

# Primer kolokvijumskog zadatka

## Uvod u Proteus

8086 – I deo gradiva

IV termin

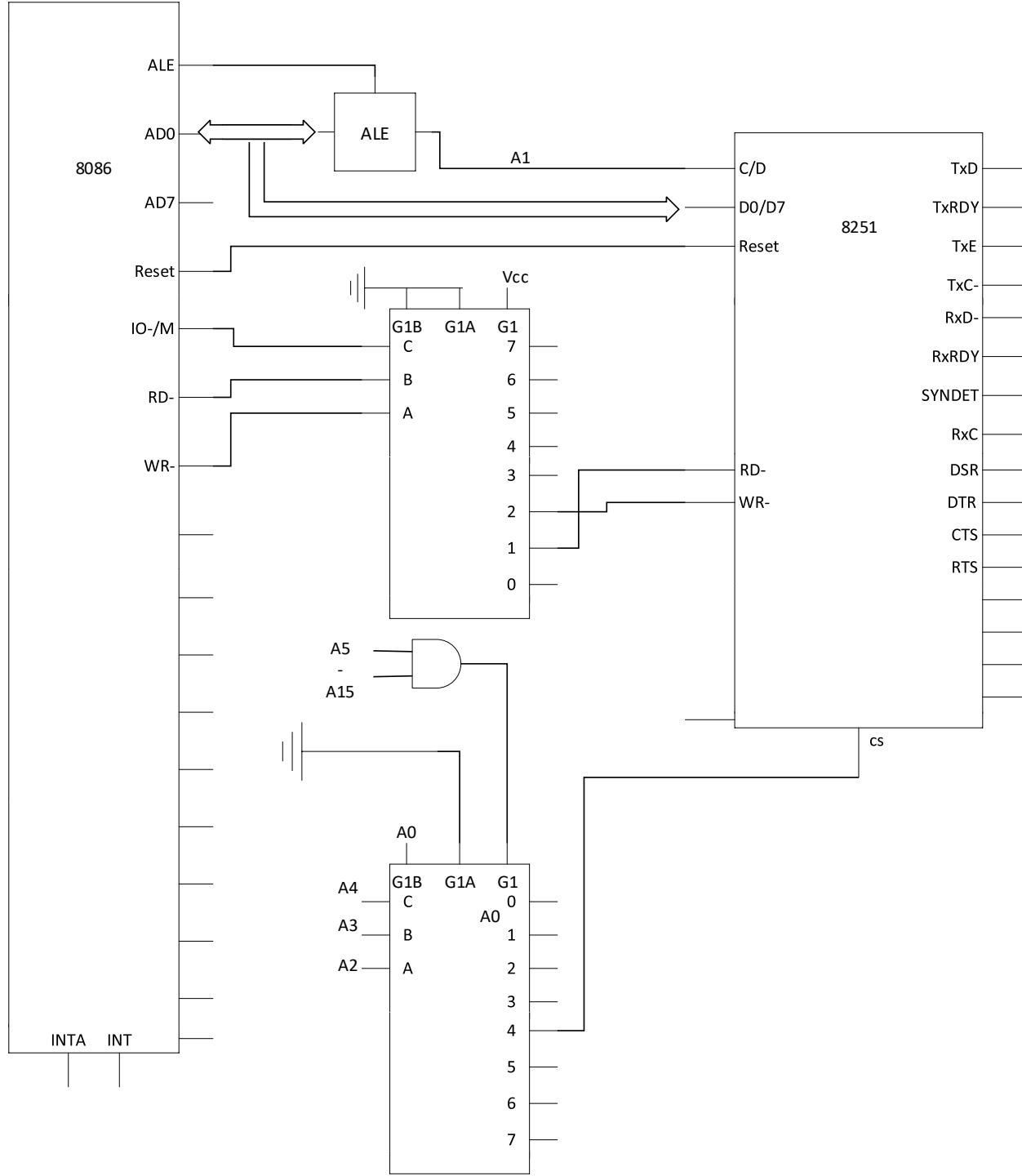
# Zadatak 4

- Za mikroprocesor iAPX8086 projektovati sistem koji obavlja prenos uz zamenu podataka. Prima se neprekidan niz podataka sa komponente 8251A i šalje nazad tako što se svaki primljeni podatak koji je jednak sa podatkom na lokaciji SEC zamenjuje sekvencom '0xFF 0xFF 0xFF' u odlaznom nizu.
- Ostali podaci se neizmenjeni šalju nazad. Prijem i slanje sekvenci obaviti koristeći ispitne petlje. Za privremeno čuvanje podataka koristiti niz veličine 128 na lokaciji BUFFER.
- U slučaju da u prijemnom baferu nema dovoljno mesta za prijem novih podataka ignorisati primljene podatke sve do oslobađanja mesta u baferu.
- Podaci se primaju i šalju kao **osmobitni** sa **parnim** bitom parnosti brzinom **64x** i slanjem **dva stop bita**.
- U slučaju pojave greške u prijemu obustaviti dalji prijem, isprazniti bafer slanjem do tada primljenih podataka i završiti program. Obezbediti prijem novih podataka kada komponenta nije spremna za slanje i obrnuto. Komponenta 8251A je na UI adresi 0xFFFF0.

# Adrese

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	8251 (FFF0)
1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	8251 (FFF2)

# Šema



# Mode word (8251)

- 0xFF

S2	S1	EP	PEN	L2	L1	B2	B1
Broj stop bitova: 01 – 1 stop bit 10 – 1.5 stop bit 11 – 2 stop bita		Bit parnosti: 00 – disable 01 – neparna 10 – disable 11- parna		Broj bitova po karakteru: 00-5b 01-6b 10 -7b 11-8b		Baud rate factor: 00-Sync 01-1x 10-16x 11-64x	
1	1	1	1	1	1	1	1

# Command word (8251)

- 0x15

EH	IR	RTS	ER	SBRK	RXE	DTR	TXEN
0:Normal 1:Hunt mode	0:Normal 1:Internal reset	0:DTR->1 1: DTR->0	0:Normal 1: Reset error flag	0:Normalan režim 1:Slanje karaktera prekida	Omogućiti prijemnik	0:DTR->1 1: DTR->0	Omogućiti predajnik
0	0	0	1	0	1	0	1

# Rešenje

```
name zad4
data segment
    broj1 db 0
    buffer db 128 dup (?)
    sec db 10h
data ends

stek segment
    bos dw 256 dup (?)
    tos label word
stek ends

code segment
    assume cs:code, ds:data, ss:stek
start:
    mov ax, data
    mov ds, ax
    mov ax, stek
    mov ss, ax
    lea sp, tos
```

```
;cli
mov al, 00h
mov dx, 0FFF2h
out dx, al
out dx, al
out dx, al
mov al, 40h
out dx, al
mov al, FFh
out dx, al
mov al, 15h
out dx, al
;sti
xor si, si
xor di, di
```

# Glavna petlja

petlja:

```
mov dx, 0FFF2h
in al, dx
test al, 38h
jnz greska
test al, 02h
jz slanje
call primi
```

slanje:

```
test al, 01h
jz petlja
call posalji
jmp petlja
```

greska:

```
call flush
mov ax, 4c02h
int 21h
```

- Status 8251

DSR	SYNDET BRKDET	FE	OE	PE	TxE	RxRDY	TxRDY
Data set ready	Sync detect/break error	Framing error	Overrun error	Parity error	Predajnik prazan	Prijemnik spreman	Predajnik spreman

- Tri pomoćne procedure

- primi
- posalji
- flush (za obradu greške)



# Prijem podataka

- 7fh
  - 0111 1111 - po modulu dužine bafera
  - Kada postane 1000 0000, obriše se i krene od 0

```
primi proc near
    pushf
    push ax
    push dx
    mov dx, 0FFF0h
    in al, dx
    cmp al, sec
    je specijalan
    cmp brojel, 128
    je kraj
    mov buffer[di], al
    inc di
    and di, 7fh
    inc brojel
    jmp kraj
specijalan:
    cmp brojel, 125
    ja kraj
    mov buffer[di], 0FFh
    inc di
    and di, 7fh
    mov buffer[di], 0FFh
    inc di
    and di, 7fh
    mov buffer[di], 0FFh
    inc di
    and di, 7fh
kraj:
    pop dx
    pop ax
    popf
    ret
primi endp
```

# Slanje podataka

```
posalji proc near
    pushf
    push ax
    push dx
    mov dx, 0FFF0h
    cmp brojel, 0
    je kraj
    mov al, buffer[si]
    out dx, al
    inc si
    and si, 7fh
    dec brojel
kraj:
    pop dx
    pop ax
    popf
posalji endp
```

# Flush

- Poziva se kada dođe do greške u prijemu podataka

```
flush proc near
    pushf
    push ax
    push dx
    push cx
    xor cx, cx
    mov cl, broj1
petlja2:
cekaj:
    mov dx, FFF2h
    in al, dx
    test al, 01h
    jz cekaj
    mov dx, FFF0h
    mov al, bufer[si]
    out dx, al
    inc si
    and si, 7fh
    loop petlja2
    pop cx
    pop dx
    pop ax
    popf
flush endp
code ends
end start
```

# Proteus 8 okruženje

- Proteus Design Suite 8 okruženje
  - <https://www.labcenter.com/>
  - dizajn mikroračunarskih sistema (Schematic Capture pogled)
  - editor za pisanje 8086 koda (Source Code pogled)
  - simulacija izvršenja koda
  - MASM32 kompajler za asemblerske programe pisane za 8086
- Verzije
  - Prva se pojavljuje 1988. godine
  - Preporuka: 8.4 SP0
- Operativni sistem
  - Windows
- Online tutorial
  - <https://www.youtube.com/watch?v=LnkhxSQIH18>
  - <https://www.youtube.com/watch?v=MEC7iiK6nV4>

# Zadatak – Proteus primer 1

- Za mikroprocesor iAPX 8086 projektovati mikroračunarski sistem sa jednim tasterom i jednom LED diodom uz pomoć jedne komponente 8255A. Na port A treba povezati plavu diodu (pin0), a na pin 0 porta B taster.
- Inicijalno dioda ne svetli. Nakon toga posle pritiska ovog tastera treba da zasvetli odgovarajuća dioda i treba da svetli sve dok je taster pritisnut.
- Koristi se IO-mapirani ulaz/izlaz (8bit adrese). Komponenta je na adresi 00h.

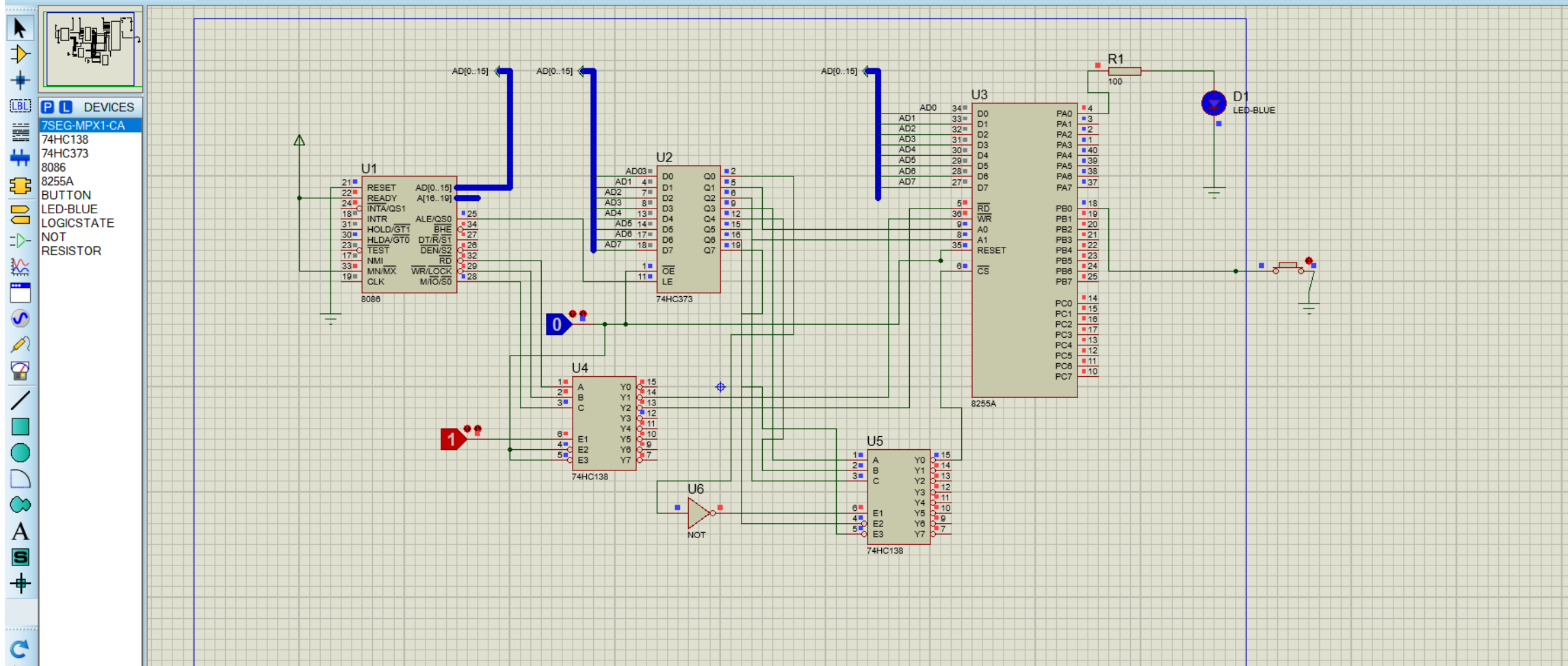
# Proteus šema

PROTEUS PRIMER1 - Proteus 8 Professional - Schematic Capture

File Edit View Tool Design Graph Debug Library Template System Help



Schematic Capture Source Code Physical Partlist View



# 8086 kod

- Konfiguraciona reč 1000 0010
- Port A izlazni
- Port B ulazni
- Taster aktivan na 0
- Dioda aktivna na 1
- Čita se sadržaj porta B
- Bit po bit se invertuje
- Šalje se na port A

```
DATA SEGMENT
    PORTA EQU 00H
    PORTB EQU 02H
    PORTC EQU 04H
    PORT_CON EQU 06H
DATA ENDS

CODE SEGMENT
    MOV AX, DATA
    MOV DS, AX
    ORG 0000H
START:
    MOV DX, PORT_CON
    MOV AL, 10000010B
    OUT DX, AL
    MOV DX, PORTA
    MOV AL, 00000000B
    OUT DX, AL
PETLJA:
    MOV DX, PORTB
    IN AL, DX
    MOV DX, PORTA
    NOT AL
    OUT DX, AL
    JMP PETLJA
CODE ENDS
END
```