

به نام خدا

تکلیف سری دهم درس ریزپردازنده

(۱)

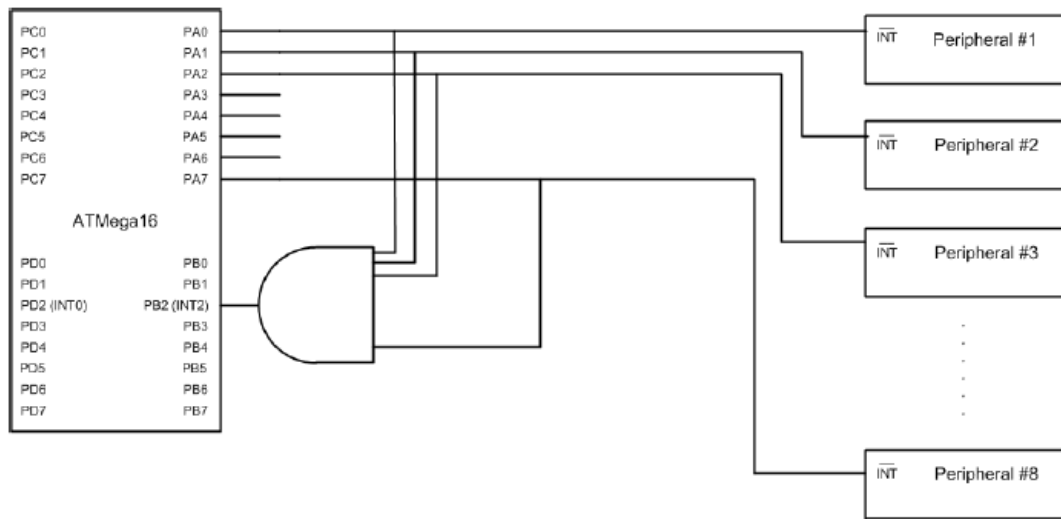
چرا نمیتوان یک لچ را به عنوان پورت ورودی و یک بافر را به عنوان پورت خروجی استفاده کرد؟

ویژگی لچ: یک قطعه دیجیتال است که ورودی را بدون اینکه تغییر بدهد در خود نگه میدارد تا زمانی که clear شود. ویژگی بافر: یک قطعه ی دیجیتال یا آنالوگ است که توان سیگنال ورودی را زیاد میکند بی آنکه مقدارش را تغییر بدهد. در حالت کلی لچ در خروجی به ما اجازه میدهد که داده ی ورودی به سرعت در خروجی قرار بگیرد. بافر در ورودی توانایی خواندن اطلاعات از ورودی و قرار دادن آن در رجیستر را به میکروکنترلر می دهد.

(۲)

چه راههایی را در عمل میتوان برای رعایت اولویت گذاری برای وسایل متقاضی وقفه استفاده نمود؟

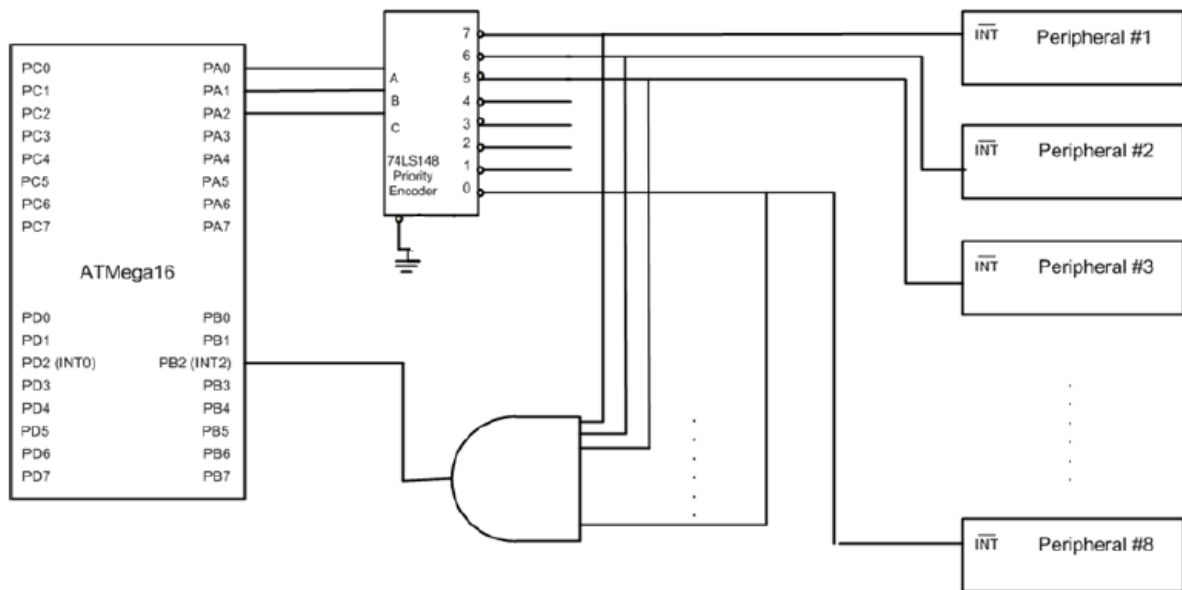
حالت کلی در خواست و رسیدگی به رقفه ها:



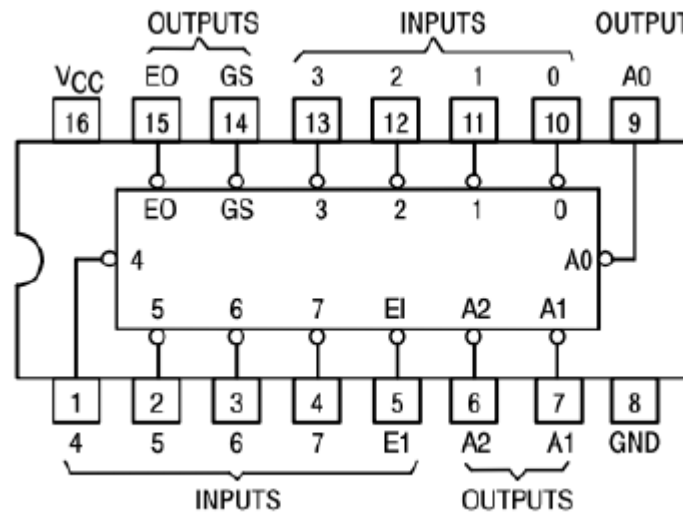
بعد از اینکه وقفه روی پایه ی INT0(PB2) دریافت شد، باید تمام بیت های پورت A را یکی یکی چک کنیم تا ببینیم کدام وسیله وقفه داده است.

اگر دو وسیله همزمان درخواست اجرای وقفه کنند:

در روش I/O وقفه گرا، اگر دو وسیله همزمان درخواست اجرای وقفه کنند، از طریق استفاده کردن از یک تراشه ی دیکودر اولویت دار (مانند 74LS148) استفاده میشود. بدین صورت که هر بار وقفه ی همزمان داده شد، وسایل وقفه دهنده که به ترتیب از بالا تا پایین قرار داده شده اند، انکودر وسیله ی با اولویت بیشتر را کدگذاری میکند و به سه بیت کم ارزش پورت A تحویل میدهد.



ساختار دیکودر اولویت دار



INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

روش دیگر، روش Daisy Chaining است:

در این روش ارتباط سریال بین وسیله ها برقرار است. وسایل هر کدام به ترتیب در زمان های خاصی سیگنال وقفه را منتشر میکنند.

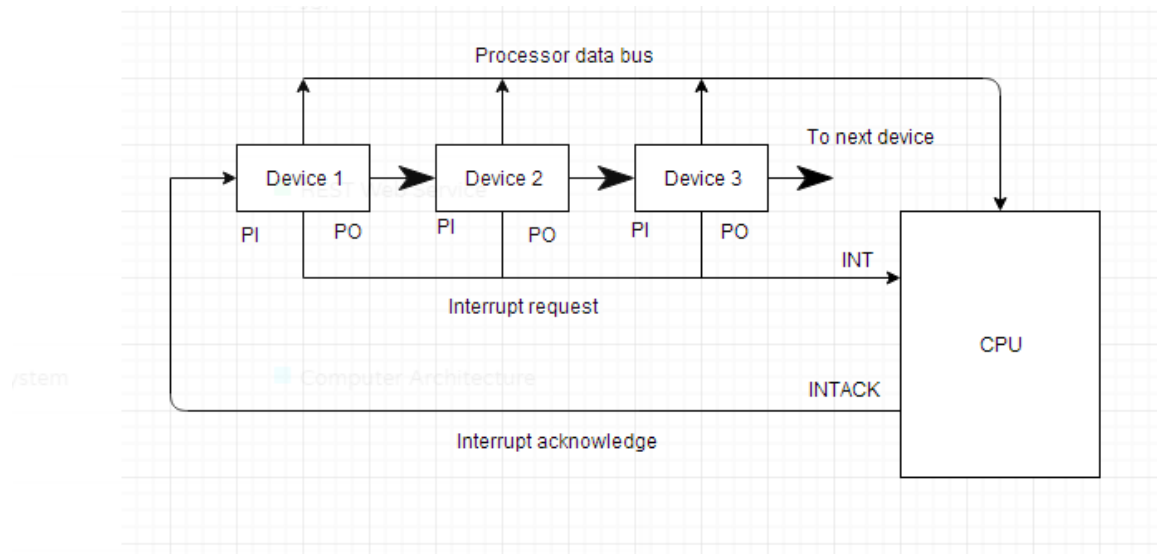
وسیله ای که بیشترین اولویت را دارد، سر زنجیره و وسیله ای که کمترین اولویت را دارد، در انتهای زنجیره قرار میگیرد.

خط وقفه برای تمامی وسایل، یکسان است (چون ارتباط سریال است)

در حالتی که هیچ سیگنال وقفه ای در کار نباشد، خط وقفه در سطح low قرار دارد.

پروسسور با فعالسازی سیگنال acknowledge به وقفه پاسخ میدهد. این سیگنال توسط وسیله ی وقفه دهنده دریافت و رسیدگی میشود و

چون به ترتب اولویت در زنجیره چیده شده اند، وقتی به وقفه رسیدگی شد، ack را به وسیله ی بعدی تحویل میدهد.



(۳)

آیا اولویت دهی به تقاضای سرویس وسایل جانبی در روش سرکشی قابل انجام است؟

در روش O/اوقفه گرا، اگر دو وسیله همزمان درخواست اجرای وقفه کنند، میتوان از طریق استفاده کردن از یک تراشه ی دیکودر اولویت دار (مانند 74LS148) صورت میگیرد بدین صورت که هر بار وقفه ی همزمان داده شد، وسایل وقفه دهنده که به ترتیب از بالا تا پایین قرار داده شده اند، انکودر وسیله ی با اولویت بیشتر را کدگذاری میکند و به سه بیت کم ارزش پورت A تحویل میدهد.

و اما در روش سرکشی :

اگر بخواهیم واحد سی پی یو را در یک سیستم مالتی تسکینگ راه اندازی کنیم، روش سرکشی برای کار با دستگاه های جانبی کندی مثل پرینتر مناسب نیست.

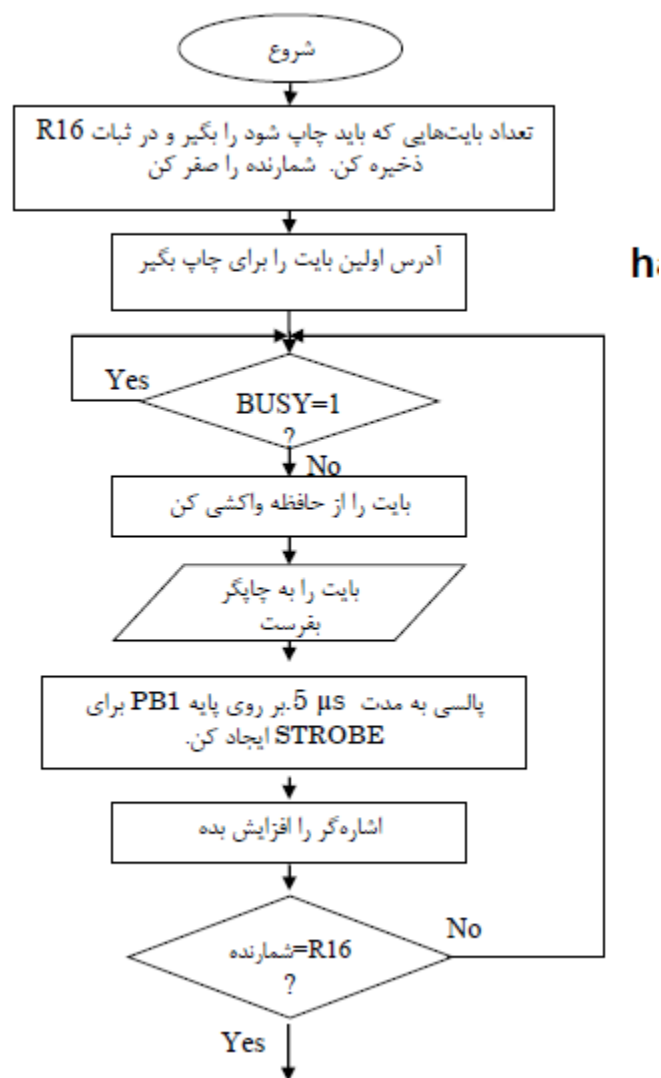
روش سرکشی در کل اگرچه پیاده سازی سخت افزاری و نرم افزاری ساده ای دارد، اما در استفاده از منابع کامپیوتری، ناکارآمد است.

چون روش سرکشی یک روش کاملاً نرم افزاری است (برخلاف روش وقفه گرا)، باید اولویت دهی به وسایل وقفه دهنده نیز، به صورت نرم افزاری کنترل شود. به این صورت که در حلقه ی موجود برای چک کردن بیت BUSY (در شکل زیر)، لازم است اول وسیله ی دارای اولویت چک شود بعد سراغ بقیه برویم. (فرض میکنیم اولویت ها را میدانیم و فقط میخواهیم رعایت شان کنیم)

پس باید در حلقه ی مورد نظر اول وسایلی را که اولویت بیشتری دارند چک کنیم و بعد سراغ بقیه برویم. (چون روش سخت افزاری نیست امکان گذاشتن دیکودر برای هندل کردن این مشکل وجود ندارد)

در حالت کلی، بله این روش قابل انجام است اما شیوه ی پیاده سازی آن ماهیتاً نرم افزاری است و نه سخت افزاری.

مثال برای پرینتر:

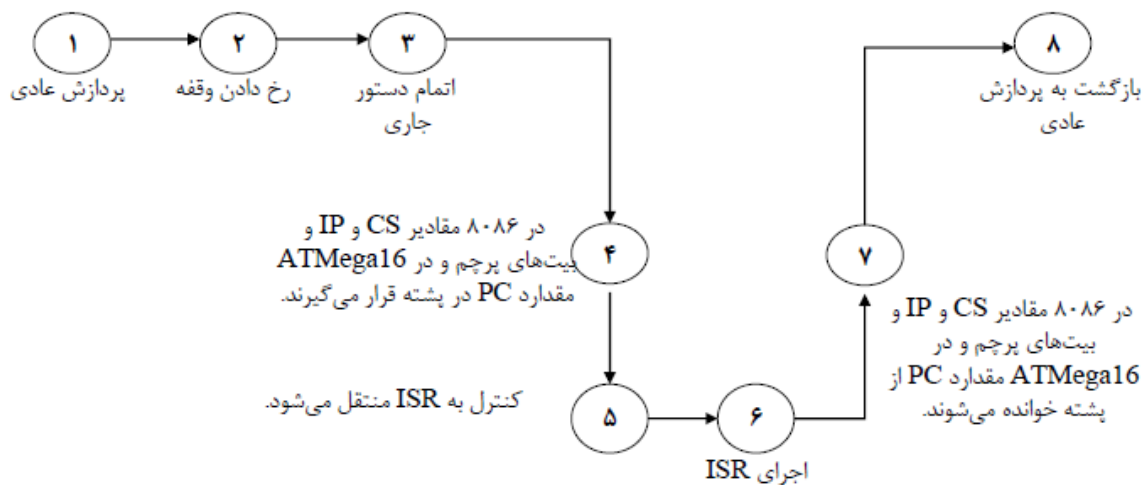


باید حلقه ها بر ای هر وسیله تکرار شوند.

(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؟

در انتهای اجرای هر دستوری، CPU خط وقفه را چک میکند و اگر این ورودی و بیت فعال ساز وقفه فعال باشند، کنترل برنامه به مکان خاصی از حافظه که روتین سرویس وقفه نامیده میشود (Interrupt service ISR routine)، منتقل میشود.



شکل ۱۷- فرآیند انجام شونده بعد از رخ دادن وقفه

به ازای هر منبع ایجاد وقفه، یک بیت فعال ساز وقفه وجود دارد که تا این بیت و بیت فعال‌ساز وقفه ی سراسری، یک نشوند، آن وقفه پذیرفته نمیشود.

در برخی موارد، یک پرچم هم وجود دارد که چنانچه آن وقفه رخ دهد، یک میشود.

آنچه که از برنامه ی اصلی، در بدو ورود به روتین وقفه باید حفظ و نگه داری شود، ثبات وضعیت (status register) است.

اطلاعات این ثبات، با ورود به روال وقفه به طور خودکار ذخیره نمیشود و هنگام بازگشت از روال وقفه، بازیابی نمیشود. اینکار میتواند توسط نرم افزار انجام بشود.

محتوای این ثبات، شامل تعدادی پرچم (مثل zero flag و carry flag و parity flag و ...) است.

(بیت های پرچم را که وضعیت کلی ریزپردازنده را نشان میدهند نگه میدارد)

در زمان ورود به روتین وقفه باید در حافظه ی برنامه ذخیره شود و در آخر بازیابی شود.

(۵)

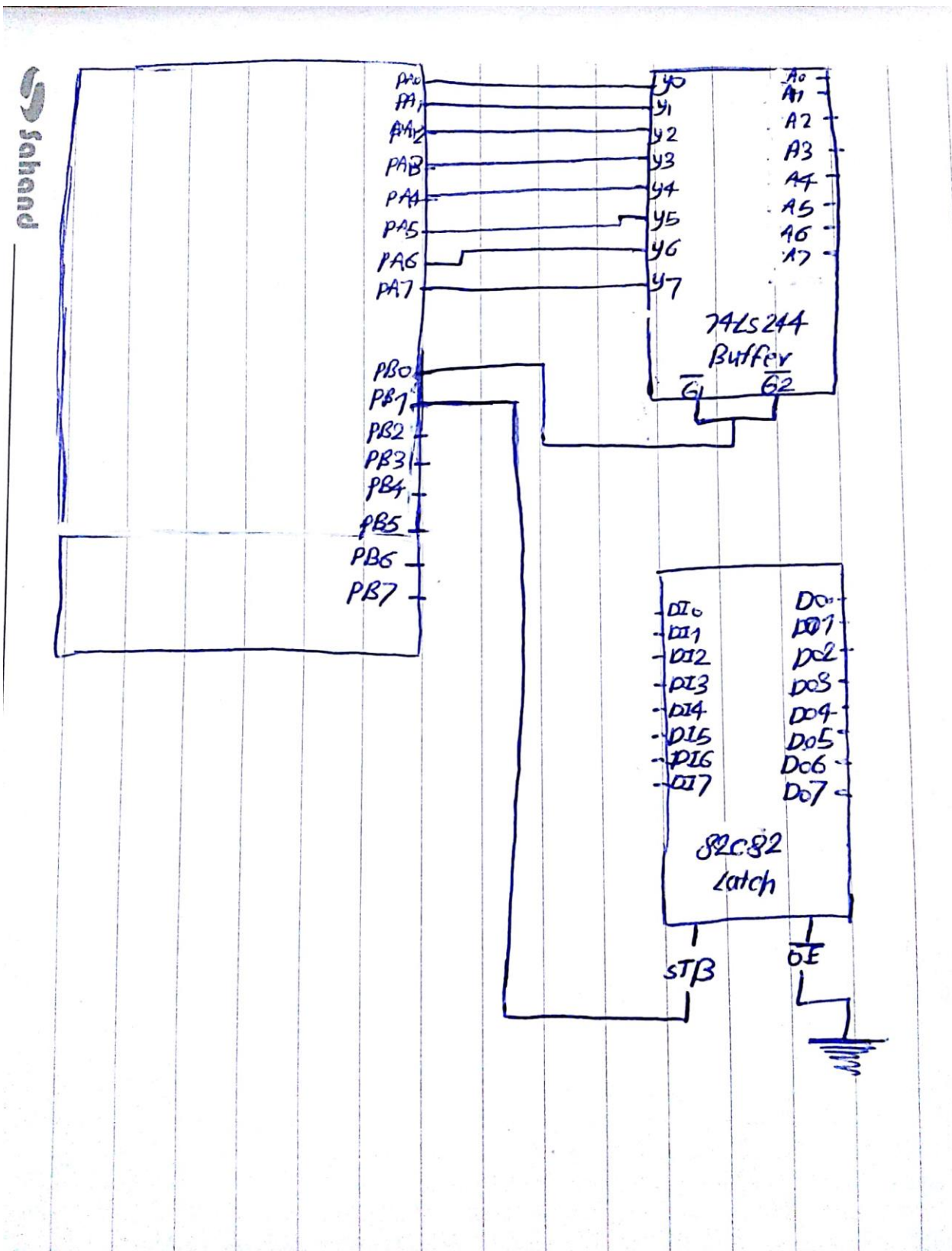
اقدامات زیر را انجام دهید.

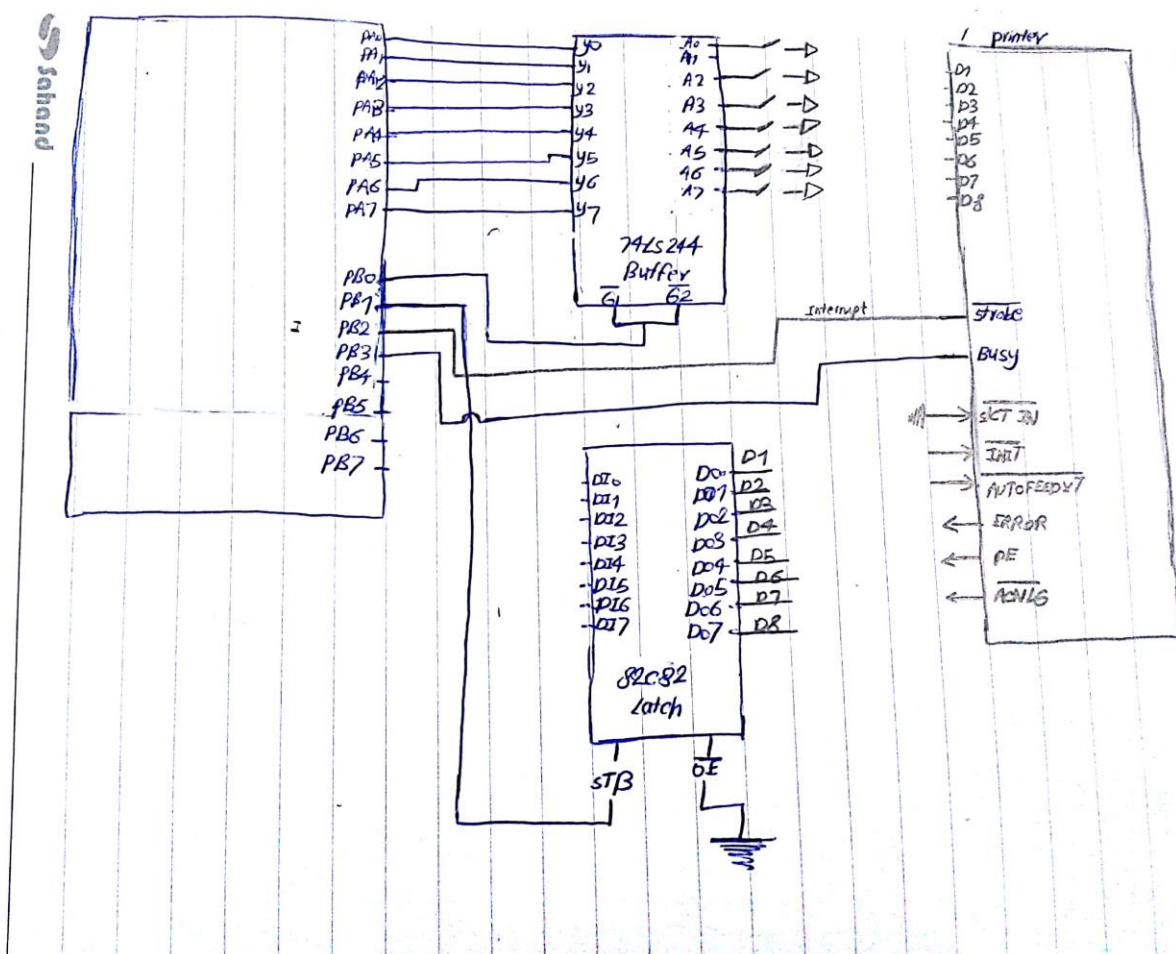
(الف)

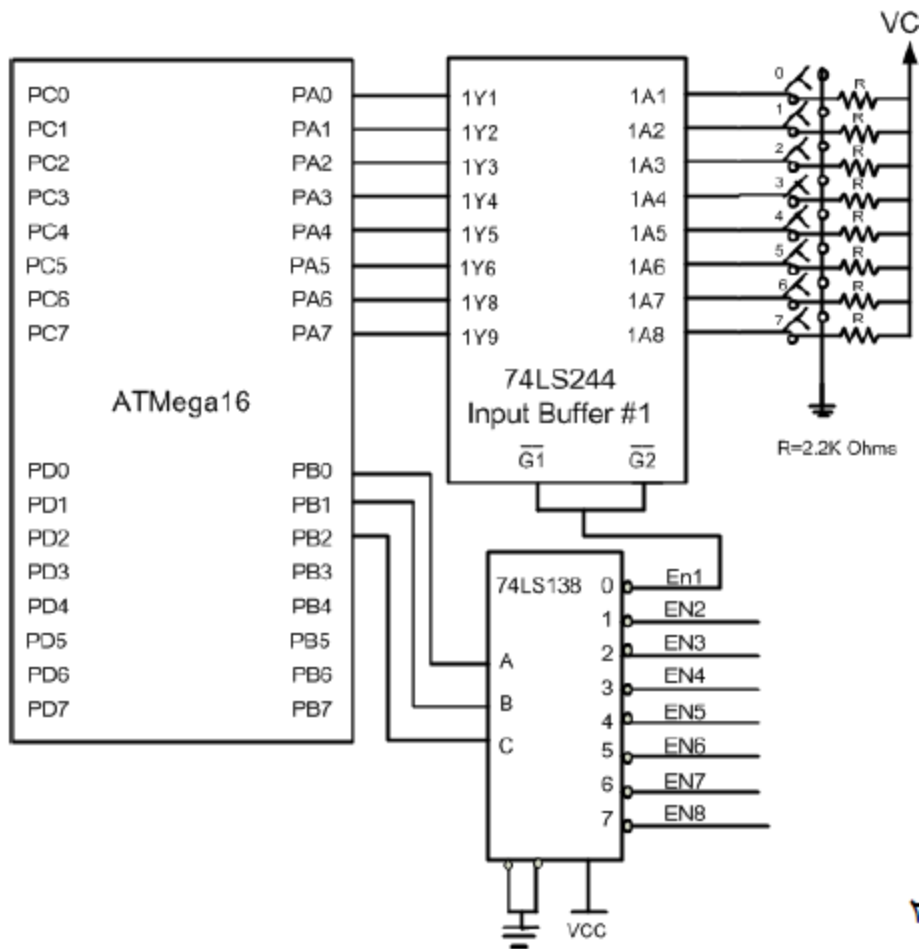
یک درگاه ورودی (بافر) 74LS244 - و یک خروجی هشت بیتی (لچ) 82C82 به نحو مناسب به میکروکنترلر ATmega16 متصل نمایید و دیکودینگ مورد نیاز برای فعالسازی هر دو را ارائه نمایید. از پورت A برای نقل و انتقال داده بین میکروکنترلر و درگاههای ورودی و خروجی و از بیتهای پورت B برای فعالسازی این درگاهها استفاده نمایید

Atmega16 +(latch) 82C82+ (buffer)74LS244

(پاسخ در صفحه ی بعد آورده شده است)



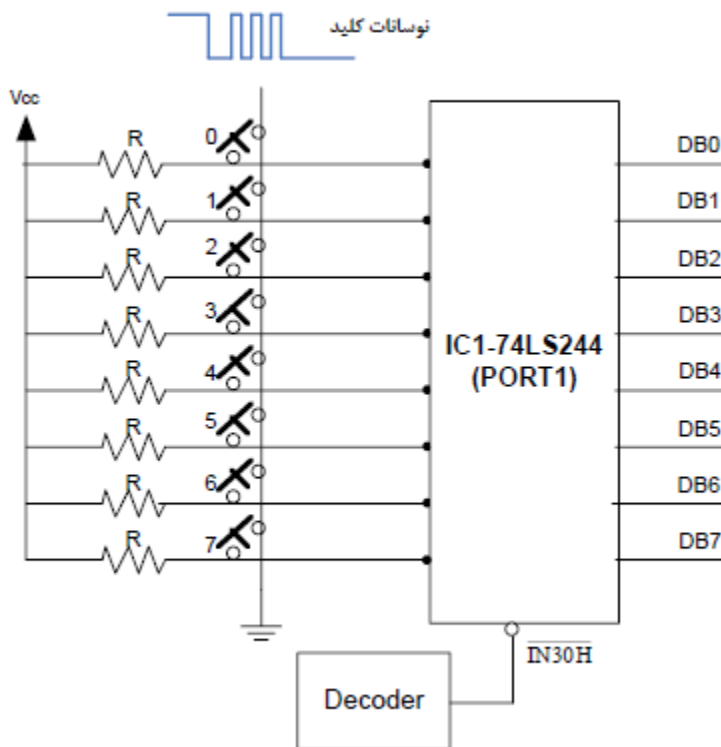




(ب)

درگاه ورودی را به ۸ کلید که هر کلید میتواند در وضعیت باز یا بسته قرار گیرد و درگاه خروجی را به یک پرینتر متصل - نمائید

برنامه دریافت داده از درگاه متصل به کلیدها و نیز برنامه نوشتن داده در درگاه متصل به پرینتر را بنویسید. برنامه را به گونهای بنویسید که زمانبندی مناسب در خواندن درگاه ورودی و نوشتن در درگاه خروجی مراعات گردد.



برنامه ی خواندن از کیبورد

:

```
Start:      LDI R24, 0b00110111
            OUT DDRB, R24 ; PORTB
```

CALL Read

```
Read :      LDI R24, 0x00
            OUT DDRA, R24 ; PORTA is Input
```

```
LDI R24, (PB0<<0) | (PB1<<1)|(PB2<<1)
OUT PORTB, R24 ;
NOP
NOP
NOP
```

```

LDI R20, 0x8 ; R20 will finally contain the No. of the pressed Key
LOOP1: IN R16, PINA ; Read Value from Input Buffer #1
        CMP R16, 0xFF
        BREQ LOOP1 ; If R16=0xFF means that no Key was Pressed
        RCALL Delay20ms ; Call a 20ms Delay if any key was pressed
LOOP2: DEC R20; ;
        LSL R16 ; Shift left the Value read from Keyboard
        BRCC LOOP3 ; Branch if Carry Flag is Cleared, so the pressed Key is
        detected
        RJMP LOOP2
LOOP3: MOV R0, R20; ; Now R0 Contains the No. of pressed key

Ret

```

برنامه ی ارتباط با پرینتر:

Read 16 Values from Address 0100H and write them to the printer.
 ; R30=01, R31=00H (Z=0x100)
 ; R16: Number of characters to be printed
 ; T1=input to latch STB setup time=10ns ; STB means CP input
 ; T2=Input to latch STB Hold time=25ns ; T3=STB Low time=25ns

CALL Printer

```

Print: LDI R24, 0xFFH
        OUT DDRA, R24 ; PORTA is Output
        LDI R20, 0x00 ; R20 as a counter for printed bytes

```

```

LDI R24, (PB0<<1) | PB1<<0)
OUT PORTB, R24 ;

```

```

:
LOOP1: SBIC PINB, 03 ; Skip if PB2 (Busy signal) is clear (Ready) [1/3] +
        RJMP LOOP1 ; Pooling of BUSY signal [3] +
        LD R21, Z+ ; [2] *
        OUT PORTA, R21 ; [1] *
        CBI PORTB, 0 ; CP=0 [1]
        SBI PORTB, 0 ; CP=1 T1, T2 and T3 times are satisfied [1]
        CBI PORTB, 1 ; STROBE=0 0.5Us=8*62.5ns [1]

```

```

NOP [1]
NOP [1]
NOP [1]
NOP [1]
NOP [1]

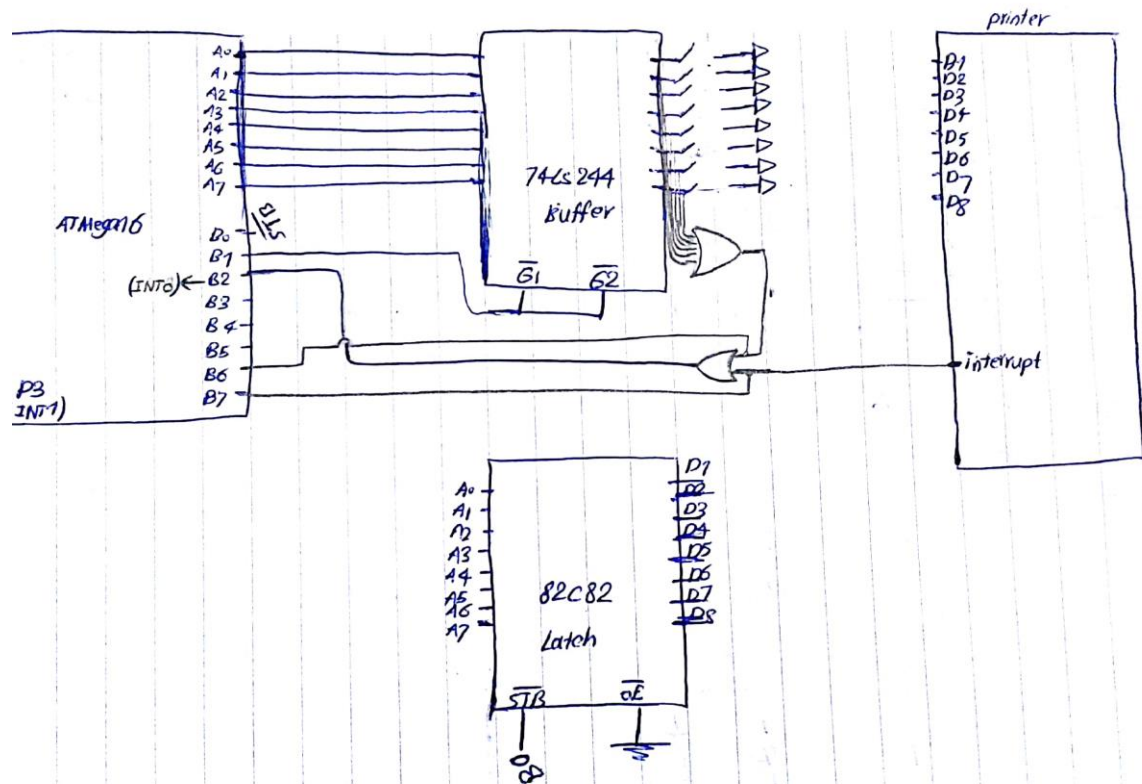
```

```
NOP [1]
NOP [1]
NOP [1]
SBI PORTB, 01 ; STROBE=1 [1]
INC R20 ; [1]
CP R20, R16 ; [1]
BRNE LOOP1 ; Check if Z Flag is 0 [1/2]
RET
```

(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 وقفهها به ورودی وقفه INT0 و تشخیص اینکه کدام وسیله (مجموعه کلیدها یا پرینتر) وقفه داده است را ارائه کنید.

(پاسخ در صفحه ی بعد)



2

sehmed

(د)

برنامه روتین وقفه INT0 - برای مدیریت وقفهها 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.

```
.org 0x002
RJMP int_handler

Int_handler :
    SBIS PINB,6
    RJMP Read
    SBIS PINB,7
    RJMP Print
RETI
```