0xFF
    out DDRB,r16 ;define PORT B as Output

    ldi r16, (1 << ISC11 | 0 << ISC10)
    out MCUCR, r16
    ldi r16, (1 << INT0)
    out GICR, r16

    ldi r16,0x01
    out PORTB,r16
;PORT Addresss valid to Data Valid in PIN = tPHL + tPZL +
;1.5 clk= 41ns+30ns+1.5(62.5)ns = 164.75
;164.75/62.5 = 2.636 so we need 3 clk
    nop
    nop
    nop

    sei
Loop:
    rjmp Loop

int0_isr:
;fist we check the keys because of high priority
    ldi r20,8 ;to count which button is pressed
    in r17,PINA
    cmp r17,0xff ;means no key was pressed
    ;so the interrupt is called because of printer
    breq printer
key_find:
    call Delay20ms
    dec r20
    LSL r17 ;left shift the value of PIN
    BRCC key_found
;if the carry bit cleared we've found it
    rjmp key_find ;again left shift to find the key
key_found:
    mov r18,r20 ;the pressed key is in r18

```



```

        ret

printer
;first we have to define PORT A as Output
    ldi r16,0xff
    out DDRA,r16
    ldi r16,0x02
    out PORTB , r16 ;output strobe High

    ldi r21,0x00 ;counter for number of characters
;give the number of pressed key to PORT A from r18
    out PORTA , r18
    CBI PORTB, 1 ;CP=0
    SBI PORTB, 1 ;CP=1
    CBI PORTB, 2 ;printer strobe Low
    ;500ns / 62.5 ns = 8 pulse clk
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    SBI PORTB, 2 ;printer strobe High
    inc r21      ;update the counter
    ;CP R21 , Number_of_values
    ;BRNE printer
    reti

```