Arhitektura računala 1 – 5. Blic ARM

5. blic – Zadaci za vježbu

1. Za procesor ARM napisan je sljedeći program:

```
        ORG 0
        MOV SP, #1<8
GLAVNI
        BL POTP
L1      SWI 123456
L2      SWI 123456
L3      SWI 123456
L4      SWI 123456
POTP    STMFD R13!, {R1, R2, R14}
        …
        LDMFD R13!, {R14, R1, R2}
        MOV PC, LR
```

Program završava na: Labeli L1.

2. Procesor ARM izvodi sljedeći programski odsječak:

```
PET     …
        MOV R0, #2
        CMP R0, R1
        BEQ PET
        MOVEQ R0, #7
        …
```

Za naredbu MOVEQ vrijedi jedna od sljedećih tvrdnji: Naredba se nikad neće izvesti.

3. Za procesor ARM napisan je potprogram POTP koji koristi drugi potprogram SQRT za računanje izraza. Sljedeći programski odsječak prikazuje potprogram POTP:

```
POTP    STMDB SP!, {LR}
        …
        BL SQRT
        …
        LDMIA R13!, {R14}
        MOV R15, R14
```

Za navedeni programski odsječak vrijedi jedna od navedenih tvrdnji: Odsječak je potpuno ispravan.

4. Procesor ARM izvodi sljedeći programski odsječak:

```
        MOV R0, #6
        MOV R1, #7
        MOV R2, #3
        STR R0, [R1, -R2]
```

Navedeni odsječak će: Pohraniti 32-bitnu vrijednost 6 na lokaciju 4, sadržaj registara ostaje isti.

5. Koja je vrijednost registra R0 nakon sljedećeg odsječka programskog koda: R0= 80_{16}

```
        ORG 0
        LDRSH R0, A
        ORG 100
A       DH 80
```

6. Na procesoru ARM naredba ADD R0, R0, R1, LSL #2 računa sljedeću funkciju: R0=R0+R1*2

7. Na RTC je spojen signal frekvencije 10 Hz. Pri inicijalizaciji jedinice RTC upisuju se sljedeće vrijednosti u registre: RTCMR=100_{10}, RTCLR=20_{10}, RTCCR=$0F_{16}$, RTCSTAT=10_2. Nakon koliko će sekundi jedinica RTC generirati zahtjev za prekid: 8

U LR (load register) upisana je vrijednost 20. U MR (match register - registar usporedbe) upisana je vrijednost 100. Budući da je na RTC spojen signal frekvencije 10 Hz, to znači da će svake sekunde vrijednost brojača biti uvećana za 10. RTC će generirati prekid kad brojač dođe do 100, a počinje brojati od 20. Razlika je 80. Pošto svake sekunde se razlika smanji za 10, proteći će 8 sekundi i RTC će generirati prekid.

8. Na jedinicu RTC spojen je signal takta frekvencije 100 Hz. RTC treba generirati zahtjev za prekid svakih 7 sekundi. U koji od registara u jedinici RTC se upisuje pogrešna vrijednost pri inicijalizaciji: Svi registri su dobro inicijalizirani (RTCMR=700_{10}, RTCSTAT=1_{16}, RTCLR=0, RTCCR=7_{10}).

Brojilo počinje brojati od 0. Brojilo postavlja prekid kada brojač dođe do 700. Pošto se svake sekunde brojilo uveća za 100 (frekvencija je 100 Hz) za sedam sekundi će se generirati prekid. Na najmanje značajan bit registra RTCCR upisana je jedinica (zbog podatka 7 = 0111) znači da će generirati prekid. Ono što se piše u registar RTCSTAT se zanemaruje pošto on služi samo za čitanje spremnosti i dojavu prihvata prekida. Svi su, dakle, registri ispravno inicijalizirani.

9. Na procesoru ARM spojen je sklop GPIO. Nakon inicijalizacije (reseta) izvodi se sljedeći programski odsječak:

```
LDR R2, [R2]        ;navedena naredba učitava adresu GPIO sklopa
MOV R0, #2
STRB R0, [R2]
SWI 123456
```

Ovaj program utječe na sklop GPIO tako da: Ne mijenja stanje na vratima A, jer je smjer ulazni.

10. Za procesor ARM napisan je glavni program i program za obradu iznimke, čiji kod izgleda ovako:

```
        ORG 0
        B 100

        ORG 18
        B 0FFF          ;IRQ

        ORG 100
        MRS R0, CPSR
        BIC R0, R0, #80
        MSR CPSR_C, R0
PET1    BNE PET1
        SWI 123456

        ORG 0FFF
PET2    B PET2
        SWI 123456
```

Ako se po uključenju procesora dogodio niz iznimaka: RESET, IRQ, RESET, IRQ, izvođenje programa nakon određenog vremena zaustavit će se na adresi: PET2

Znači ako se prvo javi reset onda program skače na adresu 0. Od tamo opet skače na 100 kako je javljeno te kada se javi IRQ skoči na adresu 18 te zatim na FFF gdje se vrti, zatim bi se trebao opet javit reset te opet isponova sve što sam napisao i onda će ostati program na beskonačnoj petlji PET2.

11. Na procesoru ARM kad se dogodi RESET, u registar PC se smješta adresa: PC=00000000

12. Prije izvođenja naredbe LDR R0, [R1], R2, LSL #1 u registrima su bile vrijednosti $R1=100_{16}$ i $R2=1_{16}$. Nakon izvođenja naredbe LDR sadržaji registara bit će: $R0=mem(100_{16})$, $R1=102_{16}$, $R2=1_{16}$.

Jer su oko R1 registra zagrade, što označava da se registar mijenja nakon obrade (odnosno povećava se za dva u ovom slučaju zbog ovog drugog registra koji je pomnožen sa 2).

Učitaj u R0 sa adrese koja je u registru R1, zatim povećaj adresu R1 za R2*2 ti je kratki opis ovog odsječka

13. Na procesoru ARM izvodi se sljedeći program.

```
        ORG 0
        MOV R13, #10<8
        BL P1
L1      SWI 123456

P1      STR R14, [R13], #4
        BL P2
        LDR R14, [R13, #-4]!
        MOV PC, LR
L2      SWI 123456

P2      MOV PC, LR
```

Kako će on završiti: Procesor će izvesti naredbu SWI 123456 na labeli L1.

vrh "stoga" je na 10<8 => 1000 (16), to je manje od 1<16 =>10000 (16) na sto inace inicijaliziramo stog, tako da je ovo i dalje u okej rasponu memorijskom
P2 ne radi nista tako da mozemo zanemarit BL P2
pa je potprogram P1
P1 ; spremi na "stog"
; ucitaj isto to sa stoga
; vrati se natrag
s obzirom da je u kontekstu ocuvan registar r14 vraca se na naredbu nakon BL P1, sto je HALT na labeli L1

14. Na procesoru ARM izvodi se sljedeći programski odsječak:

```
PRG     MOV R0, #28
        LDMDB R0, {R1, R2}
        …
        ORG 20
        DW 102          ;adresa 20
        DW 103          ;adresa 24
        DW 104          ;adresa 28
```

Nakon izvođenja instrukcije LDMDB, sadržaji registara R0 i R1 bit će redom: 28 i 102

5. blic – Pitanja s materijala

1. Što će se upisati u R0?

```
    ADD R0, R0, R0, LSL #2
    ADD R0, R0, R0, LSL #1
```

Prvo dobijemo da je R0=R0+4*R0=5R0, pa onda R0=5R0 + 2*5R0=15*R0.

2. Koja naredba uzrokuje upravljački hazard na ARM-u? BL

3. Što radi LDRSB? Predznačno proširenje učitanog bajta.

4. Što je HADDR? 32-bitna adresna sabirnica, komponenta od AHB.

5. Na procesoru ARM se dogodi IRQ, u registar PC se smješta: PC=0x00000018.

6. Na koju sabirnicu je spojen GPIO na ARM-u: APB.

7. Što će se upisati u R14 i PC nakon RESET-a: R14= unpredictable_value, PC=0x00000000.

8. Koji je registar rezerviran za PC: R15.

9. Na koju sabirnicu se spaja memorija: AHB.

10. Pomoću koje naredbe možemo učitati više 32-bitnih podataka odjednom: LDM.

11. Na procesoru ARM kad se prihvati prekid (IRQ) registri PC i R14 poprimaju sljedeće vrijednosti: R14=adresa sljedeće naredbe + 4, PC=0x00000018.

1. Na RTC jedinicu spojen je signal takta frekvencije 100 Hz. RTC treba generirati zahtjev za prekid svakih 8 sekundi. U registre se upisuje RTCMR=900, RTCSTAT=8, RTCLR=100, RTCCR=8. U koji od registara u RTC jedinici se upisala pogrešna vrijednost pri inicijalizaciji: RTCCR.

Na RTCCR trebaš poslati 1 na najniži bit da omogućiš prekid.. 8=1000, dakle 0 je na najnižem bitu, pa je prekid onemogućen.

2. U sustavu procesora ARM oznaka HREADY opisuje: dio AHB sabirnice koji je širine 1 bit.

3. Naredba LDM kod procesora ARM: učitava jedan ili više 32-bitovnih podataka iz memorije.

4. Kod koje naredbe procesora ARM dolazi do pojave strukturnog hazarda: LDRH.

5. Nakon izvođenja sljedećeg programa u ATLAS-u, sadržaj registra R0 će biti: R0=1

```
        ORG 0
        SUBS R0, R0, R0
        BLEQ P
        SWI 123456
P       STMFD R13!, {R14, R0}
        ADD R0, R0, #1
        MOV PC, LR
```

6. Na procesoru ARM izvodi se programski odsječak, početne vrijednosti registara R6=8 i R5=2. Koja je od tvrdnji točna: ne znamo da li će se izvesti skok.

```
ADD R6, R6, R5
BHI KRAJ
```

1.Namjena navedenog programskog odsječka za ARM je: množenje R0 sa 15.

```
ADD R1, R0, R0, LSL #4
SUB R0, R1, R0, LSL #1
```

2. Na RTC je spojen signal frekvencije 100 Hz. RTC treba generirati zahtjev za prekid svakih 6 sekundi. U registre se upisuje RTCMR=600, RTCSTAT=6, RTCLR=0, RTCCR=6. U koji od registara u RTC-u je upisana pogrešna vrijednost pri inicijalizaciji: RTCCR

3. U odnosu na most, za sabirnicu PWDATA vrijedi: pripada sabirnici APB, širina je 32 bita, izlazi iz mosta.

4. GPIO sklop može postaviti zahtjev za prekid ARM-u samo onda kad je: GPIO nema mogućnost generiranja prekida (nema prekidni priključak).

5. Kada ARM prihvati brzi prekid FIQ, među ostalim se događa i ovo: PC se sprema u LR_fiq, a CPSR se sprema u SPSR_fiq.

6. Nakon što ARM izvede navedeni programski odsječak, u R0 će biti vrijednost: R0=2

```
MOVS R0, #1
ADDNE R0, R0, #1
```

7. Kako će završiti sljedeći program simuliran u ATLAS-u: zavrtit će se u beskonačnoj petlji na labeli L5

```
      ORG
L1    BL L
L2    HALT
L     ORRS R3, R3, #88
L3    MOVEQ PC, LR
L4    BL L5
L5    MOV PC, LR
L6    HALT
```

8. Koja je vrijednost u R0 nakon izvođenja ovog odsječka: R0=0x FFFF FF90

```
      ORG
      LDRSB R0, A
      HALT
      ORG 100
A     DH 3090
```

9. Nakon uključenja ARM izvodi ovaj programski odsječak. Koliko perioda traje njegovo izvođenje: 6

```
        ORG 0
        CMP R0, R0
        BHI Y
        ADD R0, R0, R0
Y       EOR R3, R4, R5
```

Svaka ti naredba traje 1 period (uvjet se ne izvršava jer R0 nije veće od R0) i na kraju dodaš +2 radi protočne strukture (to uvijek ide).

Org ne uzima ciklus, za skok je 3 kad je uvjet zadovoljen jer tada se neće izvršavati sljedeća učitana naredba nego mora skočiti na adresu skoka i tamo učitati novu naredbu i time gubi još 2 perioda.

EQ provjerava zastavicu Z, i to bi vrijedilo samo ako je R0 imao vrijednost 0 sto ne mozes znati iz isjecka.
A inace da vrijedi bi bilo: cmp 1, beq 3, eor 1 i na kraju +2 = 7.

Hazardi i koje ih naredbe izazivaju:
STRUKTURNI HAZARD: STR, STM, LDR i LDM
UPRAVLJAČKI HAZARDI: B i BL
PODATKOVNI: oni su u ARM9

Trajanje perioda pojedinih naredbi:
LDR - 3
LDM - n+2
STR - 2
STM - n+1

B i BL - ukoliko je uvjet zadovoljen - 3 perioda, inače 1

Ostale traju 1 period

Te napomena, zbog protočne strukture ARM-a prva naredba traje 3 perioda.

Pitanja s fer1 programa (nezz za točnost)
1.Kako će završiti sljedeći program: Procesor neće izvesti niti jednu HALT naredbu.

```
        ORG 0
        MOV R13, #10<8
        BL P1
L1      HALT
P1      STR R14, [R13], #4
        BL P2
        MOV PC, LR
L2      HALT
P2      MOV PC, LR
```

2. Sadržaj registara R1 i R2 prije izvođenja instrukcije LDR R0, [R1, R2, LSL #1] je R1=10016 i R2=116. Nakon izvođenja instrukcije sadržaj registara je: R0=mem(102_{16}), R1=100_{16}, R2=1_{16}.

3. Koja je od navedenih tvrdnji točna za procesor ARM: Kod ugnježđenih poziva potprograma potrebno je spremiti registar R14 na stog.

4. Na procesoru ARM izvodi se sljedeći program:

```
        ORG 0
        BL LAB1
        MOV R1, #0
LAB1    BL LAB2
        MOV PC, LR
LAB2    MOV PC, LR
```

Koja je tvrdnja točna: Naredba za povratak uz potprograma LAB1 pozivat će samu sebe.

Prvi put kad se pozove LAB1 u LR registar se sprema povratna adresa iz LAB1. Prilikom poziva LAB2 u isti registar sprema se povratna adresa iz LAB2 preko adrese koja bila unutra zapisana. Prilikom povratka iz LAB2, program ce skociti na naredbu poslije BL LAB2, a to je opet MOV PC, LR.. ali unutra je jos uvijek povratna adresa za LAB2 (ona prije je prepisana) te opet skace na isto mjesto i opet poziva MOV PC, LR.. i tako u beskonacnost

5. Razlika između B i BL instrukcije je u sljedećem: BL instrukcija sprema povratnu adresu u R14.

6. Procesor ARM ima ukupno: 37 registara.

7. Nakon izvođenja naredbe LDR R0, [R1, R2] sadržaj registara je sljedeći: R0=mem(R1+R2), R1=R1

8. Prilikom izvođenja instrukcije BL ne mijenja se sadržaj registra: R13.

9. Koji od navedenih registara nije sastavni dio GPIO sklopa na ARM-u: GPIOCCDR.

10. GPIO sklop za procesor ARM zauzima: 4 memorijske lokacije.

11. Za povratak iz FIQ prekida treba izvesti sljedeću naredbu: SUBS PC, R14, #4.

12. Vrata GPIO sklopa na procesoru ARM su: 8-bitna.

13. Koja od navedenih tvrdnji za procesor ARM nije istinita: Nije moguće izvesti povratak u program iz iznimke.

14. U sabirničkom periodu tipa S (S-period) događa se sljedeće: Procesor želi prenijeti blok podataka.

15. Koja od navedenih tvrdnji za procesor ARM je istinita: GPIO je sklop koji omogućava čitanje i pisanje podataka s vanjskih ulaza.

16. U sabirničkom periodu N (N-period) događa se sljedeće: Procesor zahtjeva prijenos podataka sa adrese koja nije u svezi sa adresom koja se koristila u prethodnom periodu.

17. Na procesoru ARM, GPIO se s procesorom spaja preko: AMBA APB sabirnice.

18. U AMBA sabirničkom sustavu, povezivanje sporijih vanjskih uređaja u sustav ostvaruje se sabirnicom: APB.

19. Kada procesor ARM dohvati nepostojeću instrukciju, on će prijeći u način rada: Undefined.

20. Koja od navedenih komponenti ne spada u sastavni dio RTC jedinice: Sve navedene jedinice su sastavni dio RTC jedinice (brojilo, sklop za usporedbu, registar usporedbe, AMBA APB sučelje).

21. Koja od sljedećih komponenti nije sastavni dio RTC jedinice: AMBA AHB sučelje.

22. Na RTC jedinicu spojen je signal takta frekvencije 10 Hz. RTC treba generirati zahtjev za prekid svakih 5 sekundi. Koji od registara u RTC jedinici je pogrešno inicijaliziran: svi registri su dobro inicijalizirani (u RTCCR se upisuje 1, u RTCLR se upisuje 0, u RTCSTAT se upisuje 1, u RTCMR = 50).

23. Na RTC je spojen signal frekvencije 1 Hz. Sadržaj registara u RTC jedinici je RTCMR=20, RTCLR=10, RTCCR=1. Nakon koliko će sekundi RTC jedinica generirati zahtjev za prekid: 10.

24. Nakon što ARM izvede navedeni programski odsječak, u R0 će biti vrijednost: R0=2

```
MOVS R0, #1
ADDNE R0, R0, #1
```

Piše u šaliću da MOVS postavlja zastavice N, Z i C. NE ispituje stanje zastavice !Z, dakle ako je !Z=1 izvrši naredbu. Kako je MOVS postavio zastavicu Z=0 onda je uvjet valjan i naredba se izvršava.