ARM

1. Na displayu ispistati brojanje od 0 do 9

.global \_start

\_start:

 ldr r0, =DISP

 mov r1, #0 // Counter

 ldr r2, =nums

\_loop:

 ldrb r3, [r2, r1]

 str r3,[r0]

 add r1, r1, #1

 // Check if the counter is 10

 cmp r1, #11

 beq \_counter\_reset

\_delay:

 add r5,r5,#1

 cmp r5,#204800

 bne \_delay

 mov r5,#0

 b \_loop

\_counter\_reset:

 mov r1, #0

 b \_loop

.data

.equ DISP, 0xff200020

// To get these numbers, search online for 7 segment display codes

nums:

.byte 0x3f // 0

.byte 0x06 // 1

.byte 0x5b // 2

.byte 0x4f // 3

.byte 0x66 // 4

.byte 0x6d // 5

.byte 0x7d // 6

.byte 0x07 // 7

.byte 0x7f // 8

.byte 0x6f // 9

2. Brojanje u nazad (od 9 do 0)

.global \_start

\_start:

 ldr r0, =DISP

 mov r1, #9 // Counter

 ldr r2, =nums

\_loop:

 ldrb r3, [r2, r1]

 str r3, [r0]

 // Check if the counter is negative

 cmp r1, #-1

 beq \_counter\_reset

 sub r1, r1, #1

\_delay:

 add r5, r5, #1

 cmp r5, #204800

 bne \_delay

 mov r5, #0

 b \_loop

\_counter\_reset:

 mov r1, #9

 b \_loop

.data

.equ DISP, 0xff200020

// To get these numbers, search online for 7 segment display codes

nums:

.byte 0x3f // 0

.byte 0x06 // 1

.byte 0x5b // 2

.byte 0x4f // 3

.byte 0x66 // 4

.byte 0x6d // 5

.byte 0x7d // 6

.byte 0x07 // 7

.byte 0x7f // 8

.byte 0x6f // 9

3. Brojanje svakog drugog u bilo kojem rasponu

.global \_start

\_start:

 ldr r0, =DISP

 mov r1, #0 // Counter

 ldr r2, =nums

\_loop:

 ldrb r3, [r2, r1]

 str r3,[r0]

 // Check if the counter is 11

 cmp r1, #11

 // Branch Greater Than number

 bgt \_counter\_reset

 add r1, r1, #2

\_delay:

 add r5,r5,#1

 cmp r5,#204800

 bne \_delay

 mov r5,#0

 b \_loop

\_counter\_reset:

 mov r1, #0

 b \_loop

.data

.equ DISP, 0xff200020

// To get these numbers, search online for 7 segment display codes

nums:

.byte 0x3f // 0

.byte 0x06 // 1

.byte 0x5b // 2

.byte 0x4f // 3

.byte 0x66 // 4

.byte 0x6d // 5

.byte 0x7d // 6

.byte 0x07 // 7

.byte 0x7f // 8

.byte 0x6f // 9

4. Brojanje parnih u bilo kojem rasponu

Isto kao i rjesenje treceg zadatka

5. Brojanje neparnih u bilo kojem rasponu

Isto kao i treci/cetvrti, samo sto treba promjeniti da se u r1 registar pri resetiranju

brojaca i na pocetku programa stavi vrijednost #1

6. Brojanje na nekoj od mogućih pozicija, npr. brojanje na 3. displayu

.global \_start

\_start:

 ldr r0, =DISP

 mov r1, #0 // Counter

 ldr r2, =nums

 mov r7, #16 // Display position (N/8)

 // 0 is pos 0, 8 is pos 1, 16 is pos 2

 // and if we want to access 4/5/6/7, we

 // have to store it in the second byte of

 // The display

 mov r5, #0 // Delay counter

\_loop:

 ldrb r3, [r2, r1]

 mov r6, r3

 mov r6, r6, lsl r7

 str r6, [r0]

 // str r3, [r0]

 add r1, r1, #1

 // Check if the counter is 10

 cmp r1, #11

 beq \_counter\_reset

\_delay:

 add r5, r5, #1

 cmp r5, #204800

 bne \_delay

 mov r5, #0

 b \_loop

\_counter\_reset:

 mov r1, #0

 b \_loop

.data

.equ DISP, 0xff200020

// To get these numbers, search online for 7 segment display codes

nums:

.byte 0x3f // 0

.byte 0x06 // 1

.byte 0x5b // 2

.byte 0x4f // 3

.byte 0x66 // 4

.byte 0x6d // 5

.byte 0x7d // 6

.byte 0x07 // 7

.byte 0x7f // 8

.byte 0x6f // 9

6. Brojanje parnih brojeva

.equ SEGMENT\_ADR, 0xFF200020

.equ DELAY\_N, 2000000

.global \_start

\_start:

 ldr r3, =SEGMENT\_ADR

 mov r0, #0

loop:

 ldr r1, =table

 ldr r2, [r1, r0, LSL #2]

 str r2, [r3, #0]

 ldr r4, =DELAY\_N

delay:

 subs r4, r4, #1

 bne delay

 add r0, r0, #2

 cmp r0, #0x08

 ble loop

exit:

 b exit

.data

table:

 .word 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x4F

ZA NEPARNE SAMO PROMIJENIT KOMBINACIJE DOLE

7. Koristeći sedam segmentni display izbrojite od 1-9 ali na drugom mjestu zdesna uz dodanu kratku petlju koja će omogućiti da između brojki postoji vremenski razmak. (hint: LSL)

.equ SEGMENT\_ADR, 0xFF200020

.equ DELAY\_N, 2000000

.global \_start

\_start:

 ldr r3, =SEGMENT\_ADR

 mov r0, #1

loop:

 ldr r1, =table

 ldr r2, [r1, r0, LSL #2]

 lsl r2, r2, #8

 str r2, [r3, #0]

 ldr r4, =DELAY\_N

delay:

 subs r4, r4, #1

 bne delay

 add r0, r0, #1

 cmp r0, #9

 ble loop

exit:

 b exit

.data

table:

 .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F

8. Koristeći sedam segmentni display izbrojite od 9-1 ali na drugom mjestu zdesna uz dodanu kratku petlju koja će omogućiti da između brojki postoji vremenski razmak. (hint: LSL)

.equ SEGMENT\_ADR, 0xFF200020

.equ DELAY\_N, 2000000

.global \_start

\_start:

 ldr r3, =SEGMENT\_ADR

 mov r0, #9

loop:

 ldr r1, =table

 ldr r2, [r1, r0, LSL #2]

 lsl r2, r2, #8

 str r2, [r3, #0]

 ldr r4, =DELAY\_N

delay:

 subs r4, r4, #1

 bne delay

 sub r0, r0, #1

 cmp r0, #0

 bge loop

exit:

 b exit

.data

table:

 .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F