

PCS3432 - Laboratório de Processadores

Tarefa - E2

Bruno Mariz - 11261826

3.10.4 Register-swap algorithm

The EOR instruction is a fast way to swap the contents of two registers without using an intermediate storage location such as a memory location or another register. Suppose two values A and B are to be exchanged. The following algorithm could be used: $A = A \oplus B$ $B = A \oplus B$ $A = A \oplus B$ Write the ARM code to implement the above algorithm, and test it with the values of $A = 0xF631024C$ and $B = 0x17539ABD$. Show your instructor the contents before and after the program has run.

2.6 tarefa E2 - código + saída gdb (INDIVIDUAL) -entrega durante o laboratório

Sem zipar nada, entreguem os arquivos que correspondem ao exercício 3.10.4:

Retorne o seu código que resolve 3.10.4 (Register-swap algorithm). No PDF temos os valores: $A = 0xF631024C$ and $B = 0x17539ABD$. O seu programa deve colocar A no registrador R1 e B em R2, além disso: Caso seu NUSP for: par: troque A pelo seu NUSP e mantenha B. ímpar: troque B pelo seu NUSP e mantenha A.

Exemplo:

Supondo que seu NUSP seja 1234 (decimal), você pode colocar seu NUSP em r1 da seguinte forma:
`LDR r1, =1234`

Programa utilizado na tarefa:

```
.text
.globl main
main:
    LDR r1, =0x11261826
    LDR r2, =0x17539ABD
    BL firstfunc
    MOV r0, #0x18
    SWI 0x0
firstfunc:
    EOR r1, r1, r2
    EOR r2, r1, r2
    EOR r1, r1, r2
    MOV pc, lr
```

Estado dos registradores antes de rodar a função:

```
Register group: general
r0      0x1      1      r1      0x11261826      287709222
r2      0x17539abd      391355069      r3      0x103c8      66504
r4      0x103f4      66548      r5      0x0      0
r6      0x102d8      66264      r7      0x0      0
r8      0x0      0      r9      0x0      0
r10     0xff7ee000      -8462336      r11     0x0      0
r12     0xfffeedf8      -70152      sp      0xfffeed80      0xfffeed80
lr      0xff6657b4      -10070092      pc      0x103d0      0x103d0 <main+8>
cpsr    0x60000010      1610612752      fpscr   0x0      0
fpsid   0x410430f0      1090793712      fpexc   0x40000000      1073741824
AMAIRO S 0x0      0      AFSR0_EL1 0x0      0

11261826.s
1      .text
2      .globl main
3      main:
B+ 4      LDR    r1, =0x11261826
5      LDR    r2, =0x17539ABD
>6      BL     firstfunc
7      MOV    r0, #0x18
8      SWI    0x0
9      firstfunc:
10     EOR    r1, r1, r2
11     EOR    r2, r1, r2

remote Thread 1.15031 In: main L6 PC: 0x103d0
(gdb) b main
Ponto de parada 1 at 0x103c8: file 11261826.s, line 4.
(gdb) c
Continuing.

Breakpoint 1, main () at 11261826.s:4
(gdb) s
(gdb) s
(gdb) █
```

Estado dos registradores após rodar a função:

```
Register group: general
r0      0x1      1      r1      0x17539abd      391355069
r2      0x11261826      287709222      r3      0x103c8      66504
r4      0x103f4      66548      r5      0x0      0
r6      0x102d8      66264      r7      0x0      0
r8      0x0      0      r9      0x0      0
r10     0xff7ee000      -8462336      r11     0x0      0
r12     0xfffeedf8      -70152      sp      0xfffeed80      0xfffeed80
lr      0x103d4      66516      pc      0x103d4      0x103d4 <main+12>
cpsr    0x60000010      1610612752      fpscr   0x0      0
fpsid   0x410430f0      1090793712      fpexc   0x40000000      1073741824
AMAIRO S 0x0      0      AFSR0_EL1 0x0      0

11261826.s
1      .text
2      .globl main
3      main:
B+ 4      LDR    r1, =0x11261826
5      LDR    r2, =0x17539ABD
6      BL     firstfunc
>7      MOV    r0, #0x18
8      SWI    0x0
9      firstfunc:
10     EOR    r1, r1, r2
11     EOR    r2, r1, r2

remote Thread 1.15031 In: main L7 PC: 0x103d4
(gdb) b main
Ponto de parada 1 at 0x103c8: file 11261826.s, line 4.
(gdb) c
Continuing.

Breakpoint 1, main () at 11261826.s:4
(gdb) s
(gdb) s
(gdb) n
(gdb) █
```

É possível observar que os registradores r1 e r2 foram trocados após a execução da função.

Isso acontece pois $x \oplus x = 0$, portantoo

$a = a \oplus b$

$b = a \oplus b$, que equivale $a: a \oplus b \oplus b = a$

$a = a \oplus b$, que equivale $a: a \oplus b \oplus a = b$

