

2.4.16. Kvadrat 16-bitnog broja u formatu 2'k

Rješenje:

SQR

| | |
|----------------------|---|
| STMFD R13!, {R0, R1} | ;spremanje konteksta |
| LDRSH R0, [R13, #8] | ;ucitavanje broja i predznacno proširivanje |
| LDR R1, [R13, #0C] | ;ucitavanje adrese |
| MUL R0, R0, R0 | ;racunanje kvadrata |
| STR R0, [R1] | ;spremanje rezultata |
| LDMFD R13!, {R0, R1} | ;obnova konteksta |
| ADD R13, R13, #8 | ;brisanje argumenata |
| MOV PC, LR | ;povratak |

2.4.17. (FRISC 2.5.12.) Dijeljene uzastopnim oduzimanjem i rastavljanje na proste faktore

Rješenje:

DIV

| | |
|------------------|----------------------|
| STMFD R13!, {R2} | ;spremanje konteksta |
| MVN R2, #0 | ;R2 = -1 , FFFFFFFF |

LOOP

| | |
|-----------------|----------------------|
| ADD R2, R2, #1 | ;racunanje rezultata |
| SUBS R0, R0, R1 | |
| BPL LOOP | |

| | |
|----------------|--------------|
| ADD R1, R0, R1 | ;R1 ostatak |
| MOV R0, R2 | ;R0 rezultat |

| | |
|------------------|-------------------|
| LDMFD R13!, {R2} | ;obnova konteksta |
| MOV PC, LR | |

FACT

| | |
|--------------------------------------|---|
| STMFD R13!, {R0, R1, R2, R3, R4, LR} | ;spremanje konteksta i LR jer se poziva |
| MOV R2, #2 | ;DIV, R2 djelitelj |

PETLJ

| | |
|------------|-----------------|
| MOV R3, R0 | ;R3 cuva broj |
| MOV R4, R1 | ;R4 cuva adresu |
| MOV R1, R2 | ;R1 djelitelj |
| BL DIV | ;podjeli |

| | |
|------------|-----------------------------|
| CMP R0, #0 | ;ako je gotovo onda na kraj |
| BEQ VAN | |

NEDIJELI

| | |
|--------------|-----------------------------------|
| CMP R1, #0 | ;ako je ostatak 0 onda je djeljiv |
| MOVNE R0, R3 | ;ako nije vrati broj |
| MOVNE R1, R4 | ;vrati adresu |

| | | |
|--------|--------------------------------------|--------------------------------|
| | ADDNE R2, R2, #1 | ;povecaj djelitelj |
| | BNE PETLJ | |
| DIJELI | MOVEQ R1, R4 | ;ako je, vrati adresu |
| | STREQ R2, [R1], #4 | ;spremi i povecaj adresu |
| | BEQ PETLJ | |
| VAN | STR R0, [R1] | ;na kraju zakljucajmo blok s 0 |
| | LDMFD R13!, {R0, R1, R2, R3, R4, LR} | ;obnova konteksta |
| | MOV PC, LR | |

2.4.18. (FRISC 2.5.13.) Oduzimanje i dijeljene brojeva u dvostrukoj preciznosti

Rješenje:

| | | |
|---------|------------------------------|--|
| | ORG 0 | |
| MAIN | MOV SP, #1<16 | ;inicijalizacija stoga |
| | MOV R0, #10<8 | ;1000 = adresa prvog |
| | ADD R1, R0, #8 | ;1008 = adresa drugog |
| | ADD R2, R1, #8 | ;1010 = adresa rezultata |
| | BL DIV_DBL | |
| | HALT | |
| DIV_DBL | STMFD R13!, {R3, R4, R5 LR} | ;spremanje konteksta |
| | MVN R3, #0 | ;R3 = -1, donji dio rezultata |
| | MOV R5, #0 | ;R5 gornji diorezultata |
| LOOP | ADDS R3, R3, #1 | |
| | ADC R5, R5, #0 | ;ako je doslo do preljeva dodaj u gornji dio |
| | BL SUB_DBL | |
| | LDR R4, [R2, #4] | ;ucitaj gornji dio rezultata oduzimanja |
| | MOVS R4, R4, LSL #1 | ;stavi najvisi bit u C |
| | BCC LOOP | ;ako C=0 rez je pozitivan , ponovo petlja |
| | STR R3, [R2] | ;spremi donji dio |
| | STR R5, [R2, #4] | ;spremi gornji dio |
| | LDMFD R13!, {R3, R4, R5, LR} | ;obnova konteksta |
| | MOV PC, LR | |
| SUB_DBL | STMFD R13!, {R3, R4, R5, R6} | ;spremanje konteksta |
| | LDMIA R0, {R3, R4} | ;R3 nizi dio prvog, R4 viši dio prvog |
| | LDMIA R1, {R5, R6} | ;R5 nizi dio drugog, R6 viši dio drugog |

| | | |
|-------|------------------------------|---|
| | SUBS R3, R3, R5 | ;oduzmi niži dio |
| | SBC R4, R4, R6 | ;oduzmi viši s preljevom |
| | STMIA R2, {R3, R4} | ;spremi rezultat |
| | LDMFD R13!, {R3, R4, R5, R6} | ;obnova konteksta |
| | MOV PC, LR | |
| PRVI | `ORG 1000 | |
| DRUGI | DW 10FA, 02BB | ;nizi prvog (1000), viši prvog (1004) |
| REZ | DW 0FF10, FFFF | ;nizi drugog (1008), viši drugog (100C) |
| | `DS 8 | ;niži rez (1010}, viši rez (1014) |

2.4.19. (FRISC 2.5.14.) Potprogram za izračun funkcije sa četiri parametra

Rješenje:

| | | |
|------|--------------------------|--|
| MAIN | MOV SP, #1<16 | |
| | MOV R6, #1<12 | ;R6 = adresa |
| LOOP | LDR R0, [R6] | ;ucitaj podatak |
| | BL FUNC | ;pozovi FUNC |
| | STR R0, [R6], #4 | ;spremi, zatim uvecaj adresu |
| | CMP R6, #2<12 | ;provjera kraja |
| | BNE LOOP | |
| | HALT | |
| FUNC | STMFD R13!, {R1, R2, R4} | |
| | MOV R4, #0 | ;R4 brojac za petlju |
| | MOV R2, #0 | ;R2 = y |
| PET | CMP R4, #4 | |
| | BEQ VAN | |
| | AND R1, R0, #0FF | ;najnizi bajt |
| | MOV R0, R0, LSR #8 | ;pomak broja |
| | MOV R1, R1, LSL #24 | ;svi pomaci i rotacije su u DEKADSKOJ bazi |
| | MOV R1, R1, ARS #24 | ;predznacno prosirivanje |
| | MOV R1, R1, LSL R4 | ;množenje s potencijom broja 2 |
| | ADD R2, R2, R1 | ;dodavanje rezultatu |
| | ADD R4, R4, #1 | ;povecanje potencije |
| | B PET | |
| VAN | MOV R0, R2 | ;rezultat je u R0 |
| | LDMFD R13!, {R1, R2, R4} | |
| | MOV PC, LR | |

2.4.20. (FRSC 2.5.15.) Zrcalni bitovi i provjera palindroma

Rješenje:

MAIN

```
MOV SP, #1<16
MOV R0, #5<8      ;R0 = 500
MOV R1, #2<8      ; R1 = %D512
STMFD R13!, {R0, R1}
BL PALIN
ADD SP, SP, #8     ;brisanje argumenata sa stoga
MOV R1, #45<4     ;R1 = 450
STR R0, [R1]
HALT
```

PALIN

```
STMFD R13!, {R1, R2, R3, R4, LR}
MOV R4, #0
LDR R2, [SP, #0D20] ;R2 = broj podataka
LDR R1, [SP, #0D24] ;R1 = adresa
```

LOOP

```
LDR R0, [R1], #4    ;ucitaj i zatim povecaj adresu
MOV R3, R0          ;spremi broj u R3
```

BL INVERZ

CMP R3, R0

JESU

```
ADDEQ R4, R4, #1    ;ako su jednaki povecaj brojac palindroma
```

NISU

```
SUBS R2, R2, #1
```

BNE LOOP

MOV R0, R4

LDMFD SP!, {R1, R2, R3, R4, LR}

MOV PC, LR

INVERZ

STMFD SP!, {R1, R2, R3}

MOV R3, #0D32

MOV R2, #0

PETLJ

AND R1, R0, #1

;najnizi bit u R0

MOV R0, R0, LSR #1

;pomici broj u desno

ORR R2, R2, R1

;stavi taj bit u R2

MOV R2, R2, LSL #1

;pomici R2 u lijevo

SUBS R3, R3, #1

;i tako 32 puta

BNE PETLJ

MOV R0, R2

;rezultat u R0

LDMFD SP!, {R1, R2, R3}

MOV PC, LR

2.4.21. (FRISC 2.5.16.) Potprogram za pretvorbu iz zapisa BCD u NBC

```
MAIN
    MOV SP, #1<16
    MOV R6, #2<12           ;R6 = 2000
XYZ
    LDR R0, [R6]             ;ucitaj podatak
    BL BCD2NBC
    STR R0, [R6], #4         ;spremi i uvecaj adresu
    CMP R6, #25<8           ;provjera kraja
    BNE XYZ
    HALT

BCD2NBC
    STMFD SP!, {R1, R2, R3}
    MOV R3, #4               ;R3 brojac petlje
    MOV R2, #0               ;R2 rezultat
LOOP
    MOV R0, R0, ROR #24      ;rotacija u lijevo za 8, najvisi bajt na najnize mjesto
    AND R1, R0, #0FF         ;taj bajt u R1
    ADD R2, R2, R1            ;dodaj u R2
    MOV R1, R2, LSL #3       ;pomnozi s 8
    ADD R1, R1, R2            ;dodaj R2 = *9
    ADD R2, R2, R1            ;dodaj R2 = *10
    SUBS R3, R3, #1          ;i tako 4 puta
    BNE LOOP

    MOV R0, R2               ;rezultat u R0
    LDMFD SP!, {R1, R2, R3}
    MOV PC, LR
```

2.4.22. (FRISC 2.5.17.) Brojenje pojavljivanja 2-bitnog uzorka u broju

Rješenje:

```
MAIN
    MOV SP, #1<16           ;inicijalizacija stoga
    MOV R3, #1<12           ;R3 = 1000
    MOV R4, #15<8           ;R4 = 1500
XYZ
    LDR R0, [R3], #4         ;ucitaj broj
    MVN R1, #0               ;R1=-1
PETLJ
    ADD R1, R1, #1           ;R1 = 00, pa 01, pa 10, pa 11 i
    BL CNT2BIT
    STRB R2, [R4], #1        ;spremnje rezultata
    CMP R1, #%B11           ;ako je obrađen zadnji uzorak (11) idi na sljedeci broj
    BNE PETLJ

    CMP R3, #12<8           ;jesmo li dosli do kraja bloka
```

| | | |
|---------|-------------------------|---------------------------------------|
| | MVNNE R1, #0 | ;ako nismo resetiraj uzorak |
| | BNE XYZ | ;i kreni ponovo |
| CNT2BIT | STMFD SP!, {R0, R3, R5} | |
| | MOV R3, #D31 | ;brojac petlje |
| | MOV R2, #0 | ;brojac pojavljivanja |
| LOOP | AND R5, R0, #B11 | ;dva najniza bita |
| | CMP R5, R1 | |
| JESU | | ;ako su jednaki uzorku povecaj brojac |
| | ADDEQ R2, R2, #1 | |
| NISU | | |
| | MOV R0, R0, LSR #1 | ;pimici broj za 1 u desno |
| | SUBS R3, R3, #1 | ;i tako 31 put |
| | BNE LOOP | |
| | LDMFD SP!, {R0, R3, R5} | |
| | MOV PC, LR | |

2.4.23. (FRISC 2.5.18.) pronalazak podniza u nizu bitova

Rješenje:

| | | |
|------|-----------------------------|--|
| MAIN | MOV SP, #1<16 | ;inicijalizacija stoga |
| | MOV R6, #5<8 | ;R6 = 500 |
| | MOV R1, #0A1 | ;R1 = A1 |
| XYZ | LDR R0, [R6], #4 | ;ucitaj podatak i zatim povecaj adresu |
| | BL SUBS | |
| | CMP R0, 0 | ;ako nema podnica onda je R0 = 0 |
| NEMA | | |
| | MOVEQ R0, #0 | |
| | STREQ R0, [R6, #-4] | ;ako nema zamjeni broj nulom |
| IMA | | |
| | CMP R6, #1<12 | ;provjera kraja |
| | BNE XYZ | |
| | HALT | |
| SUBS | | |
| | STMFD SP!, {R2, R3, R4, R5} | |
| | MOV R3, #D24 | ;brojac pomaka |
| | MOV R2, #1 | ;šetajuca jedinica |
| | MOVR4, #0 | ;rezultat |
| LOOP | | |
| | AND R5, R0, #0FF | ;izdvojimo najnizi bajt |
| | CMP R5, R1 | ;usporedimo s podnizom |
| JESU | | ;ako je pronasao podniz |
| | ORREQ R4, R4, R2 | ;stavlja 1 na mjesto pronalaska |

| | | |
|------|-----------------------------|----------------------------------|
| NISU | MOVEQ R4, R4, ROR #26 | ;rotiramo u desno za 26 |
| | ADDEQ R4, R4, #1 | ;povecemo brojac pojavljivanja |
| | MOVEQ R4, R4, ROR #6 | ;rotiramo u lijevo za 26 |
| | MOV R0, R0, LSR #1 | ;pomicemo broj za 1 desno |
| | MOV R2, R2, LSL #1 | ;pomicemo jedinicu za 1 u lijevo |
| | SUBS R3, R3, #1 | ;i tako 24 puta |
| | BNE LOOP | |
| | MOV R0, R4 | |
| | LDMFD SP!, {R2, R3, R4, R5} | |
| | MOV PC, LR | |

2.4.24. (FRISC 2.5.19.) Zbrajanje brojeva u zapisu nepomičnog zareza

Rješenje:

| | | |
|---------|-----------------------------|-----------------------------------|
| MAIN | MOV SP, #1<16 | ;inicijalizacija stoga |
| | MOV R6, #6<8 | ;R6 adresa podataka |
| | LDRB R1, [R6] | ;čitaj i stavi na stog prvi |
| | STMFD SP!, {R1} | |
| | LDRB R1, [R6, #1] | ;drugi |
| | STMFD SP!, {R1} | |
| | BL ZBR_PLD | |
| | ADD SP, SP, #8 | ;ukloni |
| | LDRB R1, [R6, #2] | ;ucitaj treci |
| | STMFD SP!, {R0, R1} | ;stavi zbroj prva dva i treci |
| | BL ZBR_PLD | |
| | ADD SP, SP, #8 | ;ukloni |
| | MOV R1, #5<8 | ;adresa gdje treba zapisati |
| | STR R0, [R1] | ;zapiši |
| | HALT | |
| ZBR_PLD | STMFD SP!, {R1, R2, R3, R4} | |
| | LDR R1, [SP, #0D16] | ;ucitavanje operandi |
| | LDR R3, [SP, #0D20] | |
| | AND R2, R1, #0F | ;R2 decimalni dio |
| | MOV R1, R1, LSR #4 | ;R1 nedecimalni dio |
| | AND R4, R3, #0F | ;isto kao gore samo drugi operand |
| | MOV R3, R3, LSR #4 | |
| | ADD R2, R2, R4 | ;zbroji decimalni dio |
| | AND R4, R2, #10 | ;provjeri ima li preljeva |
| | IMA_C | |

| | | |
|--------|-----------------------------|-------------------------------------|
| NEMA_C | ADDNE R1, R1, #1 | ;ako ima dodaj 1 na nedecimalni dio |
| | ANDNE R2, R2, #0F | ;i makni preljev sa decimalnog |
| | ADD R1, R1, R3 | ;zbroji nedecimalni dio |
| | MOV R1, R1, LSL #4 | ;pomakni ga za 4 u lijevo |
| | OR R0, R1, R2 | ;dodaj decimalni dio |
| | LDMFD SP!, {R1, R2, R3, R4} | |
| | MOV PC, LR | |

2.4.25. (FRISC 2.5.20.) Izračun cjelobrojnog dijela broja u zapisu IEEE

Rješenje:

| | | |
|----------|---------------------------------|--|
| IEEE_INT | STMFD SP!, {R0, R2, R3, R4, R5} | |
| | MOV R2, R0, ROR #31 | |
| | AND R2, R2, #1 | ;R2 predznak |
| | MOV R3, R0, LSR #23 | |
| | AND R3, R3, #0FF | |
| | SUB R3, R3, #0D127 | ;R3 eksponent |
| | MOV R0, R0, LSL #9 | ;R0 decimaln dio |
| | CMP R3, #0 | |
| MINUS | MOVMI R1, #0 | ;ako je eksponent manji od 0, cjelobrojni dio je 0 |
| | BMI VAN | |
| NULA | MOVEQ R1, #1 | ;ako je eksponent 0, cjelobrojni dio je 1 |
| | BEQ VAN | |
| PLUS | MOV R4, #1 | |
| | MOV R4, R4, ROR #1 | |
| | MOV R1, #1 | |
| LOOP | MOV R1, R1, ROR #31 | |
| | MOV R4, R4, ASR #1 | |
| | SUBS R5, R3, #1 | |
| | BNE LOOP | |
| | MOV R4, R4, LSL #1 | |
| | AND R0, R0, R4 | ;decimalni dio koji postaje cjelobrojan |
| | RSB R4, R3, #0D32 | ;R4 broj pomaka u desno da se dio stavi na pocetak |
| | MOV R0, R0, LSR R4 | |
| | ORR R1, R0, R1 | ;dodavanje i one jedinice |
| PREDZNAK | | |

| | |
|-------|---------------------------------|
| | CMP R2, #0 |
| MINUS | |
| | RSB R1, R1, #0 |
| PLUS | |
| | LDMFD SP!, {R0, R2, R3, R4, R5} |

3.1.13. Očitavanje i ispisivanje temperature

Rješenje:

| | | |
|-----------|-------------------|--|
| | `ORG 0 | |
| | B MAIN | |
| LCDWR | STMFD SP!, {R4} | |
| | AND R4, R4, #7F | ;spusti najvisi bit na nula |
| | STRB R4, [R2] | ;posalji na vrata A |
| | ORR R4, R4, #80 | ;postavi na 1 |
| | STRB R4, [R2] | |
| | AND R4, R4, #7F | ;spusti na 0 |
| | STRB R4, [R2] | |
| | LDMFD SP!, {R4} | |
| | MOV PC, LR | |
| | `ORG 100 | |
| GPIO | DW 0FFFF0000 | ;adresa GPIO sklopa |
| MAIN | | |
| | MOV SP, #1<16 | |
| | MOV R2, #1<8 | |
| | LDR R2, [R2] | ;u R2 je adresa GPIO sklopa |
| INIT_GPIO | | |
| | MOV R0, #0FF | ;vrata A svi bitovi su izlazni |
| | STR R0, [R2, #8] | |
| | MOV R0, #7F | ;vrata B: najvisi izlazni, ostali ulazni |
| | STR R0, [R2, #0C] | |
| CITAJ | | |
| | LDR R0, [R2, #4] | |
| | ANDS R1, R0, #40 | ;bit 6 je signal nove temp |
| | BEQ CITAJ | |
| PISI | | |
| | MOV R1, #80 | |
| | STR R1, [R2, #4] | ;bit 7 u visoko |
| | MOV R1, #0 | |
| | STR R1, [R2, #4] | ;bit 7 u nisko |
| | MOV R4, #0D | |
| | BL LCDWR | ;ocistimo LCD |
| | AND R0, R0, #3F | ;izdvojimo temp u donjih 6 bita |
| | CMP R0, #0D35 | ;15 stupnjeva |
| | MOVLO R4, #48 | ;ako je manje onda H |

| | |
|----------------|--|
| CMP R0, #0xD45 | ;25 stupnjeva |
| MOVHI R4, #56 | ;ako je vise onda V |
| BL LCDWR | ;upisimo znak koji treba, ako je izmđu samo ce ponovo obrisati |
| MOV R4, #0A | ;upis znaka za ispis na LCD |
| BL LCDWR | |
| B CITAJ | |

3.1.14. Rampa za parkiralište

Rješenje:

| | | |
|----------|----------------------|--|
| | `ORG 0 | |
| | B MAIN | |
| | `ORG 1C | ;odabiremo brzi prekid |
| FIQ | LDR R8, RTC | ;u R8 adresa RTC |
| | MOV R9, #0 | |
| | STR R9, [R8, #8] | ;prihvat prekida |
| | STR R9, [R8, #0C] | ;reset brojaca |
| | STR R9, KRAJ | ;proslo je 5 sekundi |
| | SUBS PC, LR, #4 | ;povratak iz prekida |
| GPIO | DW 0FFFF0000 | ;adresa GPIO sklopa |
| RTC | DW 0FFFF1000 | ;adresa RTC sklopa |
| KRAJ | DW 1 | ;oznaka prolaska 5 sekundi |
| MAIN | | |
| | LDR R1, GPIO | ;R1 adresa GPIO |
| | LDR R2, RTC | ;R2 adresa RTC |
| INITGPIO | | ;na vratima A su senzori, na 2. bitu za dolazak, na 1. |
| | MOV R0, #1 | ;za odlazak , a kontrola rampe je na 0., dakle 0. je |
| | STR R0, [R1, #8] | ;izlazni ostali ulazni |
| WDOLAZAK | | ;cekanje da dode automobil |
| | LDR R0, [R1] | |
| | ANDS R0, R0, #0xB100 | |
| | BEQ WDOLAZAK | |
| DOSO | | ;kada dođe podignemo rampu |
| | MOV R0, #1 | |
| | STR R0, [R1] | |
| INIT RTC | | ;inicijalizacija RTC |
| | MOV R0, #5 | ;5 sekundi, frekvencija je 1Hz |
| | STR R0, [R2, #4] | |
| | MOV R0, #1 | ;omogucavanje postavljanja prekida u RTC |
| | STR R0, [R2, #10] | |
| | MRS R0, SPSR | ;omogucavanje FIQ u SPSR |
| | BIC R0, R0, #40 | |

MSR SPSR_c, R0

LOOP

LDR R0, KRAJ ;cekanje prolaska 5 sekundi
CMP R0, 0
BNE LOOP

WODLAZAK

;cekanje odlaska nakon proteka 5 sekundi

LDR R0, [R1]
ANDS R0, R0, #%B010
BEQ WODLAZAK

OTISO

MOV R0, #0 ;spuštanje rampe
STR R0, [R1]
MOV R0, #1 ;ponovno postavljanje KRAJ
STR R0, KRAJ
B WODLAZAK ;cekanje dolaska sljedeceg automobila