

PCS3432 - Laboratório de Processadores

Planejamento - E2

Bruno Mariz - 11261826

Execução do item-2-2.s

Antes de alterar o ADD (utilizando docker)

```
(gdb) target sim
Connected to the simulator.
(gdb) load
(gdb) b main
Breakpoint 1 at 0x8218: file item-2-2.s, line 4.
(gdb) r
Starting program: /home/student/src/e2/planejamento/a.out

Breakpoint 1, main () at item-2-2.s:4
4          MOV      r0, #15
Current language:  auto; currently asm
(gdb) p/x $cpsr
$1 = 0x60000013
(gdb) s
firstfunc () at item-2-2.s:11
11          ADD      r0, r0, r1
(gdb) p/x $cpsr
$2 = 0x60000013
(gdb) s
12          MOV      pc, lr
(gdb) p/x $cpsr
$3 = 0x60000013
(gdb) s
main () at item-2-2.s:7
7          MOV      r0, #0x18
```

```

(gdb) p/x $cpsr
$1 = 0x60000013
(gdb) s
firstfunc () at item-2-2.s:11
11          ADD      r0, r0, r1
(gdb) p/x $cpsr
$2 = 0x60000013
(gdb) s
12          MOV      pc, lr
(gdb) p/x $cpsr
$3 = 0x60000013
(gdb) s
main () at item-2-2.s:7
7           MOV      r0, #0x18
(gdb)

```

Após alterar ADD para ADDS (utilizando arm utils)

```

(gdb) b main
(gdb) c
(gdb) p/x $cpsr
$1 = 0x60000010
(gdb) s
(gdb) s
(gdb) s
firstfunc () at item-2-2-alterado.s:11
(gdb) p/x $cpsr
$1 = 0x60000010

```

Register group: general

r0	0xf	15	r1	0x14	20
r2	0xffffec5c	-70564	r3	0x103c8	66504
r4	0x103ec	66540	r5	0x0	0
r6	0x102d8	66264	r7	0x0	0

item-2-2-alterado.s

```

10      firstfunc:
>11      ADDS      r0, r0, r1
12      MOV      pc, lr

```

remote Thread 1.136831 In: firstfunc
L11 PC: 0x103e0
(gdb) p/x \$cpsr
\$1 = 0x60000010
(gdb) s
(gdb) s
(gdb) s
firstfunc () at item-2-2-alterado.s:11
(gdb) p/x \$cpsr
\$2 = 0x60000010
(gdb)

printscreen1: antes de rodar a instrução ADDS

\$cpsr = 0x60000010

NZCV = 0110

```
(gdb) s
(gdb) p/x $cpsr
$3 = 0x10
```

The screenshot shows the GDB interface with the following content:

- Register group: general**

Register	Hex Value	Decimal Value	Register	Hex Value	Decimal Value
r0	0x23	35	r1	0x14	20
r2	0xffffec5c	-70564	r3	0x103c8	66504
r4	0x103ec	66540	r5	0x0	0
r6	0x102d8	66264	r7	0x0	0
- Assembly instructions:**
 - 11 ADDS r0, r0, r1
 - >12 MOV pc, lr
- Thread Information:** remote Thread 1.136831 In: firstfunc L12 PC: 0x103e4
- GDB Commands and Output:**

```
(gdb) s
(gdb) s
firstfunc () at item-2-2-alterado.s:11
(gdb) p/x $cpsr
$2 = 0x60000010
(gdb) s
(gdb) p/x $cpsr
$3 = 0x10
(gdb)
```

printscreen2: depois de rodar a instrução ADDS

`$cpsr = 0x10`

Resultado do CPSR após ADDS:

`NZCV = 0000`

Isso pode ser explicado pois a soma $15+15 = 30$ ($\$r2 = 0x14$) resultou em um número positivo, não nulo, sem carry e sem overflow.