تکلیف 10 درس ریز پردازنده

آریا بنایی زاده 9431029

# 1.

ماهیت لچ و بافر متفاوت هستند و بسته به کاربرد هر کدام در جای مخصوص استفاده میشوند اما شاید مهمترین دلیل آن این باشد که برای استفاده از لچ نیاز به low و high کردن بیت cp داریم تا داده بعدی وارد آن شود ( یعنی باید بدانیم چه موقعی میخواهیم داده جدید در آن ذخیره شود) و چون هنگام آمدن ورودی این را نمیدانیم از لچ برایورت ورودی استفاده کنیم.

# 2.

بهترین راه استفاده از یک PRIORITY ENCODER است که هر چه دستگاه اولویت بیشتری دارد باید در مرتبه کمتری آن را قرار دهیم.

میتوان به وسیله چک کردن پورت های متناظر با دستگاه ها نیز اولویت را مشخص کرد؛ مثلا از پورت صفر شروع به چک کردن بیت READY میکنیم و تا پورت هفت پیش میرویم. دستگاه پر اهمیت تر را در پورت صفر قرار میدهیم و همینطور به ترتیب اهمیت پیش میرویم.

# 3.

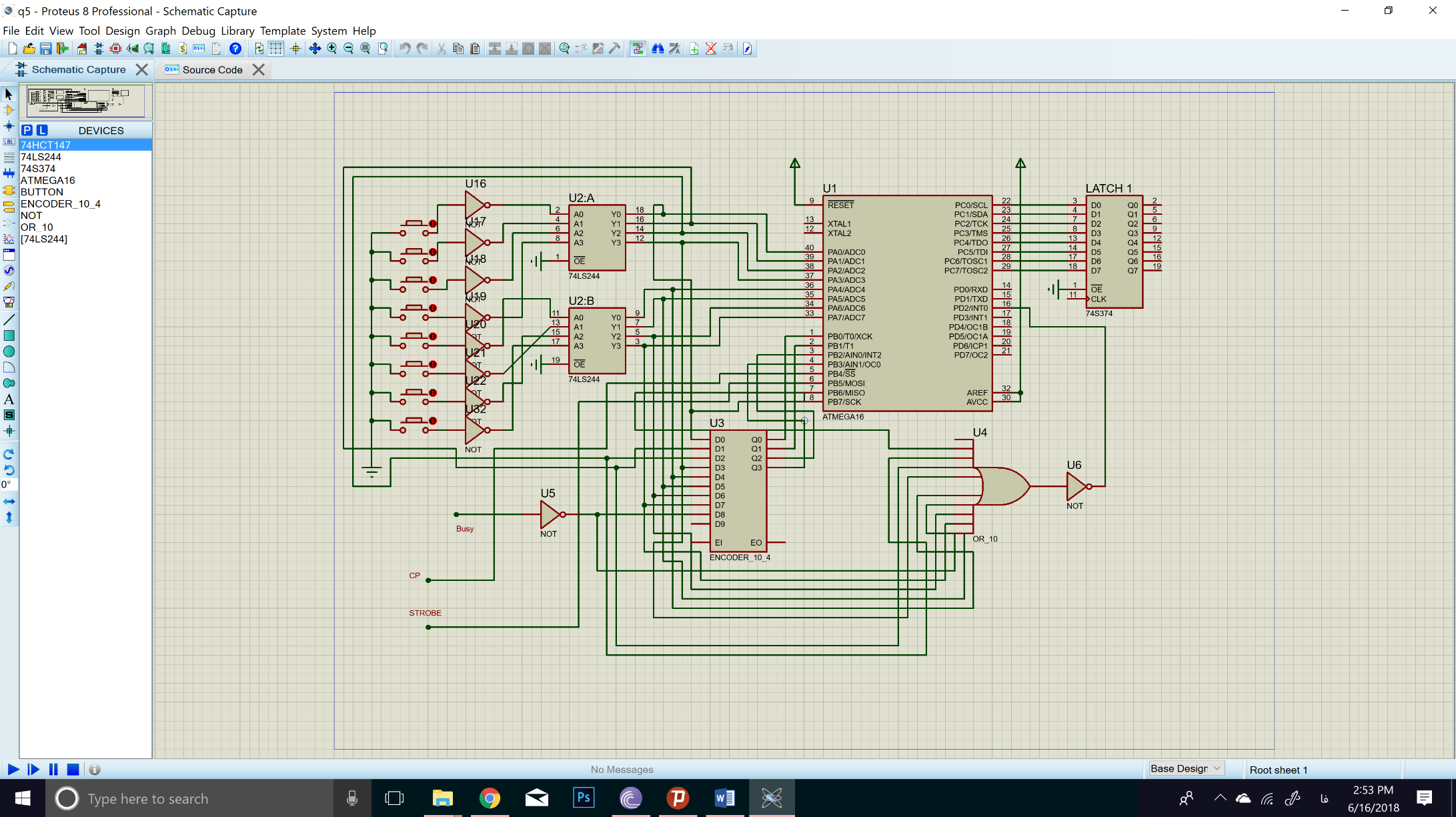
بله میتوان دائم پورت متناظر با دستگاه ها را چک کرد و هرکدام در حالت حاضر بود روتین مربوط به ارتباط با آن دستگاه استفاده شود. نحوه استفاده و اولویت بندی در سوال دو توضیح داده شده است.

# 4.

چون پردازنده پس از اجرای روتین وقفه باید به ادامه برنامه اصلی بازگردد قبل از رفتن به سراغ روتین وقفه آدرس مربوط به دستوری که در برنامه اصلی باید ادامه دهد(PC) را در پشته ذخیره میکند و پس از اتمام وقفه آن را بازیابی میکند.

# 5.

قطعات را با توجه به شکل به هم وصل میکنیم:

بافر ورودی را به پورت A و بافر خروجی را به پورت C وصل کرده کلید های کیبورد را به بافر ورودی وصل میکنیم و خروجی لچ را به بخش دیتای پرینتر و پورت ها PB4 را به CP و PB5 را به Strobe وصل میکنیم. یک priority encoder را به 4 بیت اول پورت B متصل میکنیم (از انکودر 10 به 4 استفاده میکنیم) حال به هشت بیت اول آن کلید های کیبورد و به بیت نهم کلید BUSY پرینتر وصل میکنیم حال برای ایجاد وقفه از یک مدار or استفاده میکنیم و تمام کلید ها و سیگنال busy را به آن میدهیم (بیت آن را بر عکس میکنیم) در نهایت نتیجه or را برعکس کرده و به وقفه خارجی صفر میدهیم. حل از طریق وقفه صفر و انکودر مینوانیم بفهمیم که به کیبورد باید سرویس بدهیم یا به پرینتر.

با توجه به تنظیمات انجام شده کد برنامه را به صورت زیر اعمال میکنیم:

تنظیمات اولیه : (ست کردن وقفه ها)

.org 0x000

rjmp SP\_INIT

.org 0x0002

rjmp intCheck

indSP\_INIT:

LDI R16, low(RAMEND)

OUT SPL, R16

LDI R16, high(RAMEND)

OUT SPH, R16

Start:

init:

LDI R24, 0x00

OUT DDRA, R24 ; PORTA is Input

LDI R24, 0b11110000

OUT DDRB, R24 ; PORTB ddr set

LDI R24, 0xFF

OUT DDRC, R24 ; PORTC is Output

LDI R24, 0x00 ;

OUT PORTB, R24 ;

ldi r16,(1<<INT0) ; set int 0 flags

out GICR,r16

ldi r16,(1<<ISC11)|(0<<ISC10)out MCUCR,r16 ; activates on falling edge

out MCUCR,r16

sei

NOP

NOP

NOP

تابع مربوط به اجرای وقفه:

intCheck:

in r16,PORTB

CMPI r16,8

BREQ PRINT

RJMP FINDKEY

تابع چک کردن کیبورد:

FINDKEY:

LDI R20, 0x8 ; R20 will finally contain the No. of the pressed Key

LOOP1:

IN R17, PINA ; Read Value from Input Buffer #1

CMP R17, 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 R17 ; 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

تابع پرینت:

Print:

LDI R20, 0x00 ; R20 as a counter for printed bytes

LOOP4:

OUT PORTA, R17

CBI PORTB, 4 ; CP=0

SBI PORTB, 4 ; CP=1 T1, T2 and T3 times are satisfied

CBI PORTB, 5 ; STROBE=0 0.5Us=8\*62.5ns

NOP

NOP

NOP

NOP

NOP

NOP

NOP

NOP

SBI PORTB, 5 ; STROBE=1

INC R20

CP R20, R16

BRNE LOOP1 ; Check if Z Flag is 0

rjmp LOOP