

Zadatak 5

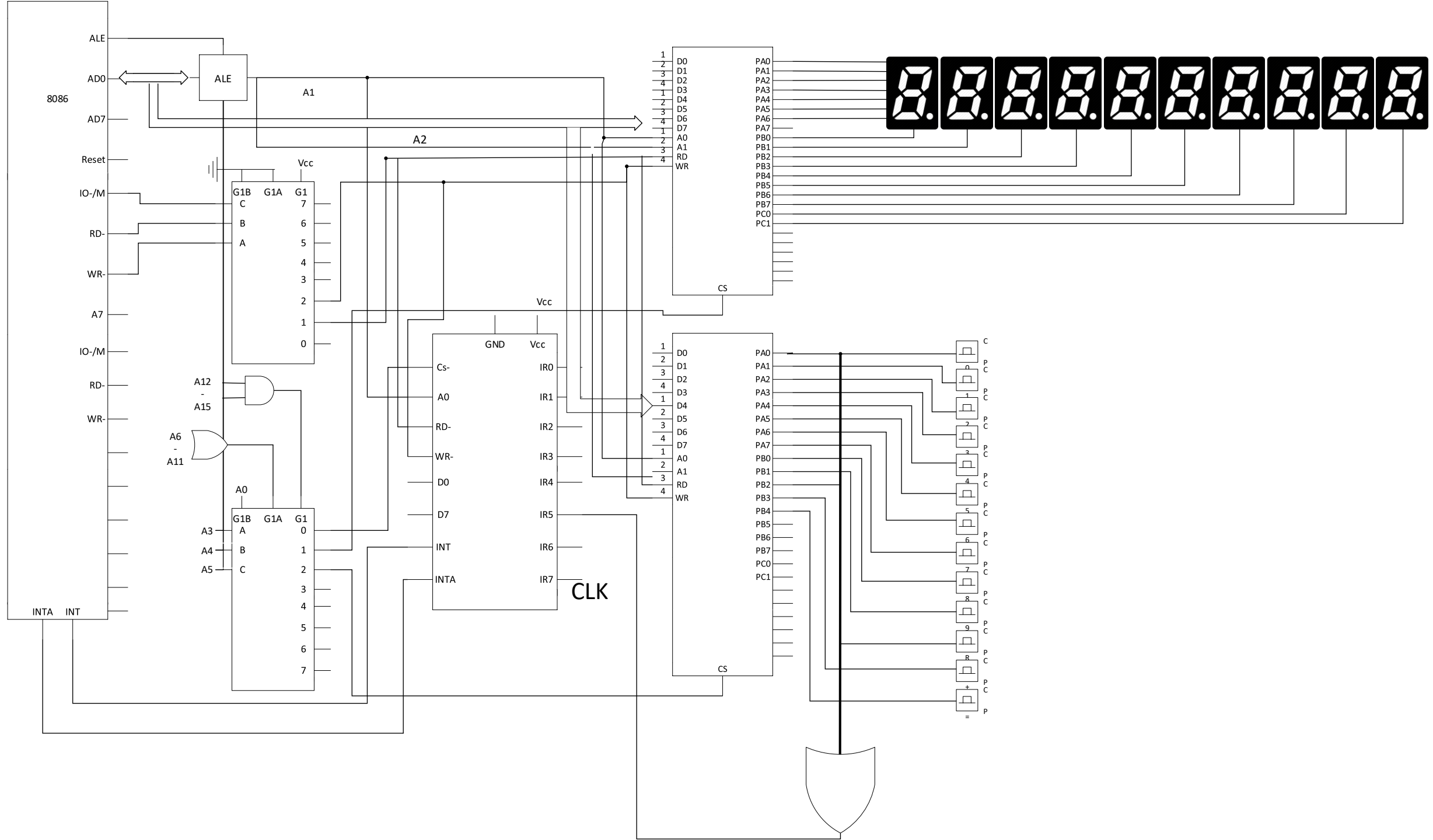
- Projektovati mikroračunarski sistem za kasu na bazi mikroprocesora iAPX8086 i dve 8255 komponente. Sistem sadrži 10 sedmosegmentnih displeja i tastaturu sa tasterima od 0 do 9, +, = i RESET taster. Pritiskom na RESET briše se trenutno stanje, dok pritiskom na = generiše prikaz zbira svih unetih iznosa.
- Napisati procedure koje će na displejima ispisivati ono što je otkucano na tastaturi, kao i ukupan iznos za plaćanje.
- Komponenta 8259 je na adresi F000h, 8255-1 na adresi F008H, 8255-2 na adresi F010H, a redni broj prekida najvišeg prioriteta je 117
- Za realizaciju displeja koristiti tehniku osvežavanja
- Jedan pin odgovara jednom tasteru

Adrese

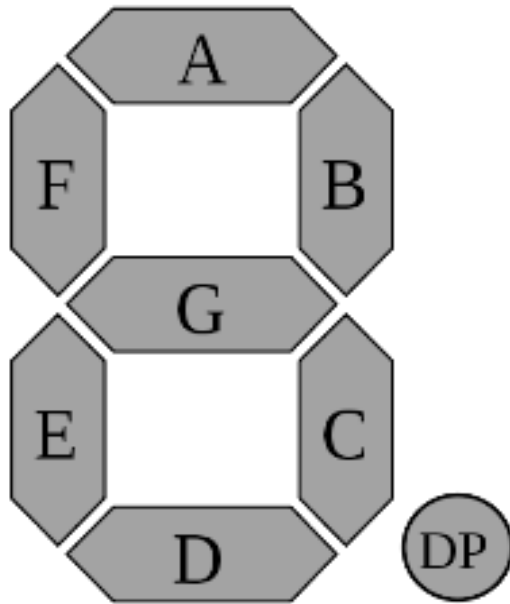
Komponenta	$A_{15} - A_{12}$	$A_{11} - A_8$	$A_7 - A_4$	$A_3 - A_0$	
8259	1111	0000	0000	0000	F000h
	1111	0000	0000	0010	F002h
	1111	0000	0000	1100	F00Ch PC
8255-1	1111	0000	0000	1110	F00Eh CTRL
	1111	0000	0000	1000	F008h PA
	1111	0000	0000	1010	F00Ah PB
	1111	0000	0001	0100	F014h PC
8255-2	1111	0000	0001	0110	F016h CTRL
	1111	0000	0001	0000	F010h PA
	1111	0000	0001	0010	F012h PB

Povezivanje

- 7s displeji: 8255-1
 - Segmenti povezani na PORTA A0-A6
 - Pojedinačni displeji povezani na PORTB (B0-B7) i PORTC (C0 i C1)
- Tasteri: 8255-2
 - 0-7 povezani na PORTA
 - 8, 9, RESET, +, = na PORTB



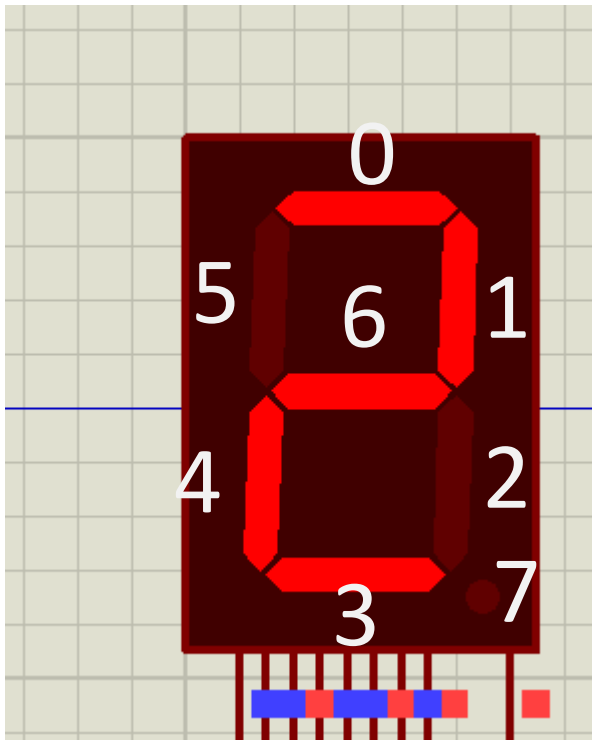
7s displej



PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0		
	a	b	c	d	e	f	g	code	value
0	1	1	1	1	1	1	0	7Eh	0
0	0	1	1	0	0	0	0	30h	1
0	1	1	0	1	1	0	1	6Dh	2
0	1	1	1	1	0	0	1	79h	3
0	0	1	1	0	0	1	1	33h	4
0	1	0	1	1	0	1	1	5Bh	5
0	1	0	1	1	1	1	1	5Fh	6
0	1	1	1	0	0	0	0	70h	7
0	1	1	1	1	1	1	1	7Fh	8
0	1	1	1	1	0	1	1	7Bh	9

7s displej u Proteus-u

Segment - aktivan na 0



PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	code
1	1	0	0	0	0	0	0	0
1	1	1	1	1	0	0	1	1
1	0	1	0	0	1	0	0	2
1	0	1	1	0	0	0	0	3
1	0	0	1	1	0	0	1	4
1	0	0	1	0	0	1	0	5
1	0	0	0	0	0	1	0	6
1	1	0	1	1	0	0	0	7
1	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	9

Rešenje

- DEFC:
 - kodovi cifara
- POZC:
 - gde je unos cifre
- DISPLAY:
 - niz prikazanih cifara
 - Ako je 1357, uneto u mem DISPLAY je 7531
- ROTOR:
 - koji će displej biti osvežen

```
EXTRN Taster_proc:far, Osvez_proc:far
PUBLIC DISPLAY, ZBIR, ROTOR, DEFC, POZC
```

```
DATA segment
```

```
    DEFC db 7eh, 30h, 6dh, 79h, 33h, 5bh,
    5fh, 70h, 7fh, 7bh
```

```
    DISPLAY db 10 dup (0)
```

```
    ZBIR db 10 dup (0)
```

```
    ROTOR dw 0
```

```
    POZC dw 0
```

```
DATA ends
```

```
STACK segment
```

```
    bos dw 256 dup (?)
```

```
    tos label word
```

```
STACK ends
```

```
CODE segment
```

```
    assume CS:CODE, DS:DATA, SS:STACK
```

```
start:
```

```
    mov AX, DATA
```

```
    mov DS, AX
```

```
    mov AX, STACK
```

```
    mov SS, AX
```

```
    lea SP, tos
```

```
    cli
```

8259 inicijalizacija

```
; 8259 inic.  
; ICW1  
mov DX, 0F000h  
mov AL, 13h  
out DX, AL  
; ICW2  
mov DX, 0F002h  
mov AL, 70h  
out DX, AL  
; ICW4  
mov AL, 03h  
out DX, AL  
; OCW1  
mov AL, 5Fh  
out DX, AL
```


ICW1

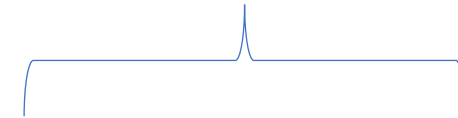
- 0x13

A7	A6	A5	1	LTIM	ADI	SNGL	IC4
A7-A5 adrese vektora interapta (samo za 80/85 mod) Za 8086 nebitno – stavljamo sve 0				0:Okidanje na ivicu (uzlaznu) 1:Okidanje na nivo (jedinica)	Nije relevantno za 8086	0:više od 1, kaskadno 1: jedan 8259	0: ICW4 nije potreban 1: ICW4 potreban
0	0	0	1	0	0	1	1

ICW2

- 117 = 01110101(bin)
- 0x70

Nije relevantno u toku
inicijalizacije



Interrupt	A7	A6	A5	A4	A3	D2	D1	D0
IR0	0	1	1	1	0	0	0	0
IR1						0	0	1
IR2						0	1	0
IR3						0	1	1
IR4						1	0	0
IR5						1	0	1
IR6						1	1	0
IR7						1	1	1

ICW4

- 0x03

D7	D6	D5	SFNM	BUF	M/S	AEOI	mPM
0	0	0	0: NOT SPECIAL FULLY NESTED 1: SPECIAL FULLY NESTED	0X – NON BUFFERED 10 – BUFFERED MODE/SLAVE 11 – BUFFERED MODE/MASTER		0: NORMAL 1: AUTO	0: MCS-80/85 MODE 1: 8086/8088 MODE
0	0	0	0	0	0	1	1

OCW1

- 0x5F
- Taster – viši prioritet
 - 117 = 01110101(bin)
- Refresh - niži
 - 119 = 01110111(bin)
- $112 + 5 = 117$
- $112 + 7 = 119$

M7	M6	M5	M4	M3	M2	M1	M0
0	1	0	1	1	1	1	1

Inicijalizacija 8255-1 i 8255-2

```
    ; 8255-1 inic.  
    mov DX, 0F00Eh  
    mov AL, 80h  
    out DX, AL  
    ; 8255-2 inic.  
    mov DX, 0F016h  
    mov AL, 92h  
    out DX, AL
```

Struktura kontrolne reči 8255-1

- Displej – output A, B i C
- 0x80

D7	D6	D5	D4	D3	D2	D1	D0
0- SET/RESET	00 – MODE0 01- MODE1		0 -PORTA OUT	0 – PORTC HIGHER OUT	0 - MODE0	0 – PORTB OUT	0 – PORTC LOWER OUT
1 - I/O MODE	1X – MODE2		1 – PORTA IN	1 – PORTC HIGHER IN	1 - MODE1	1- PORTB IN	1 - PORTC LOWER IN
	MOD GRUPE A		GRUPA A		MOD GRUPE B	GRUPA B	
1	0	0	0	0	0	0	0

Struktura kontrolne reči 8255-2

- Tastatura – input A i B
- 0x92

D7	D6	D5	D4	D3	D2	D1	D0
0- SET/RESET	00 – MODE0 01- MODE1 1X – MODE2 MOD GRUPE A		0 -PORTA OUT	0 – PORTC HIGHER OUT	0 - MODE0	0 – PORTB OUT	0 – PORTC LOWER OUT
1 - I/O MODE			1 – PORTA IN	1 – PORTC HIGHER IN	1 - MODE1	1- PORTB IN	1 - PORTC LOWER IN
			GRUPA A		MOD GRUPE B		GRUPA B
1	0	0	1	0	0	1	0

Inicijalizacija tabele prekida

- $4 * 117 = 468$
- $4 * 117 + 2 = 470$
- $117 + 2 = 119$
- $119 * 4 = 476$
- $119 * 4 + 2 = 478$

Proteus:

MOV SI, BROJ HEX

MOV [SI], ...

```
; Tabela prekida inic.
mov [468], OFFSET Taster_proc
mov [470], SEG Taster_proc
mov [476], OFFSET Osvez_proc
mov [478], SEG Osvez_proc
sti
main:
    jmp main
mov AX, 4C02h int 21h
CODE ends
end start
```


Osvežavanje displeja

- Prekidne i pomoćne procedure

```
; prekidne i pomocne procedure
PUBLIC Osvez_proc, Taster_proc
EXTRN DISPLAY:byte, ZBIR:byte, ROTOR:byte, DEFC:byte, POZC:word

PCODE SEGMENT
    assume CS: PCODE
```

- Prekidna procedura za osvežavanje

- ROTOR

- Shift jedinice za broj pozicija ROTOR
- 0x0200-0000 0010 0000 0000

```
Osvez_proc proc far
    pushf
    push AX
    push CX
    push DX
    push SI
    cli
    mov AL, ROTOR
    inc AL
    cmp AL, 10
    jl daljel
    mov AL, 0
daljel:
    mov ROTOR, AL
    mov AH, 0
    mov SI, AX
    mov AL, DISPLAY[SI]
    mov SI, AX
    mov AL, DEFC[SI]
    mov DX, 0F008h
    out DX, AL
    mov AX, 0200h
    mov CL, ROTOR
    shr AX, CL
    mov DX, 0F00Ah
    out DX, AL
    mov DX, 0F00Ch
    mov AL, AH
    out DX, AL
    sti
    pop SI
    pop DX
    pop CX
    pop AX
    popf
    iret
Osvez_proc endp
```

Pritisak tastera

- Prekidna procedura
- Prva petlja
 - Da li taster na PORTA?
 - SHIFT udesno i traži poziciju 1
 - BX čuva vrednost od 0 do 7
 - 0-7
- PORTB
 - Koji je to na PORTB?
 - 8, 9, RESET, +, =
 - Posebne labele

```
Taster_proc proc far
    assume CS: PCODE
    pushf
    push AX
    push DX
    push BX
    cli
    mov DX, 0F010h
    in AL, DX
    cmp AL, 0
    je portb
    mov BX, 0
    poredi:
        cmp AL, 1
        je nasao
        shr AL, 1
        inc BX
        jmp poredi
    nasao:
        call Cifra_proc
        jmp kraj
    portb:
        mov DX, 0F012h
        in AL, DX
        cmp AL, 1
        je losam
        cmp AL, 2
        je ldevet
        cmp AL, 4
        je lreset
        cmp AL, 8
        je lplus
        jmp ljednako
    losam:
        mov BX, 8
        call Cifra_proc
        jmp kraj
    ldevet:
        mov BX, 9
        call Cifra_proc
        jmp kraj
    lplus:
        call Plus_proc
        jmp kraj
    ljednako:
        call Jednako_proc
        jmp kraj
    lreset:
        call Reset_proc
    kraj:
        sti
        pop BX
        pop DX
        pop AX
        popf
        iret
Taster_proc endp
```

Cifra

- Pomoćna procedura
- Umeće cifru na prvo
- Ostatak SHIFT udesno (DISPLAY)

Primer

- 123 na displeju prikazano
- 1234 nakon unosa 4
- Niz pre
 - 321
- Niz posle
 - 4321

```
Cifra_proc proc near
pushf
push SI
push AX
mov SI, POZC
cmp SI, 10
je kraj
petlja:
    cmp SI, 0
    je novacifra
    dec SI
    mov AL, DISPLAY[SI]
    inc SI
    mov DISPLAY[SI], AL
    dec SI
    jmp petlja
novacifra:
    mov DISPLAY[SI], BL
    inc POZC
kraj:
    pop AX
    pop SI
    popf
    ret
Cifra_proc endp
```

Reset

- Pomoćna procedura

```
Reset_proc proc near
pushf
push SI
push CX
xor SI, SI
mov CX, 10
petlja:
    mov byte ptr DISPLAY[SI], 0
    mov byte ptr ZBIR[SI], 0
    inc SI
loop petlja

pop CX
pop SI
popf ret
Reset_proc endp
```

Jednako

- Pomoćna procedura
- Poslednje uneto doda na prethodi zbir i kopira niz zbir u DISPLAY

```
Jednako_proc proc near
pushf
push SI
push AX
push CX
call Plus_proc
xor SI, SI
mov CX, 10
petlja:
    mov AL, ZBIR[SI]
    mov DISPLAY[SI], AL
    inc SI
loop petlja
pop CX
pop AX
pop SI
popf
ret
Jednako_proc endp
```

Plus

- Pomoćna procedura
- Sabira do tada sakupljeno i poslednje uneto

```
Plus_proc proc near
    pushf
    push AX
    push SI
    xor SI, SI
    mov AH, 0
    petlja:
        mov AL, DISPLAY[SI]
        add AL, ZBIR[SI]
        add AL, AH
        mov AH, 0
        cmp AL, 10
        jl v2
        mov AH, 1
        sub AL, 10
        v2:
        mov ZBIR[SI], AL
        mov DISPLAY[SI], 0
        inc SI
        cmp SI, 9
    jbe petlja
    mov POZC, 0
    pop SI
    pop AX
    popf
    ret
Plus_proc endp
PCODE ends END
```

Interrupt code u Proteus-u

- Sve u main.asm
- PROC FAR procedure
- Tabela prekida

MOV SI, HEXADR_OFFSET(IP)

MOV [SI], OFFSET PREKID

MOV SI, HEXADR_SEG(CS)

MOV [SI], SEG PREKID

```
PROCED SEGMENT
PREKID PROC FAR
    ASSUME CS:PROCED, DS:DATA
    ORG HEX_ADDRESS1
;OVDE IDE PREKIDNA PROCEDURA
    IRET
PREKID ENDP
PROCED ENDS

DATA SEGMENT
; OVDE IDU MAKROI I PROMENLJIVE
DATA ENDS

; STACK
STEK SEGMENT STACK
;STEK SEGMENT
STEK ENDS

;MAIN CODE
CODE    SEGMENT PUBLIC 'CODE'
    ASSUME CS:CODE, DS:DATA, SS:STEK
    ORG HEX_ADDRESS2
START:
;GLAVNI PROGRAM
CODE    ENDS
END START
```