

**J. ĐORĐEVIĆ, D. MILIĆEV, D. BOJIĆ,
A. MILENKOVIĆ, B. NIKOLIĆ,
Z. RADIVOJEVIĆ, M. OBRADOVIĆ**

ARHITEKTURA RAČUNARA

ZBIRKA REŠENIH ZADATAKA

Beograd 2005.

1 PREKIDI

1.1 ZADATAK

Posmatra se procesor sa vektorisanim mehanizmom prekida. Adrese prekidnih rutina se čuvaju u tabeli adresa prekidnih rutina (IV tabeli) koja počinje na adresi 0 i ima 256 ulaza. Kapacitet operativne memorije je 64 Kbajta, širina reči memorije je jedan bajt, a 16-bitni podaci se u memoriju smeštaju tako da se na nižoj adresi nalazi viši bajt, a na višoj adresi niži bajt. Prekidne rutine za periferije PER1, PER2 i PER3 počinju na adresama 5678h, 3456h i 1234h, respektivno. Ulazi 5, 3 i 1 u IV tabeli su dodeljeni periferijama PER1, PER2 i PER3, respektivno.

a) Nacrtati deo operativne memorije koji se nalazi na ulazima 0 do 5 u IV tabeli, označiti adrese relevantnih memorijskih lokacija i popuniti ih odgovarajućim vrednostima.

b) Nabrojati korak po korak šta se sve dešava u procesoru od trenutka kada je stigao zahtev za prekid od periferije do trenutka kada se u registru PC nalazi početna adresa prekidne rutine.

c) Objasniti za svaki korak nabrojan u tački b) da li ga obavlja hardver ili softver.

d) Objasniti kako se obavlja povratak iz prekidne rutine i nabrojati korake koji se tom prilikom izvode.

Rešenje:

a) Kapacitet operativne memorije je 64Kbajta, tj. 2^{16} reči, pa su adrese široke 16 bita, odnosno dve 8-bitne reči. Interapt vektor tabela sastoji se od ulaza koji sadrže početne adrese prekidnih rutina. Kako su adrese široke dve reči, svaki ulaz tabele zauzima dve lokacije u memoriji, na nižoj adresi nalazi se viši bajt adrese prekidne rutine, a na višoj adresi niži bajt adrese prekidne rutine. Adresa početka IV tabele se čuva u registru IVTP, to je adresa 0000h. Ako broj ulaza i koji odgovara nekoj periferiji PER_j pomnožimo sa 2 i tu vrednost dodamo na sadržaj registra IVTP dobijamo adresu a na kojoj je smešten viši bajt adrese na kojoj počinje prekidna rutina periferije PER_j , na adresi $a+1$ nalazi se niži bajt adrese na kojoj počinje prekidna rutina za PER_j . Deo IV tabele na ulazima od 0 do 5 prikazan je na slici 1.1. Ulaz 1 dodeljen je periferiji PER1, na adresi 0002h nalazi se viši bajt adrese početka prekidne rutine za PER1(12h), a na adresi 0003h niži bajt adrese početka prekidne rutine za PER1(34h). Periferiji PER2 dodeljen je ulaz 3, što znači da se na adresi 0006h nalazi viši bajt adrese početka prekidne rutine za PER2(34h), a na adresi 0007h niži bajt adrese početka prekidne rutine za PER2(56h). Ulaz 5 dodeljen je periferiji PER3, na adresi 000Ah nalazi se viši bajt adrese početka prekidne rutine za PER3(56h), a na adresi 000Bh niži bajt adrese početka prekidne rutine za PER3(78h).

broj ulaza u IV tabelu	memorijska adresa	sadržaj	opis sadržaja
5	000Bh	78h	niži bajt adrese prekidne rutine za PER3
	000Ah	56h	viši bajt adrese prekidne rutine za PER3
4	0009h		
	0008h		
3	0007h	56h	niži bajt adrese prekidne rutine za PER2
	0006h	34h	viši bajt adrese prekidne rutine za PER2
2	0005h		
	0004h		
1	0003h	34h	niži bajt adrese prekidne rutine za PER1
	0002h	12h	viši bajt adrese prekidne rutine za PER1



Slika 1.1. Izgled dela operativne memorije na ulazima od 0 do 5 u IV tabeli

Napomena: Pogrešno je ulaze numerisati počev od 1.

b) Kada stigne zahtev za prekid najpre se završi sa izvršavanjem tekuće instrukcije. Nakon toga se nastavlja sa izvršavanjem dodatnih koraka koji su potrebni da bi se ① sačuvao kontekst procesora i ② izračunala adresa prekidne rutine. Kontekst procesora (PC, PSW i programski dostupni registri) se čuva na vrhu steka. Adresa prekidne rutine se računa sledećom sekvencom akcija:

procesor šalje inta signal (*interrupt acknowledge*) periferiji,
periferija odgovara tako što procesoru pošalje svoj broj ulaza u IV tabelu,
procesor izračuna adresu ulaza u IV tabelu (broj ulaza se pretvori u pomeraj i doda se na sadržaj IVTP registra) i

pročita se adresa prekidne rutine iz IV tabele i smesti se u PC.

Treba dodati da veoma često nakon koraka ① i ② procesor obavlja i korak ③. U ovom koraku:

bit I (maskirajući prekidi dozvoljeni) u registru PSW se resetuje,
bit T (prekid posle svake instrukcije) u registru PSW se resetuje i
u bitove L (prioritet tekućeg programa) u registru PSW se upisuje prioritet prekidne rutine na koju se skače.

c) Odgovor u tački b) podrazumeva da se svi koraci izvode hardverski. Međutim, neki od koraka pod ① mogu da se izvedu softverski. Ovo je slučaj kod programski dostupnih registara i ponekad kod registra PSW koje u tom slučaju treba programski sačuvati na početku prekidne rutine. Programski dostupni registri se obično čuvaju softverski ako ih je mnogo. U suprotnom, oni se čuvaju hardverski. Registar PSW se mora čuvati hardverski ako postoji korak ③. U suprotnom, može se čuvati ili hardverski ili softverski. U većinu slučajeva registar PSW se čuva hardverski. Koraci ② i ③ se uvek izvode hardverski.

d) Povratak iz prekidne rutine se izvodi posebnom instrukcijom RTI (*Return From Interrupt*). Ova instrukcija restaurira sa steka kontekst procesora:

programski dostupni registri se restauriraju ako su sačuvani hardverski,
registar PSW se restaurira ako je sačuvan hardverski i
registar PC je restauriran.

Ako se programski dostupni registri i PSW ne restauriraju instrukcijom RTI, moraju se restaurirati softverski na kraju prekidne rutine pre instrukcije RTI.

1.2 ZADATAK

Memorija nekog računara je kapaciteta 4 G (giga) 16-bitnih reči. Adresibilna jedinica je 16-bitna reč, a 32-bitni brojevi se u memoriju smeštaju tako da je na nižoj adresi nižih 16 bita. Procesor je jednoadresni, ulazno/izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. Interrapt vektor tabela ima 64 K ulaza i počinje od adrese 0 operativne memorije. Na procesor su vezane tri periferije, PER1, PER2 i PER3 kojima treba dodeliti ulaze 3, 5 i 7 u vektor tabeli, i kojima odgovaraju prekidne rutine na adresama 12345678h, 23456789h i 3456789Ah, respektivno. Adrese registara u kojima se čuvaju brojevi ulaza su 0h, 4h i 8h, respektivno.

a) Napisati deo programa kojim se inicijalizuju brojevi ulaza za navedene periferije.

b) Nacrtao izgled prvih 8 ulaza u vektor tabeli, označiti adrese relevantnih lokacija i upisati sadržaje u njih.

c) Napisati deo programa kojim se inicijalizuje ulaz 5 u vektor tabeli.

d) Koju vrednost šalje procesoru periferija PER2 kada joj procesor odobri zahtev za prekidom?

e) Opisati i predstaviti programom postupak kojim se i periferiji PER2 dodeljuje ista prekidna rutina kao i periferiji PER3.

Rešenje:

a) Brojevi ulaza čuvaju se u za to namenjenim registrima periferije. To su navedeni registri na adresama 0h, 4h i 8h. Ovi registri se, podrazumevano, nalaze u ulazno/izlaznom adresnom prostoru, pa im se pristupa OUT instrukcijama. Traženi deo programa je prikazan na slici 1.2.

```

LOAD #3 ;napuni akumulator konstantom 3-broj ulaza za PER1
OUT 0h ;prebaci sadržaj akumulatora u reg. periferije PER1
LOAD #5 ;napuni akumulator konstantom 5-broj ulaza za PER2
OUT 4h ;prebaci sadržaj akumulatora u reg. periferije PER2
LOAD #7 ;napuni akumulator konstantom 7-broj ulaza za PER3
OUT 8h ;prebaci sadržaj akumulatora u reg. periferije PER3

```

Slika 1.2. Inicijalizacija brojeva ulaza za PER1, PER2 i PER3

Ovaj program izvršava procesor u toku inicijalizacije sistema, pre nego što se dozvoli opsluživanje prekida. Podrazumeva se, zato, da su u toku izvršavanja ovog programa svi prekidi maskirani. Tek pošto su brojevi ulaza smešteni u registre periferija, periferije se mogu ispravno odazvati na dozvolu inta procesora za prekid.

b) Adresni prostor je veličine 2^{32} reči, pa su adrese široke 32 bita, odnosno dve 16-bitne reči. Interapt vektor tabela sastoji se od ulaza koji sadrže adrese prekidnih rutina. Kako su adrese široke dve reči, svaki ulaz tabele zauzima dve lokacije u memoriji. Tabela počinje od adrese 0, pa se adresa ulaza i dobija množenjem i sa 2. Adrese prekidnih rutina se u odgovarajuće ulaze tabele smeštaju tako da je na nižoj adresi niža reč adrese prekidne rutine, a na višoj adresi viša reč adrese prekidne rutine. Veličina tabele od 64 K ulaza ne znači ništa za rešenje zadatka, ali se može videti da je broj ulaza tačno 16-bitni binarni broj. Vektor tabela je prikazana na slici 1.3. Ulaz 3 dodeljen je periferiji PER1, na adresi 00000006h nalazi niža reč adrese početka prekidne rutine za PER1(5678h), a na adresi 00000007h viša reč adrese početka prekidne rutine za PER1(1234h). Periferiji PER2 dodeljen je ulaz 5, što znači da se na adresi 0000000Ah nalazi niža reč adrese početka prekidne rutine za PER2(6789h), a na adresi 0000000Bh viša reč adrese početka prekidne rutine za PER2(2345h). Ulaz 7 dodeljen je periferiji PER3, na adresi 0000000Eh nalazi se niža reč adrese početka prekidne rutine za PER3(789Ah), a na adresi 0000000Fh viša reč adrese početka prekidne rutine za PER3(3456h).

broj ulaza u IV tabelu	memorijska adresa	sadržaj	opis sadržaja
7	0000000Fh	3456h	viša reč adrese prekidne rutine za PER3
	0000000Eh	789Ah	niža reč adrese prekidne rutine za PER3
6	0000000Dh		
	0000000Ch		
5	0000000Bh	2345h	viša reč adrese prekidne rutine za PER2
	0000000Ah	6789h	niža reč adrese prekidne rutine za PER2
4	00000009h		
	00000008h		
3	00000007h	1234h	viša reč adrese prekidne rutine za PER1
	00000006h	5678h	niža reč adrese prekidne rutine za PER1
2	00000005h		
	00000004h		
1	00000003h		
	00000002h		
0	00000001h		
	00000000h		

Slika 1.3. Izgled dela operativne memorije na ulazima od 0 do 7 u IV tabeli

c) Potrebno je u ulaz broj 5, koji počinje na adresi 0Ah, upisati adresu prekidne rutine. Preciznije, u lokaciju 0000000Ah treba upisati nižu reč adrese (6789h), a u lokaciju 0000000Bh višu reč adrese (2345h). Kako ništa nije rečeno o skupu instrukcija procesora, pretpostavlja se da procesor nema instrukcije za rad direktno sa 32-bitnim veličinama, pa se tražena inicijalizacija mora izvesti iz dva prenosa 16-bitnih reči. Vektor tabela se nalazi u memoriji, dakle u memorijskom adresnom prostoru. Traženi deo programa je prikazan na slici 1.4.

```
LOAD #2345h ;napuni akumulator konstantom 2345h
STORE 0000000Bh ;prebaci sadržaj akumulatora na adresu 0000000Bh
LOAD #6789h ;napuni akumulator konstantom 6789h
STORE 0000000Ah ;prebaci sadržaj akumulatora na adresu 0000000Ah
```

Slika 1.4. Inicijalizacija ulaza 5 u vektor tabeli

I ovaj deo programa se, najčešće, izvršava tokom inicijalizacije sistema. I ovde se podrazumeva da su prekidi maskirani, jer ako to nije slučaj, može se dogoditi da prekid stigne u trenutku kada je jedna reč adrese upisana u tabelu, a druga nije. Tada bi procesor prešao na izvršavanje programa sa pogrešne adrese. Ako bi procesor imao instrukciju za prenos 32-bitne reči u memoriju, to se ne bi moglo dogoditi, iako se prenos interno obavlja iz dva puta, jer se instrukcija izvršava do kraja pre nego što se pređe na obradu prekida.

d) Periferija šalje svoj broj ulaza procesoru kada joj procesor odobri zahtev za prekidom. To je uvek slučaj. Odgovor je, dakle, 5.

e) Da bi dve periferije imale istu prekidnu rutinu, dovoljno je da im ulazi u vektor tabeli ukazuju na istu prekidnu rutinu. Dovoljno je, zato, u ulaz periferije PER2 upisati istu vrednost kao i u ulaz periferije PER3. To se postiže programom prikazanim na slici 1.5.

```
LOAD #3456h ;napuni akumulator konstantom 3456h
STORE 0000000Bh ;prebaci sadržaj akumulatora na adresu 0000000Bh
LOAD #789Ah ;napuni akumulator konstantom 789Ah
STORE 0000000Ah ;prebaci sadržaj akumulatora na adresu 0000000Ah
```

Slika 1.5. Dodela iste vrednosti u ulaz PER2 kao i u ulazu PER3

Isto se postiže i programom prikazanim na slici 1.6.

```
LOAD 0000000Fh ;napuni akumulator sadržajem adrese 0000000Fh
STORE 0000000Bh ;prebaci sadržaj akumulatora na adresu 0000000Bh
LOAD 0000000Eh ;napuni akumulator sadržajem adrese 0000000Eh
STORE 0000000Ah ;prebaci sadržaj akumulatora na adresu 0000000Ah
```

Slika 1.6. Dodela iste vrednosti u ulaz PER2 kao i u ulazu PER3

Drugi, takođe ispravan, ali manje uobičajen način da se postigne traženo je da se periferiji PER2 dodeli isti broj ulaza u tabeli kao i periferiji PER3. To se postiže programom prikazanim na slici 1.7.

```
LOAD #7 ;napuni akumulator konstantom 7
OUT 4h ;prebaci sadržaj akumulatora na adresu 4h u
;ulazno/izlaznom adresnom prostoru
```

Slika 1.7. Dodela istog broja ulaza za periferiju PER2 kao i za periferiju PER3

I za ove programe važe iste napomene o potrebi maskiranja prekida. Opisana tehnika promene vektora u interapt vektor tabeli je često primenjivana u sistemskom programiranju. To je način da se preuzme kontrola nad nekim prekidom (“presretanje prekida” ili “preusmeravanje prekida”).

1.3 ZADATAK

Memorija nekog računara je kapaciteta 64 KB. Adresibilna jedinica je bajt, a 16-bitni brojevi se u memoriju smeštaju tako da je na nižoj adresi viši bajt. Procesor je jednoadresni i ima naredbe koje obavljaju prenos samo jednog bajta ka/iz memorije, ulazno/izlazni i memorijski

adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. Interapt vektor tabela ima 8 ulaza i počinje od adrese 0 operativne memorije.

Procesor ima 4 ulazne linije $IRQ_{3..0}$ za maskirajuće prekide koji se prioritiraju, pri čemu je ulaz IRQ_0 najvišeg prioriteta. Ovim linijama dodeljeni su fiksno ulazi 4 do 7 u vektor tabeli, respektivno. Nemaskirajući i interni prekidi su višeg prioriteta od maskirajućih, i zauzimaju ulaze 0 do 3 u tabeli. U PSW procesora postoji bit I (*Interrupt Enable*) koji se briše u mikroprogramu za obradu prekida, a programski kontroliše naredbama $INTE$ (*Interrupt Enable*, postavlja I na 1) i $INTD$ (*Interrupt Disable*, briše I), kao i biti $L_{2..0}$ koji sadrže nivo prekida koji se trenutno opslužuje. Takođe postoji 4-bitni registar IMR (*Interrupt Mask Register*) kod koga jedinica na bitu i znači da je dozvoljen prekid sa linije IRQ_i . Ovaj registar inicijalno sadrži sve jedinice.

Na linije IRQ_1 i IRQ_3 su vezane periferije PER_1 i PER_3 , respektivno, a ostale linije su slobodne. Treba vezati još uređaj T—vremenski brojač koji periodično generiše signal zahteva za prekid. Potrebno je obezbediti da se prekid od vremenskog brojača može opslužiti i u toku izvršavanja neke od prekidnih rutina za periferije PER_1 i PER_3 . Ove rutine počinju na adresama 100h i 300h, respektivno.

- Na koji ulaz $IRQ_{3..0}$ treba vezati signal zahteva za prekid vremenskog brojača?
- Napisati deo programa kojim se inicijalizuje ulaz periferije PER_1 u vektor tabeli.
- Prekidna rutina periferije PER_1 je prikazana na slici 1.8.

```
INTH1:IN    OFF00h
          STORE (Dest)+
          ...
          RTI
```

Slika 1.8. Prekidna rutina periferije PER_1

Ako u toku izvršavanja instrukcije $STORE$ stigne signal prekida od vremenskog brojača, da li će se ovaj prekid odmah opslužiti? Obrazložiti odgovor.

- Izmeniti datu prekidnu rutinu periferije PER_1 tako da odgovor na prethodno pitanje bude suprotan.
- Šta treba uraditi u prekidnoj rutini periferije PER_3 tako da se obezbedi da ona ne bude prekidana od zahteva sa periferije PER_1 , ali da bude prekidana od zahteva sa vremenskog brojača?

Rešenje:

a) Da bi prekid od vremenskog brojača T mogao biti opslužen i u toku izvršavanja neke od prekidnih rutina drugih periferija, prekid od uređaja T treba da bude višeg prioriteta od ostalih. Kako je IRQ_0 jedini slobodni ulaz višeg prioriteta od zauzetih, na ovaj ulaz treba vezati signal zahteva za prekidom od uređaja T.

b) Kapacitet operativne memorije je 64Kbajta, tj. 2^{16} reči, pa su adrese široke 16 bita, adresibilna jedinica je bajt, pa svaki ulaz u vektor tabeli zauzima dve lokacije. Vektor tabela počinje od adrese 0. Periferiji PER_1 dodeljen je ulaz 5. Ulaz 5 je na adresama 000Ah, 000Bh. Adresa prekidne rutine je 0100h, a smešta se u tabelu tako da je na nižoj adresi viši bajt. Što znači da ćemo na adresu 000Ah smestiti viši bajt adrese prekidne rutine $PER_1(10h)$, a na adresu 000Bh niži bajt prekidne rutine za $PER_1(00h)$. Program za inicijalizaciju je prikazan na slici 1.9.

```
LOAD #01h          ;napuni akumulator konstantom 10h
STORE 000Ah        ;prebaci sadržaj akumulatora na adresu 000Ah
LOAD #00h          ;napuni akumulator konstantom 00h
STORE 000Bh        ;prebaci sadržaj akumulatora na adresu 000Bh
```

Slika 1.9. Inicijalizacija ulaza 5 u vektor tabeli

c) Kako se u mikroprogramu za opsluživanje prekida bit maske I u registru PSW briše, odmah po ulasku u neku prekidnu rutinu svi maskirajući prekidi su maskirani (ne opslužuju se). Zato se u datoj prekidnoj rutini pristigli prekid ne opslužuje.

d) Da bi se u toku date prekidne rutine mogao izvršiti maskirajući prekid (višeg prioriteta), treba demaskirati prekide odmah po ulasku u prekidnu rutinu. To se radi instrukcijom `INTE`. Izmenjena prekidna rutina je prikazana na slici 1.10.

```
INTH1:INTE                ;demaskiraju se svi maskirajući prekidi
      IN      0FF00h
      STORE  (Dest)+
      ...
      RTI
```

Slika 1.10. Izmenjena prekidna rutina periferije PER1

e) Osim postavljanja bita I na 1, treba selektivno maskirati prekid od periferije PER1. To se postiže upisom 0 u odgovarajući bit 1 registra IMR. U bit 0 registra IMR treba postaviti 1, kako bi bio dozvoljen prekid od uređaja T. Vrednost bita 2 registra IMR nije bitna, a vrednost bita 3 određuje da li će se prihvatiti novi prekid od iste periferije ili ne. Kako ovo nije posebno definisano, u IMR treba upisati bb01 (binarno, b označava bilo koju vrednost).

1.4 ZADATAK

Adrese i podaci računara su široki po 16 bita. Procesor je jednoadresni, operiše samo sa 16-bitnim veličinama, a mehanizam prekida je vektorisan. Interapt vektor tabela ima 8 ulaza i počinje od adrese na koju ukazuje registar IVTP (*Interrupt Vector Table Pointer*) čija je vrednost 3.

Ne postoje interni ni nemaskirajući prekidi. Procesor ima 8 ulaznih linija IRQ0...IRQ7 za maskirajuće prekide koji se prioritiraju, pri čemu je ulaz IRQ7 najvišeg prioriteta. Ovim linijama dodeljeni su fiksno ulazi 0 do 7 u vektor tabeli, respektivno. U PSW procesora postoji bit I (*Interrupt Enable*) koji se briše u mikroprogramu za obradu prekida, a programski kontroliše naredbama `INTE` (*Interrupt Enable*, postavlja I na 1) i `INTD` (*Interrupt Disable*, briše I), kao i biti L koji sadrže nivo prioriteta dela programa koji se trenutno izvršava. Procesor nikada ne prihvata prekid istog ili nižeg nivoa prioriteta nego što je tekući nivo. U mikroprogramu za opsluživanje prekida na stek se stavljaju PSW i PC tim redom.

Na linije IRQ0 i IRQ1 su vezane periferije PER0 i PER1, respektivno. Ovim periferijama dodeljene su prekidne rutine koje su prikazane na slici 1.11., gde je $X=A$ za PER0, odnosno $X=B$ za PER1, adrese prekidnih rutina za periferije (dve kopije u memoriji). Izgled dela memorije počev od lokacije 0 zaključno sa 7 je prikazan na slici 1.12.

adresa	instrukcija
X+0	INTE
X+1	PUSHA
X+2	LOAD MemCnt
X+4	INCA
X+5	STORE MemCnt
X+7	POPA
X+8	INTD
X+9	RTI

Slika 1.11. Prekidna rutina periferije PER0(X=A) i PER1(X=B)

adresa	sadržaj
0000h	5638h
0001h	2E78h
0002h	1334h
0003h	6423h
0004h	23F7h
0005h	0010h
0006h	1001h
0007h	7AB0h

Slika 1.12. Izgled dela memorije

a) Koliko iznose A i B?

b) Posmatra se sledeća sekvenca izvršavanja naredbi (navedene su adrese naredbi) i događaja: počinje prekidna rutina za PER0, $A + 0$, $A + 1$, prekid od PER1, $B + 0$, $B + 1$, ..., $B + 9$, $A + 2$. U toku izvršavanja naredbe $A + 2$ dolazi ponovo zahtev za prekid od PER1. Koja se naredba sledeća izvršava (posle $A + 2$)? Koji je odgovor na ovo pitanje ako se naredba `INTD` na adresi $B + 8$ zameni naredbom `INTE`?

c) Ako se na sâm početak prekidne rutine za PER1, ispred naredbe `INTE`, dodaju naredbe prikazane na slici 1.13.

adresa	instrukcija
B+0	CLRA ;Clear Accumulator (ACC=0)
B+1	PUSHA ;Push Accumulator
B+2	PUSHA

Slika 1.13. Naredbe dodate na početak prekidne rutine za PER1

a ostatak ostane nepromenjen, sa koje adrese se nastavlja izvršavanje programa po povratku iz prekidne rutine PER1 u sekvenci izvršavanja iz prethodne tačke?

d) Ne menjajući dalje već izmenjenu rutinu iz prethodne tačke, napisati deo glavnog programa kojim se eliminiše efekat izmene iz prethodne tačke.

e) Koliko je bita L potrebno u PSW i zašto?

Rešenje:

a) Adrese su široke 16-bita, adresibilna jedinica je 16-bitna reč, pa svaki ulaz u vektor tabeli zauzima jednu lokaciju. Vektor tabela počinje na adresi 3 (toliko iznosi IVTP), a periferyama PER0 i PER1 su dodeljeni ulazi 0 i 1, respektivno. Ovi ulazi nalaze se na adresama 0003h i 0004h, a sadržaji ovih adresa predstavljaju adrese prekidnih rutina. Tako je $A = 6423h$, $B = 23F7h$.

b) Kada prihvati prekid, procesor u svom mikroprogramu obriše bit maske I u registru PSW. Prva instrukcija prekidne rutine postavi ovaj bit na 1, i time omogući prekide. Zatim dolazi do prekida višeg prioriteta koji se prihvata. Pre ulaska u prekidnu rutinu, hardverski se (procesor to radi u mikroprogramu) na steku čuvaju PSW i PC. Time je na steku sačuvana i vrednost $I = 1$ unutar registra PSW.

Pri kraju prekidne rutine za PER1, bit I se briše naredbom `INTD`, čime se maskiraju prekidi. Međutim, ova naredba nema nikakvog uticaja na dalji tok izvršavanja prekidne rutine za PER0 u koju se procesor vraća posle prekidne rutine za PER1, jer se PSW, a sa njim i I, restauriraju sa steka hardverski. Zato je, po povratku u prekidnu rutinu za PER0 bit I ponovo 1, pa se novi prekid od PER1 prihvata.

Sledeća naredba u sekvenci je, dakle, naredba sa adrese $B + 0 = 23F7h$. Na odgovor, naravno, nema nikakvog uticaja zamena naredbe `INTD` naredbom `INTE`.

c) Po ulasku u novu prekidnu rutinu, na stek se stavljaju tri nule (dve nove instrukcije `PUSH` i jedna stara). Na završetku, prekidna rutina sa steka skida samo jednu nulu, tako da su na vrhu steka još dve nule. Kada dođe do instrukcije `RTI`, procesor "slepo" uzima dve reči sa vrha steka i tretira ih kao (smešta ih u) PC i PSW. Posle toga, PC ukazuje na adresu 0, sa koje će procesor pročitati sledeću instrukciju. To znači da će se procesor "vratiti" na "program" na adresi 0. Ovo je, naravno, greška.

d) Greška opisana u prethodnoj tački može se jednostavno sprečiti tako što se procesor spreči da skoči na prve tri instrukcije izmenjene rutine. Umesto toga, procesor treba da, po prihvatanju prekida, odmah skoči na stari deo prekidne rutine, dakle na adresu $B + 3 = 23FAh$. To se postiže promenom vektora u vektor tabeli za datu periferyu, što je prikazano na slici 1.14.

```
LOAD #23FAh ;napuni akumulator konstantom 23FAh
STORE 0004h ;prebaci sadržaj akumulatora na adresu 0004h
```

Slika 1.14. Promena vektora u vektor tabeli za periferyu PER1

Diskusija:

Alternativno rešenje se sastoji u tome da se na adresu 0 upiše kôd instrukcije `RTI` (*Return from Interrupt*). Time se postiže da procesor, kada skoči na adresu 0 po povratku iz prekidne rutine, još jednom “slepo” izvrši skidanje sa steka dve reči koje će tretirati kao PC i PSW, a to su upravo prave vrednosti. Međutim, ovo rešenje krije jednu suptilnu grešku. Naime, ne sme se, u opštem slučaju, dozvoliti da se u PSW upiše tako “destruktivna” i neodređena vrednost kao što je nula. To se još uvek dešava kod ovog rešenja. PSW sadrži i druge bite osim pomenutih, koji mogu da imaju jako bitno značenje za režim rada procesora. Upisom nule u PSW se procesor prevodi u sasvim nepredvidiv režim, na primer neočekivano prelazi u sistemski ili korisnički režim, ili u *trap* režim i slično.

Jedino sasvim ispravno rešenje je, dakle, ne dozvoliti da procesor dođe do pogrešno dodatih instrukcija, odnosno treba da ih obavezno izbegne, a to se postiže promenom vektora.

e) Ključna rečenica u zadatku koja daje odgovor na postavljeno pitanje je da “Procesor nikada ne prihvata prekid istog ili nižeg nivoa prioriteta nego što je tekući nivo”. Naime, zbog ovoga nije dovoljno da PSW sadrži 3 bita L za 8 nivoa prioriteta. Kada bi se prekidu `IRQ0` dodelio nivo kodovan sa 000, koji se takođe mora dodeliti i glavnom programu, prekid sa `IRQ0` se jednostavno nikad ne bi opslužio. Evo zašto. Procesor počinje inicijalno da radi na najnižem nivou, a to je 0. Ako stigne prekid sa `IRQ0`, procesor ne može znati da se trenutno nalazi u glavnom programu, a ne u prekidnoj rutini za `IRQ0`, čiji se nivo koduje takođe sa 0. Kako procesor ne prihvata prekid istog nivoa, ovaj prekid se ne može prihvatiti.

Potrebno je, dakle, obezbediti 9 nivoa prioriteta, za čije je kodovanje potrebno 4 bita. Glavni program, odnosno početna vrednost nivoa, ima kôd 0000 binarno.

Diskusija:

Ovo još uvek ne znači da je kodovanje ostalih nivoa jednoznačno određeno. Ako bi se, prosto, nivo prekida `IRQ0` kodovao sa 1, `IRQ1` sa 2 itd., to bi zahtevalo relativno složenu mrežu koja koduje zahteve `IRQi` u navedene brojeve (to je, u stvari, mreža za inkrementiranje vrednosti `IRQi` kodovane običnim koderom prioriteta). Nije potrebno toliko komplikovati hardver. Dovoljno je, na primer, da jedan L bit (na primer L_0), kada je postavljen na 1, ukazuje da je procesor u prekidnoj rutini. Tako bi se nivoi kodovali sa: glavni program sa 0000, `IRQ0` sa 0001, `IRQ1` sa 0011, `IRQ2` sa 0101, ..., `IRQ7` sa 1111. Za ovakvo kodovanje dovoljan je običan koder prioriteta koji koduje `IRQi` sa tri bita, plus jedan bit L_0 koji se postavlja na 1 kada se prihvata prekid.

1.5 ZADATAK

Adresna i magistrala podataka računara su 16-bitne. Procesor je jednoadresni, operiše samo sa 16-bitnim veličinama, a mehanizam prekida je vektorisan. Interapt vektor tabela ima 6 ulaza (0 do 5) i počinje od adrese na koju ukazuje registar `IVTP` (*Interrupt Vector Table Pointer*) čija je vrednost 2.

Procesor ima 2 ulazne linije `IRQ0` i `IRQ1` za spoljne nemaskirajuće prekide koji se prioritiraju, pri čemu je ulaz `IRQ1` višeg prioriteta. Ovim linijama se mogu dodeliti proizvoljni ulazi u vektor tabeli. U bitima 1 i 0 PSW procesora postoje biti L_1 i L_0 koji sadrže nivo prioriteta dela programa koji se trenutno izvršava (01 za `IRQ0` i 11 za `IRQ1`). Procesor ne prihvata prekid istog nivoa kao što je tekući. Nema mogućnosti selektivnog maskiranja prekida. U mikroprogramu za opsluživanje prekida na stek se stavljaju PC i PSW tim redom. Jedine naredbe koje operišu sa PSW su `PUSH PSW` i `POP PSW`.

Na linije `IRQ0` i `IRQ1` su vezane periferije `PER0` i `PER1`, respektivno. Ovim periferijama dodeljene su prekidne rutine koje počinju na adresama 23F7h i 7AB0h, respektivno, i koje su prikazane na slici 1.15 (X označava adresu početka prekidne rutine). Izgled dela memorije od lokacije 0 do 7 je prikazan na slici 1.16.

adresa	instrukcija
X+0	PUSHA

X+1	LOAD	MemCnt
X+3	INCA	
X+4	STORE	MemCnt
X+6	POPA	
X+7	RTI	

Slika 1.15. Prekidna rutina periferije PER0(X=237Fh) i PER1(X=7AB0h)

adresa	sadržaj
0000h	23F7h
0001h	7AB0h
0002h	1334h
0003h	6423h
0004h	23F7h
0005h	0010h
0006h	1001h
0007h	7AB0h

Slika 1.16. Izgled dela memorije

- Zašto se naredba na adresi $X + 1$ koduje sa dve reči?
- Koji broj šalje periferija PER1 kada joj procesor odobri zahtev za prekidom?
- Kakvu promenu u opisanoj hardverskoj konfiguraciji treba izvršiti da bi se obezbedilo da prekidna rutina od PER0 ne bude prekidana od prekida sa PER1? Ne menjati date rutine, IVTP, niti vektor tabelu.
- Da li treba nešto uraditi programski posle izmene iz prethodne tačke da bi se svakoj periferiji dodelila prekidna rutina koja joj je bila dodeljena pre te izmene? Ako treba, navesti šta; ako ne treba, objasniti zašto.

Rešenje:

a) Naredba `LOAD MemCnt` zauzima dve reči, jer koristi memorijsko direktno adresiranje za operand. Kod memorijskog direktnog adresiranja, u instrukciji se nalazi adresa memorijske lokacije u kojoj je operand, za šta je neophodna jedna reč. Za kôd instrukcije je potrebna bar još jedna reč. Prva reč ove instrukcije, dakle, sadrži kôd operacije i informacije o načinu adresiranja, a druga reč sadrži adresu operanda `MemCnt`.

b) Adrese su široke 16-bita, adresibilna jedinica je 16-bitna reč, pa svaki ulaz u vektor tabeli zauzima jednu lokaciju. Vektor tabela počinje na adresi 2 (toliko iznosi IVTP), i ima 6 ulaza. Ovi ulazi zauzimaju lokacije počev od 2 zaključno sa 7. Jedine lokacije iz ovog opsega koje sadrže date adrese prekidnih rutina su lokacije 4 i 7, za periferije PER0 i PER1 respektivno. Ovim lokacijama odgovaraju ulazi 2 i 5. Periferija PER1 šalje svoj broj ulaza kada joj procesor odobri zahtev za prekidom, a to je broj 5.

c) Jedini način da se obezbedi da se prekidna rutina od PER0 ne prekida od prekidne rutine od PER1 je da se periferiji PER0 dodeli viši prioritet nego periferiji PER1. To se obezbeđuje vezivanjem zahteva od PER0 na liniju višeg prioriteta IRQ1, a zahteva od PER1 na liniju nižeg prioriteta IRQ0.

d) Kako su periferije zadržale svoje brojeve ulaza u vektor tabeli, a ulazi u vektor tabeli zadržali svoje sadržaje (adrese prekidnih rutina), to su i posle hardverskog prevezivanja periferije zadržale svoje stare prekidne rutine, pa ništa u programu ne treba menjati.

1.6 ZADATAK

Memorija računara je veličine 64 KB, a adresiranje je bajtovsko. Procesor je jednoadresni, operiše samo sa 8-bitnim veličinama, a 16-bitni broj (adresa) se u memoriju smešta tako da je na nižoj adresi niži bajt. Mehanizam prekida je vektorisan. Tabela adresa prekidnih rutina (IVT) ima 4 ulaza (0, 1, 2 i 3) i počinje fiksno od adrese 0 operativne memorije. Sadržaj dela memorije prikazan je na slici 1.17.

adresa	sadržaj
0000h	23h

0001h	F7h
0002h	F7h
0003h	23h
0004h	23h
0005h	23h
0006h	F7h
0007h	F7h

Slika 1.17. Sadržaj dela memorije

Od unutrašnjih prekida postoji prekid posle svake instrukcije (*trap*) i prekid usled instrukcije prekida. Broj ulaza za *trap* je fiksna i jednak 2, dok instrukcija prekida sadrži broj ulaza u svom adresnom delu. Procesor ima 2 ulazne linije IRQ0 i IRQ1 za spoljne maskirajuće prekide. Ovim linijama su dodeljeni fiksni ulazi; broj ulaza jednak je broju linije. Na linije IRQ0 i IRQ1 su vezane periferije PER0 i PER1, ali na nepoznati način. Ovim periferijama dodeljene su prekidne rutine koje počinju na adresama 23F7h (za PER0) i F723h (za PER1).

U mikroprogramu za opsluživanje prekida na stek se stavljaju registri PC i PSW, zatim se brišu bitovi I (dozvoljava maskirajućih prekida) i T (*trap* režim) registra PSW.

a) Na koje linije IRQ su vezane periferije PER0 i PER1?

b) Napisati instrukciju prekida koja izaziva izvršavanje prekidne rutine za PER0.

c) Procesor u *trap* režimu počinje da izvršava deo glavnog programa koji je prikazan na slici 1.18. Naredbe NOP (*No Operation*, bez dejstva) i RTI zauzimaju svaka po jedan bajt memorije. Napisati niz naredbi (navesti njihove adrese) u sekvenci koja počinje izvršavanjem prve NOP naredbe glavnog programa, a završava se neposredno pred početak izvršavanja treće NOP naredbe glavnog programa. Napomena: u toku izvršavanja sekvence nema zahteva za maskirajućim prekidima.

```
;Glavni program-početna adresa 2000h
NOP
NOP
NOP
;Trap rutina-početnu adresu odredi sam!
NOP
RTI
```

Slika 1.18. Glavni program

d) Ukratko objasniti sve promene vrednosti bita T registra PSW do kojih dolazi u okviru tačke c).

e) Kratko objasniti zašto prekid od periferije PER1 mora biti onemogućen u toku izvršavanja sekvence naredbi glavnog programa, koja ima za cilj da, promenom ulaza tabele IVT, periferiji PER1 dodeli novu prekidnu rutinu, kao što je prikazano na slici 1.19.

```
LOAD #12h
STORE NižiBajtOdgovarajućegUlazaIVT
LOAD #34h
STORE VišiBajtOdgovarajućegUlazaIVT
```

Slika 1.19. Sekvenca naredbi glavnog programa

Rešenje:

a) S obzirom na to da linijama IRQ0 i IRQ1, prema uslovu zadatka, fiksno odgovaraju ulazi 0 i 1 tabele IVT, treba odrediti koje se adrese nalaze u ulazima 0 i 1 tabele IVT da bi se odgovorilo na postavljeno pitanje. Memorija je bajtovski organizovana, tako da adresa (kao 16-bitna veličina) zauzima dve sukcesivne memorijske lokacije pri čemu se na lokaciji sa nižom adresom nalazi niži bajt podatka (u ovom slučaju adrese prekidne rutine). Prema tome, ulaz 0 tabele IVT obuhvata lokacije memorije sa adresama 0 i 1 i u njemu se, prema slici 1.17 nalazi adresa F723h, dok ulaz 1 obuhvata adrese 2 i 3 i u njemu se nalazi adresa 23F7h. Prema tome, na liniju 0 vezana je periferija PER1, a na liniju 1 periferija PER0.

b) Instrukcija prekida kod vektorisanog mehanizma prekida u svom adresnom delu sadrži broj ulaza u tabelu IVT na osnovu koga se određuje adresa prekidne rutine na uobičajen način. Da bi se izazvalo izvršenje prekidne rutine periferije PER0, u programu je potrebno navesti instrukciju

```
INT #1
```

pri čemu je INT simbolička oznaka za instrukciju prekida, a 1 predstavlja broj ulaza u IVT koji odgovara periferiji PER0.

c) *Trap* režim rada procesora je takav režim rada u kome se po izvršenju svake instrukcije programa izvršava prekidna rutina za ovu vrstu prekida (takozvana *trap* rutina). U konkretnom slučaju, *trap* prekidima je fiksno dodeljen ulaz 2 u IVT što znači da *trap* rutina počinje od adrese 2323h. Prema svemu rečenom, niz instrukcija predstavljen adresama na kojima se instrukcije nalaze dat je na slici 1.20.

adresa	komentar
2000h	prva NOP instrukcija glavnog programa
2323h	NOP instrukcija <i>trap</i> rutine
2324h	RTI instrukcija-povratak iz <i>trap</i> rutine
2001h	druga NOP instrukcija glavnog programa
2323h	NOP instrukcija <i>trap</i> rutine
2324h	NOP instrukcija <i>trap</i> rutine

Slika 1.20. Niz naredbi koje počinju prvom NOP naredbom glavnog programa, a završavaju se neposredno pred početak izvršavanja treće NOP naredbe glavnog programa

d) U programskoj statusnoj reči PSW procesora postoji bit T za kontrolu *trap* režima rada procesora. U glavnom programu vrednost bita T je 1, što označava da je aktivan *trap* režim. U mikroprogramu za opsluživanje prekida, posle čuvanja vrednosti PC i PSW iz glavnog programa na steku, bit T se briše da bi se izbeglo prekidanje *trap* režima. Izvršavanjem poslednje naredbe *trap* rutine, naredbe RTI (*return from interrupt*) vraćaju se sa steka sadržaji registara PSW i PC za glavni program čime se bit T opet postavlja na 1.

e) Ukoliko prekid nastane posle druge instrukcije u sekvenci, u okviru opsluživanja prekida biće izvršen skok na adresu F712h, s obzirom da je to trenutni sadržaj nultog ulaza tabele IVT. Ova adresa ne odgovara ni staroj ni novoj prekidnoj rutini.

1.7 ZADATAK

Adresni prostor nekog računara je veličine 16 M (mega) 16-bitnih reči. Adresibilna jedinica je 16-bitna reč, a 32-bitni brojevi se u memoriju smeštaju tako da je na nižoj adresi nižih 16 bita. Procesor je jednoadresni, akumulator je veličine 16 bita, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IV (*Interrupt Vector*) tabela ima 64 K ulaza i počinje od adrese 0 operativne memorije. Na procesor su vezane dve periferije, PER1 i PER2. Adrese registara u kojima se čuvaju brojevi ulaza u IV tabeli ovih periferija su 6h i 7h, respektivno. Početak glavnog programa, kojim se inicijalizuje sistem, je prikazan na slici 1.21.

```
LOAD #3
OUT 6h
LOAD #1
OUT 7h
LOAD #4h
STORE 6h
LOAD #2h
STORE 7h
```

Slika 1.21. Inicijalizacija sistema

a) Koju vrednost šalje procesoru periferija PER1 kada joj procesor pošalje signal potvrde zahteva za prekid?

b) Napisati deo programa kojim se periferiji PER2 dodeljuje prekidna rutina na adresi 234567h.

c) Prikazati izgled prvih 8 ulaza u IV tabeli, označiti adrese relevantnih lokacija i upisati sadržaje u njih.

d) Na kojoj adresi se nalazi prekidna rutina za periferiju PER1?

e) Da li se iz postavke zadatka može odgovoriti na pitanje “Koja periferija ima prioritetni prekid?” Ako može, dati direktan odgovor na to pitanje; ako ne može, kratko objasniti zašto ne može.

Rešenje:

a) Periferija uvek šalje svoj broj ulaza. Taj broj nalazi se u registru na adresi 6h ulazno/izlaznog prostora. U datom inicijalizacionom programu, u ovaj registar je upisana vrednost broja ulaza, a to je 3.

b) Periferiji PER2 dodeljen je ulaz 1 u IVT, što se vidi iz inicijalizacionog programa: u registar periferije na adresi 7h upisana je vrednost 1. Adresni prostor iznosi $16 M = 2^{24}$ reči, adresibilna jedinica je reč, pa je adresna reč široka 24 bita. Kako je veličina memorijske reči jedna 16-bitna reč, za smeštanje adrese u jedan ulaz IVT potrebne su dve lokacije. Dakle, svaki ulaz IVT zauzima dve lokacije, IVT počinje od adrese 0, pa je ulaz 1 na adresama 2 i 3. Kako se na nižu adresu smešta nižih 16 bita adrese, upis adrese prekidne rutine za PER2 u ulaz 1 IVT izvršava program prikazan na slici 1.22.

```
LOAD #4567h ;napuni akumulator konstantom 4567h
STORE 000002h ;prebaci sadržaj akumulatora na adresu 000002h
LOAD #23h ;napuni akumulator konstantom 23h
STORE 000003h ;prebaci sadržaj akumulatora na adresu 000003h
```

Slika 1.22. Upis adrese prekidne rutine za PER2 u ulaz 1 IV tabele

c) Prema zaključcima iz prethodne tačke, i kako su u inicijalizacionom programu na lokacije 6 i 7 IVT upisane vrednosti 4 i 2, izgled IVT je dat na slici 1.23.

broj ulaza u IV tabelu	memorijska adresa	sadržaj	opis sadržaja
7	00000Fh		
	00000Eh		
6	00000Dh		
	00000Ch		
5	00000Bh		
	00000Ah		
4	000009h		
	000008h		
3	000007h	0002h	viših 8 bita adrese prekidne rutine za PER1
	000006h	0004h	nižih 16 bita adrese prekidne rutine za PER1
2	000005h		
	000004h		
1	000003h	0023h	viših 8 bita adrese prekidne rutine za PER2
	000002h	4567h	nižih 16 bita adrese prekidne rutine za PER2
0	000001h		
	000000h		

Slika 1.23. Izgled IV tabele

d) Prema rezultatima prethodne tačke, adresa prekidne rutine za PER1 je 020004h.

e) Ne može. Odgovor zavisi od načina vezivanja signala za prekid od periferija, na IRQ_i ulaze zahteva za prekid procesora, kao i od načina prioritiranja IRQ_i ulaza unutar samog procesora. Ni jedno ni drugo nije specifikovano u zadatku.

1.8 ZADATAK

Adresni prostor nekog računara je veličine 64 KB, adresibilna jedinica je bajt, a 16-bitni brojevi se u memoriju smeštaju tako da je na nižoj adresi niži bajt. Mehanizam prekida je vektorisan. IV (*Interrupt Vector*) tabela počinje od adrese na koju ukazuje registar IVTP. *Trap* prekidu pripada fiksno ulaz 2 u IVT. Pri prekidu se na steku čuvaju PSW i PC tim redom. Stek raste prema nižim lokacijama. Svi registri opšte namene (*Ri*) su 8-bitni. Procesor poseduje instrukcije `TRPE` (*Trap Enable*) i `TRPD` (*Trap Disable*) za dozvolu, odnosno zabranu *trap*-a. Instrukcije `TRPE` i `TRPD` ne reaguju na prekide (ni na *trap*). Dat je deo programa na slici 1.24, *trap* prekidna rutina na slici 1.25, i deo memorije počev od adrese 0 na slici 1.26. Za vreme izvršavanja datog dela programa, nema drugih prekida osim *trap*-a.

adresa	instrukcija
FF00h	TRPD
FF01h	MOV R0, #2
FF03h	TRPE
FF04h	DEC R0
FF05h	JNZ FF03h
FF08h	TRPD

Slika 1.24. Deo glavnog programa

adresa	instrukcija
00A0h	POP R1
00A1h	PUSH R1
00A2h	RTI

Slika 1.25. *Trap* prekidna rutina

adresa	sadržaj
0000h	00h
0001h	A0h
0002h	00h
0003h	0Ah
0004h	00h
0005h	A0h
0006h	00h

Slika 1.26. Deo memorije počev od adrese 0

- Koja se vrednost nalazi na adresi FF07h?
- Koja se vrednost nalazi u registru IVTP (ta vrednost je veća ili jednaka od 0, a manja od 7)?
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od FF00h, zaključno sa FF08h.
- Koja se vrednost nalazi u registru R1 neposredno pre prvog izvršavanja naredbe `RTI` na adresi 00A2h?
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od FF00h, sve dok su adrese poznate, ako se umesto *trap* rutine sa slike 1.25 koristi rutina sa slike 1.27.

adresa	instrukcija
00A0h	POP R1
00A1h	PUSH #2
00A3h	RTI

Slika 1.27. Prekidna rutina

Rešenje:

- Lokacije na adresama FF05h, FF06h i FF07h zauzima instrukcija `JNZ FF03h`. Kako se u ovoj instrukciji koristi memorijsko direktno adresiranje, sigurno je da bar dva bajta zauzima zapis same adrese FF03h. To su svakako druga dva bajta instrukcije, na adresama FF06h i

FF07h, jer prvi bajt sigurno sadrži kod operacije. Kako se 16-bitni brojevi smeštaju tako da je na višoj adresi viši bajt, sledi da je na adresi FF07h sadržaj FFh.

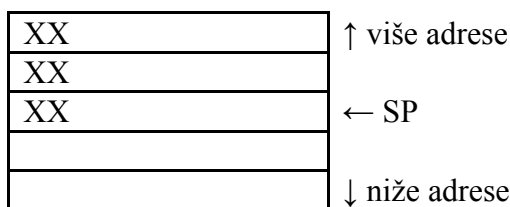
b) Kako je *trap* prekidu fiksno dodeljen ulaz 2, a adresa prekidne rutine je 00A0h, potrebno je odrediti početak IV tabele prema uslovima iz zadatka. Potrebno je u datom segmentu memorije najpre pronaći redom bajtove A0h i 00h, jer se 16-bitni brojevi smeštaju u redosledu niži pa viši bajt. Tražene vrednosti nalaze se na dva mesta: počev od adrese 1 i počev od adrese 5. Kako svaki ulaz zauzima 2 bajta, adresa početka ulaza u IVT određuje se prema formuli: $a = IVTP + 2 \cdot i$, gde je i broj ulaza. Uvek važi da je $IVTP \geq 0$. Zbog ovoga, adresa 1 ne dolazi u obzir, pa ostaje da je ulaz za *trap* prekid na adresi 5. Odavde sledi da je IVTP jednako 1.

c) Kako *trap* rutina ne menja vrednost registra R0, instrukcije od adrese FF03h do FF05h izvršiće se dva puta. Sekvenca adresa instrukcija je data na slici 1.28 (sve vrednosti su heksadecimalne).

adrese	objašnjenje
FF00h, FF01h, FF03h, FF04h	Instrukcije glavnog programa TRPD, MOV, TRPE i DEC
00A0h, 00A1h, 00A2h,	<i>Trap</i> prekidna rutina
FF05h,	Instrukcija glavnog programa JNZ
00A0h, 00A1h, 00A2h,	<i>Trap</i> prekidna rutina
FF03h, FF04h	Instrukcije glavnog programa TRPE(ne reaguje na <i>trap</i>) i DEC
00A0h, 00A1h, 00A2h,	<i>Trap</i> prekidna rutina
FF05h,	Instrukcija glavnog programa JNZ
00A0h, 00A1h, 00A2h,	<i>Trap</i> prekidna rutina
FF08h	Instrukcija glavnog programa TRPD

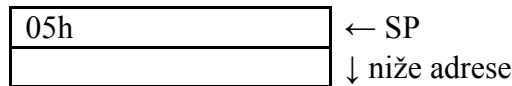
Slika 1.28. Sekvenca adresa instrukcija

d) Izgled steka na početku glavnog programa predstavljen je na slici 1.29. Stek raste prema nižim lokacijama, a predpostavljeno je da SP pokazuje na poslednju zauzetu lokaciju. Sa XX predstavljen je sadržaj koji se nalazio na steku pre početka izvršavanja instrukcije FF00h glavnog programa, taj sadržaj nam je nepoznat i nije od značaja za zadatak. Pre prelaska na *trap* prekidnu rutinu, na stek se stavlja PSW, pa zatim PC. Izgled steka nakon stavljanja PSW i PC prikazan je na slici 1.30. Kako je PC 16-bitni (jer su adrese 16-bitne), na vrh steka se stavlja ova dva bajta. Stek raste prema nižim lokacijama, a na nižu adresu se smešta niži bajt 16-bitne reči, na vrhu steka, pri ulasku u *trap* rutinu, nalazi se niži bajt adrese instrukcije koja je sledeća po redu za izvršavanje u glavnom programu. To je vrednost 5 (instrukcija na adresi FF05h je sledeća). Upravo ova vrednost se skida sa steka u *trap* rutini instrukcijom POP i smešta u R1, a zatim vraća na stek radi regularnog povratka u glavni program. Zbog toga je tražena vrednost registra R1 jednaka 5. Izgled steka nakon instrukcije POP prikazan je na slici 1.31.

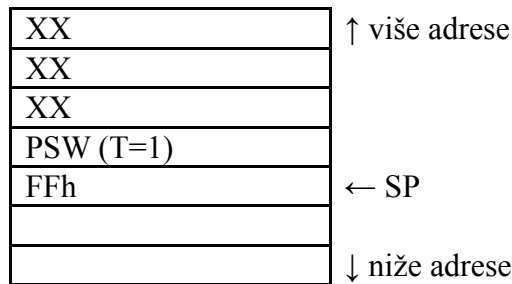


Slika 1.29. Izgled steka na početku glavnog programa





Slika 1.30. Izgled steka neposredno pre prelaska na *trap* prekidnu rutinu



Slika 1.31. Izgled steka nakon instrukcije `POP`, u registru R1 se nalazi 05h

e) Sada se u *trap* rutini, umesto vrednosti 5, na vrh steka stavlja vrednost 2. Zbog toga se, pri povratku iz ove rutine, u PC smešta vrednost FF02h. To je adresa sledeće instrukcije koja se izvršava. Na ovoj adresi nalazi se deo koda instrukcije `MOV R0, #2`, tačnije njen drugi bajt, u kome je verovatno neposredna vrednost 2 (ili čak i deo koda operacije i načina adresiranja). U svakom slučaju, ne zna se kako će procesor tumačiti vrednost koja se nalazi na adresi FF02h, kada je dohvati kao prvi bajt instrukcije. Sekvenca adresa je tako: FF00h, FF01h, FF03h, FF04h, 00A0h, 00A1h, 00A3h, FF02h, dalje nepoznato.

1.9 ZADATAK

Adresni prostor nekog računara je veličine 4 GB (giga bajta). Adresibilna jedinica je bajt, a višebajtni brojevi se u memoriju smeštaju tako da je na nižoj adresi niži bajt. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IV (*Interrupt Vector*) tabela ima 256 ulaza, a registar IVTP ima vrednost 3. Procesor poseduje dva ulaza za spoljne nemaskirajuće prekide, IRQ0 i IRQ1, pri čemu je IRQ1 višeg prioriteta, na koje su vezane periferije PER0 i PER1, respektivno. Ne postoji maskiranje prekida. Procesor prihvata prekid istog ili višeg nivoa nego što je tekući. Pri prekidu se na steku čuva najpre 8-bitni PSW, a zatim PC. Stek raste prema višim adresama. Bit L je u razredu 0 PSW. Instrukcija `RTI` reaguje na prekid.

Broj ulaza za PER1 je 9. Adresa registra PER0 u kome se čuva broj ulaza je 04h. Adresa prekidne rutine za PER0 je 0A00h, a za PER1 je 0B00h. Prekidne rutine za PER0 i PER1 izgledaju sasvim isto, kao na Slici 1.32. Posmatra se glavni program na slici 1.33, i sledeći scenario izvršavanja. Instrukcija `INCA` na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 1. instrukcije stiže zahtev za prekid od PER0, u toku 3. zahtev od PER1, u toku 4. zahtev od PER0. Izgled dela memorije počev od adrese 0 dat je na slici 1.34.

adresa	instrukcija
0A00h	INCA
0A01h	RTI

Slika 0.32. Prekidne rutine

adresa	instrukcija
0100h	CLRA
0101h	INCA

Slika 0.33. Glavni program

adresa	sadržaj	adresa	sadržaj
00000000h	00h	00000008h	0Ah
00000001h	0Ah	00000009h	00h

00000002h	00h	0000000Ah	00h
00000003h	00h	0000000Bh	0Ah
00000004h	00h	0000000Ch	00h
00000005h	0Ah	0000000Dh	A0h
00000006h	00h	0000000Eh	00h
00000007h	00h	0000000Fh	00h

Slika 0.34. Izgled dela memorije počev od adrese 0

- Napisati deo programa kojim se periferiji PER0 dodeljuje broj ulaza.
- Napisati deo programa kojim se inicijalizuje ulaz u IVT za PER1.
- Dati sekvencu adresa instrukcija koje se redom izvršavaju po datom scenariju.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka u trenutku izvršavanja 4. instrukcije. Za sačuvanu vrednost PSW dati samo vrednost bita L. Označiti vrh steka.
- Procesor je modifikovan tako da postoji bit I u PSW, koji se briše pri prihvatanju prekida. Prekidi sa IRQ0 i IRQ1 su maskirajući. Sve ostalo je isto. Dati sekvencu adresa instrukcija koje se redom izvršavaju po datom scenariju.

Rešenje:

a) Adresni prostor je veličine $4\text{ GB} = 2^{32}\text{ B}$, adresibilna jedinica je bajt, pa je adresna reč široka 32 bita. Zbog toga ulaz u IVT zauzima 4 lokacije. IVT počinje od adrese 00000003h, jer je to vrednost registra IVTP. Višebajtni brojevi se smeštaju na nižu adresu niži bajt, a adresa prekidne rutine za PER0 je 0000A00h. Tako ulaz 0 u IVT počinje od adrese 00000003h, ulaz 1 od adrese 00000007h, itd. Jedino počev od adrese 00000007h nalazi se tražena vrednost (00h, 0Ah, 00h, 00h), pa je zato broj ulaza PER0 jednak 1. Program koji dodeljuje ovaj broj ulaza treba da upiše vrednost 1 u registar na adresi 04h, i prikazan je na slici 1.35.

```
LOAD #1 ;napuni akumulator konstantom 1h
OUT 04h ;prebaci sadržaj akumulatora na adresu 04h u
;ulazno/izlaznom adresnom prostoru
```

Slika 0.35. Dodela broja ulaza za PER0

b) Ulaz u IVT za PER1 je 9, pa počinje od adrese $9 \cdot 4 + 3 = 39$. Adresa prekidne rutine za PER1 je 0000B00h. U lokacije 39, 40, 41 i 42 treba, dakle, redom upisati vrednosti 00h, 0Bh, 00h i 00h. Program koji inicijalizuje ulaz u IVT za PER1 je prikazan na slici 0.6.

```
LOAD #0Bh ;napuni akumulator konstantom 0Bh
STORE 40 ;prebaci sadržaj akumulatora na adresu 40
CLRA ;obriši sadržaj akumulatora ACC=00h
STORE 39 ;prebaci sadržaj akumulatora na adresu 39
STORE 41 ;prebaci sadržaj akumulatora na adresu 41
STORE 42 ;prebaci sadržaj akumulatora na adresu 42
```

Slika 0.36. Inicijalizacija ulaza u IVT za PER1

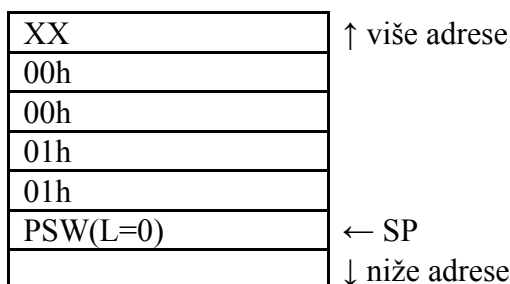
c) U toku izvršavanja instrukcije na adresi 100h glavnog programa stiže prekid od PER0 i prihvata se. Tako se ulazi u prekidnu rutinu PER0, što znači da se izvršava, kao druga, instrukcija CLRA na adresi 0A00h, a zatim, kao treća, instrukcija RTI na adresi 0A01h. U toku ove instrukcije stiže zahtev od PER1 koji je višeg prioriteta, pa kako ne postoji maskiranje, on se prihvata. Zato se kao četvrta izvršava instrukcija CLRA na adresi 0B00h. U toku njenog izvršavanja stiže ponovo zahtev za prekid od PER0 koji je nižeg nivoa prioriteta, pa se ignoriše. Izvršava se zato sledeća instrukcija na adresi 0B01h, pa se tek posle restauracije PC i PSW sa steka, koji zapravo predstavljaju kontekst glavnog programa, prihvata prekid od PER0. Tako se izvršava prekidna rutina za PER0 i posle toga vrši povratak na narednu instrukciju glavnog programa. Tražena sekvencu je prikazana na slici 1.37.

0100h,	Glavni program, instrukcija CLRA
0A00h, 0A01h,	Prekidna rutina za PER0, instrukcije INCA, RTI
0B00h, 0B01h,	Prekidna rutina za PER1, instrukcije INCA, RTI

0A00h, 0A01h, Prekidna rutina za PER0, instrukcije INCA, RTI
 0101h Glavni program, instrukcija INCA

Slika 0.37. Sekvenca adresa instrukcija koje se redom izvršavaju po datom scenariju

d) Četvrta instrukcija date sekvence je prva instrukcija prekidne rutine PER1. Pre toga je napušten kontekst glavnog programa i izvršen skok u prekidnu rutinu PER0. Kada je prihvaćen prekid od PER1, tekuća instrukcija RTI iz rutine PER0 je završena, što znači da su sa steka vraćene vrednosti konteksta glavnog programa. Kada se obrađuje prekid od PER1, na stek se ponovo stavljaju iste te vrednosti PC i PSW, pa je vrh steka u traženom trenutku prikazan na slici 1.38.



Slika 0.38. Sadržaj svih poznatih lokacija na vrhu steka u toku izvršavanja 4. instrukcije

e) Prva izmena nastaje pri obradi prekida od PER1, posle završetka instrukcije RTI. Kako je ova instrukcija vratila kontekst glavnog programa, u kome su prekidi bili dozvoljeni (jer je prihvaćen prekid od PER0), pa prema tome vraćena i vrednost 1 bita I u PSW, prekid od PER1 će se prihvatiti. Dalje je izvršavanje potpuno isto, jer se prekid od PER0 ne prihvata tokom izvršavanja prekidne rutine od PER1, jer je nižeg prioriteta. Ovaj prekid biće prihvaćen tek po završetku prekidne rutine, što znači na istom mestu kao i u tački pod c). Odgovor je, dakle, potpuno isti kao u tački pod c).

1.10 ZADATAK

Adresni prostor nekog računara je veličine 8 GB (giga bajta). Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi niža reč. Procesor je dvoadresni, ulazno-izlazni adresni prostor je memorijski preslikan, a mehanizam prekida je vektorisan. IV (*Interrupt Vector*) tabela ima 256 ulaza i nalazi se na najvišim adresama adresnog prostora. Procesor poseduje dva ulaza za spoljne maskirajuće prekide, IRQ0 i IRQ1, pri čemu je IRQ1 višeg prioriteta, na koje su vezane periferije PER0 i PER1, respektivno. U PSW postoji bit I (*Interrupt Enable*) koji se briše u mikroprogramu za obradu prekida. Adrese 8-bitnih registara u kojima se čuvaju brojevi ulaza PER0 i PER1 su 02h i 04h, respektivno. Deo glavnog programa koji vrši inicijalizaciju brojeva ulaza i vektora u IV tabeli dat je na slici 1.39 (oznaka ind predstavlja registarsko indirektno adresiranje sa pomerajem). Prekidna rutina za PER0 data je na slici 0.40 (NOP je instrukcija bez dejstva—*No Operation*).

```

MOVE #04h, 02h
MOVE #02h, 04h
MOVE #FFFFh, R0
LOOP: MOVE R0, ind(R0, FFFF0000h)
DEC R0
JNZ LOOP

```

Slika 0.39. Deo glavnog programa koji vrši inicijalizaciju brojeva ulaza i vektora u IVT

```

NOP
MOVE #0Ah, FFFFFFFE04h
MOVE #A0h, FFFFFFFE05h
MOVE #AAh, FFFFFFFE04h
RTI

```

Slika 0.40. Prekidna rutina za PER0

- a) Na kojoj adresi (heksadecimalno) počinje IV tabela?
- b) Na kojoj adresi (heksadecimalno) počinje prekidna rutina za PER0?
- c) Ako u toku izvršavanja druge instrukcije prekidne rutine za PER0 stigne zahtev za prekid od PER1, na koju će adresu “skočiti” procesor po prihvatanju ovog prekida?
- d) Odgovoriti na pitanje iz prethodne tačke, ako je prva instrukcija prekidne rutine za PER0, umesto `NOE`, instrukcija `INTE` (*Interrupt Enable*).
- e) Ako se u prvoj instrukciji datog dela glavnog programa, umesto konstante #04h navede konstanta #0204h, a sve ostalo ostane isto, da li se odgovor na pitanje iz tačke b) menja?

Rešenje:

a) Adresni prostor je veličine $8\text{ GB} = 2^3 \cdot 2^{30}\text{ B} = 2^{32}\text{ W}$, gde je W adresibilna jedinica—jedna 16-bitna reč. Prema tome, adresna reč je veličine 32 bita, adresibilna reč je 16-bitna, pa svaki ulaz u IVT zauzima dve lokacije. Kako adresni prostor obuhvata adrese počev od 0 zaključno sa FFFFFFFFh, a IVT ima $256 = 100\text{h}$ ulaza, i svaki ulaz zauzima dve lokacije, sledi da IVT počinje od adrese FFFFFFFFh – $100\text{h} \cdot 2 + 1 = \text{FFFFFFE00h}$.

b) Dati inicijalizacioni program upisuje u registar na adresi 02h, koji pripada PER0, broj 04h, što predstavlja broj ulaza za PER0. Taj ulaz 4 u IVT počinje od adrese FFFFFFFE00h + $4 \cdot 2 = \text{FFFFFFE08h}$ i zauzima lokacije FFFFFFFE08h i FFFFFFFE09h.

Sa druge strane, petlja inicijalizacionog programa u lokaciju na adresi FFFFxxxxh upisuje broj xxxxh, gde je xxxh između 0 i FFFFh. Kako se 32-bitni brojevi tumače tako što je na nižoj adresi niža reč, sledi da je adresa prekidne rutine za PER0 jednaka FE09 FE08h.

c) Prema objašnjenjima iz prethodnih tačaka, ulaz u IVT koji pripada PER1 zauzima lokacije FFFFFFFE04h i FFFFFFFE05h. Pri ulasku u prekidnu rutinu za PER0 bit I se briše, pa je prekid od PER1 maskiran, što znači da će biti prihvaćen tek po završetku ove rutine. Ova rutina ukupno menja vrednost vektora za PER1 tako da je njegova konačna vrednost 00A000AAh, što upravo predstavlja odgovor na postavljeno pitanje.

d) Sada je u toku prekidne rutine za PER0 prekid od PER1 dozvoljen, pa će se prihvatiti odmah po završetku druge instrukcije, jer je višeg prioriteta. U tom trenutku je u IVT izmenjena samo jedna “polovina” vektora za PER1, i to njegova niža reč, pa on zbog toga iznosi FE05000Ah, što je i traženi odgovor.

e) Ne. Pri prihvatanju broja ulaza, procesor uzima u obzir samo niži bajt sa magistrale podataka, jer IVT ima samo 256 ulaza. Zbog toga izmena višeg bajta broja ulaza nema efekta.

1.11 ZADATAK

Adresni prostor nekog računara je veličine 16 GB (giga bajta). Adresibilna jedinica je 32-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. Magistrala je asinhrona, a ciklus na magistrali može da traje proizvoljno dugo vreme. Procesor prihvata podatak sa magistrale podataka kada signal FC postane 1. IV (*Interrupt Vector*) tabela ima 4 ulaza i počinje od adrese 0. Procesor poseduje dva ulaza za spoljne maskirajuće prekide, IRQ0 i IRQ1, pri čemu je IRQ1 višeg prioriteta, na koje su vezane periferije PER0 i PER1, respektivno. U PSW postoji bit I (*Interrupt Enable*) u razredu 1 koji se briše u mikroprogramu za obradu prekida i bit L u razredu 0 koji sadrži nivo prioriteta tekućeg izvršavanja. Ne postoji selektivno maskiranje prekida. Procesor prihvata prekid istog nivoa kao što je tekući. Instrukcije `INTE` i `RTI` ne reaguju na prekid. Pri prekidu se na steku čuvaju PC i PSW tim redom. Adrese registara u kojima se čuvaju brojevi ulaza PER0 i PER1 su 02h i 04h, respektivno. Sadržaj dela memorijskog adresnog prostora počev od adrese 0 dat je na slici 0.41. Prekidna rutina za PER0 data je na slici 1.42, a za PER1 na Slici 1.43. Posmatra se sledeći scenario izvršavanja: u toku izvršavanja instrukcije INCA dužine jedne reči na adresi 100h glavnog programa stiže zahtev za prekid od PER0 i prihvata se. Instrukcija na adresi A00Ah označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 2. instrukcije stiže zahtev za prekid od

PER1, u toku 6. ponovo zahtev od PER0. Instrukcija na adresi 101h glavnog programa je instrukcija ADD. Sve vrednosti na slikama su heksadecimalne.

adresa	sadržaj	adresa	sadržaj
00000000h	0Ah	00000004h	0Ah
00000001h	0Ah	00000005h	A0h
00000002h	0A0Ah	00000006h	0Ah
00000003h	A00Ah	00000007h	0Ah

Slika 0.41. Sadržaj dela memorijskog adresnog prostora počev od adrese 0

adresa	instrukcija
A00Ah	INTE
A00Bh	INCA
A00Ch	RTI

Slika 0.42. Prekidna rutina za preiferiju PER0

adresa	instrukcija
0A0Ah	POPA
0A0Bh	ORA #1
0A0Ch	PUSHA
0A0Dh	RTI

Slika 0.43. Prekidna rutina za periferiju PER1

a) Nacrtati vremenski oblik svih relevantnih signala na magistrali računara (A, D, RD, WR, M/IO, FC) i signala inta1, od trenutka kada procesor odobrava zahtev za prekid preko IRQ1, do završetka tog ciklusa na magistrali?

b) Koja vrednost se nalazi na magistrali podataka u ciklusu iz tačke a)?

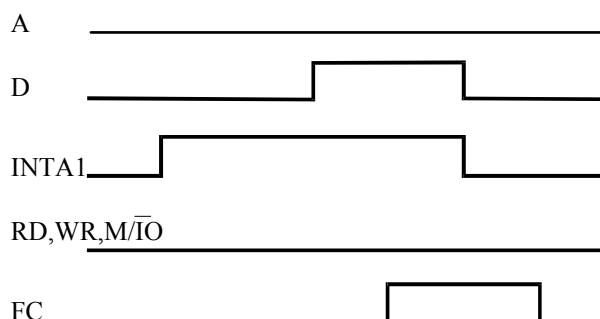
c) Napisati sekvencu instrukcija kojom se periferiji PER0 dodeljuje broj ulaza u IVT.

d) Prikazati sadržaj svih poznatih lokacija na vrhu steka neposredno posle izvršavanja 5. instrukcije. Za sačuvanu vrednost PSW dati samo vrednost dva bita najmanje težine. Označiti vrh steka.

e) Dati sekvencu adresa instrukcija koje se redom izvršavaju po datom scenariju, zaključno sa 9. (devetom).

Rešenje:

a) Ciklus na magistrali koji se odvija kada procesor odobrava zahtev za prekid određenoj periferiji predstavlja očitavanje broja ulaza u IVT te periferije. Ovaj ciklus se razlikuje od ciklusa čitanja po tome što procesor ne zadaje adresu lokacije na adresnoj magistrali, već adresira datu periferiju neposredno slanjem signala inta1 prema toj periferiji. Ova periferija šalje preko magistrale podataka svoj broj ulaza u IVT. Signali RD, WR i M/IO su neaktivni, a signal FC ima uobičajenu ulogu. Prema tome, vremenski oblik signala je prikazan na slici 0.44.



Slika 0.44. Vremenski oblik svih relevantnih signala na magistrali računara

b) Na magistrali podataka se nalazi broj ulaza u IVT koji pripada PER1, koja je vezana na IRQ1. Ovaj broj ulaza možemo odrediti prema sadržaju IVT koja počinje od adrese 0, sadrži 4 ulaza, a svaki ulaz zauzima 1 lokaciju. Ovo se vidi iz uobičajene računice: $16 \text{ GB} = 2^{34} \text{ B} = 2^{32} \text{ W}$, gde je W adresibilna jedinica—jedna 32-bitna reč, pa je adresna reč

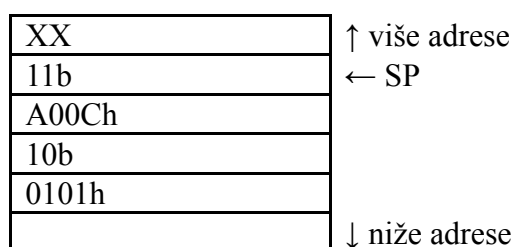
široka 32 bita, koliko iznosi i širina jedne adresibilne jedinice. Kako se jedino na adresi 2 u IVT nalazi vrednost 0A0Ah adrese prekidne rutine za PER1, sledi da je traženi broj ulaza 2.

c) Kao i u prethodnoj tački, dolazi se do zaključka da je broj ulaza u IVT za PER0 jednak 3. Traženi program je prikazan na slici 1.45.

```
LOAD #3 ;napuni akumulator konstantom 3
OUT 02h ;prebaci sadržaj akumulatora na adresu 02h u
;ulazno/izlaznom adresnom prostoru
```

Slika 0.45. Dodela broja ulaza u IV tabeli za PER0

d) Dati scenario se odvija ovako: izvršava se instrukcija glavnog programa na adresi 100h, zatim se prihvata prekid od PER0, pa se izvršavaju instrukcije na adresama A00Ah i A00Bh. Tada stiže prekid od PER1 koji je višeg prioriteta i dozvoljen je (jer je izvršena instrukcija INTE), pa se prihvata. Zatim se izvršavaju instrukcije na adresama 0A0Ah, 0A0Bh i 0A0Ch. Ove instrukcije menjaju reč na vrhu steka tako što postavljaju najniži bit na 1, a ostale bite ne menjaju. U tom trenutku se na vrhu steka nalazi sačuvan kontekst (PC i PSW) iz prekidne rutine za PER0, a ispod njega kontekst glavnog programa. U oba slučaja bit I je bio postavljen na 1, jer su prekidi prihvatani. L bit je u glavnom programu imao vrednost 0. Izgled vrha steka je dat na slici 1.46.



Slika 0.46. Izgled vrha steka neposredno posle izvršavanja 5. instrukcije

e) Prvih pet instrukcija je navedeno u prethodnoj tački. Scenario se dalje nastavlja ovako. Izvršava se instrukcija RTI na adresi 0A0Dh. U toku njenog izvršavanja stiže prekid of PER0, ali se ne prihvata jer ova instrukcija ne reaguje na prekid. Ovaj zahtev za prekidom se pamti, da bi se prihvatio kada svi uslovi za to budu zadovoljeni.

Tako se izvršavanje vraća na instrukciju na adresi A00Ch, jer je to vrednost koja je skinuta sa steka i smeštena u PC. Izvršava se instrukcija RTI, koja opet ne reaguje na zapamćeni prekid, pa se izvršavanje vraća u glavni program, na instrukciju na adresi 101h. Kako je ovo instrukcija ADD, svi uslovi za prihvatanje zapamćenog prekida su zadovoljeni: ova instrukcija reaguje na prekid, bit I je postavljen na 1, i prekid od PER0 je istog nivoa prioriteta kao i glavni program, pa se prihvata. Tražena sekvenca je data na slici 1.47.

A00Ah, A00Bh, 0A0Ah, 0A0Bh, 0A0Ch, 0A0Dh, A00Ch, 101h, A00Ah

Slika 1.47. Sekvenca adresa instrukcija koje se izvršavaju po datom scenariju

Diskusija:

Ako pretpostavimo da instrukcija RTI reaguje na prekid, prekid od PER0 se opet neće prihvatiti posle završetka instrukcije RTI na adresi 0A0Dh, jer je u tom trenutku vraćen kontekst sa vrha steka, a time i bit L postavljen na 1 (jer je to izmenjena vrednost na vrhu steka). Ova vrednost koduje viši nivo prioriteta nego što ga ima prekid od PER0. Ovo predstavlja način da se “nasilno” izmeni nivo prioriteta nekog izvršavanja, promenom vrednosti zapamćenog konteksta na vrhu steka.

1.12 ZADATAK

Adresni prostor nekog računara je veličine 128 KB. Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je

vektorisan. IV (*Interrupt Vector*) tabela ima 4 ulaza i počinje od adrese na koju ukazuje registar IVTP čija je trenutna vrednost 2. Procesor poseduje četiri ulaza za spoljne maskirajuće prekide, IRQ0...IRQ3, pri čemu je IRQ3 najvišeg prioriteta, na koje su vezane periferije PER0...PER3, respektivno. U PSW postoji bit I (*Interrupt Enable*) u razredu 2 koji se briše u mikroprogramu za obradu prekida i biti L u razredima 1...0 koji sadrže nivo prioriteta tekućeg izvršavanja. Pri prekidu se na steku čuvaju samo PC i PSW. Za selektivno maskiranje prekida služi registar IMR u kome biti 3...0 odgovaraju redom ulazima IRQ3...IRQ0, dok ostali biti nisu značajni. Instrukcija `STOREA IMR` smešta sadržaj akumulatora u IMR, dok instrukcija `LOADA IMR` smešta sadržaj IMR u akumulator. Instrukcije `INTE` i `RTI` ne reaguju na prekid. Adrese registara u kojima se čuvaju brojevi ulaza PER0, PER1 i PER2 su 02h, 03h i 04h, respektivno. Sadržaj dela memorijskog adresnog prostora počev od adrese 0 dat je na slici 1.48. Prekidna rutina za PER0 data je na slici 1.49, a za PER2 na slici 1.50. Posmatra se sledeći scenario izvršavanja: u toku izvršavanja instrukcije dužine jedne reči na adresi 100h glavnog programa stiže zahtev za prekid od PER0 i prihvata se. U toku izvršavanja instrukcije na adresi A00Dh stiže zahtev za prekid od PER2.

adresa	sadržaj
0000h	000Ah
0001h	000Ah
0002h	0A0Ah
0003h	A00Ah

Slika 0.48. Sadržaj dela memorijskog adresnog prostora počev od adrese 0

adresa	instrukcija
A00Ah	INTE
A00Bh	LOADA #3
A00Dh	STOREA IMR
A00Eh	RTI

Slika 0.49. Prekidna rutina za PER0

adresa	instrukcija
0A0Ah	RTI

Slika 0.50. Prekidna rutinaza PER1

- Koji je sadržaj registra PER0 koji se nalazi na adresi 02h, a koji registra PER2 na adresi 04h?
- Napisati sekvencu instrukcija kojom se periferiji PER1 dodeljuje broj ulaza 2 u IVT.
- Napisati sekvencu instrukcija kojom je inicijalizovan vektor u IVT za PER1, prema uslovu iz prethodne tačke.
- Dati sekvencu adresa instrukcija koje se redom izvršavaju po datom scenariju, sve dok su te adrese poznate.
- Ako u toku izvršavanja instrukcije na adresi A00Dh, istovremeno sa prekidom od PER2 stigne i prekid od PER0, koja se instrukcija (dati njenu adresu) izvršava neposredno posle instrukcije na adresi A00Dh?

Rešenje:

a) Adresni prostor je veličine $128 \text{ KB} = 2^{17} \text{ B} = 2^{16} \text{ W}$, gde je W adresibilna jedinica—jedna 16-bitna reč. Prema tome, adresna reč je široka 16 bita, kolika je širina i adresibilne jedinice. Zato svaki ulaz u IVT zauzima jednu lokaciju. IVT počinje od adrese 0002h. Kako se na adresi 02h u ulazno-izlaznom adresnom prostoru nalazi adresa prekidne rutine za PER2, a na adresi 03h za PER0, ulaz u IVT koji odgovara PER0 je 1, a PER2 je 0. Ovo su upravo tražene vrednosti sadržaja registara periferija u kojima se čuvaju brojevi ulaza.

b) Traženi program je prikazan na slici 1.51.

```
LOADA #2      ;napuni akumulator konstantom 2
OUTA 03h     ;prebaci sadržaj akumulatora na adresu 03h u
```

;ulazno/izlaznom adresnom prostoru

Slika 1.51. Sekvenca instrukcija kojom se se perifereiji PER1 dodeljuje broj ulaza 2 u IVT

c) Perifeiji PER1 pripada prekidna rutina na adresi 0Ah, jer je njen ulaz u IVT na adresi 4, prema objašnjenjima iz tačke a). Traženi program je dat na slici 1.52

```
LOADA #0Ah ;napuni akumulator konstantom 0Ah
STOREA 04h ;prebaci sadržaj akumulatora na adresu 04h
```

Slika 0.52. Sekvenca instrukcija kojom je inicijalizovan vektor u IVT za PER1

d) Prekidna rutina za PER0 smešta u registar IMR binarnu vrednost 0011, što znači da su prekidi sa IRQ0 i IRQ1 dozvoljeni, a prekid sa IRQ2 maskiran. Prema tome, prekid od PER2 se neće prihvatiti tokom ove prekidne rutine, a ni po povratku u glavni program, jer ista vrednost ostaje u IMR. Tražena sekvenca je data na slici 1.53.

100h, A00Ah, A00Bh, A00Dh, A00Eh, 101h

Slika 0.53. Sekvenca adresa instrukcija koje se redom izvršavaju po datom scenariju

e) Prekid od PER0 je dozvoljen, a istog je nivoa prioriteta kao i tekuće izvršavanje. Kako u PSW postoje samo dva bita L za kodovanje pet nivoa: nivo glavnog programa i 4 prekida, sigurno je da procesor prihvata prekid istog nivoa prioriteta 0. Zato se prekid od PER0 prihvata i prelazi na instrukciju na adresi A00Ah.

1.13 ZADATAK

Adresni prostor nekog računara je veličine 8 GB (gigabajta). Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi viša reč. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IV (*Interrupt Vector*) tabela ima 16 ulaza i počinje od adrese na koju ukazuje registar IVTP čija je trenutna vrednost 2. Procesor poseduje četiri ulaza za spoljne maskirajuće prekide, IRQ0...IRQ3, pri čemu je IRQ3 najvišeg prioriteta, na koje su vezane perifrije PER0...PER3, respektivno. Liniji IRQ i odgovara fiksno ulaz i u IVT. U PSW postoji bit I (*Interrupt Enable*) u razredu 2 koji se briše u mikroprogramu za obradu prekida i biti L koji sadrže nivo prioriteta tekućeg izvršavanja. Ne postoji selektivno maskiranje prekida. Pri obradi prekida se na steku čuvaju PC i PSW. Sadržaj dela memorijskog adresnog prostora počev od adrese 0 dat je na slici 1.54. Prekidna rutina za PER0 data je na slici 1.55., a za PER2 na slici 1.56. Posmatra se sledeći scenario izvršavanja dat sekvencom adresa instrukcija koje se redom izvršavaju: 100 (glavni program), $X + 0$, $X + 1$, $Y + 0$, $X + 0$, $X + 1$.

adresa	sadržaj	adresa	sadržaj
00000000h	000Ah	00000004h	000Bh
00000001h	000Ah	00000005h	0A0Ah
00000002h	0A0Ah	00000006h	000Bh
00000003h	A00Ah	00000007h	000Bh

Slika 0.54. Sadržaj dela memorijskog adresnog prostora počev od adrese 0

adresa	instrukcija
X+0	INTE
X+1	INCA
X+2	RTI

Slika 0.55. Prekidna rutina za PER0

adresa	instrukcija
Y+0	RTI

Slika 0.56. Prekidna rutina za PER1

a) Koliko iznosi X, a koliko Y?

- b) Napisati sekvencu instrukcija kojom je inicijalizovan vektor u IVT za PER1.
- c) Da li instrukcija `RTI` reaguje na prekid? Obrazložiti?
- d) Da li procesor prihvata prekid istog nivoa prioriteta kao što je tekući ili ne? Obrazložiti.
- e) Ako se pretpostavi da posle date sekvence više nije bilo zahteva za prekid i da je na lokaciji 100h instrukcija `INCA`, napisati nastavak date sekvence, sve dok su adrese poznate.

Rešenje:

- a) Kao i u prethodnim zadacima, dolazi se do vrednosti $X = 0A0AA00Ah$ i $Y = 000B000Bh$.
- b) Tražena sekvenca je prikazana na slici 1.57.

```
LOADA #000Bh
STOREA 0004h
LOADA #0A0Ah
STOREA 0005h
```

Slika 0.57. Sekvenca instrukcija kojom je inicijalizovan vektor u IVT za PER1

- c) Da. Posle $Y + 0$ bi trebalo da se izvršava $X + 2$, ali se izvršava $X + 0$. Jedini mogući razlog je prekid koji se dogodio u toku izvršavanja instrukcije `RTI`.
- d) Da. Prekid iz c) je stigao od PER0 i prihvaćen je u trenutku kada je instrukcija `RTI` već povratila vrednost bita L nivoa prioriteta istog ovog prekida. Naime, instrukcija `RTI`, kao i svaka druga instrukcija, najpre završava svoje delovanje, koje se u ovom slučaju sastoji iz povratka konteksta procesora sa steka, a tek potom obrađuje prekid, u smislu da odlučuje da li se prekid prihvata, prema tekućem kontekstu.
- e) Nastavak date sekvence je: $X + 2, X + 3, X + 2, X + 3, 0101h$.

1.14 ZADATAK

Procesor podržava spoljašnje maskirajuće prekide, za koje zahtevi dolaze po linijama IRQ0 do IRQ7 procesora, pri čemu su svi ulazi istog nivoa prioriteta. Pri prekidu se na steku čuvaju PC, PSW i svi registri opšte namene R0 do R7, tim redom, i svi maskirajući prekidi se onemogućavaju brisanjem bita I u PSW. Za potrebe maskiranja prekida postoji 32-bitni registar IMR. Biti 0 do 7 ovog registra maskiraju ulaze IRQ0 do IRQ7, redom; ostali biti nisu značajni. Jedine instrukcije za rad sa ovim registrom su: `MOV IMR, Ri` i `MOV Ri, IMR` gde je R_i jedan od R0 do R7. Dejstvo ovih instrukcija, vidljivo za programera, je sledeće. Instrukcija `MOV IMR, Ri` prebacuje osam najmlađih bita registra R_i u najmlađe bite registra IMR, a ostale bite registra IMR postavlja na nulu. Instrukcija `MOV Ri, IMR` prebacuje 32 bita registra IMR u registar R_i . Osim toga, postoje još i instrukcije `INTE` i `INTD` za dozvolu/maskiranje svih maskirajućih prekida.

Potrebno je obezbediti da zahtev na liniji IRQ_i bude višeg prioriteta od $IRQ(i + 1)$, za svako i , odnosno da se u toku prekidne rutine za IRQ_i ne prihvataju prekidi sa ulaza IRQ_j , gde je $j \geq i$, a prihvataju svi prekidi sa IRQ_k , gde je $k < i$. Kako ne postoji hardversko prioritiranje prekida, potrebno je ovo prioritiranje izvesti softverski. Napisati naredbe koje treba navesti na početku i na kraju svake prekidne rutine (koja odgovara IRQ_i), da bi se obezbedilo zahtevano softversko prioritiranje.

Rešenje:

Zahtevano softversko prioritiranje može se obezbediti korišćenjem mehanizma selektivnog maskiranja prekida: kada se uđe u prekidnu rutinu koja odgovara IRQ_i , potrebno je maskirati sve prekide kojima se dodeljuje viši nivo prioriteta. Veoma je važno da se pri izlasku iz prekidne rutine registar IMR vrati na onu vrednost koju je imao pre ulaska u prekidnu rutinu, jer njegova vrednost zapravo predstavlja deo konteksta izvršavanja, pošto on čuva na neki način informaciju o tekućem nivou prioriteta. Kako se bit I briše hardverski pri prihvatanju prekida, nema potrebe odmah na ulasku u prekidnu rutinu softverski maskirati sve prekide dok se vrše kritične operacije. Traženi segmenti su prikazani na slici 1.58.

```

;Početak prekidne rutine za ulaz IRQi
INTHi:MOV  R0, IMR          ;sačuvaj vrednost IMR na steku
        PUSH  R0
        MOV   R0, #0...01...1B ;ovde ima 32-i nula i i jedinica
        MOV   IMR, R0        ;maskiraj sve niže prekide,
        INTE                ;i dozvoli prekid svim višim;
        ...                  ;ostatak prekidne rutine
;Kraj prekidne rutine:
        POP   R0            ;restauriraj IMR
        INTD
        MOV   IMR, R0
        RTI

```

Slika 0.58. Naredbe koje treba navesti na početku i na kraju svake prekidne rutine (koja odgovara IRQ_i), da bi se obezbedilo zahtevano softversko prioritiranje

1.15 ZADATAK

Procesor je jednoadresni i ima samo jedan registar podataka (akumulator, ACC) i 16 adresnih registara, X0 do X15, svi su 32-bitni. Pored ovih registara, postoji registar PSW sa uobičajenim značenjem, koji ima indikatore N, Z, C i V, bit I (*Interrupt Enable*) i bite L1...L0. Registar X15 služi kao pokazivač steka. Stek raste prema višim adresama, a X15 pokazuje na poslednju zauzetu lokaciju.

Postoje četiri ulaza za spoljašnje maskirajuće prekide $IRQ_0...IRQ_3$, pri čemu je IRQ_3 najvišeg prioriteta. U mikroprogramu za obradu prekida na steku se čuvaju ACC, PSW i PC tim redom, a I se briše. Biti L označavaju tekući nivo prioriteta izvršavanja. Ne postoji mogućnost selektivnog maskiranja prekida. Postoji i *trap* režim, koji se zadaje instrukcijom $TRPE$, a ukida instrukcijom $TRPD$. IV tabela počinje od adrese 0.

Za opisani procesor projektuje se program za sakupljanje statistike o izvršenim instrukcijama. Ovaj program zasniva se na *trap* prekidnoj rutini, koja ima zadatak da "pročita" instrukciju koja je na redu za izvršavanje u programu koji se analizira. Sama analiza instrukcije nalazi se u potprogramu *Stat*, koji ne treba pisati. Potrebno je napisati samo *trap* prekidnu rutinu, koja ima zadatak da instrukciju koja je sledeća na redu za izvršavanje u glavnom programu smesti u lokaciju na adresi 100h, pozove proceduru *Stat* ($CALL Stat$), i po povratku iz nje vrati se u glavni program na regularan način. Eventualnu drugu reč instrukcije analiziraće procedura *Stat*. Dozvoljeno je koristiti sve načine adresiranja.

a) Napisati opisanu *trap* prekidnu rutinu.

b) Da li je moguće, u opisanom procesoru, hardverski obezbediti da se *trap* prekidna rutina ni na koji način ne može prekinuti od strane spoljašnjeg maskirajućeg prekida, čak i ako je bit I u njoj postavljen na 1 (dozvoljeni su maskirajući prekidi)? Ako jeste, precizno objasniti kako to treba izvesti. Ako nije, precizno objasniti šta treba izmeniti u procesoru, da bi to bilo moguće.

c) Da li je moguće analizirati, umesto instrukcije koja je sledeća na redu za izvršavanje, instrukciju koja je upravo izvršena (iz koje se stiglo u *trap* rutinu)? Ako jeste, precizno objasniti kako to treba izvesti. Ako nije, precizno objasniti zašto to nije moguće.

Rešenje:

a) Tražena *trap* prekidna rutina treba da očita instrukciju koja je naredna na redu za izvršavanje. Na ovu instrukciju ukazuje vrednost PC po završetku instrukcije iz koje se stiglo u rutinu. Ova vrednost PC nalazi se na vrhu steka po ulasku u rutinu. Tako *trap* rutina treba da očita sa steka vrednost PC i da očita instrukciju sa te adrese. Pretpostavljajući da postoji registarsko indirektno adresiranje preko adresnih registara X, tražena rutina je data na slici 1.59.

```

;Trap prekidna rutina
INTTP:LOAD (X15)          ;ACC := Mem[X0], stara vrednost PC
        PUSH  X0          ;sačuvaj X0
        MOV   X0,ACC      ;X0 := ACC, stara vrednost PC
        LOAD (X0)        ;ACC := Mem[X0], naredna instrukcija

```

```

MOV   X0, #100h    ;X0:= 100h
STORE(X0)          ;Mem[100h] := ACC
CALL  Stat         ;poziv potprograma Stat
POP   X0           ;povrati X0
RTI

```

Slika 0.59. *Trap* prekidna rutina

b) Da bi postavljeni zahtev bio ispunjen, potrebno je da *trap* prekid bude strogo višeg nivoa prioriteta od svih maskirajućih prekida i da utiče na L bite (iako to obično nije slučaj). Naime, pošto postoje samo dva L bita i četiri spoljašnja prekida, procesor prihvata i prekide istog nivoa prioriteta kao što je tekući. Prema tome, *trap* ne može biti nivoa 3, već mora biti nivoa 4. Da bi to bilo ispunjeno, potrebno je obezbediti tri L bita. Prema tome, u opisanom procesoru nije moguće izvesti traženo, već je potrebno procesor izmeniti tako da poseduje tri L bita, i *trap* prekidu pridružiti nivo 100b.

c) Nije moguće. U trenutku ulaska u prekidnu rutinu, poznata je adresa instrukcije koja je naredna za izvršavanje (to je vrednost PC posle završetka instrukcije). Adresu prethodne (upravo izvršene) instrukcije nije moguće znati, iz više razloga. Prvo, instrukcije su različite dužine, pa se ne može znati da li je prethodna instrukcija bila dugačka jednu ili više reči. Drugo, prethodna instrukcija je mogla biti i instrukcija skoka (uslovnog ili bezuslovnog), pa je izvorište skoka opet nepoznato. Jedini slučaj kada ovo ne važi jeste kada je prekid izazvan zbog greške u instrukciji i instrukcija nije uspeła da se izvrši do kraja (npr. deljenje nulom). Tada se pre ulaska u prekidnu rutinu na steku pamti vrednost PC koji pokazuje na instrukciju u kojoj je došlo do greške.

1.16 ZADATAK

Adresni prostor nekog računara je veličine 64 KB. Adresibilna jedinica je bajt, a višebajtni brojevi se u memoriju smeštaju tako da je na nižoj adresi viši bajt. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. Svi registri procesora su 16-bitni. IV (*Interrupt Vector*) tabela počinje od adrese na koju ukazuje registar IVTP, čija je početna vrednost 0. Procesor poseduje dva ulaza za spoljne maskirajuće prekide, IRQ0 i IRQ1, pri čemu je IRQ1 višeg prioriteta, na koje su vezane periferije PER0 i PER1, respektivno. Postojanje prekida se pamti (procesor reaguje na impuls). Ne postoji maskiranje prekida. Instrukcija RTI reaguje na prekid. Adrese registara u kojima se čuvaju brojevi ulaza PER0 i PER1 su 02h i 04h, respektivno, s tim da procesor pri očitavanju broja ulaza uzima u obzir samo niži bajt 16-bitne reči. Broj ulaza za PER0 je 2, a za PER1 je 0. Sadržaj dela operativne memorije dat je na slici 1.60., a prekidne rutine za PER0 i PER1 date su na slikama 1.61. i 1.62., respektivno. U toku izvršavanja jednobajtne instrukcije *INCA* na adresi 100h glavnog programa stižu istovremeno prekidi od obe periferije.

adresa	sadržaj	adresa	sadržaj
0000h	00h	0005h	0Ah
0001h	0Ah	0006h	00h
0002h	0Ah	0007h	A0h
0003h	0Ah	0008h	00h
0004h	A0h	0009h	0Ah

Slika 0.60. Sadržaj dela operativne memorije

adresa	instrukcija
X+0	LOADW #4 ;ACC := 4
X+4	STOREW IVTP ;IVTP := ACC
X+6	RTI

Slika 0.61. Prekidna rutina za PER0

adresa	instrukcija
Y+0	RTI ;2 bajta

Slika 0.62. Prekidna rutina za PER1

- a) Koliko bajtova zauzima IV tabela?
- b) Napisati sekvencu instrukcija kojima se periferijama PER0 i PER1 dodeljuju brojevi ulaza.
- c) Napisati sekvencu instrukcija kojima je inicijalizovan vektor u IVT za PER0.
- d) Napisati sekvencu adresa instrukcija koje se izvršavaju po navedenom scenariju, sve do povratka u glavni program.
- e) Ako se posle završenog scenarija iz tačke d) (po povratku u glavni program) odmah ponovo istovremeno jave prekidi od obe periferije, napisati nastavak sekvence adresa instrukcija koje se izvršavaju, sve dok su one poznate. Pretpostaviti da je i na adresi 101h instrukcija INCA.

Rešenje:

a) Rečeno je da procesor uzima samo jedan bajt kao broj ulaza, kada potvrđuje prekid. To znači da vrednosti broja ulaza mogu biti 0...255. Procesor će taj broj ulaza iskoristiti da pristupi vektor tabeli. Prema tome, vektor tabela sadrži 256 ulaza, bez obzira što postoje samo dva ulazna signala za spoljne prekide. Tim prekidima mogu odgovarati bilo koji od 256 ulaza. Kako svaki ulaz zauzima dva bajta (adrese su 16-bitne), IVT zauzima 512 bajtova.

b) Sekvenca je sledeća:

```
LOADW #2      ;napuni akumulator konstantom 2
OUTW  2h      ; prebaci sadržaj akumulatora na adresu 2h u
           ;ulazno/izlaznom adresnom prostoru
LOADW #0      ;napuni akumulator konstantom 0
OUTW  4h      ; prebaci sadržaj akumulatora na adresu 4h u
           ;ulazno/izlaznom adresnom prostoru
```

Slika 0.63. Sekvenca instrukcija kojima se periferijama PER0 i PER1 dodeljuju brojevi ulaza

c) Sekvenca je sledeća:

```
LOADW #A00Ah   ;napuni akumulator konstantom A00Ah
STOREW 4h      ;prebaci sadržaj akumulatora na adresu 4h
```

Slika 0.64. Sekvenca instrukcija kojima je inicijalizovan vektor u IVT za PER0

d) Prekid od PER1 je višeg prioriteta, pa se prihvata najpre ovaj prekid, a zatim prekid od PER0. Tražena sekvenca je prikazana na slici 1.65.

100h, 000Ah, A00Ah, A00Eh, A010h, 101h

Slika 0.65. Sekvenca adresa instrukcija koje se izvršavaju po navedenom scenariju, sve do povratka u glavni program.

e) U prekidnoj rutini za PER0 menja se vrednost IVTP na 4. Čitajući novu vektor tabelu, može se zaključiti da su sada prekidne rutine praktično zamenjene. Kada se drugi put obrađuje prekid, sprovodi se potpuno ista procedura: procesor prihvata prekid od PER1, uzima od periferije broj ulaza, određuje vektor iz IVT pomoću tekuće vrednosti IVTP i skače na prekidnu rutinu. Nastavak sekvence je prikazan na slici 1.66.

101h, A00Ah, A00Eh, A010h, 0Ah, 102h

Slika 0.66. Nastavak sekvence adresa instrukcija koje se izvršavaju

1.17 ZADATAK

Adresni prostor nekog računara je veličine 4 GB. Adresibilna jedinica je bajt, a višebajtni brojevi se u memoriju smeštaju tako da je na nižoj adresi viši bajt. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. Svi registri procesora su 32-bitni. Pri prekidu se na steku čuvaju PC i PSW tim redom. Stek raste prema višim lokacijama. U registru PSW postoji bit L koji čuva nivo prioriteta tekućeg izvršavanja. Instrukcija RTI reaguje na prekid. IV (*Interrupt Vector*) tabela sadrži 4 ulaza i počinje od adrese na koju ukazuje registar IVTP, čija je vrednost 2. Procesor poseduje dva ulazna

signala za spoljne maskirajuće prekide, IRQ0 i IRQ1, pri čemu je IRQ1 višeg prioriteta, na koje su vezane periferije PER0 i PER1, respektivno. Ne postoji maskiranje prekida. Postojanje zahteva za prekid se pamti (procesor reaguje na impuls). Adrese registara u kojima se čuvaju brojevi ulaza PER0 i PER1 su 02h i 04h, respektivno. Broj ulaza za PER1 je 1. Sadržaj dela operativne memorije dat je na slici 1.67, a prekidne rutine za PER0 i PER1 date su na slikama 1.68. i 1.69., respektivno. Posmatra se sledeći scenario: u toku izvršavanja 4-bajtna instrukcije INCA na adresi 100h glavnog programa stiže prekid od PER1; ova instrukcija INCA označena je kao prva (1.); u toku izvršavanja 2. instrukcije stiže prekid od PER0, a u toku 4. ponovo prekid od PER1. Pretpostaviti da su u glavnom programu na adresama 104h i 108h takođe instrukcije INCA.

adresa	sadržaj	adresa	sadržaj
00000000h	000Ah	0000000Ah	00A0h
00000002h	0A0Ah	0000000Ch	000Ah
00000004h	0AA0h	0000000Eh	A00Ah
00000006h	0AAAh	00000010h	00AAh
00000008h	A00Ah	00000012h	0A0Ah

Slika 0.67. Sadržaj dela operativne memorije

adresa	instrukcija
A00A00AAh	RTI

Slika 0.68. Prekidna rutina za PER0

adresa	instrukcija
X+0	INCA
X+4	RTI

Slika 0.69. Prekidna rutina za PER1

- Koji broj šalje periferija PER0 kada joj procesor pošalje potvrdu prekida?
- Napisati sekvencu instrukcija kojima je inicijalizovan vektor u IVT za PER1.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka posle završetka 3. instrukcije prema datom scenariju. Označiti vrh steka. Za PSW dati samo vrednost bita L.
- Napisati sekvencu adresa instrukcija koje se izvršavaju po navedenom scenariju, sve dok su adrese poznate.
- Odgovoriti na pitanje pod d) ukoliko se pretpostavi da instrukcija RTI ne reaguje na prekid.

Rešenje:

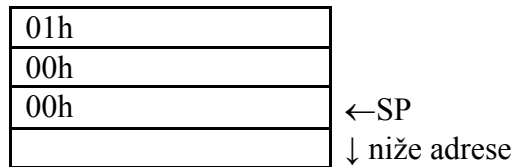
- Periferija šalje svoj broj ulaza. Pregledom sadržaja memorije koju zauzima IVT, uzimajući u obzir da je IVTP=2 i da svaki ulaz zauzima 4 bajta, da je adresa 32-bitna, a da se višebajtni brojevi smeštaju tako da je na nižoj adresi viši bajt, dobija se da je broj ulaza za PER0 jednak 3.
- Tražena sekvenca je data na slici 1.70.

```
LOAD #0AAAA00Ah
STORE 6h
```

Slika 0.70. Sekvencu instrukcija kojima je inicijalizovan vektor u IVT za PER1

- Prekid od PER0 u toku izvršavanja instrukcije INCA prekidne rutine za PER1 se ne prihvata odmah, već tek posle završetka instrukcije RTI. Posle završetka te instrukcije, stanje na steku je prikazano na slici 1.71.

XX	↑ više adrese
PSW0	
PSW1	
PSW2	
PSW3 (L=0)	
04h	



Slika 0.71. Sadržaj svih poznatih lokacija na vrhu steka posle završetka 3. instrukcije

d) Tražena sekvenca je prikazana na slici 1.72.

100h, 0AAA A00Ah, 0AAA A00Eh, A00A 00AAh, 0AAA A00Ah, 0AAA A00Eh, 104h, 108h

Slika 0.72. Sekvenca adresa instrukcija koje se izvršavaju po navedenom scenariju

e) Sada instrukcija RTI ne reaguje na prekid, pa se prekid obrađuje tek posle prve naredne instrukcije. Posle povratka u glavni program na adresu 104h iz prekidne rutine za PER1 (ovo je 4. instrukcija u toku koje stiže zahtev i od PER1), postoje dva zahteva za prekid, od obe periferije. Prihvatiće se zato zahtev od PER1, jer je višeg prioriteta. Sekvenca je prikazana na slici 1.73.

100h, 0AAA A00Ah, 0AAA A00Eh, 104h, 0AAA A00Ah, 0AAA A00Eh, 108h, A00A 00AAh, 10Ch

Slika 0.73. Sekvencu adresa instrukcija koje se izvršavaju kada RTI ne reaguje na prekid

1.18 ZADATAK

Adresni prostor nekog računara je veličine 1GB. Adresibilna jedinica je 16-bitna reč. Instrukcije mogu imati najviše jedan memorijski operand. Sve instrukcije koje nemaju memorijske operande staju u jednu reč, a kod onih koje imaju memorijski operand adresa tog operanda se nalazi neposredno iza reči sa instrukcijskim kodom. Višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi viša reč. Procesor je dvoadresni, stek raste prema nižim adresama, SP ukazuje na prvu slobodnu adresu, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IVT (Interrupt Vector Table) počinje od adrese na koju ukazuje registar IVTP čija je vrednost 80h. Procesor poseduje tri ulaza za spoljne prekide različitih prioriteta, IRQ0, IRQ1 i IRQ2, na koje su vezane periferije PER0, PER1 i PER2, respektivno. Ne postoji maskiranje prekida, a prekidi istog prioriteta se odmah prihvataju. Pri obradi prekida se na steku čuvaju PSW i PC (prvo se na stek stavlja PSW, pa zatim PC; registar PSW je 16-bitni). Prekidna rutina za PER0 data je na slici 1.74., za PER1 na slici 1.75., a za PER2 na slici 1.76. Adrese registara u kojima se čuvaju brojevi ulaza u IVT za PER0, PER1 i PER2 su 80h, 81h i 82h, respektivno. Adresa upravljačkog registra periferije PER0, u kome je bit 0 Start bit, je A0h, a adresa registra podataka PER0 je A2h.

```
Int0: IN R1,A2h
      STORE (F000h),R1
      LOAD R1,F000h
      INC R1
      STORE F000h,R1
      LOAD R1,F001h
      DEC R1
      STORE F001h,R1
      JNZ R1,Back
      OUT A0h,#0
      STORE F002h,#1
Back: RTI
```

Slika 1.74. Prekidna rutina za PER0

```
Int1: LOAD R1,F002h
```

```

INC    R1
STORE F002h,R1
RTI

```

Slika 1.75. Prekidna rutina za PER1

```

Int2: LOAD  R1,F001h
        INC  R1
        STORE F002h,R1
        RTI

```

Slika 1.76. Prekidna rutina za PER2

Napomena: Kod svih instrukcija, prvi operand je uvek i odredišni.

a) Napisati deo programa kojim se periferyama PER0, PER1 i PER2 dodeljuje broj ulaza 2, 3 i 4 u IVT, respektivno.

b) Napisati deo programa kojim se inicijalizuje vektor u IVT za PER0, PER1 i PER2 u skladu sa prethodnom tačkom, ako prekidna rutina za PER0 počinje na adresi A00h, prekidna rutine za PER1 je smeštena neposredno iza prekidne rutine za PER0, a prekidna rutina za PER2 neposredno iza prekidne rutine za PER1.

c) Šta bi trebalo dodati na početku i kraju prekidnih rutina (pre prve naredbe, odnosno ispred RTI) da bi se izvršavanje programa ispravno nastavilo i nakon povratka iz prekidne rutine (to znači da nakon povratka u prekidnu rutinu program nastavlja izvršavanje kao da se prekid nije dogodio)? Obrazložiti!

d) Posmatra se sledeći scenario: U toku izvršavanja naredbe `INC R1` koja se nalazi na adresi 9000h glavnog programa, periferija PER0 generiše prekid. Vrednost registra SP pre prekida je bila FFFFh, a na adresi F002h u glavnoj memoriji se nalazila vrednost 0. U toku izvršavanja naredbe `OUT A0h,#0` prekidne rutine za PER0, stiže drugi prekid (ne zna se od koje periferije) i više se ne dešava ni jedan prekid. Pod pretpostavkom da je zadovoljen uslov iz tačke c), odrediti koja vrednost se nalazi na lokaciji F002h neposredno po povratku u glavni program? Da li je na osnovu toga moguće zaključiti koja periferija je generisala prekid?

e) Prateći scenario iz tačke d), uz pretpostavku da važi uslov iz tačke c), prikazati kompletan sadržaj steka neposredno pre povratka u prekidnu rutinu periferije PER0. Označiti vrh steka.

Rešenje:

a) Traženi program prikazan je na slici 1.77.

```

OUT 80h,#2
OUT 81h,#3
OUT 82h,#4

```

Slika 1.77. Dodela broja ulaza u IVT za PER0, PER1 i PER2

b) Traženi program prikazan je na slici 1.78.

```

STORE 84h,#0h
STORE 86h,#0h
STORE 88h,#0h
STORE 85h,#A00h
STORE 87h,#Axxh ; (xx=3+3+3+1+3+3+1+3+3+4+4+1=3210=20h)
STORE 89h,#Ayyh ; (yy=xx+3+1+3+1=4010=28h)

```

Slika 1.78. Inicijalizacija vektora u IVT za PER0, PER1 i PER2

Adresu prekidne rutine za PER1 računamo tako što na adresu prekidne rutine za PER0 (A00h) dodamo broj reči potrebnih za smeštanje instrukcija prekidne rutine za PER0 (xxh). Instrukcije koje nemaju memorijske operande staju u jednu reč, za smeštanje instrukcija koje imaju memorijske operande biće potrebne 3 reči (jedna reč za smeštanje instrukcijskog koda i dve reči za smeštanje adrese operanda), a za smeštanje instrukcija koje imaju memorijski i neposredni operand biće potrebne 4 reči (jedna reč za smeštanje instrukcijskog koda, dve reči za smeštanje adrese operanda i još jedna reč za smeštanje neposrednog operanda). Ako na adresu prekidne

rutine za PER1 (Axxh) dodamo broj reči potrebnih za smeštanje instrukcija prekidne rutine za PER1 (yyh), dobićemo adresu prekidne rutine za PER2.

c) Sve tri prekidne rutine menjaju sadržaj registra R1. Pošto po povratku iz svake prekidne rutine sadržaj registra R1 mora da se restaurira (kako bi program nastavio kao da prekida nije ni bilo), na početku prekidnih rutina za PER0 i PER1 treba staviti `PUSH R1`, odnosno ispred `RTI` treba staviti `POP R1`.

d) PER0 nije mogla generisati prekid jer je prethodno zaustavljena upisom 0 u Start bit upravljačkog registra te periferije. Prekidna rutina periferije koja generiše prekid počinje da radi tek kad se završi naredba `OUT A0h, #0` iz prekidne rutine PER0. Ako je to bila prekidna rutina PER1: sa adrese F002h čita vrednost 0 koja je bila važeća u glavnom programu, inkrementira tu vrednost i upisuje nazad 1. Ako je prekid generisala periferija PER2: čita 0 koje je upisala prekidna rutina periferije PER0 pre izlaska iz petlje, inkrementira je i upisuje 1 na adresu F002h. Dakle, obe prekidne rutine upisuju istu vrednost 1 na adresu F002h. Na kraju, bez obzira na sve, `STORE F002h, #1` upisuje 1, pa i da su bile različite vrednosti na ovoj adresi pre toga, posle nje više se ne može razlikovati koja od prekidnih rutina ju je izvršila.

e) Izgled steka prikazan je na slici 1.79. Pošto prekinuta naredba `INC R1` glavnog programa nema memorijske operande, ona je dužine jedne reči, tako da sledeća naredba počinje na adresi 9001h, i ta adresa (adresa zauzima dve reči => 0000 9001h) se stavlja na stek. Uz pretpostavku da važi uslov iz tačke c) instrukcija `PUSH R1` sa početka prekidne rutine za PER0, stavlja R1 na stek. Sa zz je predstavljen pomeraj adrese povratka (`STORE F002h, #1`) u prekidnu rutinu periferije PER0 u odnosu na adresu početka prekidne rutine za periferiju PER0, $zz=28_{10}=1Ch$.

memorijska adresa	sadržaj
FFFFFh	PSW (GP)
FFFFEh	9001h
FFFFDh	0000h
FFFFCh	R1 (GP)
FFFFBh	PSW (PER0)
FFFAh	0Azzh
FFF9h	0000h
FFF8h	

←SP

Slika 1.79. Izgled steka neposredno pre povratka u prekidnu rutinu za PER0

1.19 ZADATAK

Adresni prostor nekog računara je veličine 2MB. Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi viša reč. Procesor je jednoadresni, akumulator je 16-bitni, stek raste prema nižim adresama, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IV (Interrupt Vector) počinje od adrese na koju ukazuje registar IVTP čija je vrednost 80h. Procesor poseduje tri ulaza za spoljne prekide različitih prioriteta, IRQ0, IRQ1 i IRQ2, na koje su vezane periferije PER0, PER1 i PER2, respektivno. Ne postoji maskiranje prekida, a prekidi istog prioriteta se odmah prihvataju. Pri obradi prekida se na steku čuvaju PSW i PC (prvo se na stek stavlja PSW, pa zatim PC; registar PSW je 16-bitni). Prekidna rutina za PER0 data je na Slici 1.80., za PER1 na Slici 1.81., a za PER2 na Slici 1.82.. Adrese registara u kojima se čuvaju brojevi ulaza u IVT za PER0, PER1 i PER2 su 80h, 81h i 82h, respektivno. Adresa upravljačkog registra periferije PER0, u kome je bit 0 Start bit, je A0h, a adresa registra podataka PER0 je A2h.

```
Int0: IN  A2h
      AND 8080h
      STORE (F0000h)
```



```

INC F0000h
DEC F0001h
JNZ Back
OUT A0h
INCA
STORE F0002h
Back: RTI

```

Slika 1.80. Prekidna rutina za PER0

```

Int1: LOAD F0002h
      INCA
      STORE F0002h
      RTI

```

Slika 1.81. Prekidna rutina za PER1

```

Int2: POPA
      INCA
      STORE F002h
      PUSHA
      RTI

```

Slika 1.82. Prekidna rutina za PER2

a) Napisati deo programa kojim se periferyama PER0, PER1 i PER2 dodeljuje broj ulaza 2, 3 i 4 u IVT, respektivno.

b) Napisati deo programa kojim se inicijalizuje vektor u IVT za PER0, PER1 i PER2 prema prethodnoj tački, ako labele Int0, Int1 i Int2 označavaju adrese F00h, F20h i F25h respektivno.

c) Šta bi trebalo dodati na početku i kraju prekidnih rutina (pre prve naredbe, odnosno ispred RTI) da bi se izvršavanje programa ispravno nastavilo i nakon povratka?

d) Posmatra se sledeći scenario: U toku izvršavanja naredbe `INCA` na adresi 9000h glavnog programa, periferija PER0 generiše prekid. Vrednost registra SP pre prekida je bila FFFFh, a na adresi F0002h u glavnoj memoriji se nalazila vrednost 0. U toku izvršavanja naredbe `STORE F0002h` prekidne rutine za PER0, stiže drugi prekid (ne zna se od koje periferije) i više se ne dešava ni jedan prekid. Pod pretpostavkom da je zadovoljen uslov iz tačke c), odrediti koja vrednost se nalazi na lokaciji F0002h neposredno po povratku u glavni program? Da li je na osnovu toga moguće zaključiti koja periferija je generisala prekid?

e) Prateći scenario iz tačke d), uz pretpostavku da važi uslov iz tačke c), prikazati kompletan sadržaj steka neposredno pre povratka u prekidnu rutinu periferije PER0. Označiti vrh steka.

Rešenje:

a) Traženi program prikazan je na slici 1.83.

```

LOAD #2
OUT 80h
LOAD #3
OUT 81h
LOAD #4
OUT 82h

```

Slika 1.82. Dodela broja ulaza u IVT za PER0, PER1 i PER2

b) Traženi program prikazan je na slici 1.83.

```

LOAD #0
STORE 84h
STORE 86h
STORE 88h
LOAD #F00h
STORE 85h
LOAD #F20h

```

```

STORE 87h
LOAD #F25h
STORE 89h

```

Slika 1.83. Inicijalizacija vektora u IVT za PER0, PER1 i PER2

c) Sve tri prekidne rutine menjaju sadržaj akumulatora. Pošto po povratku iz svake prekidne rutine sadržaj akumulatora mora da se restaurira (kako bi program nastavio kao da prekida nije ni bilo), na početku prekidnih rutina za PER0 i PER1 treba staviti `PUSHA`, odnosno ispred `RTI` treba staviti `POPA`. Kod prekidne rutine za PER2 na početku treba staviti 2 puta `PUSHA` i na kraju 2 puta `POPA`, jer ima jedno `POPA` na početku i `PUSHA` na kraju.

d) PER0 nije mogla generisati prekid jer je prethodno zaustavljena. Prekidna rutina periferije koja generiše prekid počinje da radi tek kad se završi naredba `STORE F0002h` iz prekidne rutine PER0. Ako je to bila prekidna rutina PER1: sa adrese `F0002h` čita 1, inkrementira tu vrednost i upisuje nazad 2. Ako je prekid generisala periferija PER2: čita 1 koje je prethodno stavljeno na stek (2 puta `PUSHA` – po uslovu iz tačke c), inkrementira ga i upisuje 2 na adresu `F0002h`. Dakle, obe prekidne rutine upisuju istu vrednost na adresu `F0002h`, pa se ne može razlikovati koja periferija je generisala prekid.

e) Izgled steka prikazan je na slici 1.84. Pošto prekinuta naredba `INCA` glavnog programa nema memorijske operande, ona je dužine jedne reči, tako da sledeća naredba počinje na adresi `9001h`, i ta adresa (adresa zauzima dve reči => `0000 9001h`) se stavlja na stek. Uz pretpostavku da važi uslov iz tačke c) instrukcija `PUSHA` sa početka prekidne rutine za PER0, stavlja akumulator na stek. Adresa povratka u prekidnu rutinu za PER0 je `F1Bh` i ta adresa (adresa zauzima dve reči => `0000 0F1Bh`) se stavlja na stek.

memorijska adresa	sadržaj
FFFFFh	PSW (GP)
FFFFEh	9001h
FFFFDh	0000h
FFFFCh	A (GP)
FFFFBh	PSW (PER0)
FFFAh	0F1Bh
FFF9h	0000h
FFF8h	

←SP

Slika 1.84. Izgled steka neposredno pre povratka u prekidnu rutinu za PER0

1.20 ZADATAK

Adresni prostor nekog računara je veličine 8 GB. Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi viša reč. Procesor je jednoadresni, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IV (Interrupt Vector) tabela počinje od adrese 0. Procesor poseduje dva ulaza za spoljne maskirajuće prekide, `IRQ0..1`, pri čemu je `IRQ0` višeg prioriteta, na koje su vezane periferije PER0 i PER1, na nepoznat način. Periferijama se mogu dodeliti proizvoljni ulazi u IVT, u opsegu 0 do 3. U PSW postoji bit I u razredu 0 koji se briše u mikroprogramu za obradu prekida. Ne postoji selektivno maskiranje prekida. Procesor ne prihvata prekid istog nivoa kao što je tekući. Pri obradi prekida se na steku čuvaju PC i 16-bitni PSW tim redom. Stek raste prema nižim adresama. Registri PER0 i PER1 u kojima se čuvaju brojevi ulaza nalaze se redom na adresama 4 i 0. Deo inicijalizacione procedure (glavni program) dat je na slici 1.85., prekidna rutina za PER0 data je na slici 1.86., a za PER1 na slici 1.87.. Posmatra se sledeća sekvenca instrukcija koje su izvršene (date su njihove adrese, heksadecimalno): ..., `10C`, `10E`, `AA00`, `AA01`, `BB00`, `BB02`, `BB04`, `BB06`, `BB08`. Dalje više nema spoljašnjih prekida.

adresa instrukcija

...	...
100h	LOAD #2
102h	OUT 0
104h	LOAD #0
106h	OUT 2
108h	LOAD #1
10Ah	OUT 4
10Ch	LOAD #3
10Eh	OUT 6
110h	INT #3
112h	...

Slika 1.85. Deo inicijalizacione procedure (glavni program)

adresa	instrukcija
AA00h	?
AA01h	INCA
AA02h	RTI

Slika 1.86. Prekidna rutina za periferiju PER0

adresa	instrukcija
BB00h	LOAD #0h
BB02h	STORE 6
BB04h	LOAD #AA02h
BB06	STORE 7
BB08	RTI

Slika 1.87. Prekidna rutina za periferiju PER0

- Na koji način su PER0 i PER1 vezane na IRQ0..1? Kratko obrazložiti kako se zaključak izvodi.
- Napisati sekvencu instrukcija kojima su inicijalizovani vektori u IVT za PER0 i PER1.
- Koja instrukcija se nalazi na adresi AA00h? Kratko obrazložiti kako se zaključak izvodi.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka posle završetka kompletne instrukcije na adresi BB06h scenarija (posle završetka faze obrade prekida te instrukcije). Označiti vrh steka. Za sačuvanu vrednost PSW naznačiti samo vrednost bita I.
- Napisati nastavak sekvence adresa instrukcija koje se izvršavaju, sve dok su one poznate.

Rešenje:

- Na osnovu sekvence instrukcija koje su izvršene vidi se da u toku izvršavanja prekidne rutine za PER0 prelazi se, posle instrukcije `INCA`, na prekidnu rutinu za PER1. Jedini razlog za ovaj prelazak može biti spoljašnji prekid od PER1. Kako je ovaj prekid prihvaćen, PER1 je vezana na ulaz višeg prioriteta, a to je IRQ0, a PER0 je onda vezana na IRQ1.
- Traženi program prikazan je na slici 1.88.

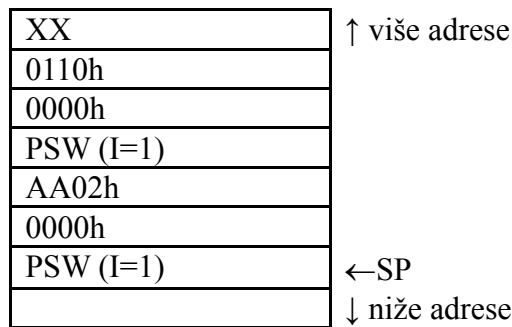
```

LOAD #0h
STORE 2
LOAD #AA00h
STORE 3
LOAD #0h
STORE 4
LOAD #BB00h
STORE 5

```

Slika 1.88. Inicijalizacija vektora u IVT za PER0 i PER1

- Prema obrazloženju iz tačke a), prekid od PER1 je prihvaćen unutar prekidne rutine za PER0, što znači da je prekid bio dozvoljen (bit I je 1). Kako se bit I hardverski briše prilikom prihvatanja prekida, jedini način da on postane ponovo 1 je izvršavanje instrukcije `INTE` unutar prekidne rutine za PER0. Dakle, na adresi AA00h nalazi se instrukcija `INTE`.
- Izgled steka prikazan je na slici 1.89.



Slika 1.89. Izgled steka posle završetka kompletne instrukcije na adresi BB06h

e) Nakon završetka instrukcije na adresi BB08h (RTI), sledi povratak u prekidnu rutinu PER0 i sledeća instrukcija koja će se izvršiti je na adresi AA02h (takodje RTI), nakon koje sledi povratak u glavni program. U glavnom programu se izvršava instrukcija na adresi 110h (INT #3), koja generiše prekid čija se adresa prekidne rutine nalazi u ulazu 3 IVT. Ta vrednost je AA00h i ona je postavljena u prekidnoj rutini periferije PER1. Dakle, nastavak sekvence adresa instrukcija koje se izvršavaju je: AA02h, 110h, AA02h, 112h.

1.21 ZADATAK

Adresni prostor nekog računara je veličine 32 MB. Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi viša reč. Procesor je jednoadresni i reaguje na nivo signala zahteva za prekid. Ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. Ne postoji maskiranje prekida. IV (Interrupt Vector) tabela počinje od adrese na koju ukazuje registar IVTP čija je vrednost 10h. Na procesor su vezane dve periferije, PER1 i PER2, pri čemu je PER1 vezana na ulaz za prekid višeg prioriteta. Adrese registara u kojima se čuvaju brojevi ulaza u IV tabeli ovih periferija su 6h i 7h, respektivno. Na slici 1.90. dat je glavni program, na slici 1.91. prekidna rutina za PER1, a na slici 1.92. za PER2 (sve vrednosti adresa su heksadecimalne). Svaka instrukcija traje pet faza, a svaka faza traje tri takta. Postojanje prekida se ispituje u prvom taktu poslednje faze instrukcije (prekid se obrađuje u toj fazi). Prvim taktom sekvence smatra se prvi takt izvršavanja instrukcije na adresi 200h. Periferije PER1 i PER2 postavljaju i ukidaju svoje signale zahteva za prekid u sledećim taktovima:

PER1: 33, 39, 93, 104 (postavlja u 33., ukida u 39., postavlja u 93., ukida u 104. taktu)
 PER2: 21, 29 (postavlja u 21., ukida u 29. taktu)

adresa	instrukcija
200h	LOAD #2
202h	DEC
203h	JNZ PC(x)
205h	HALT

Slika 1.90. Glavni program

adresa	instrukcija
A000h	RTI

Slika 1.91. Prekidna rutina za periferiju PER1

adresa	instrukcija
B00Bh	ORA #0
B00Dh	RTI

Slika 1.92. Prekidna rutina za periferiju PER2

a) U instrukciji na adresi 203h koristi se PC-relativno adresiranje, a njome se vrši uslovni skok na instrukciju na adresi 202h. Oznaka x predstavlja nepoznatu vrednost 16-bitnog pomeraja u drugom komplementu. Koji je sadržaj lokacije 204h (napisati heksadecimalno)?

b) Napisati deo programa kojim se perifერიји PER1 dodeljuje broj ulaza 4, a PER2 broj ulaza 6 u IVT.

c) Napisati deo programa kojim se inicijalizuje vektor u IVT za PER1 prema prethodnoj tački. Sve instrukcije operišu nad 16-bitnim operandima.

d) Napisati sekvencu adresa koje se izvršavaju po datom scenariju, počev od instrukcije na adresi 200h, sve dok su one poznate.

e) Isto kao u prethodnoj tački, samo ako procesor sada reaguje na impuls (tačnije na uzlaznu ivicu) umesto na nivo signala zahteva za prekid, a sve ostalo je isto.

Rešenje:

a) Pošto se u registru PC nalazi 205h, a vrši se uslovni skok na adresu 202h, nepoznata 16-bitna vrednost x predstavljena u drugom komplementu je FFFDh ili $(-3)_{10}$. Instrukcija `JNZ PC(x)` zauzima dve reči. U prvoj reči na adresi 203h nalazi se kod operacije, a u drugoj na adresi 204h se nalazi 16-bitni pomeraj x , odnosno FFFDh.

b) Traženi program prikazan je na slici 1.93.

```
LOAD #4
OUT 6h
LOAD #6
OUT 7h
```

Slika 1.93. Dodela broja ulaza u IVT za PER1 i PER2

c) Traženi program prikazan je na slici 1.94.

```
LOAD #A000h
STORE 19h
LOAD #0
STORE 18h
```

Slika 1.94. Inicijalizacija vektora u IVT za PER1

d) Periferija PER2 postavlja signal zahteva za prekid u 21. taktu, odnosno u toku 2. instrukcije (2. faza, 3.takt) i ukida ga u 29. taktu, odnosno u toku 2. instrukcije (5. faza, 2. takt). Postojanje prekida će se ispitati u 5. fazi i 1. taktu instrukcije. Pošto je u tom taktu 2. instrukcije signal zahteva za prekid od periferije PER2 aktivan prekid će biti prihvaćen. Dakle biće izvršene dve instrukcije glavnog programa na adresama 200h, 202h i sledeća instrukcija koja će se izvršiti biće prva instrukcija prekidne rutine za periferiju PER2, na adresi B00Bh. U 33. taktu, odnosno u toku 3. instrukcije (1.faza, 3. takt) periferija PER1 postavlja svoj signal zahteva za prekid, ali ga ukida u 39. taktu, odnosno u toku 3. instrukcije (3.faza, 3.takt) pre nego što se ispita postojanje prekida u 3. instrukciji. Do 7. instrukcije, više nema zahteva za prekidom, tako da se izvršava sledeća instrukcije prekidne rutine periferije PER2, na adresi B00Dh i pošto je to `RTI` instrukcija izvršava se povratak u glavni program. Sledeća instrukcija koja se izvršava je instrukcija glavnog programa na adresi 203h, i vrši se skok na instrukciju na adresi 202h. U toku 7. instrukcije na adresi 203h u 1. fazi, 3. taktu (93. takt) periferija PER1 postavlja svoj signal zahteva za prekid i ukida ga u toku iste instrukcije u 5. fazi, 2. taktu (104. takt). S obzirom da se u predhodnom taktu ispitivalo postojanje prekida, zahtev za prekid od periferije PER1 će se prihvatiti. Sekvenca adresa koje se izvršavaju po datom scenariju predstavljena je na slici 1.95.

200h, 202h, B00Bh, B00Dh, 203h, 202h, 203h, A000h, 205h

Slika 1.95. Sekvenca adresa koje se izvršavaju po datom scenariju

f) U slučaju kada procesor reaguje na impuls umesto na nivo signala zahteva za prekid zahtevi za prekid od pojedinih periferija se pamte i njihovo postojanje se ispituje u prvom taktu poslednje faze instrukcije. To znači da će se za razliku od predhodne tačke sada prihvatiti i prvi zahtev za prekid od periferije PER1. Sekvenca adresa u ovom slučaju je predstavljena na slici 1.96.

200h, 202h, B00Bh, A000h, B00Dh, 203h, 202h, A000h, 203h, 205h

Slika 1.96. Sekvenca adresa koje se izvršavaju u slučaju kada procesor reaguje na impuls

1.22 ZADATAK

Adresni prostor procesora je veličine 128KB, adresibilna jedinica je 16-bitna reč, a višerečni brojevi se smeštaju tako da je na višoj adresi niža reč. Procesor je jednoadresni sa razdvojenim memorijskim i ulazno/izlaznim adresnim prostorima, a mehanizam prekida je vektorisan. IV (Interrupt Vector) tabela počinje od adrese na koju ukazuje registar IVTP (Interrupt Vector Table Pointer), a registar IVTP ima vrednost 2. Procesor ima tri ulazne linije IRQ0, IRQ1 i IRQ2 za spoljne maskirajuće prekide, pri čemu je IRQ0 najvišeg prioriteta, a IRQ2 najnižeg prioriteta, na koje su vezane periferije PER0, PER1 i PER2, respektivno. Njima su pridruženi ulazi 2, 3 i 4 u IV tabelu, respektivno. Ne prihvata se prekid istog nivoa prioriteta. Adrese 16-bitnih registara u kojima se čuvaju brojevi ulaza su 10h, 20h i 30h, respektivno. U PSW-u postoji bit I (Interrupt Enable) koji se briše u mikroprogramu za obradu prekida, kao i određen broj L bita. Pri prekidu se na steku čuvaju PSW i PC tim redom. Stek raste prema nižim lokacijama. Akumulator je 16-bitni. Instrukcije INTE, INTD, RTI i INT ne reaguju na prekide. Instrukcija INT ne menja nivo prioriteta tekućeg programa. Dat je deo glavnog programa na slici 1.97., prekidne rutine na slici 1.98., izgled dela memorije počev od adrese 0 dat je na slici 1.99.. Instrukcija na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 2. instrukcije stiže zahtev za prekid po liniji IRQ2, a u toku 5. po liniji IRQ0. Na početku su svi biti PSW-a postavljeni na 0. Ne postoji IMR registar.

adresa	instrukcija
0100h	LOAD #2h
0102h	INTE
0103h	INCA
0104h	ADD #2h
0106h	INT #3h
0108h	INTD

Slika 1.97. Izgled dela glavnog programa

adresa	instrukcija
1000h	PUSHA
1001h	INTE
1002h	LOAD 1h
1004h	INCA
1005h	STORE 1h
1007h	POPA
1008h	RTI
1009h	INTE
100Ah	INCA
100Bh	RTI
100Ch	INCA
100Dh	RTI

Slika 1.98. Prekidne rutine

adresa	sadržaj
0000h	0000h
0001h	0001h
0002h	1000h
0003h	1008h
0004h	1000h
0005h	100Ch
0006h	1009h

Slika 1.99. Izgled dela memorije

- a) Na kojim adresama započinju prekidne rutine za linije IRQ0, IRQ1 i IRQ2, respektivno?
- b) Napisati deo programa kojim se inicijalizuje ulaz 3 u vektor tabeli.
- c) Napisati deo programa kojim se dodeljuju brojevi ulaza navedenim periferijama.
- d) Dati sekvencu adresa instrukcija koje se redom izvršavaju po datom scenariju.
- e) Prikazati sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 6. instrukcije. Za sačuvanu vrednost PSW dati samo vrednosti bita I i L. Naznačiti u kome smeru raste stek.

Rešenje:

a) Adresni prostor procesora je veličine 128KB, odnosno 2^{16} reči, pošto je adresibilna jedinica 16-bitna reč. Odatle sledi da su adrese 16-bitne i da svaki ulaz IV tabele zauzima jednu lokaciju u memoriji. IV tabela počinje od adrese na koju ukazuje registar IVTP, a vrednost registara IVTP je 2. Na linije IRQ0, IRQ1, IRQ2 su redom vezane periferije PER0, PER1, PER2 i njima su pridruženi ulazi 2, 3 i 4 u IVT, respektivno. To znači da će se adrese na kojima započinju prekidne rutine za linije IRQ0, IRQ1 i IRQ2 nalaziti na memorijskim lokacijama 4h (IVTP+2), 5h (IVTP+3) i 6h (IVTP+4) respektivno. Odnosno prekidna rutina za liniju IRQ0 započinje na adresi 1000h, za liniju IRQ1 na adresi 100Ch i za IRQ2 na adresi 1009h.

b) Traženi program prikazan je na slici 1.100.

```
LOAD #100Ch
STORE 5h
```

Slika 1.100. Deo programa kojim se inicijalizuje ulaz 3 u vektor tabeli

c) Traženi program prikazan je na slici 1.101.

```
LOAD #2h
OUT 10h
LOAD #3H
OUT 20h
LOAD #4h
OUT 30h
```

Slika 1.101. Deo programa kojim se dodeljuju brojevi ulaza periferijama PER0, PER1 i PER2

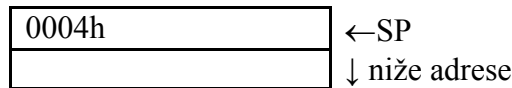
d) U toku izvršavanja 2. instrukcije stiže zahtev za prekid po liniji IRQ2. Pošto je 2. instrukcija instrukcija INTE koja ne reaguje na prekid, prekid će biti prihvaćen tek posle 3. instrukcije. U toku izvršavanja 5. instrukcije stiže zahtev za prekid po liniji IRQ0 i prihvata se. Nakon završetka obe prekidne rutine i povratka u glavni program 15. instrukcija po redu izvršavanja biće instrukcija na adresi 106h, odnosno instrukcija INT #3h, koja generiše prekid čija se adresa prekidne rutine nalazi u ulazu 3 IV tabele (na adresi 5h). Tražena sekvencu adresa instrukcija predstavljena je na slici 1.102.

100h, 102h, 103h, 1009h, 100Ah, 1000h, 1001h, 1002h, 1004h, 1005h, 1007h, 1008h, 100Bh, 104h, 106h, 100Ch, 100Dh, 108h, 109h

Slika 1.102. Sekvenca adresa instrukcija koje se redom izvršavaju po datom scenariju

e) Nakon izvršavanja 6. instrukcije na steku se nalazi PSW i PC glavnog programa kao i PSW i PC prekidne rutine za PER2. Na vrhu steka se nalazi vrednost akumulatora jer je 6. instrukcija po redosledu izvršavanja instrukcija PUSHA. Izgled steka nakon izvršavanja 6. instrukcije prikazan je na slici 1.103. Stek raste ka nižim lokacijama.

XX	↑ više adrese
PSW (I=1, L=00)	
0104h	
PSW (I=1, L=01)	
100Bh	



Slika 1.103. Izgled steka nakon izvršavanja 6. instrukcije

1.23 ZADATAK

Adresni prostor procesora je veličine 8GB, adresibilna jedinica je 32-bitna reč, a 64-bitni brojevi se smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni, a mehanizam prekida je vektorisan. Interapt vektor (IV) tabela ima 8 fiksnih ulaza i počinje od adrese 4h. Procesor ima dve ulazne liniju IRQM0 i IRQM1 za spoljne maskirajuće prekide, pri čemu je IRQM0 višeg prioriteta, i jednu ulaznu liniju IRQN za spoljne nemaskirajuće prekide. Njima su pridruženi ulazi 3, 4 i 2 u IV tabelu, respektivno. Ulaz 5 se koristi u svim ostalim slučajevima. Nemaskirajući prekidi su višeg prioriteta od maskirajućih. Prekidne rutine započinju na sledećim adresama: 1000h, 1002h, 1004h i 1006h, respektivno. U PSW-u postoje biti I (Interrupt Enable) i T (Trap) koji se brišu u mikroprogramu za obradu prekida, kao i određen broj L bita. Pri prekidu se na steku čuvaju PSW i PC tim redom. Stek raste prema nižim lokacijama. Akumulator je 32-bitni. Instrukcije INTE, INTD, TRPE i TRPD ne reaguju na prekide. Ne prihvata se prekid istog nivoa prioriteta. Ne postoji registar maske IMR. Dat je deo glavnog programa na slici 1.104 i prekidne rutine na slici 1.105. Instrukcija na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 2. instrukcije stiže zahtev za prekid po liniji IRQM1, u toku 5. po liniji IRQM0, a u toku 7. po liniji IRQN. Na početku su svi biti PSW-a postavljeni na 0.

adresa	instrukcija
0100h	LOAD #0h
0101h	STORE 2h
0102h	TRPE
0103h	INTE
0104h	ADD #2h
0105h	INTD
0106h	STORE #3h
0107h	TRPD
0108h	INCA

Slika 1.104. Deo glavnog programa

adresa	instrukcija
1000h	INCA
1001h	RTI
1002h	POPA
1003h	PUSHA
1004h	DECA
1005h	RTI
1006h	PUSHA
1007h	LOAD 2h
1008h	INCA
1009h	STORE 2h
100Ah	POPA
100Bh	RTI

Slika 1.105. Prekidne rutine

- Nacrtati izgled svih 8 ulaza u vektor tabeli, označiti adrese relevantnih lokacija i upisati sadržaje u njih.
- Napisati deo programa kojim se inicijalizuje ulaz 4 u vektor tabeli.
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od adrese 0100h.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 8. instrukcije. Za sačuvanu vrednost PSW dati samo vrednosti bita I, T i L. Naznačiti u kome smeru raste stek.
- Koja će se vrednost nalaziti na lokaciji 2h nakon izvršenja sekvence pod c)?

Rešenje:

a) Izgled svih 8 ulaza u vektor tabeli prikazan je na slici 1.106.

broj ulaza u IV tabelu	memorijska adresa	sadržaj	opis sadržaja
7	000Bh		
6	000Ah		
5	0009h	1006h	adresa prekidne rutine za ostale slučajeve
4	0008h	1002h	adresa prekidne rutine za IRQM1
3	0007h	1000h	adresa prekidne rutine za IRQM0
2	0006h	1004h	adresa prekidne rutine za IRQN
1	0005h		
0	0004h		

Slika 1.106. Izgled svih 8 ulaza u vektor tabeli

b) Traženi program prikazan je na slici 1.107.

```
LOAD #1002h
STORE 8h
```

Slika 1.107. Deo programa kojim se inicijalizuje ulaz 4 u vektor tabeli

c) Tražena sekvenca naredbi prikazana je na slici 1.108.

adrese	objašnjenje
100h, 101h, 102h, 103h, 104h, TRPE, INTE	Instrukcije glavnog programa LOAD, STORE,
1000h, 1001h, 1004h, 1005h, 1002h, 1003h, 1004h, 1005h, 1006h, 1007h, 1008h, 1009h, 100Ah, 100Bh, 105h, 106h, 1006h, 1007h, 1008h, 1009h, 100Ah, 100Bh, 107h, 108h, 109h	Prekidna rutina za IRQM0 Prekidna rutina za IRQN Prekidna rutina za IRQM1 Trap prekidna rutina Instrukcije glavnog programa INTD i STORE Trap prekidna rutina Instrukcije glavnog programa

Slika 1.108. Sekvenca naredbi koje se redom izvršavaju počev od adrese 0100h

d) Nakon izvršavanja 8. instrukcije na steku se nalazi PSW i PC glavnog programa. Izgled steka prikazan je na slici 1.109.

XX	↑ više adrese
PSW (I=1, L=00)	
0105h	← SP
	↓ niže adrese

Slika 1.109. Sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 8. instrukcije

e) Glavni program (instrukcije na adresi 100h i 101h) na lokaciju 2h upisuje 0. U trap prekidnoj rutini (instrukcije na adresi 1007h, 1008h i 1009h) vrednost sa lokacije 2h se inkrementira. Trap prekidna rutina se poziva dva puta u toku izvršavanja sekvence pod c). Odatle sledi da je MEM[2]=2.

1.24 ZADATAK

Adresni prostor procesora je veličine 16GB, adresibilna jedinica je 32-bitna reč, a 64-bitni brojevi se smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni, a mehanizam prekida je vektorisan. Interapt vektor (IV) tabela ima 4 fiksna ulaza i počinje od adrese 2h. Procesor ima jednu ulaznu liniju IRQM za spoljne maskirajuće prekide i jednu ulaznu liniju IRQN za spoljne nemaskirajuće prekide. Njima su pridruženi ulazi 0 i 1 u IV tabelu, respektivno. Ulaz 2 u IV tabeli se upotrebljava u slučaju prekida posle svake instrukcije (TRAP), a ulaz 3 u svim ostalim slučajevima. Nemaskirajući i interni prekidi su višeg prioriteta od maskirajućih. PSW postoji bit I (Interrupt Enable) koji se briše u mikroprogramu za obradu prekida. Pri prekidu se na steku čuvaju PSW i PC tim redom. Stek raste prema nižim lokacijama. Akumulator je 32-bitni. Instrukcije INTE, INTD, RTI i TRPE ne reaguju na prekide. Dat je deo glavnog programa na slici 1.110, prekidne rutine na slici 1.111, izgled dela memorije počev od adrese 0 dat je na slici 1.112. Instrukcija INTE na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 2. instrukcije stiže zahtev za prekid po liniji IRQN, a u toku 5. po liniji IRQM. Na početku su svi biti PSW-a postavljeni na 0.

adresa	instrukcija
0100	INTE
0101	LOAD #1h
0102	INCA
0103	STORE #1h
0104	DECA
0105	INTD

Slika 1.110. Deo glavnog programa

adresa	instrukcija
1000h	INCA
1001h	PUSHA
1002h	POPA
1003h	RTI
1004h	LOAD 1h
1005h	OR #FFh
1006h	STORE 1h
1007h	RTI
1008h	POPA
1009h	INCA
100Ah	PUSHA
100Bh	RTI

Slika 1.111. Prekidne rutine

adresa	sadržaj
0000h	1001h
0001h	100Ah
0002h	1000h
0003h	1004h
0004h	1006h
0005h	1008h

Slika 1.112. Izgled dela memorije počev od adrese 0

- Na kojim adresama započinju prekidne rutine za linije IRQM i IRQN, respektivno?
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od adrese 0100h.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 7. instrukcije. Za sačuvanu vrednost PSW dati samo vrednost bita I. Naznačiti u kome smeru raste stek.
- Koja će se vrednost nalaziti na lokaciji 1h nakon izvršenja sekvence pod c)?

Rešenje:

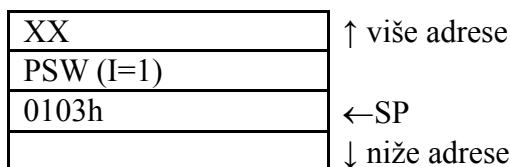
a) Adresni prostor procesora je veličine 128KB, odnosno 2^{16} reči, pošto je adresibilna jedinica 16-bitna reč. Odatle sledi da su adrese 16-bitne i da svaki ulaz IV tabele zauzima jednu lokaciju u memoriji. IV tabela počinje od adrese na koju ukazuje registar IVTP, a vrednost registara IVTP je 2. Procesor ima jednu ulaznu liniju IRQM za spoljne maskirajuće prekide i jednu ulaznu liniju IRQN za spoljne nemaskirajuće prekide. Njima su pridruženi ulazi 0 i 1 u IV tabelu, respektivno. To znači da će se adrese na kojima započinju prekidne rutine za linije IRQM i IRQN nalaziti na memorijskim lokacijama 2h (IVTP+0) i 3h (IVTP+1) respektivno. Odnosno prekidna rutina za liniju IRQM započinje na adresi 1000h, a za liniju IRQN na adresi 1004h.

b) U toku izvršavanja 2. instrukcije stiže zahtev za prekid po liniji IRQN i zahtev će biti prihvaćen. Zahtev za prekid koji stiže po liniji IRQM u toku izvršavanja 5. instrukcije neće biti prihvaćen jer je bit I=0. Nakon povratka u glavni program iz prekidne rutine za IRQN i izvršenja instrukcije na adresi 0102h zahtev za prekid koji je stigao po liniji IRQM biće prihvaćen. Nakon povratka u glavni program, instrukcija na adresi 0103h (STORE #1h) je nekorektna (neposredno adresiranje kao odredište), generiše se prekid čija adresa prekidne rutine je smeštena u ulazu 3 IV tabele (svi ostali slučajevi), odnosno instrukcija na adresi 1008h se izvršava sledeća. Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h prikazana je na slici 1.113.

adrese	objašnjenje
0100h, 0101h,	Instrukcije glavnog programa INTE, LOAD
1004h, 1005h, 1006h, 1007h,	Prekidna rutina za IRQN
0102h,	Instrukcija glavnog programa INCA
1000h, 1001h, 1002h, 1003h,	Prekidna rutina za IRQM
0103h,	Instrukcija glavnog programa STORE
1008h, 1009h, 100Ah, 100Bh,	Prekidna rutina za prekid nastao usled nekorektnosti prilikom izvršavanja instrukcije na adresi 0103h
0104h, 0105h	Instrukcije glavnog programa INTD, DECA

Slika 1.113. Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h

c) Nakon izvršavanja 7. instrukcije na steku se nalazi PSW i PC glavnog programa. Izgled steka prikazan je na slici 1.114.



Slika 1.114. Sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 8. instrukcije

d) Nakon izvršenja instrukcija na adresama 1004h, 1005h i 1006h u prekidnoj rutini za IRQN, vrednost na adresi 1h se menja. Na kraju sekvence pod c) MEM[1]=10FFh.

1.25 ZADATAK

Adresni prostor nekog računara je veličine 128KB (kilo bajta). Adresibilna jedinica je 16-bitna reč, a višerečni brojevi se u memoriju smeštaju tako da je na nižoj adresi viša reč. Procesor je jednoadresni, akumulator je 16-bitni, stek raste prema nižim adresama, ulazno-izlazni i memorijski adresni prostori su razdvojeni, a mehanizam prekida je vektorisan. IV (Interrupt Vector) počinje od adrese na koju ukazuje registar IVTP, čija je vrednost 02h. Instrukcije RTI, INTE i INT ne reaguju na prekid. U PSW postoji bit I (Interrupt Enable) koji se briše u mikroprogramu za obradu prekida, kao potreban broj L bita. Procesor poseduje tri ulaza za spoljne maskirajuće prekide različitih prioriteta, IRQ0, IRQ1 i IRQ2, pri čemu je IRQ0 najvišeg prioriteta pri ugneždivanju, kao i liniju IRQ za spoljnje nemaskirajuće prekide. Na liniju IRQ0 je vezana periferija PER0, a na liniju IRQ periferija PER. Ne postoji selektivno maskiranje prekida, a prekidi istog prioriteta se odmah prihvataju. Pri obradi prekida se na steku čuvaju PSW i PC

(prvo se na stek stavlja PSW, pa zatim PC; registar PSW je 16-bitni). Adresa 16-bitnog registara u kojem se čuvaju broj ulaza u IVT za PER0 je 80h. Liniji IRQ odgovara fiksni ulaz u IV tabelu 0. Dat je deo glavnog programa na slici 1.115, prekidna rutina za PER0 data je na Slici 1.116, a za PER na Slici 1.117, izgled dela memorije počev od adrese 0 dat je na Slici 1.118. Instrukcija INTE na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 3. instrukcije stižu zahtevi za prekid od PER0 i od PER. Na početku su svi biti PSW-a postavljeni na 0.

adresa	instrukcija
0100	INTE
0102	LOAD #2h
0104	INCA
0105	DECA
0107	INCA

Slika 1.115. Deo glavnog programa

adresa	instrukcija
X	INTE
X + 1	INT #5
X + 2	RTI

Slika 1.116. Prekidna rutina za PER0

adresa	instrukcija
Y	PUSHA
Y + 1	INCA
Y + 2	POPA
Y + 3	RTI

Slika 1.117. Prekidna rutina za PER

adresa	sadržaj
0000	01h
0001	A1h
0002	BBh
0003	00h
0004	CDh
0005	00h
0006	00h
0007	CFh

Slika 1.118. Izgled dela memorije počev od adrese 0

- Napisati deo programa kojim se periferiji PER0 dodeljuje broj ulaza 2 u IVT.
- Na kojim adresama započinju prekidne rutine za periferije PER0 i PER.
- Koji opseg adresa zauzima IV (Interrupt Vector) tabela?
- Dati sekvencu adresa instrukcija koje se izvršavaju po datom scenariju, sve dok su one poznate.
- Prateći scenario iz tačke d) prikazati kompletan sadržaj steka neposredno nakon završetka 10-te instrukcije. Za sačuvanu vrednost PSW dati samo vrednost bita I.

Rešenje:

- Traženi program prikazan je na slici 1.119.

```
LOAD #2
OUT 80h
```

Slika 1.119. Deo programa kojim se periferiji PER0 dodeljuje broj ulaza 2 u IVT

b) Adresni prostor procesora je veličine 128KB, odnosno 2^{16} reči, pošto je adresibilna jedinica 16-bitna reč. Odatle sledi da su adrese 16-bitne i da svaki ulaz IV tabele zauzima jednu lokaciju u memoriji. IV tabela počinje od adrese na koju ukazuje registar IVTP, a vrednost registara IVTP je 2. Periferiji PER0 je pod a) dodeljen broj ulaza 2 u IVT, a periferija PER je vezana na liniju IRQ kojoj odgovara fiksni ulaz 0 u IVT. To znači da će se adrese na kojima započinju prekidne rutine za periferije PER0 i PER nalaziti na memorijskim lokacijama 4h (IVTP+2) i 2h (IVTP+0) respektivno. Odnosno prekidna rutina za periferiju PER0 počinje na adresi 00CDh, a za periferiju PER na adresi 00BBh.

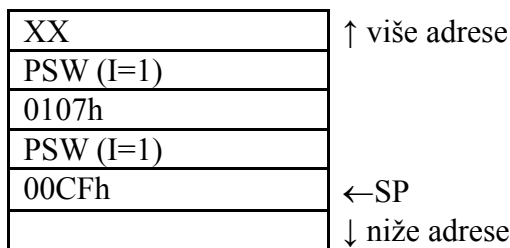
c) Registar u kome se čuva broj ulaza u IVT za PER0 je 16-bitni, to znači da IVT zauzima 2^{16} adresa, opseg 0000h-FFFFh (0002h-0001h).

d) Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h prikazana je na slici 1.120.

adrese	objašnjenje
0100h, 0101h, 0104h	Instrukcije glavnog programa INTE, LOAD, INCA
00BBh, 00BCh, 00BDh, 00BEh,	Prekidna rutina za PER
0105h,	Instrukcija glavnog programa DECA
00CDh, 00CEh,	Instrukcije prekidne rutine za PER0 INTE, INT
00CFh,	Prekidna rutina za prekid nastao kao rezultat izvršavanja instrukcije INT
00CFh	Instrukcija prekidne rutine za PER0 RTI
0107h	Instrukcija glavnog programa INCA

Slika 1.120. Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h

e) Neposredno nakon izvršavanja 10-te instrukcije na steku se nalaze PSW i PC glavnog programa, kao i PSW i PC prekidne rutine za PER0. Izgled steka prikazan je na slici 1.121.



Slika 1.121. Izgled steka neposredno nakon izvršavanja 10-te instrukcije

1.26 ZADATAK

Adresni prostor procesora je veličine 128KB, adresibilna jedinica je 16-bitna reč, a 32-bitni brojevi se smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni, a mehanizam prekida je vektorisan. Interrapt vektor tabela ima 4 ulaza i počinje od adrese na koju ukazuje registar IVTP (Interrupt Vector Table Pointer), a registar IVTP ima vrednost 4. Procesor ima dve ulazne linije IRQ0 i IRQ1 za spoljne nemaskirajuće prekide pri čemu je ulaz IRQ0 višeg prioriteta na koje su nakačene periferije PER0 i PER1 respektivno. Bit L je u razredu 0 8-bitnog PSW. U PSW postoji bit I (Interrupt Enable) koji se briše u mikroprogramu za obradu prekida. Pri prekidu se na steku čuvaju PSW i PC tim redom. Stek raste prema nižim lokacijama. Akumulator je 16-bitni. Instrukcija INTE ne reaguju na prekide. Dat je deo glavnog programa na slici 1.122, prekidna rutina za IRQ0 na slici 1.123, prekidna rutina za IRQ1 slici 1.124, izgled dela memorije počev od adrese 0 dat je na slici 1.125. Instrukcija na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 3. instrukcije stižu zahtevi za prekid od PER0 i od PER1. Na početku su svi biti PSW-a postavljeni na 0.

adresa	instrukcija
0100	LOAD #03h
0102	ADD 3h
0104	DECA
0105	STORE 3h
0107	INCA

Slika 1.122. Deo glavnog programa

adresa	instrukcija
X	INTE
X + 1	INCA
X + 2	RTI

Slika 1.123. Prekidna rutina za IRQ0

adresa	instrukcija
Y	INTE
Y + 1	POPA
Y + 2	PUSHA
Y + 3	RTI

Slika 1.124. Prekidna rutina za IRQ1

adresa	sadržaj
0000	0001h
0001	AB00h
0002	B000h
0003	00h
0004	CDh
0005	A0h
0006	00h
0007	00h
0008	AAh

Slika 1.125. Izgled dela memorije počev od adrese 0

- Po kojim linijama periferija šalje procesoru broj ulaza u IV tabelu?
- Na kojim adresama započinju prekidne rutine za periferije PER0 i PER1 ukoliko im odgovaraju brojevi ulaza u IVT 3 i 1, respektivno?
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od adrese 0100h.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 5. instrukcije. Za sačuvanu vrednost PSW dati samo vrednost bita L. Naznačiti u kome smeru raste stek.
- Koja će se vrednost nalaziti na lokaciju 3h nakon izvršenja sekvence pod c)?

Rešenje:

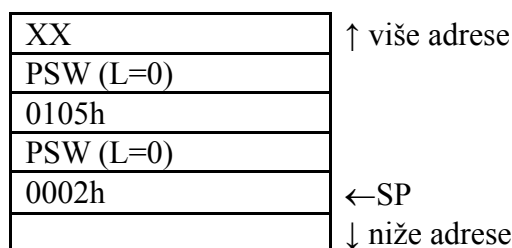
- IVT ima 4 ulaza, broj ulaza se šalje po najniže dve linije podataka, D[1..0].
- Adresni prostor procesora je veličine 128KB, odnosno 2^{16} reči, pošto je adresibilna jedinica 16-bitna reč. Odatle sledi da su adrese 16-bitne i da svaki ulaz IV tabele zauzima jednu lokaciju u memoriji. IV tabela počinje od adrese na koju ukazuje registar IVTP, a vrednost registara IVTP je 4. To znači da će se adrese na kojima započinju prekidne rutine za periferije PER0 i PER1 nalaziti na memorijskim lokacijama 7h (IVTP+3) i 5h (IVTP+1) respektivno. Odnosno prekidna rutina za periferiju PER0 počinje na adresi 0000h, a za periferiju PER1 na adresi 00A0h.
- Tražena sekvencu prikazana je na slici 1.126.

adrese
0100h, 0102h, 0104h,
0000h, 0001h,
00A0h, 00A1h, 00A2h, 00A3h,
0002h,
0105h, 0107h

objašnjenje
Instrukcije glavnog programa LOAD, ADD, INCA
Instrukcije prekidne rutine za PER0 INTE, INCA
Prekidna rutina za PER1
Instrukcija prekidne rutine za PER0 RTI
Instrukcije glavnog programa STORE, INCA

Slika 1.126. Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h

d) Neposredno nakon izvršavanja 5-te instrukcije na steku se nalaze PSW i PC glavnog programa, kao i PSW i PC prekidne rutine za PER0. Izgled steka prikazan je na slici 1.127.



Slika 1.127. Izgled steka neposredno nakon izvršavanja 5-te instrukcije

e) Instrukcija `STORE` na adresi 0105h glavnog programa smešta sadržaj akumilatora(2h) na adresu 3h. Nakon izvršenja sekvence pod c) `MEM[3]=2h`

1.27 ZADATAK

Adresni prostor procesora je veličine 128KB, adresibilna jedinica je 16-bitna reč, a 32-bitni brojevi se smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni, a mehanizam prekida je vektorisan. Interapt vektor (IV) tabela ima 8 fiksna ulaza i počinje od adrese 2h. Procesor ima dve ulazne linije IRQ0 i IRQ1 za spoljne maskirajuće prekide, pri čemu je IRQ0 najvišeg prioriteta, na koje su vezane periferije PER0 i PER1, respektivno. Njima su pridruženi ulazi 0 i 1 u IV tabelu, respektivno. Ulaz 2 u IV tabeli se upotrebljava u slučaju prekida posle svake instrukcije (TRAP), a ulaz 3 u svim ostalim slučajevima. Nemaskirajući i interni prekidi su višeg prioriteta od maskirajućih. U PSW-u postoje biti I (Interrupt Enable) i T (Trap) koji se brišu u mikroprogramu za obradu prekida, kao i određen broj L bita. Pri prekidu se na steku čuvaju PSW i PC tim redom. Stek raste prema nižim lokacijama. Akumulator je 16-bitni. Dat je deo glavnog programa na slici 1.128, prekidne rutine na slici 1.129, izgled dela memorije počev od adrese 0 dat je na slici 1.130. Instrukcija na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 4. instrukcije stiže zahtev za prekid po linijama IRQ0 i IRQ1. Na početku su svi biti PSW-a postavljeni na 0. Ne postoji IMR registar.

adresa	instrukcija
0100	INTE
0101	LOAD #2h
0103	DIV 1h
0105	STORE 1h
0107	DECA
0108	INTD

Slika 1.128. Deo glavnog programa

adresa	instrukcija
1000h	INTA
1001h	POPA
1002h	ADD #2
1003h	PUSHA
1004h	RTI
1005h	PUSHA
1006h	INCA
1007h	POPA
1008h	RTI
1009h	INTE
100Ah	INCA
100Bh	RTI

Slika 1.129. Prekidne rutine

adresa	sadržaj
0000	0001h
0001	0000h
0002	1005h
0003	1008h
0004	100Ah
0005	1000h

Slika 1.130. Izgled dela memorije počev od adrese 0

- Na kojim adresama započinju prekidne rutine za linije IRQ0 i IRQ1, respektivno?
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od adrese 0100h.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 7. instrukcije. Za sačuvanu vrednost PSW dati samo vrednost bita I i L. Naznačiti u kome smeru raste stek
- Koja će se vrednost nalaziti na lokaciji 1h nakon izvršenja sekvence pod c)?

Rešenje:

1.28 ZADATAK

Adresni prostor procesora je veličine 8GB, od kojih je najviših 1GB rezervisani za ROM memoriju, adresibilna jedinica je 16-bitna reč, a višerečni brojevi se smeštaju tako da je na nižoj adresi niža reč. Procesor je jednoadresni sa razdvojenim memorijskim i ulazno/izlaznim adresnim prostorima, a mehanizam prekida je vektorisan. IV (Interrupt Vector) tabela počinje od adrese na koju ukazuje registar IVTP (Interrupt Vector Table Pointer), a registar IVTP ima vrednost 4. Procesor ima tri ulazne linije IRQ0, IRQ1 i IRQ2 za spoljne maskirajuće prekide, pri čemu je IRQ0 najvišeg prioriteta, a IRQ2 najnižeg prioriteta, na koje su vezane periferije PER0, PER1 i PER2, respektivno, kojima treba dodeliti ulaze 4, 5 i 7 u vektor tabeli, i kojima odgovaraju prekidne rutine na adresama 1007h, 1010h i 1005h, respektivno. Adrese 8-bitnih registara u kojima se čuvaju brojevi ulaza su 0h, 4h i 8h, respektivno. Ne prihvata se prekid istog nivoa prioriteta. Ulaz 0 u IV tabelu se koristi u svim ostalim slučajevima. U PSW-u postoji bit I (Interrupt Enable) koji se briše u mikroprogramu za obradu prekida, kao i određen broj L bita. Pri prekidu se na steku čuvaju ACC, PSW i PC tim redom. Stek raste prema nižim lokacijama. Akumulator je 16-bitni. Dat je deo glavnog programa na slici 1.133, prekidne rutine na slici 1.134, izgled dela memorije počev od adrese 0 dat je na slici 1.135. Instrukcija na adresi 0100h označena je kao 1. (prva) po redosledu izvršavanja, a svaka sledeća instrukcija koja se izvršava označena je sledećim rednim brojem. U toku izvršavanja 5. instrukcije stiže zahtev za prekid po liniji IRQ2, u toku 6. po liniji IRQ1, a u toku 7. po liniji IRQ0. Na početku su svi biti PSW-a postavljeni na 0. U registru maske IMR bit i odgovara liniji IRQi; vrednost ovog registra je inicijalno 5h. Instrukcija RTI reaguje na prekid.

adresa	instrukcija
0100h	INTE
0101h	LOAD 1h
0104h	DECA
0105h	STORE 1h
0108h	INCA
0109h	INCA
010Ah	INTD

Slika 1.133. Deo glavnog programa

adresa	instrukcija
1000h	POPA
1001h	ADD #3h
1003h	PUSHA
1004h	RTI
1005h	INCA
1006h	RTI


```

1007h   PUSHA
1008h   LOAD #2h
100Ah   STORE (1h)
100Dh   POPA
100Eh   RTI
100Fh   RTI
1010h   RTI
1011h   RTI

```

Slika 1.134. Prekidne rutine

adresa	sadržaj
0000h	0001h
0001h	0000h
0002h	FFFFh
0003h	0000h
0004h	1000h
0005h	0000h

Slika 1.135. Izgled dela memorije počev od adrese 0

- Napisati deo programa kojim se dodeljuju brojevi ulaza navedenim periferijama.
- Nacrtati izgled prvih 8 ulaza u vektor tabeli, označiti adrese relevantnih lokacija i upisati sadržaje u njih.
- Napisati sekvencu adresa naredbi koje se redom izvršavaju, počev od adrese 0100h.
- Prikazati sadržaj svih poznatih lokacija na vrhu steka nakon izvršavanja 9. instrukcije. Za sačuvanu vrednost PSW dati samo vrednost bita I i L. Naznačiti u kome smeru raste stek.
- Koja će se vrednost nalaziti na lokaciji 0001h nakon izvršenja sekvence pod c)?

Rešenje:

- Traženi program prikazan je na slici 1.136.

```

LOAD #04h
OUTB 0000 0000h
LOAD #05h
OUTB 0000 0004h
LOAD #07h
OUTB 0000 0008h

```

Slika 1.136. Deo programa kojim se dodeljuju brojevi ulaza navedenim periferijama

- Izgled prvih 8 ulaza u vektor tabeli prikazan je na slici 1.137.

broj ulaza u IV tabelu	memorijska adresa	sadržaj	opis sadržaja
7	0000 0013h	0000h	viša reč adrese prekidne rutine za PER2
	0000 0012h	1005h	niža reč adrese prekidne rutine za PER2
6	0000 0011h		
	0000 0010h		
5	0000 000Fh	0000h	viša reč adrese prekidne rutine za PER1
	0000 000Eh	1010h	niža reč adrese prekidne rutine za PER1
4	0000 000Dh	0000h	viša reč adrese prekidne rutine za PER0
	0000 000Ch	1007h	niža reč adrese prekidne rutine za PER0
3	0000 000Bh		
	0000 000Ah		
2	0000 0009h		
	0000 0008h		
1	0000 0007h		

0	0000 0006h	
	0000 0005h	0000h
	0000 0004h	1000h

Slika 1.137. Izgled prvih 8 ulaza u vektor tabeli

c) Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h, prikazana je na slici 1.138. U toku 10. instrukcije, odnosno instrukcije na adresi 100Ah, prekidne rutine za PER0, generiše se prekid, zbog pokušaja upisa u ROM memoriju.

adrese	objašnjenje
0100h, 0101h, 0104h, 0105h, 0108h,	Instrukcije glavnog programa INTE, LOAD, DECA, STORE
1005h, 1006h,	Prekidna rutina za PER2
1007h, 1008h, 100Ah,	Instrukcije PUSHA, LOAD, STORE prekidne rutine za PER0
1000h, 1001h, 1003h, 1004h,	Prekidna rutina za prekid nastao zbog pokušaja upisa u ROM memoriju
100Dh, 100Eh,	Instrukcije POPA, RTI prekidne rutine za PER0
0109h, 010Ah	Instrukcije glavnog programa INCA, INTD

Slika 1.138. Sekvenca adresa naredbi koje se redom izvršavaju, počev od adrese 0100h

d) Nakon izvršavanja 9. instrukcije na steku se nalazi ACC, PSW, PC glavnog programa i vrednost iz akumulatora, kao posledica 8. instrukcije (PUSHA). Izgled steka prikazan je na slici 1.139.

XX	↑ više adrese	
0000h		
PSW (I=1, L=00)		
0000h		
0109h		
0000h		← SP
		↓ niže adrese

Slika 1.139. Izgled steka neposredno nakon izvršavanja 9-te instrukcije

e) Instrukcija STORE na adresi 0105h glavnog programa smešta sadržaj akumulatora(FFFFh) na adresu 1h. Nakon izvršenja sekvence pod c) MEM[1]=FFFFh

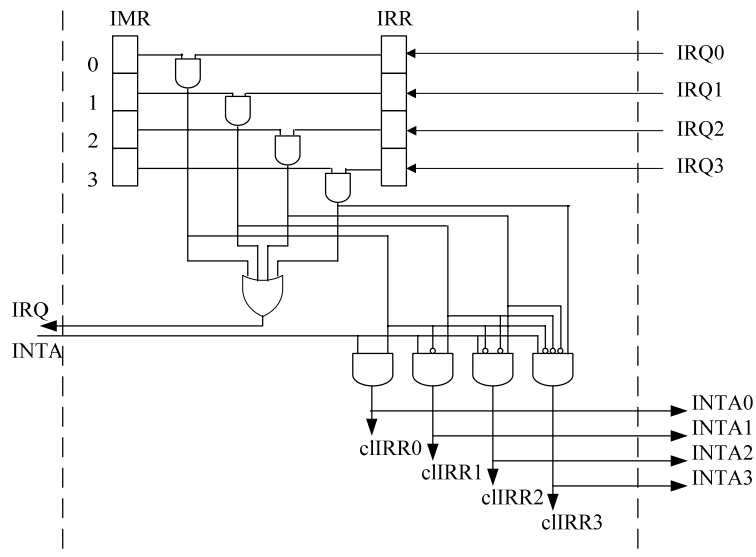
1.29 ZADATAK

Jednoadresni 16-bitni procesor poseduje samo jednu liniju IRQ za spoljašnji prekid kome može da se dodeli proizvoljan ulaz u IVT. Postoji još i linija INTA kojom procesor odobrava prekid i zahteva postavljanje broja ulaza na magistralu podataka. Poseban uređaj, kontroler prekida, služi za paralelnu arbitraciju do 4 zahteva za prekid. Kontroler ne podržava ugnežđavanje prekida sa prioriteto, već samo prioritiranje trenutno postojećih zahteva. Kontroler poseduje i registar maske, koji je vezan za port 0FF0h U/I adresnog prostora, i čija 6 najniža bita maskiraju zahteve za prekid na ulazima kontrolera.

- Nacrtati principijelnu strukturnu šemu kontrolera.
- Koje modifikacije treba izvršiti da bi kontroler podržavao i ugnežđavanje sa prioriteto.
- Napisati deo programa kojim se maskiraju prekidi na ulazima IRQ1 i IRQ2 kontrolera, a ostali dozvoljavaju.
- Prikazati način vezivanja 16 linija zahteva za prekid na procesor, pomoću odgovarajućeg broja datih kontrolera.

Rešenje:

- Tražena šema prikazana je na slici 1.140.



Slika 1.140. Principijelna strukturna šema kontrolera

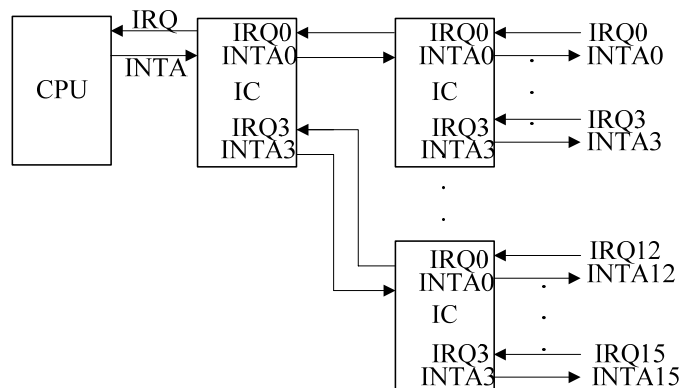
b) Treba uvesti registar ISR (Inerrupt Service Register) kontrolera, i liniju IS procesora, koja se aktivira kada procesor izvrši instrukciju RTI. ISR ima 4 bita, jedinica u bitu i znači da se opslužuje prekid IRQ_i. Zahtev sa IRQ_i se prosleđuje na IRQ ako ne postoji i u ISR u razredu j višeg prioriteta (j<i). Kada stigne INTA upisuje se 1 u odgovarajući razre ISR. Kada stigne IS briše se 1 najvišeg prioriteta u ISR.

c) Traženi program prikazan je na slici 1.141.

```
LOAD #1001b
OUT 0FF0h
```

Slika 1.141. Deo programa kojim se maskiraju prekidi na ulazima IRQ1 i IRQ2 kontrolera, a ostali dozvoljavaju.

d) Način vezivanja 16 linija zahteva za prekid na procesor, pomoću odgovarajućeg broja datih kontrolera prikazan je na slici 1.142.



Slika 1.142. Način vezivanja 16 linija zahteva za prekid na procesor

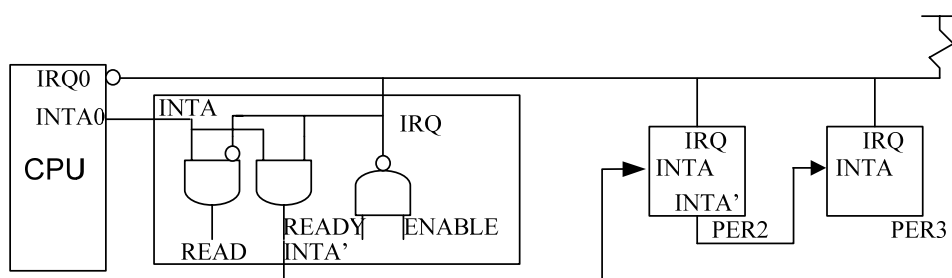
1.30 ZADATAK

Procesor poseduje 4 linije zahteva za prekid IRQ0...IRQ3, kojima se mogu dodeliti proizvoljni ulazi u IVT. Na liniju IRQ0 vezani su zahtevi za prekid od 3 periferije, a na ostale po jedan.

- Nacrtati principijelnu strukturnu šemu ovog sistema.
- U slučaju da postoji više zahteva za prekidom, kako se određuje zahtev najvišeg prioriteta?

Rešenje:

a) Principijelna strukturna šema prikazana je na slici 1.143.



Slika 1.143. Daisy chaining

b) Prioritete IRQ0 do IRQ3 procesor razrešava interno, koderom prioriteta; kada pošalje INTA0 najprioritetnija periferija sa linije šalje svoj broj ulaza (aktivan je njen READ signal).