

U ovom dokumentu nalaze se tri zadatka iz Zbirke iz međuispita iz 2011., 2010. i 2009. Budući da su prije 2012. postojala dva međuispita, gradivo je bilo drugačije raspoređeno. Prvi međuispit imao je samo opće programske zadatke, a drugi je imao ulazno-izlazne programe, ali je uključivao i vanjske jedinice PIO, CT i DMA koje se sada obrađuju tek nakon međuispita. Zato u Zbirci postoji manji broj zadataka kakvi se sada pojavljuju u međuispitu.

Zbog toga smo tri zadatka sa sklopovima PIO i CT (koja smo mogli prilagoditi tako da koriste samo općenite vanjske jedinice) preformulirali i ponovno riješili tako da postoji nešto više tipičnih zadataka prilagođenima sadašnjem gradivu međuispita.

Dodatno su zadatci prilagođeni novom FRISC-u uvedenom 2015. godine.

3.4.6. Jedna prekidna i tri uvjetne vanjske jedinice

(modificirani zadatak iz 2MI09, umjesto PIO je stavljena prekidna VJP)

Riješen: DA Težina: ★★★

Na procesor FRISC spojene su uvjetne ulazne vanjske jedinice VJ1 (adresa FFFF1000) i VJ2 (adresa FFFF2000), uvjetna izlazna vanjska jedinica VJ3 (adresa FFFF3000) i izlazna prekidna VJP (spojen na INT, adresa FFFF4000) koja može primiti 8-bitne podatke.

Napisati glavni program koji neprestano prima podatke od VJ1 i VJ2 (jedinice su međusobno nezavisne). Svaki primljeni podatak treba poslati na VJ3. Ako je podatak djeljiv sa 4, treba povećati brojač prenesenih podataka **BROJAC4** pohranjen na memorijskoj lokaciji 1000₁₆. Na primjer, ako su preneseni podatci 1₁₀, 24₁₀, 5₁₀, 6₁₀ i 8₁₀ brojač sadrži vrijednost 2 jer su samo 24₁₀ i 8₁₀ djeljivi sa 4.

Napisati prekidni potprogram za obradu prekida jedinice VJP, koji na VJP šalje sadržaj brojača **BROJAC4** i ponovno postavlja brojač na 0. Pretpostavlja se da sadržaj brojača neće prijeći opseg od 8 bita.

Potprogrami trebaju čuvati stanje registara.

Prijedlog rješenja:

Glavni program posluhuje sve tri uvjetne vanjske jedinice, a redoslijed je sličan već pokazanim zadacima s uvjetnim vanjskim jedinicama (na primjer, zadatak **Error! Reference source not found.**). U beskonačnoj petlji potrebno je prozivati vanjske jedinice VJ1 i VJ2 jer su nezavisne.

Nakon uspješnog dobivanja podatka od VJ1 ili VJ2 treba poslužiti VJ3 i provjeriti djeljivost sa 4. Kako se ne bi unutar petlje prozivanja dva puta pisao isti odsječak programa koji ovo rješava, taj zadatak je izdvojen u potprogram **OBRADI**. U ovom potprogramu čeka se na spremnost jedinice VJ3 i šalje joj se podatak, a također se provjera je li broj djeljiv sa 4 – u tom slučaju povećava se varijabla **BROJAC4**.

Kada VJP postane spremna (dojavom prekida), na njega treba poslati podatak (vrijednost brojača). U samom prekidnom potprogramu treba prihvatiti prekid, a na kraju dojaviti kraj posluživanja sklopa.

Rješenje:

VJ1	EQU	0FFFF1000	; adrese od tri uvjetne jedinice
VJ1CL	EQU	0FFFF1004	
VJ2	EQU	0FFFF2000	
VJ2CL	EQU	0FFFF2004	
VJ3	EQU	0FFFF3000	
VJ3CL	EQU	0FFFF3004	
VJP_DATA	EQU	0FFFF4000	; adrese od VJP
VJP_IACK	EQU	0FFFF4004	
VJP_IEND	EQU	0FFFF4008	
	ORG	0	
POCETAK	MOVE	10000, SP	; inicijalizacija stoga
	JP	GLAVNI	; skok na glavni program
	ORG	8	; prekidni vektor
	DW	200	

GLAVNI	MOVE	%B 10000, SR	; omogućavanje prekida INT
			; prozivanje VJ1 i VJ2
PET1	LOAD	R0, (VJ1CL)	; VJ1 spremna?
	CMP	R0, 0	
	JP_EQ	PET2	; ako VJ1 nije spremna, ispitaj VJ2
	LOAD	R1, (VJ1)	; učitati podatak u R1
	STORE	R1, (VJ1CL)	; obrisati spremnost VJ1
	CALL	OBRADI	; poslati R1 na VJ3
PET2	LOAD	R0, (VJ2CL)	; VJ2 spremna?
	CMP	R0, 0	
	JP_EQ	PET1	; ako VJ2 nije spremna, ispitaj VJ1
	LOAD	R1, (VJ2)	; učitati podatak u R1
	STORE	R1, (VJ2CL)	; obrisati spremnost VJ2
	CALL	OBRADI	; poslati R1 na VJ3
	JP	PET1	; povratak na ispitivanje spremnosti VJ1
			; potprogram za prijenos na VJ3, u R1 je podatak za slanje na VJ3
OBRADI	PUSH	R0	; spremanje konteksta
PET3	LOAD	R0, (VJ3CL)	; VJ3 spremna?
	CMP	R0, 0	
	JP_EQ	PET3	; čekaj dok VJ3 ne postane spremna
	STORE	R1, (VJ3)	; poslati podatak iz R1
	STORE	R1, (VJ3CL)	; obrisati spremnost VJ3
	AND	R1, %B 11, R0	; ispitati djeljivost podatka
	JP_NZ	NAZAD	; ako nije djeljiv sa 4, izlazak iz potprog.
	LOAD	R0, (BROJAC4)	; učitati brojač
	ADD	R0, 1, R0	; povećati brojač
	STORE	R0, (BROJAC4)	; spremiti brojač
NAZAD	POP	R0	; obnova konteksta
	RET		; povratak iz potprograma
			; prekidni potprogram
IRQ	ORG	200	
	PUSH	R0	; spremanje konteksta (SR se ne mijenja)
	STORE	R0, (VJP_IACK)	; potvrda prekida
	LOAD	R0, (BROJAC4)	; učitati brojač
	STORE	R0, (VJP_DATA)	; poslati na PIO
	MOVE	0, R0	; upisati 0 u brojač
	STORE	R0, (BROJAC4)	
	STORE	R0, (PIOIEND)	; dojava kraja posluživanja
	POP	R0	; obnova konteksta
	RETI		; povratak iz prekida
	ORG	1000	
BROJAC4	DW	0	; varijabla

Komentar rješenja:

U ovom rješenju za potprogram **OBRADI** odabrano je da se podatak šalje preko registra **R1**. Provjera je li broj djeljiv sa 4 obavlja se maskiranjem podatka vrijednošću 11_2 – ako su zadnja dva bita podatka ničice, podatak je djeljiv sa 4. U zadatcima s potprogramima često se kao

varijabla – brojač – koristi određena memorijska lokacija. Prilikom promjene vrijednosti varijable, ne smije se zaboraviti prvo učitati, a kasnije i spremiti novu vrijednost u memoriju. U prekidnom potprogramu nije potrebno spremati **SR**, jer naredbe **MOVE**, **LOAD** i **STORE** ne mijenjaju zastavice, što je naznačeno komentarom u rješenju.

Riješite zadatak tako da i VJP radi u uvjetnom načinu. Obratite pažnju na kojim sve mjestima treba ispitivati spremnost VJP Riješite zadatak tako da jedinica VJ3 nije uvjetna nego bezuvjetna, a VJP neka radi u prekidnom načinu. Riješite zadatak tako da jedinica VJ3 nije uvjetna nego bezuvjetna, a VJP neka radi u uvjetnom načinu.

3.6.8. Tipkovnica spojena na prekidnu jedinicu, zaslon i zvučnik

(modificirani zadatak iz 2MI10, umjesto PIO je stavljena prekidna VJP)

Riješen: DA Težina: ★★

Na procesor FRISC su spojene tri vanjske jedinice: prekidna ulazna jedinica VJP, uvjetna jedinica ZASLON i bezuvjetna jedinica ZVUCNIK. Na VJP je spojena tipkovnica, a VJP je spojena na INT. Svaki puta kada je na tipkovnici pritisnuta tipka, jedinici VJP se šalje 8-bitni ASCII kôd znaka pritisnute tipke, a VJP generira prekid.

Napisati program koji beskonačno pomoću prekida prima znakove sa VJP (tipkovnice) i šalje ih na zaslon. Jedinica ZASLON prima znakove u zapisu ASCII i prikazuje ih. Pretpostaviti da je jedinica ZASLON vrlo brza u odnosu na tipkovnicu.

Samo ako je sa VJP primljen znak zvonca (ASCII kod 07₁₆ = BEL, *bell*), onda znak ne treba slati na ZASLON, već treba aktivirati jedinicu ZVUCNIK slanjem bilo kojeg podatka.

Glavni program izvodi praznu petlju. Adrese sklopova odabrati proizvoljno.

Prijedlog rješenja:

Iako se zadatak na prvi pogled možda čini kompliciran, pažljivijim čitanjem vidi se da se radi o jednoj bezuvjetnoj, jednoj uvjetnoj, i jednoj prekidnoj jedinici, a zadani način posluživanja je vrlo jednostavan. Preporučuje se kod ovakvih, problemskih zadataka, prvo razlučiti bitne informacije i predložiti si način rada sklopova.

Zadatak se svodi na primanje znaka od VJP (pomoću prekida) nakon čega se, ovisno o vrijednosti znaka, uključuje zvučnik (bezuovjetno) ili se znak šalje na zaslon (uvjetno). Što se tiče tipkovnice, ona se u programu ne mora posluživati jer je spojena na VJP. U prekidnom potprogramu se znak prima i ispituje. Ako je primljen ASCII-kôd 7, bezuvjetno se aktivira zvučnik, a u suprotnom treba čekati na spremnost zaslona i poslužiti ga kao i bilo koju drugu uvjetnu izlaznu jedinicu. U glavnom programu treba izvoditi praznu petlju.

Rješenje:

VJP_DATA	EQU	0FFFF1000	; adrese vanjskih jedinica
VJP_IACK	EQU	0FFFF1004	
VJP_IEND	EQU	0FFFF1008	
ZASL_POD	EQU	0FFFF2000	
ZASL_ISP	EQU	0FFFF2004	
ZASL_BRIS	EQU	0FFFF2004	
ZVUCNIK	EQU	0FFFF3000	
POCETAK	ORG	0	
	MOVE	10000, SP	; stog
	JP	GLAVNI	; skok na glavni
	ORG	8	; adresa prekidnog vektora
	DW	1000	; prekidni vektor
GLAVNI	MOVE	%B 10000, SR	; omogućavanje prekida INT
PET	JR	PET	; prazna petlja

	ORG	1000	; prekidni potprogram za VJP
PREKIDNI	PUSH	R0	; čuvanje konteksta
	MOVE	SR, R0	
	PUSH	R0	
	PUSH	R1	
	STORE	R0, (VJP_IACK)	; dojava prihvata prekida VJP
	LOAD	R1, (VJP_DATA)	; učitavanje kôda utipkanog znaka iz VJP
	CMP	R1, 7	; je li učitani kôd znaka BEL?
	JP_NE	ZASLON	; ako nije BEL, znak se šalje na zaslon
ZVUCNIK	STORE	R1, (ZVUCNIK)	; inače zazvoni
	JR	VAN	; skok van, ništa se ne ispisuje na zaslon
ZASLON	LOAD	R0, (ZASL_ISP)	; čekanje na spremnost zaslona
	OR	R0, R0, R0	
	JR_Z	ZASLON	
	STORE	R1, (ZASL_POD)	; slanje podatka na zaslon
	STORE	R0, (ZASL_BRIS)	; brisanje spremnosti zaslona
VAN	STORE	R0, (VJP_IEND)	; dojava kraja posluživanja VJP
	POP	R1	; obnova konteksta
	POP	R0	
	MOVE	R0, SR	
	POP	R0	
	RETI		; povratak

Komentar rješenja:

U prekidnom potprogramu potrebno je dojaviti prihvati prekida jedinici VJP, pročitati vrijednost kôda tipke sa VJP, te ga usporediti s kôdom znaka BEL – s brojem 7. Ako je primljen BEL, na bezuvjetnu vanjsku jedinicu ZVUCNIK treba samo poslati bilo koju vrijednost i izaći iz prekidnog potprograma. No, ako je primljen bilo koji drugi znak, potrebno ga je ispisati na jedinicu ZASLON. S obzirom da je ZASLON uvjetna jedinica, prvo treba u petlji čekati na spremnost, a tek kada je jedinica spremna treba poslati vrijednost primljenog znaka i nakon toga joj obrisati spremnost. Prilikom izlaska iz prekidnog potprograma treba dojaviti kraj posluživanja jedinici VJP.

U zadatku je zadano da zaslon radi znatno brže nego što znakovi dolaze s tipkovnice. Na taj način posluživanje zaslona neće blokirati primanje znakova pomoću VJP, jer bi se u protivnom mogli gubiti utipkani znakovi, dok se čeka na spremnost zaslona.

Riješite zadatak s razlikom da je VJP spojena na NMI. Dodatno, na kraju primanja svakog retka od 80 znakova treba na zaslon poslati i dodatni znak za prijelaz u novi redak (ASCII kôd $0A_{16}$ = LF, *line feed*). Po primanju nul-znaka (ASCII kôd 00_{16} = NUL, *null*) treba zaustaviti procesor.

3.6.17. Alarm za bicikl

(modificirani zadatak 2MI11, umjesto PIO je stavljena bezuvjetna BVJ, a umjesto CT je stavljena prekidna PVJ)

Riješen: DA Težina: ★★★

Alarmni uređaj za bicikl se sastoji od procesora FRISC, uvjetne vanjske jedinice te bezuvjetne jedinice BVJ i prekidne jedinice PVJ. Uvjetna vanjska jedinica je senzor vibracija koji detektira pokušaj krađe bicikla. Senzor vibracija postaje spreman:

- ako nije bilo vibracija pa su počele; vrijednost podatka pročitano sa senzora bit će 1
- ako je bilo vibracija pa su prestale; vrijednost podatka pročitano sa senzora bit će 0

Na izlaznu jedinicu BVJ, koja može primiti 8-bitne podatke, spojen je zvučnik koji može raditi na sljedeće načine:

- tiši alarm: ako se postave jedinice na nižih 4 bita od BVJ
- glasniji alarm: ako se postave jedinice na svih 8 bitova od BVJ
- alarm isključen: ako se postave ničice na svih 8 bitova od BVJ

Prekidna jedinica PVJ ne služi za prijenos podataka i zauzima samo tri lokacije (prva lokacija ne služi za uobičajeni prijenos podataka nego služi za dojavu prihvata prekida). PVJ periodički generira prekide svake stotinke sekunde. Pomoću treće lokacije može se dozvoliti ili zabraniti generiranje prekida. PVJ je spojena na NMI.

Napišite program koji upravlja alarmom na sljedeći način:

Glavni program kontinuirano ispituje stanje senzora. Kada počnu vibracije, treba uključiti tiši alarm. Tada se počinje mjeriti trajanje od 3 sekunde počevši od uključivanja tišeg alarma. Ako nakon 3 sekunde vibracije nisu prestale, treba uključiti glasniji alarm. Ako vibracije prestanu (u bilo kojem trenutku), treba isključiti alarm.

Na početku rada, alarm treba biti isključen. Adrese vanjskih jedinica odabrati po volji.

Prijedlog rješenja:

Na početku i alarm (zvučnik spojen na BVJ) i prekidi trebaju biti isključeni. U glavnom programu se prvo isključuju zvučnik i zabranjuje se jedinici PVJ izazivanje prekida. Nakon toga se beskonačno uključuje tiši alarm kad započnu vibracije, ili se isključuje alarm ako vibracije prestanu. U petlji glavnog programa čeka se spremnost senzora (uvjetna vanjska jedinica). Važno je uočiti kako radi senzor vibracija. Suprotno onome što bi možda bilo intuitivno, spremnost ne znači da vibracije traju, a nespremnost ne znači da vibracija nema. Umjesto toga, spremnost senzora označava da su vibracije ili počele ili završile. Kad senzor postane spreman, pročita se iz njega podatak na temelju kojega se može ustanoviti jesu li vibracije započele ili su završile. Ova dva slučaja obrađuju se na sljedeći način.

Ako su vibracije započele, uključuje se tiši alarm, kojim se upravlja preko BVJ, slanjem odgovarajućeg podatka. Istovremeno s uključivanjem tišeg alarma, dozvoljava se jedinici PVJ da postavlja prekide, čime se započne mjeriti razdoblje od 3 sekunde. Prekidi se izazivaju svake stotinke pa u prekidnom potprogramu treba prebrojiti 300 prekida (za to se koristi lokacija BROJAC). Nakon isteka 3 sekunde u prekidnom potprogramu se uključi glasniji alarm i zabrani se daljnje generiranje prekida jedinici PVJ (jer se razdoblje od 3 sekunde treba izmjeriti samo jednokratno).

Ako su vibracije završile, onda treba isključiti alarm, bez obzira je li trenutno aktivan tiši ili glasniji alarm. Zato se na BVJ šalje ništica da se alarm isključi. Ako su vibracije prestale dok je aktivan tiši alarm, onda je sigurno aktivna prekidna jedinica PVJ koji mjeri razdoblje od 3 sekunde, pa je treba "isključiti". Ako su pak vibracije prestale dok je aktivan glasniji alarm, onda PVJ nije aktivna. U ovom rješenju glavni program će i u ovom slučaju zabraniti jedinici PVJ postavljanje prekida, tj. isključit će PVJ koja već jeste isključena (što je jednostavnije nego pamtiti je li trenutno aktivan tiši ili glasniji alarm).

Rješenje:

UVJ_DATA	EQU	0FFFF0000	; adrese vanjskih jedinica
UVJ_TEST	EQU	0FFFF0004	
BVJ	EQU	0FFFF1000	
PVJ_IACK	EQU	0FFFF2000	; dojava prihvata prekida
PVJ_IEND	EQU	0FFFF2004	; dojava kraja posluživanja
PVJ_EINT	EQU	0FFFF2008	; omogućivanje/zabranjivanje postavljanja prekida
POCETAK	ORG	0	
	MOVE	10000, SP	; stog
	JP	GLAVNI	; skok na glavni
PP	ORG	0C	; adresa nemaskirajućeg prekida NMI (za PVJ)
	; prekidni potprogram - prošla je stotinka sekunde		
	PUSH	R0	; spremanje konteksta
	MOVE	SR, R0	
	PUSH	R0	
	STORE	R0, (PVJ_IACK)	; prihvaćanje prekida
	LOAD	R0, (BROJAC)	; smanji brojač pristiglih prekida
	SUB	R0, 1, R0	
	STORE	R0, (BROJAC)	
	JR_NZ	VAN	; nije prošlo 3 sek => idi van
TRI_SEK	; proteklo je tri sekunde		
	MOVE	0, R0	; zaustavi PVJ - zabrani joj postavljanje prekida
	STORE	R0, (PVJ_EINT)	
GLASNO	MOVE	%B 11111111, R0	; uključivanje glasnijeg alarma: 11111111
	STORE	R0, (BVJ)	
VAN	STORE	R0, (PVJ_IEND)	; dojava kraja posluživanja PVJ
	POP	R0	; obnova konteksta
	MOVE	SR, R0	
	POP	R0	
	RETN		; povratak iz potprograma
BROJAC	DW	0	; brojač 300 prekida


```

; glavni program

GLAVNI    MOVE    0, R0          ; na početku je alarm isključen
          STORE   R0, (BVJ)      ; tj. zvučnik ne radi
          STORE   R0, (PVJ_EINT) ; zaustavi PVJ - zabrani joj postavljanje prekida

PETLJA    LOAD    R0, (UVJ_TEST) ; čekanje spremnosti senzora
          OR      R0, R0, R0
          JR_Z     PETLJA

ISPITAJ   LOAD    R0, (UVJ_DATA) ; čitanje stanja senzora iz UVJ
          STORE   R0, (UVJ_TEST) ; brisanje spremnosti
          CMP     R0, 0          ; ako je stanje 0 (vibracije su prestale):
          JR_EQ    ISKLJUCI      ; ... idi na isključivanje

TIHO      ; u protivnom stanje je 1 (vibracije su počele): uključi tiši alarm
          MOVE    %B 1111, R0    ; tiši alarm: slanje 1111 na BVJ (zvučnik)
          STORE   R0, (BVJ)

          ; pokreni mjerenje 3 sekunde pomoću PVJ
          MOVE    300, R0
          STORE   R0, (BROJAC)   ; inicijaliziraj brojač prekida
          MOVE    1, R0
          STORE   R0, (PVJ_EINT) ; pokreni PVJ, tj.dozvoli joj postavljanje prekida

          JR      PETLJA        ; povratak na ispitivanje spremnosti

ISKLJUCI  MOVE    0, R0          ; isključivanje brojila pa alarma
          STORE   R0, (PVJ_EINT) ; zaustavi PVJ - zabrani joj postavljanje prekida
          STORE   R0, (BVJ)      ; isključivanje alarma (zvučnika)
          JR      PETLJA

```

Komentar rješenja:

Prilikom rješavanja važno je shvatiti da PVJ služi isključivo za mjerenje vremena od uključivanja tišeg do uključivanja glasnijeg alarma (3 sekunde). Kada je uključen glasniji alarm, PVJ treba biti isključena. Također, nakon svakog isključivanja alarma, PVJ3 treba isključiti, i pri sljedećem uključivanju alarma početi brojiti ispočetka (varijabla BROJAC se ponovno inicijalizira). U prekidnom potprogramu se smanjuje vrijednost varijable BROJAC, a kad ona padne na nulu to je znak da su istekle 3 sekunde. U protivnom, se samo obavlja običan povratak iz prekidnog potprograma. Ako su istekle 3 sekunde, onda se dodatno zabranjuje daljnje prekidanje jedinici PVJ i uključi se glasni alarm.

Iako se na prvi pogled može činiti da nije potrebno provjeravati koja je vrijednost na senzoru (već samo provjeriti spremnost senzora), jer se vrijednost uvijek izmjenjuje (0 -> 1 -> 0), ipak je potrebno učitati i provjeriti vrijednost, jer ne znamo u kojem će stanju senzor biti prilikom samog uključivanja.

Razmotrimo i rubni slučaj kada nakon početnih vibracija i uključivanja tihog alarma dođe do prestanka vibracija u isto vrijeme kad ističu 3 sekunde i kad PVJ izaziva prekid.

Ako do prekida dođe neznatno prije prestanka vibracija, tada će se u prekidnom potprogramu uključiti glasniji alarm, ali će već povratkom u glavni program biti prepoznata spremnost senzora i nakon što se ispitivanjem ustanovi da su vibracije prestale, isključit će se alarm. Tako će glasni alarm trajati zanemarivo kratko – dok se izvede svega desetak naredaba (što je možda i prekratko da ga se može čuti). Ovo ponašanje može se ocijeniti ispravnim.

Ako senzor dojaviti kraj vibracija neznatno prije isteka 3 sekunde, može se dogoditi da glavni program izvodi naredbe na labelama **PETLJA**, **ISPITAJ** i **ISKLJUCI**, a da još nije stigao do naredbe **STORE R0,(PVJ_EINT)** iza labela **ISKLJUCI**. Ako u tom razdoblju dođe do isteka 3 sekunde i postavljanja prekida INT3, izvest će se prekidni potprogram u kojemu će biti uključen glasniji alarm, a povratkom u glavni program će se petlja izvesti do kraja, pri čemu će se zaustaviti već zaustavljena PVJ, te će se isključiti zvučnik i ponašanje će opet biti ispravno. Ako se uspije izvesti naredba **STORE R0,(VPJ_EINT)** iza labela **ISKLJUCI**, tada će PVJ biti zaustavljena i neće moći izazvati prekid te će se nakon toga isključiti zvučnik, što je ponovno ispravno ponašanje.

Što bi se moglo dogoditi kad bi se zamijenio redoslijed dviju naredaba **STORE** iza labela **ISKLJUCI**?