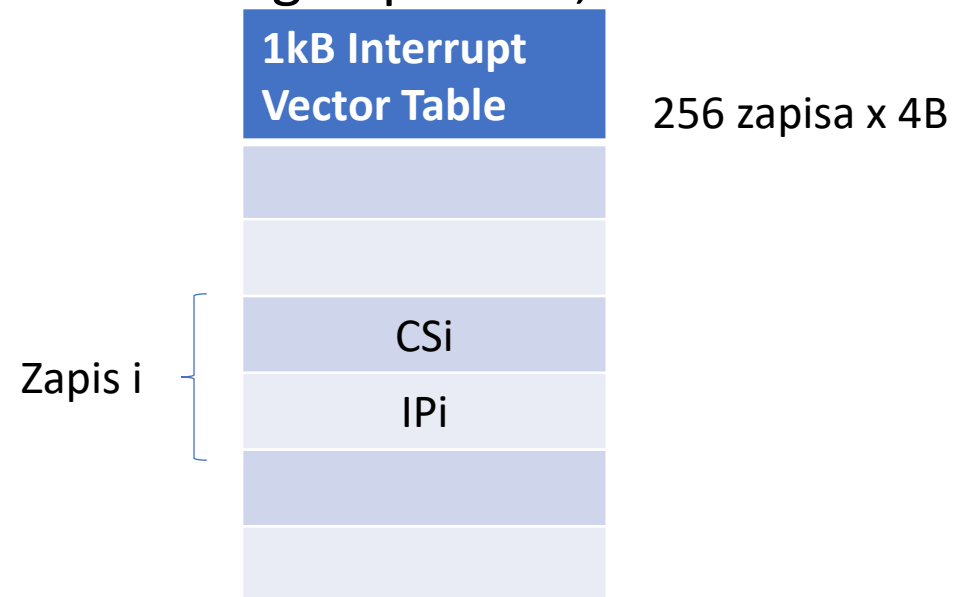


Kontroler prekida 8259

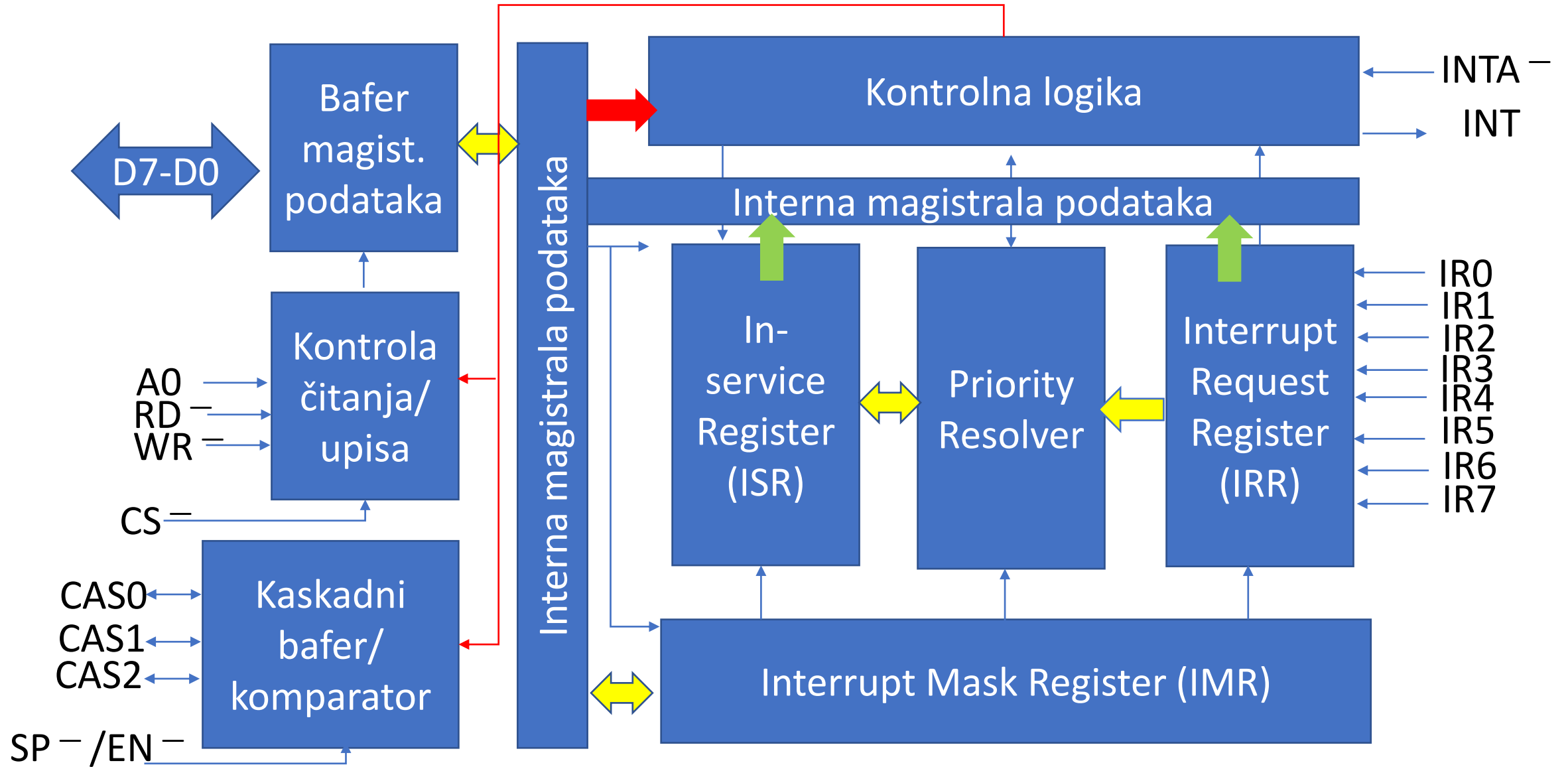
3. termin računskih vežbi

Tabela vektora prekida

- Prvi kilobajt memorije 8086 ima specijalnu namenu – tabela vektora prekida
- Zapis i tabele prekida je podeljen na 2 dela:
 - CSi koji označava adresu segmenta koda prekidne procedure
 - IPi koji označava pokazivač instrukcije prekidne procedure
- Zapis u tabeli prekida je veličine 4B, što znači da kada stigne prekid i, biće opslužen procedurom sa adresom $4i$
- Za ovakvu realizaciju prekida koristi se 8259A



Interna struktura 8259A



Blokovi mehanizma prekida

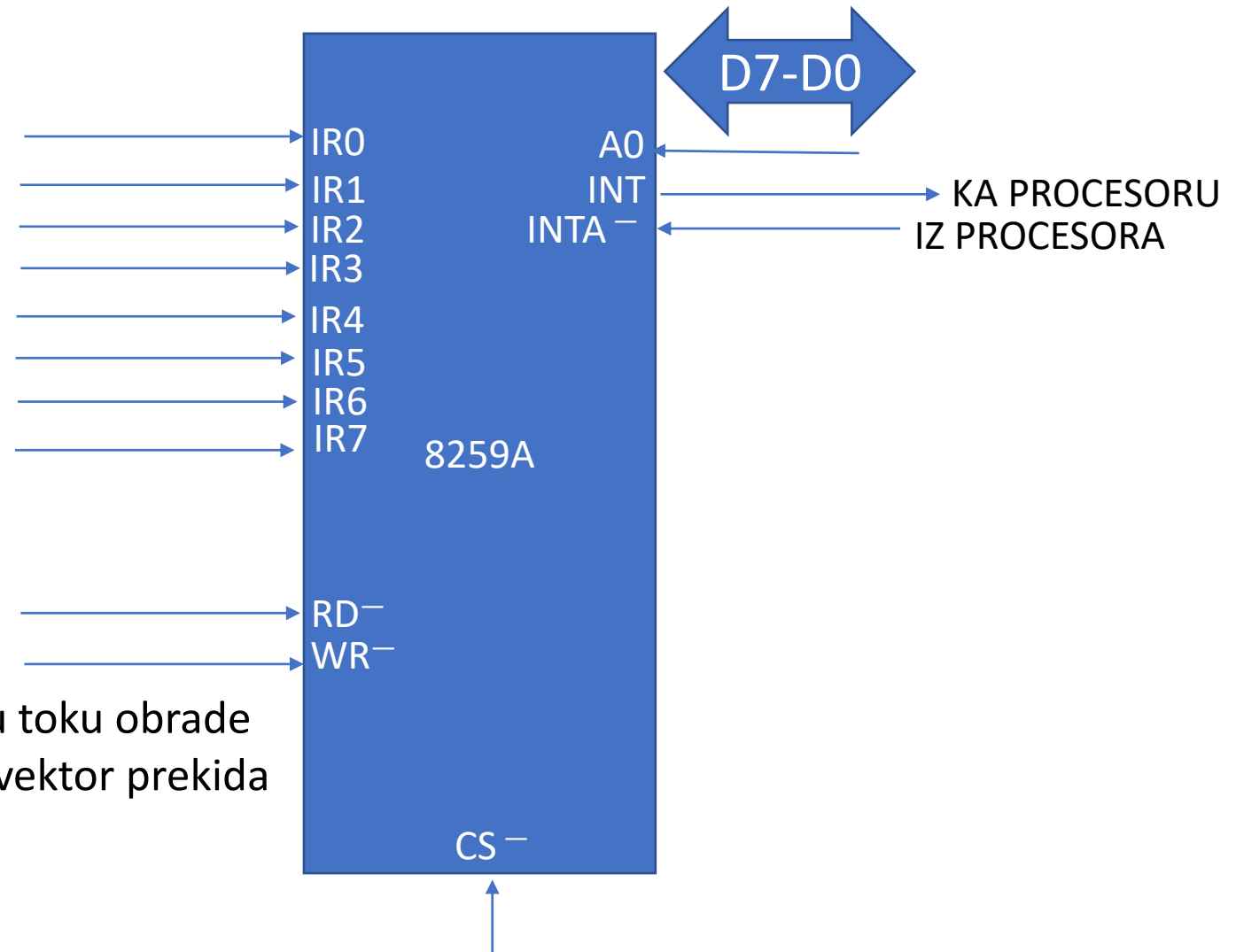
- IMR (Interrupt Mask Register)
- IRR (Interrupt Request Register)
- ISR (Inservice Register)
- Priority Resolver

Opis blokova za prekide

- IRR (Interrupt Request Register)
 - Pamćenje trenutno aktivnih zahteva za prekid
 - Svaki ulaz povezan sa odgovarajućim bitom
 - Bez obzira na mod, ulaz IR mora ostati aktivan do silazne ivice prvog impulsa INTA signala
- Priority Resolver
 - Mehanizam prioriteta prekida
- ISR (Inservice Register)
 - Za svaki ulaz po jedan bit
 - Zahtev sa i-tog ulaza se opslužuje => i-ti bit registra postavljen na 1
- IMR (Interrupt Mask Register)
 - Služi za maskiranje pojedinih ulaza
 - Redni broj bita jednak broju ulaza
 - 0 – demaskiranje
 - 1 – maskiranje

Najbitniji pinovi 8259A

- D0-D7
 - Šalje vektora prekida ka 8086
- A0
 - Adresiranje internih registara
- INT
 - Prekida proces
- INTA
 - Potvrda prekida
 - Dva impulsa:
 - Prvi: 8259 zabeleži da je prekid u toku obrade
 - Drugi: kaže mu da na liniji stavi vektor prekida



Sekvenca prihvatanja prekida

- Stiže zahtev na IR
- Šalje se zahtev na INT
- Procesor šalje na INTA 8259 prvi negativni impuls
 - Silazna ivica zamrzava stanje u svim 8259A da bi se prioriteti pravilno izračunali
 - Na uzlaznu ivicu master postavlja CAS linije
- U ISR se postavlja bit najvišeg prioriteta iz IRR (0 najveći prioritet) - koji pripada kontroleru čiji će se zahtev opslužiti
- Procesor šalje na INTA drugi negativni impuls
 - Odgovarajući 8259A šalje broj ulaza za zahtev koji je upravo prihvaćen
- Odgovarajući ISR bit se briše automatski u ovom trenutku ili se čeka odgovarajuća EOI komanda

Kontrolne reči kod 8259

- ICW1

- ICW2

- ICW3

- ICW4

- OCW1

- OCW2

- OCW3

ICW1

- A0=0

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

ICW2

- A0=1
- A7-A3: viših 5 bitova interrupt vektora
- D2-D0: setuje ih 8259 i predstavljaju redni broj IR linije na kojoj je prekid došao

Interrupt	A7	A6	A5	A4	A3	D2	D1	D0
IR0						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

ICW3

- Master, A0=1

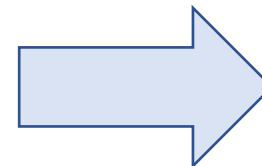
- informacija na kojim linijama postoji slave
- 1 : na odgovarajućem ulazu je slave

S7	S6	S5	S4	S3	S2	S1	S0

- Slave, A0=1

- Njegova adresa se pojavljuse na CAS linijama
- Odgovara rednom broju linije na masteru na koju je taj slave povezan

0	0	0	0	0	ID2	ID1	ID0



Interrupt	ID2	ID1	ID0
IR0	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

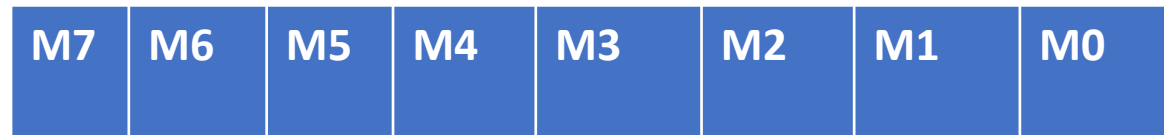
ICW4

- A0=1
- AEOI – Govori o načinu završetka prekida, tj. kako da procesor obavesti 8259 da je obrada prekida završena na strani procesora.
 - 1 (auto): aktivira automatsko brisanje bita u ISR registru po prihvatanju odgovarajućeg zahteva (uzlazna ivica drugog INTA impulsa)
 - 0 (normal): radi se o softverskom prekidu i podrazumeva se slanje OCW2 kontrolne reči neposredno pre instrukcije IRET

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

OCW1

- A0=1
- Jedinica na poziciji i označava da je prekid sa IRi zabranjen
- 1 – MASK SET
- 0 – MASK RESET



OCW2

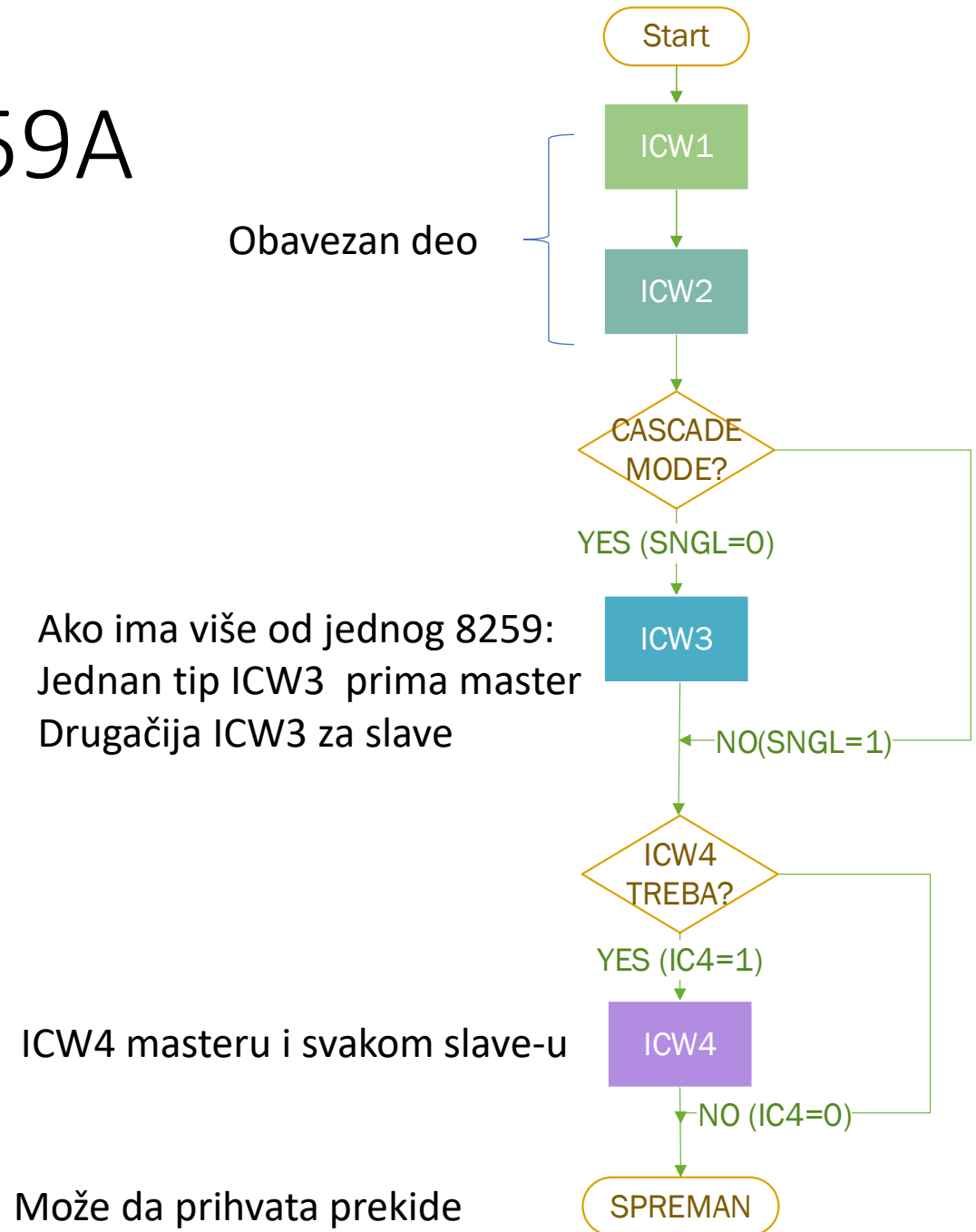
- $A_0=0$

	R	SL	EOI	0	0	L2	L1	L0
END OF INTERRUPT	000 – NON-SPECIFIC EOI COMMAND					IR nivo (0-7) na koji treba da reaguje		
AUTO-ROTATION	011 - SPECIFIC EOI COMMAND							
	101 – ROTATE ON SPECIFIC EOI COMMAND							
	100– ROTATE IN AUTOMATIC EOI MODE (SET)							
	000 – ROTATE IN AUTOMATIC EOI MODE (CLEAR)							
SPECIFIC ROTATION	111- ROTATE ON SPECIFIC EOI COMMAND							
	110 – SET PRIORITY COMMAND							
	010 – NO OPERATION							

OCW2 - objašnjenje

- Dve vrste komandi
 - NON-SPECIFIC – odnosi se na upravo odsluženi zahtev (najvećeg prioriteta)
 - SPECIFIC – zadaje se i broj ulaza na koji se komanda odnosi
 - Način da kontroler sazna kada se koja rutina završila i da osnovu toga radi sa prioritetima
- EOI:
 - Komanda za brisanje odgovarajućeg bita u ISR
 - Način da kontroler sazna kada se koja rutina završila i ispravno radi sa prioritetima
- Automatsko rotiranje:
 - U AEIOI modu, nakon opsluživanja dolazi do rotiranja prioriteta, taj nivo postaje najnižeg prioriteta.
- Specifično rotiranje tako da zadati ulaz postane najnižeg prioriteta
- Ako je upravo odslužen i-ti prioriteti su (od najvišeg ka najnižem)
 - $i+1, i+2, \dots, 7, 0, 1, \dots, i$

Sekvenca inicijalizacije 8259A



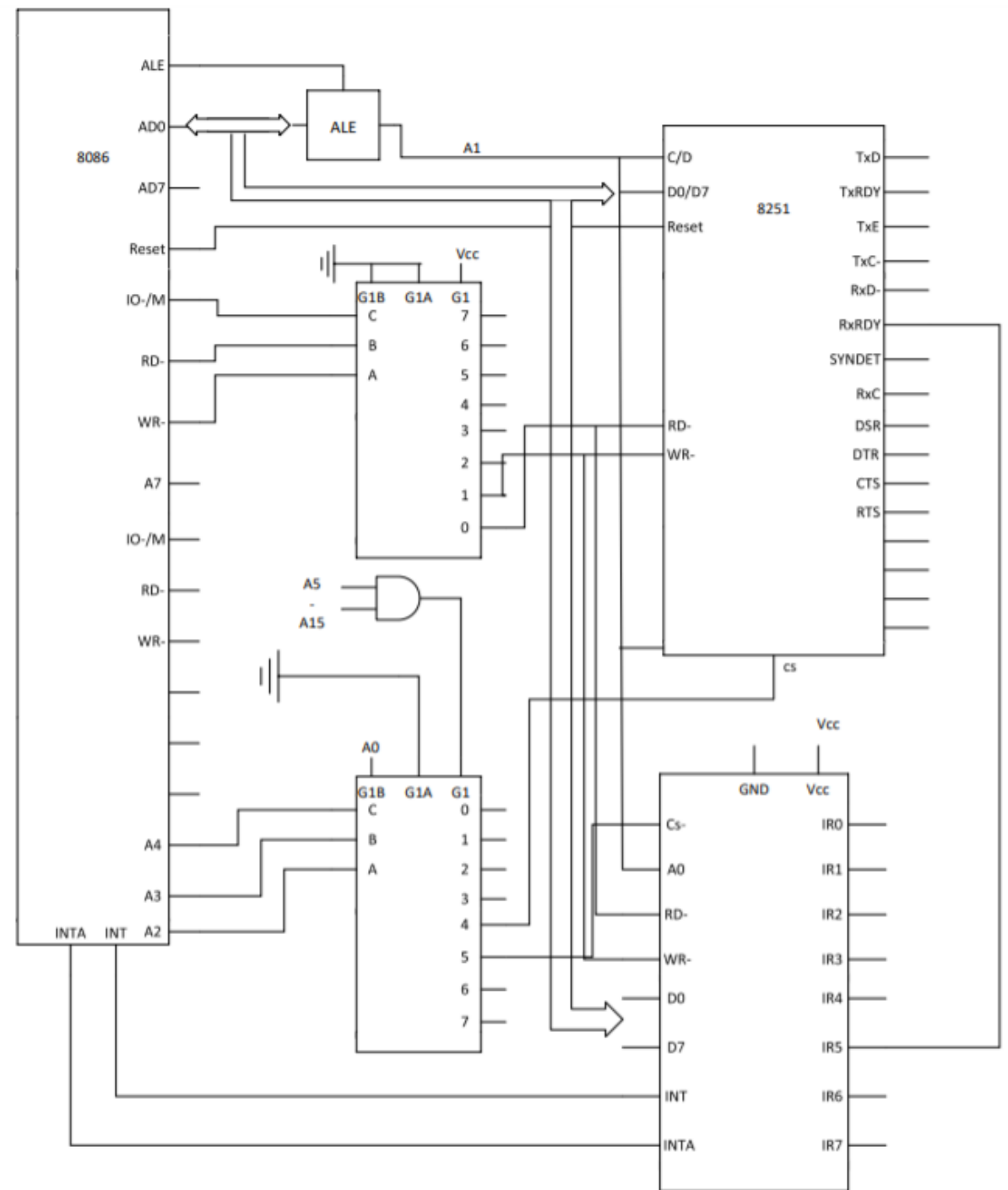
Zadatak 3

- Za mikroprocesor iAPX 8086 projektovati mikroračunarski sistem za predaju i prijem podataka koristeći komponentu 8251A. Podaci se šalju od adrese SEND, a primaju od adrese REC. Broj podataka za predaju i prijem je 64. Komponentu treba isprogramirati za istovremeni prijem i predaju 5-bitnih podataka sa parnim bitom parnosti, 1.5 stop bitom i brzinom 16x. Predaju organizovati ispitnom petljom, a prijem preko prekida 173, koji se generiše na priključku RxRDy. Komponenta 8251 nalazi se na U/I 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)
1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	8259 (FFF4)
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	8259 (FFF6)

Šema povezivanja



Mode word (8251)

- 0xB2

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	0	1	1	0	0	1	0

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

ICW1

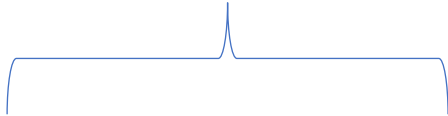
- 0x13

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

- 173 = 10101101(bin)
- 0xA8

Nije relevantno u toku
inicijalizacije



Interrupt	A7	A6	A5	A4	A3	D2	D1	D0
IR0	1	0	1	0	1	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

- 0xDF
- 173 = 10101101(bin)
- 168+5=173

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

Program – inicijalizacija i konfiguracija

```
EXTRN PREKID:FAR ;pomocna procedura za obradu prekida
PUBLIC RECV, BRULAZ ;vidljivo i iz drugih modula
```

```
DATA SEGMENT
```

```
    BRULAZ DB 0
```

```
    ;baferi primljenih i poslatih podataka
```

```
    SEND DB 64 DUP (3Fh)
```

```
    RECV DB 64 DUP (?)
```

```
DATA ENDS
```

```
STEK SEGMENT
```

```
    BOS DW 256 DUP (?)
```

```
    TOS LABEL WORD
```

```
STEK ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS:CODE, DS:DATA, SS:STEK
```

```
START:
```

```
    ;inicijalizacija segmenata
```

```
    MOV AX, DATA
```

```
    MOV DS, AX
```

```
    MOV AX, STEK
```

```
    MOV SS, AX
```

```
    LEA SP, TOS
```

```
    ;inicijalizacija komponenti
```

```
    CLI
```

```
    ;8251
```

```
    MOV AL, 00h
```

```
    MOV DX, 0FFF2h
```

```
    OUT DX, AL
```

```
    OUT DX, AL
```

```
    OUT DX, AL
```

```
    MOV AL, 40h
```

```
    OUT DX, AL
```

```
    MOV AL, B2h ;slanje mode kontrolne reci
```

```
    OUT DX, AL
```

```
    MOV AL, 15h ;slanje komandne kontrolne reci
```

```
    OUT DX, AL
```

```
    ;8259
```

```
    MOV DX, 0FFF4h
```

```
    MOV AL, 13h ;ICW1, A0=0
```

```
    OUT DX, AL
```

```
    MOV DX, 0FFF6h
```

```
    MOV AL, 0A8h ;ICW2
```

```
    OUT DX, AL
```

```
    MOV AL, 03h ;ICW4
```

```
    OUT DX, AL
```

```
    MOV AL, 0DFh ;OCW1
```

```
    OUT DX, AL
```

```
    ;inicijalizacija tabele vektora prekida
```

```
    MOV AX, OFFSET PREKID
```

```
    MOV [173*4], AX
```

```
    MOV AX, SEG PREKID
```

```
    MOV [173*4+2], AX
```

```
    MOV CX, 64
```

```
    MOV SI, 0
```

```
    STI ;dozvola prekida
```

Program - glavna petlja

```
PETLJA:
CEKAJ:
    MOV DX, 0FFF2h
    IN AL, DX ;AL -- status
    TEST AL, 38h ;test na gresku koja se moze javiti u prijemu
    JNZ GRESKA
    TEST AL, 01h ;TxRDY == 1?
    JZ CEKAJ ;ako nije spreman, cekaj da bude
    MOV DX, 0FFF0h
    MOV AL, SEND[SI]
    OUT DX, AL
    INC SI
    LOOP PETLJA ;64 prolaza
CEKAJ1: ;cekanje na prekidnu proceduru
    CMP BRULAZ 64
    JL CEKAJ1
    MOV DX, 0FFF2h ;testiranje na gresku
    IN AL, DX
    TEST AL, 38h
    JZ KRAJ

GRESKA: ;kod za upravljanje greskom
KRAJ:
    MOV AH, 4Ch
    INT 21h
CODE ENDS
END START
```

Eksterna procedura za prekide

```
;Radi paralelno i vrsi prijem podataka
EXTRN RECV:BYTE, BRULAZ:BYTE
PUBLIC PREKID
PROCED SEGMENT
PREKID PROC FAR
ASSUME CS:PROCED
    ;cuvanje konteksta procesora
    PUSHF; automatski se stavlja na stek
    PUSH AX
    PUSH DI
    PUSH DX
    MOV DX, 0FFF0h ;bez testiranja na spremnost jer je RxRDY sigurno 1
    IN AL, DX
    MOV DI, BRULAZ
    MOV RECV[DI], AL
    INC BRULAZ ;uvecanje brojaca primljenih
    ;vracanje konteksta procesora
    POP DX
    POP DI
    POP AX
    POPF
    IRET ;vracanje iz procedure
PREKID ENDP
PROCED ENDS
```