Aluno: Roberta Boaventura Andrade

NUSP: 11260832

Laboratório de Processadores – PCS3432

Planejamento do Experimento 7

Foi solicitado, para o planejamento do experimento, que fossem feitos rascunhos relativos às soluções dos exercícios do capítulo 7 do “ARM Lab Manual”. Com base nisso, foram desenvolvidos os seguintes códigos, para que pudessem ser testados e aprimorados no laboratório, durante a aula:

7.5.1

```
@ Exercicio 7.5.1

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008

    .text
    .global main

main:
    LDR     r0, =0x3FF5000      @ IOPMOD
    LDR     r1, =0xF0           @ Define leds como output
    STR     r1, [r0]

    B       ascending
    B       descending
    BAL     fim

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0

ascending:
    LDR     r0, =0x3FF5008      @ IOPDATA
    MOV     r1, #0

ascending_loop:
    MOV     r1, r1, LSL #4
    STR     r1, [r0]
    MOV     r1, r1, LSR #4
    BL      delay
    ADD     r1, r1, #1
    CMP     r1, #16
    MOVGE   r1, #0
    B       ascending_loop

descending:
    LDR     r0, =0x3FF5008      @ IOPDATA
    MOV     r1, #15

descending_loop:
    MOV     r1, r1, LSL #4
    STR     r1, [r0]
    MOV     r1, r1, LSR #4
    BL      delay
    SUB     r1, r1, #1
    CMP     r1, #-1
    MOVLE   r1, #15
    B       descending_loop

delay:
    STMFD   sp!, {r0, lr}
    LDR     r0, =0xFFFFF
    BL      delay_loop
    LDMFD   sp!, {r0, lr}
    MOV     pc, lr

delay_loop:
    CMP     r0, #0
    MOVEQ   pc, lr              @ Retorna da subrotina caso o r0 tenha
chegado em 0
    SUB     r0, r0, #1          @ Decrementa o r0 até o valor de 0 para
aplicar o delay
    B       delay_loop
```

7.5.3

```
@ Exercicio 7.5.3

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008

    .text
    .global main

main:
    LDR     r0, =0x3FF5000  @ IOPMOD
    LDR     r1, =0x1FC00    @ Define o display de 7 segmentos como output
    STR     r1, [r0]

    LDR     r0, =dados
    LDR     r2, [r0]

    LDR     r0, =0x3FF5008  @ IOPDATA

    CMP     r2, #0
    BLT     fim
    CMP     r2, #15
    BGT     fim

    MOV     r2, r2, LSL #10
    STR     r2, [r0]
    MOV     r2, r2, LSR #10

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0

dados:      .word 10
```

7.5.4

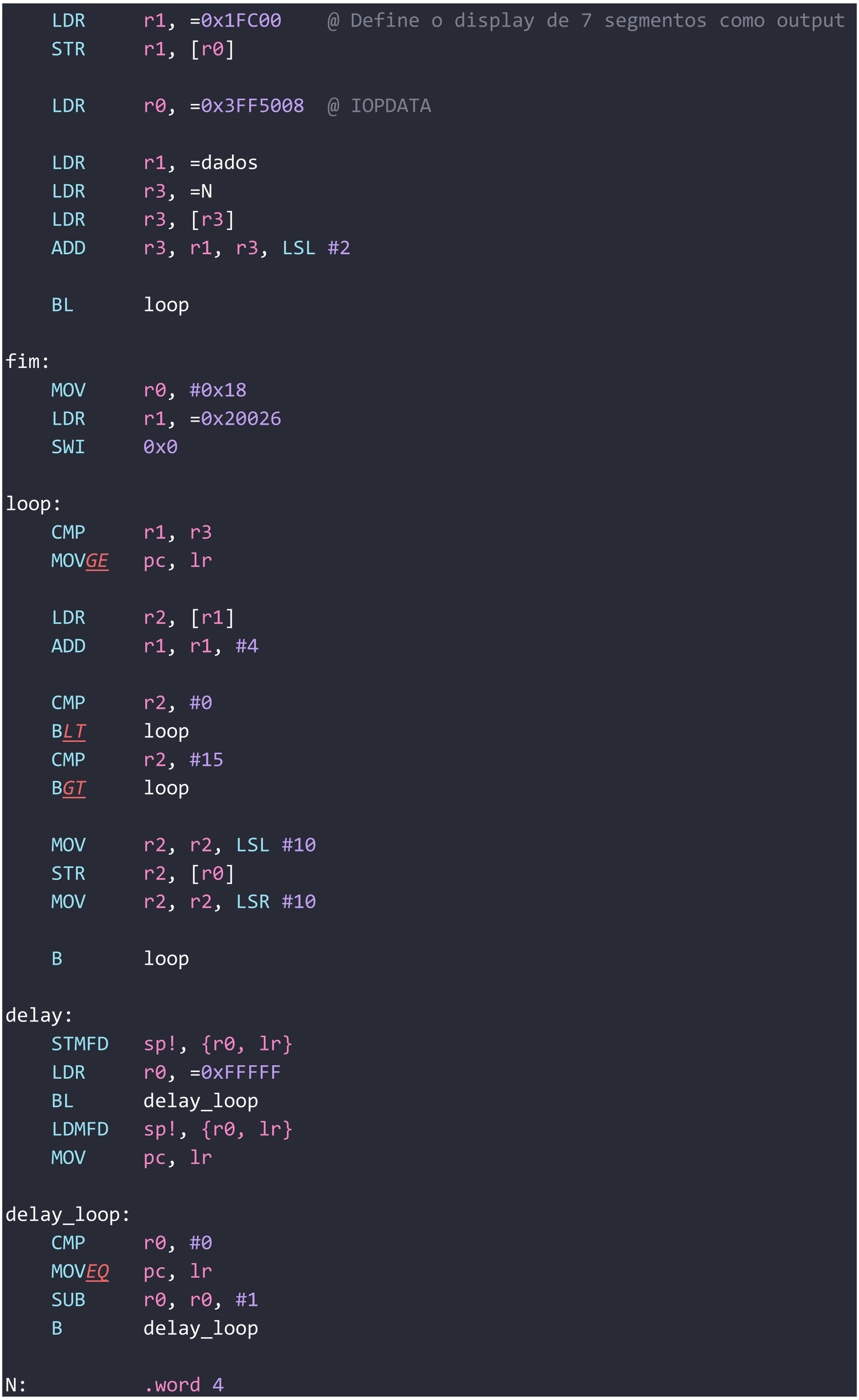
```
@ Exercicio 7.5.4

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008

    .text
    .global main

main:
    LDR     r0, =0x3FF5000  @ IOPMOD
```



```
    LDR     r1, =0x1FC00    @ Define o display de 7 segmentos como output
    STR     r1, [r0]

    LDR     r0, =0x3FF5008  @ IOPDATA

    LDR     r1, =dados
    LDR     r3, =N
    LDR     r3, [r3]
    ADD     r3, r1, r3, LSL #2

    BL      loop

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0

loop:
    CMP     r1, r3
    MOVGE   pc, lr

    LDR     r2, [r1]
    ADD     r1, r1, #4

    CMP     r2, #0
    BLT     loop
    CMP     r2, #15
    BGT     loop

    MOV     r2, r2, LSL #10
    STR     r2, [r0]
    MOV     r2, r2, LSR #10

    B       loop

delay:
    STMFD   sp!, {r0, lr}
    LDR     r0, =0xFFFFF
    BL      delay_loop
    LDMFD   sp!, {r0, lr}
    MOV     pc, lr

delay_loop:
    CMP     r0, #0
    MOVEQ   pc, lr
    SUB     r0, r0, #1
    B       delay_loop

N:          .word 4
```

```
dados:      .word 1, 2, 3, 4
```

7.5.5

```
@ Exercicio 7.5.5

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008

    .text
    .global main

main:
    LDR     r0, =0x3FF5000      @ IOPMOD
    LDR     r1, =0xF0           @ Define leds como output, dip como input
    STR     r1, [r0]

    LDR     r0, =0x3FF5008      @ IOPDATA

    LDR     r1, [r0]
    MOV     r1, r1, LSL #28
    MOV     r1, r1, LSR #28
    MOV     r1, r1, LSL #4

    STR     r1, [r0]

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0
```

7.5.6

```
@ Exercicio 7.5.6

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008
```

```
    .text
    .global main

main:
    LDR     r0, =0x3FF5000  @ IOPMOD
    LDR     r1, =0xF0       @ Seta leds como output, dip como input
    STR     r1, [r0]

    LDR     r0, =0x3FF5008  @ IOPDATA

    B       loop

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0

loop:
    LDR     r1, [r0]

    MOV     r1, r1, LSL #28
    MOV     r1, r1, LSR #28
    MOV     r1, r1, LSL #4

    STR     r1, [r0]

    B       loop
```

7.5.8

```
@ Exercicio 7.5.8

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD   = 0x3FF5000
@ IOPDATA = 0x3FF5008

    .text
    .global main

main:
    LDR     r0, =0x3FF5000  @ IOPMOD
    LDR     r1, =0x1FC00    @ Seta leds como output, dip como input
    STR     r1, [r0]
    LDR     r0, =0x3FF5008  @ IOPDATA

    LDR     r1, [r0]
    MOV     r1, r1, LSL #26
    MOV     r1, r1, LSR #26
    MOV     r1, r1, LSL #10

    STR     r1, [r0]

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0
```

7.5.9

```
@ Exercicio 7.5.9

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008

    .text
    .global main

main:
    LDR     r0, =0x3FF5000  @ IOPMOD
    LDR     r1, =0x1FC00    @ Seta leds como output, dip como input
    STR     r1, [r0]

    LDR     r0, =0x3FF5008  @ IOPDATA

    B       loop

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0

loop:
    LDR     r1, [r0]

    MOV     r1, r1, LSL #28
    MOV     r1, r1, LSR #28
    MOV     r1, r1, LSL #10

    STR     r1, [r0]

    B       loop
```

7.5.11

```
@ Exercicio 7.5.11

@ 7seg = P[16:10]   set for output
@ leds = P[7:4]     set for output
@ dip  = P[3:0]     clear for input

@ IOPMOD  = 0x3FF5000
@ IOPDATA = 0x3FF5008

@ r0 = IOPDATA
@ r1 = count
@ r2 = start
@ r3 = next

    .text
    .global main

main:
    LDR     r0, =0x3FF5000  @ IOPMOD
    LDR     r1, =0x1FC00    @ Seta 7 segment como output, dip4 como input
    STR     r1, [r0]

    LDR     r0, =0x3FF5008  @ IOPDATA
    LDR     r2, [r0]
    MOV     r2, r2, LSL #28
    MOV     r2, r2, LSR #28
    MOV     r2, r2, LSR #3

    MOV     r1, #0

    BL      loop

fim:
    MOV     r0, #0x18
    LDR     r1, =0x20026
    SWI     0x0

loop:
    LDR     r3, [r0]
    MOV     r3, r3, LSL #28
    MOV     r3, r3, LSR #28
    MOV     r3, r3, LSR #3

    CMP     r2, r3
    ADDNE   r1, r1, #1
    MOVNE   r2, r3

    MOV     r2, r2, LSL #10
    STR     r1, [r0]
    MOV     r2, r2, LSR #10

    CMP     r1, #15
    MOVGE   pc, lr

    B       loop
```