

Prezime i ime (tiskanim slovima): _____ JMBAG: _____

Izjavljujem da tijekom ispita neću od drugoga primiti niti drugome pružiti pomoć, te da se neću koristiti nedopuštenim sredstvima. Ove su radnje teška povreda Kodeksa ponašanja te mogu uzrokovati i trajno isključenje s Fakulteta. Izjavljujem da mi zdravstveno stanje dozvoljava pisanje ovog ispita. Potpis: _____.

Dozvoljeno je koristiti isključivo službene šalabahtere (popis naredaba FRISC-a i ARM-a). Programe treba pisati uredno i komentirati pojedine cjeline programa. Rješenja teorijskih zadataka treba napisati na ovaj papir. Ispit traje 100 minuta i nosi 30 bodova.

1.a (1,5 bod) Općenita 4-bitna (ne FRISC-ova) aritmetičko-logička jedinica oduzima binarne brojeve 0101 - 1101. Nakon operacije rezultat i zastavice će biti: rezultat = _____, prijenos = _____, posudba = _____, preljev = _____, ništica = _____, predznak = _____.
MORA SE VIDJETI POSTUPAK RJEŠAVANJA!!!

Mjesto za postupak:

1.b (2 boda) U memoriji FRISC-a su upisani podatci pomoću pseudonaredaba na desnoj strani. U tablici ispod napišite sadržaje navedenih bajtova u memoriji (u heksadekadskoj bazi):

ORG 100
DH +3, -3
DW 5

adresa	100	101	102	103	104	105	106	107
sadržaj								

1.c (1 bod) U mikroarhitekturi FRISC-a postoji sklop SHUFFLER. Navedite četiri naredbe za čije ispravno izvođenje je **nužan** sklop SHUFFLER: _____, _____, _____ i _____.

1.d (1 bod) Zastavica GIE u registru SR ima početno stanje 0 što označava da je INT (tj. int[0]) _____. GIE se automatski briše kada se (dogodi što?) _____. GIE se automatski postavlja kada se izvede naredba _____. IIF se automatski postavlja kada se izvede naredba _____.

1.e (2 boda) **Smjerovi** FRISC-ovih priključaka su: ADR je _____, DATA je _____, READ je _____, WRITE je _____, WAIT je _____. Čitanje/pisanje u brzu memoriju traje _____ takt(ova) clocka. Memorija mora aktivirati WAIT ako je _____. Što će napraviti FRISC ako je WAIT aktivan? _____.

1.f (2,5 boda) Napišite korake koje FRISC obavlja prilikom izvođenja naredbe STORE R2, (R5+200)

Razina dohvata:

Prva polovina periode CLOCK-a:

Druga polovina periode CLOCK-a:

Razina izvođenja:

Prva polovina periode CLOCK-a:

Druga polovina periode CLOCK-a:

1.g (0,5 boda) U **ulaznoj uvjetnoj** vanjskoj jedinici bistabil stanja postavlja se u 1 nakon što _____.

2. (7 bodova) U memoriji od adrese 1000_{16} je blok sa 100_{16} podataka u 64-bitnom formatu s bitom za predznak. Glavni program treba svaki broj iz bloka **pretvoriti** u 64-bitni broj u formatu 2^k , **obraditi** ga potprogramom OBRADI i obrađene 64-bitne brojeve u formatu 2^k **spremiti** u novi blok od adrese 2000_{16} .

Potprogram OBRADI preko stoga prima 64-bitni broj u formatu 2^k . Broj treba obraditi ovako: ako je broj pozitivan treba ga **umanjiti** za 123_{16} , ako je negativan treba ga **uvećati** za 567_{16} , a ako je ničica treba ga ostaviti **bez promjene**. Rezultat, tj. obrađeni broj (također u 64-bitnom formatu 2^k) vraća se registrima R0 (niži dio) i R1 (viši dio).

3. (12,5 bodova) Na FRISC su spojene tri uvjetne vanjske jedinice U1, U2 i U3, zatim jedna bezuvjetna vanjska jedinica BJ, te jedna prekidna vanjska jedinica PJ spojena na INT. Adrese odaberite sami.

FRISC treba **prenositi** podatke sa U1 na U2 i **prebrajati** ih. **Istovremeno** treba **prenositi** podatke sa BJ na U3. Nakon 333_{16} poslanih podataka na U2, treba zaustaviti procesor i PJ.

Kad dođe prekid od PJ, treba joj poslati broj podataka poslanih na U2.

Prezime i ime (tiskanim slovima): _____ JMBAG: _____

Izjavljujem da tijekom ispita neću od drugoga primiti niti drugome pružiti pomoć, te da se neću koristiti nedopuštenim sredstvima. Ove su radnje teška povreda Kodeksa ponašanja te mogu uzrokovati i trajno isključenje s Fakulteta. Izjavljujem da mi zdravstveno stanje dozvoljava pisanje ovog ispita. Potpis: _____.

Dozvoljeno je koristiti isključivo službene šalabahtere (popis naredaba FRISC-a i ARM-a). Programe treba pisati uredno i komentirati pojedine cjeline programa. Rješenja teorijskih zadataka treba napisati na ovaj papir. Ispit traje 100 minuta i nosi 30 bodova.

1.a (1,5 bod) Općenita 4-bitna (ne FRISC-ova) aritmetičko-logička jedinica oduzima binarne brojeve 0101 - 1101. Nakon operacije rezultat i zastavice će biti: rezultat = 1000, prijenos = 0, posudba = 1, preljev = 1, ništica = 0, predznak = 1. MORA SE VIDJETI POSTUPAK RJEŠAVANJA!!!

Mjesto za postupak:

011 - prijenosi prijenos = 0
0101 posudba = not prijenos = 1
0010 preljev = 0 xor 1 = 1
+ 1 predznak = 1
1000 ništica = 0 (jer rezultat nije 0)

1.b (2 boda) U memoriji FRISC-a su upisani podatci pomoću pseudonaredaba na desnoj strani. U tablici ispod napišite sadržaje navedenih bajtova u memoriji (u heksadekadskoj bazi):

ORG 100
DH +3, -3
DW 5

adresa	100	101	102	103	104	105	106	107
sadržaj	03	00	FD	FF	05	00	00	00

1.c (1 bod) U mikroarhitekturi FRISC-a postoji sklop SHUFFLER. Navedite četiri naredbe za čije ispravno izvođenje je **nužan** sklop SHUFFLER: LOADB, STOREB, LOADH i STOREH.

1.d (1 bod) Zastavica GIE u registru SR ima početno stanje 0 što označava da je INT (tj. int[0]) zabranjen/onemogućen. GIE se automatski briše kada se (dogodi što?) PRIHVATI (INT/zahtjev za mask.prekid/maskirajući prekid). GIE se automatski postavlja kada se izvede naredba RETI. IIF se automatski postavlja kada se izvede naredba RETN.

1.e (2 boda) **Smjerovi** FRISC-ovih priključaka su: ADR je izlazni, DATA je dvosmjerni, READ je izlazni, WRITE je izlazni, WAIT je ulazni. Čitanje/pisanje u brzu memoriju traje 1 takt(ova) clocka. Memorija mora aktivirati WAIT ako je spora/ne stigne obaviti traženu operaciju (čitanje ili pisanje). Što će napraviti FRISC ako je WAIT aktivan? umetnuti će stanje čekanja / produljiti će ciklus čitanja ili pisanja.

1.f (2,5 boda) Napišite korake koje FRISC obavlja prilikom izvođenja naredbe STORE R2, (R5+200)

Razina dohvata:

Prva polovina periode CLOCK-a:
PC -> AR
Druga polovina periode CLOCK-a:
(AR) -> IR
dekodiranje
R5, ext 200 -> ALU
ALU zbraja
PC + 4 -> PC
onemogućiti dohvat u sljedećem ciklusu

Razina izvođenja:

Prva polovina periode CLOCK-a:
ALU -> AR
R2 -> DR
Druga polovina periode CLOCK-a:
omogućiti dohvat u sljedećem ciklusu

1.g (0,5 boda) U ulaznoj uvjetnoj vanjskoj jedinici bistabil stanja postavlja se u 1 nakon što vanjski uređaj/proces pošalje podatak vanjskoj jedinici / vanjska jedinica zapamti podatak poslan od vanjskog procesa/uređaja.

2. (7 bodova) U memoriji od adrese 1000_{16} je blok sa 100_{16} podataka u 64-bitnom formatu s bitom za predznak. Glavni program treba svaki broj iz bloka **pretvoriti** u 64-bitni broj u formatu 2^k , **obraditi** ga potprogramom OBRADI i obrađene 64-bitne brojeve u formatu 2^k **spremiti** u novi blok od adrese 2000_{16} .

Potprogram OBRADI preko stoga prima 64-bitni broj u formatu 2^k . Broj treba obraditi ovako: ako je broj pozitivan treba ga **umanjiti** za 123_{16} , ako je negativan treba ga **uvećati** za 567_{16} , a ako je ničica treba ga ostaviti **bez promjene**. Rezultat, tj. obrađeni broj (također u 64-bitnom formatu 2^k) vraća se registrima R0 (niži dio) i R1 (viši dio).

```

ORG      0
MOVE     10000, SP      ;inicijalizacija stoga

MOVE     1000, R2       ; IZVOR
MOVE     2000, R3       ; ODREDISTE
MOVE     100,  R4       ; BROJAC

PETLJA   LOAD    R0, (R2) ; dohvat broja iz bloka
         LOAD    R1, (R2+4)

         OR      R1, R1, R1 ; ispitaj predznak broja
         JR_P    DALJE

NEGATIVAN SHL     R1, 1, R1 ; briši bit predznaka
         SHR     R1, 1, R1

         XOR     R0, -1, R0 ; negativne treba dvojno komplementirati
         XOR     R1, -1, R1 ; u dvostruko preciznosti

         ADD     R0, 1, R0
         ADC     R1, 0, R1

DALJE    PUSH    R0       ; parametri na stog
         PUSH    R1
         CALL    OBRADI   ; poziv
         ADD     SP, 8, SP ; iš enje parametara

         STORE   R0, (R3) ; spremi rezultat
         STORE   R1, (R3+4)

         ADD     R2, 8, R2 ; pomak adresa
         ADD     R3, 8, R3

         SUB     R4, 1, R4 ; broja petlje i kraj petlje
         JR_NZ   PETLJA

         HALT

OBRADI   ; spremanje konteksta ili bez spremanja ako pametno iskoriste registre
         LOAD    R0, (SP+8) ; dohvat parametara
         LOAD    R1, (SP+4)

         CMP     R1, 0 ; ispitivanje predznaka broja
         JR_M    NEGAT
         JR_NZ   POZIT

         CMP     R0, 0 ; za nulu treba ispitati oba dijela broja
         JR_NE   POZIT
         RET

NEGAT    ADD     R0, 567, R0 ; uve anje negativnog broja
         ADC     R1, 0, R1
         RET

POZIT    SUB     R0, 123, R0 ; smanjivanje pozitivnog broja
         SBC     R1, 0, R1
         RET ; povratak

```

3. (12,5 bodova) Na FRISC su spojene tri uvjetne vanjske jedinice U1, U2 i U3, zatim jedna bezuvjetna vanjska jedinica BJ, te jedna prekidna vanjska jedinica PJ spojena na INT. Adrese odaberite sami.

FRISC treba **prenositi** podatke sa U1 na U2 i **prebrajati** ih. **Istovremeno** treba prenositi **podatke** sa BJ na U3. Nakon 333₁₆ poslanih podataka na U2, treba zaustaviti procesor i PJ.

Kad dođe prekid od PJ, treba joj poslati broj podataka poslanih na U2.

Bitan je redoslijed posluživanja:

1. Prvo treba PROZIVATI U1 i U3.

1.1 Ako U3 postane spremna, treba je poslužiti i nastaviti s prozivanjem 1.

1.2 Ako U1 postane spremna, treba je poslužiti i prijeći na korak 2.

2. Zatim treba PROZIVATI U2 i U3.

1.1 Ako U3 postane spremna, treba je poslužiti i nastaviti s prozivanjem 2.

1.2 Ako U2 postane spremna, treba je poslužiti i prijeći na korak 1.

```
BJ      EQU    0FFFF0000
U1_D    EQU    0FFFF1000
U1_BS   EQU    0FFFF1004
U2_D    EQU    0FFFF2000
U2_BS   EQU    0FFFF2004
U3_D    EQU    0FFFF3000
U3_BS   EQU    0FFFF3004
PJ_D    EQU    0FFFF4000
PJ_IACK EQU    0FFFF4004
PJ_IEND EQU    0FFFF4008
PJ_STOP EQU    0FFFF400C

        ORG     0
        MOVE    10000, SP      ; inicijalizacija stoga
        JP      GLAVNI        ; skok preko vektora u glavni

        ORG     8              ; vektor na adresi 8
        DW      200

GLAVNI   MOVE    %B 10000, SR ; dozvoliti prekide

; prozivanje 1. i 3. uvjetne jedinice
POLL_1_3 LOAD    R0, (U1_BS)   ; ispitaj prvu
        OR      R0, R0, R0
        JR_NZ   CITAJ_1        ; posluži prvu ako je spremna

CHK3     LOAD    R0, (U3_BS)   ; ispitaj tre u
        OR      R0, R0, R0
        JR_Z    POLL_1_3      ; natrag na ispitivanje prve

        ; posluži tre u ako je spremna
        LOAD    R2, (BJ)       ; prvo uzmi podatak s bezuvjetne

        STORE   R2, (U3_D)     ; pošalji ga tre oj i obriši joj status
        STORE   R2, (U3_BS)

        JR      POLL_1_3      ; natrag na ispitivanje prve

CITAJ_1  LOAD    R1, (U1_D)     ; itaj podatak i briši status
        STORE   R0, (U1_BS)
        ; nakon posluživanja prve ide se na prozivanje 2. i 3.
```

```

; prozivanje 2. i 3. uvjetne jedinice
POLL_2_3  LOAD  R0, (U2_BS) ; ispitaj drugu
          OR     R0, R0, R0
          JR_NZ  SALJI_2      ; posluži drugu ako je spremna

          LOAD  R0, (U3_BS) ; ispitaj tre u
          OR     R0, R0, R0
          JR_Z   POLL_2_3     ; natrag na ispitivanje druge

          ; posluži tre u ako je spremna (isto kao u prvoj petlji)
          LOAD  R2, (BJ)      ; prvo uzmi podatak s bezuvjetne

          STORE R2, (U3_D)    ; pošalji ga tre o j i obriši joj status
          STORE R2, (U3_BS)

          JR     POLL_2_3     ; natrag na ispitivanja druge

SALJI_2   STORE R1, (U2_D)    ; šalji podatak i briši status
          STORE R0, (U2_BS)

          LOAD  R4, (BROJAC)
          ADD   R4, 1, R4     ; update broja a U MEMORIJI
          STORE R4, (BROJAC)
          CMP   R4, 333       ; ispitivanje kraja
          JR_EQ KRAJ
          JP    POLL_1_3     ; nakon slanja na drugu, opet prozivanje 1. i 3.

KRAJ      MOVE  0, R0        ; zaustavi prekidnu
          STORE R0, (PJ_STOP)
          HALT               ; zaustavi procesor

```

; PREKIDNI POTPROGRAM

```

ORG 200 ; adresa mora odgovarati vektoru

PUSH R0 ; spremi kontekst, SR se ne mijenja

STORE R0, (PJ_IACK) ; dojava prihvata prekida

LOAD R0, (BROJAC) ; dohvat broja a i slanje na PJ
STORE R0, (PJ_D)

STORE R0, (PJ_IEND) ; dojava kraja posluživanja

POP R0 ; obnova konteksta
RETI ; povratak iz prekida

```

```

BROJAC DW 0 ; BROJA MORA BITI U MEMORIJI

```