

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO  
Graduação em Engenharia Computação

PCS3732 - Laboratório de Processadores

Professor Jorge Kinoshita



Grupo 10 - Planejamento 1

Rode o código item-2-2.s no seu computador e acompanhe os registradores sendo modificados.

Observe as flags do CSPR (apostila cap 1) ao fazer a soma. As flags são atualizadas?

Não, as flags não são atualizadas, pois os únicos registradores que mudam após a operação são o R2 (que recebe a soma) e o PC (que iria mudar de qualquer forma, para seguir para a próxima instrução).

Isso pode ser observado nas imagens abaixo.

## Antes do ADD

The screenshot shows a debugger window with the title bar 'arthur@arthurfonseca: ~/Documents/Pol/LabProc/gcc-arm'. The window is divided into several panes. The top pane shows the 'Register group: general' with a table of registers and their values. The middle pane shows the assembly code for 'item-2-2.s'. The bottom pane shows the debugger's command line and output.

Register	Value
r0	0xf 15
r4	0x1 1
r8	0x0 0
r12	0x1fffc 2097100
fps	0x0 0
r1	0x14 20
r5	0x1ffff8 2097144
r9	0x0 0
sp	0x1ffff8 2097144
cpsr	0x6000013 16
r2	0xffffffff -1r3
r6	0x0 0
r10	0x200100 2097408
lr	0x8224 33316
r3	0xa9c8 43464
r7	0x0 0
r11	0x0 0
pc	0x8230 33328

```
7      MOV     r0, #0x18
8      LDR     r1, =0x20026
9      SWI     0x0
10     firstfunc:
11     ADD     r0, r0, r1
12     MOV     pc, lr
13
14
15
16
```

sim process 42 In: firstfunc  
Breakpoint 1 at 0x822c: file item-2-2.s, line 9.  
(gdb) b 11  
Breakpoint 2 at 0x8230: file item-2-2.s, line 11.  
(gdb) c  
Continuing.  
The program is not being run.  
(gdb) r  
Starting program: /home/student/src/a.out  
  
Breakpoint 2, firstfunc () at item-2-2.s:11  
Current language: auto; currently asm  
(gdb)

## Depois do ADD

```
arthur@arthurpfonseca: ~/Documents/Pol/LabProc/gcc-arm
File Edit View Search Terminal Help
Register group: general
r0      0x23      35      r1      0x14      20      r2      0xffffffff -1r3      0xa9c8      43464
r4      0x1       1      r5      0x1ffff8 2097144 r6      0x0        0      r7      0x0        0
r8      0x0       0      r9      0x0       0      r10     0x200100 2097408 r11     0x0        0
r12     0x1fffcc 2097100 sp      0x1ffff8 2097144 lr      0x8224     33316  pc      0x8234     33332
fps     0x0       0      cpsr    0x60000013 16

item-2-2.s
7          MOV     r0, #0x18
8          LDR     r1, =0x20026
9          SWI     0x0
10         firstfunc:
11         ADD     r0, r0, r1
12         MOV     pc, lr
13
14
15
16

sim process 42 In: firstfunc                                     Line: 12  PC: 0x8234
(gdb) b 11
Breakpoint 2 at 0x8230: file item-2-2.s, line 11.
(gdb) c
Continuing.
The program is not being run.
(gdb) r
Starting program: /home/student/src/a.out

Breakpoint 2, firstfunc () at item-2-2.s:11
Current language: auto; currently asm
(gdb) n
(gdb) █
```

**Agora, no código item-2-2.s troque ADD por ADDS. As flags são atualizadas logo depois da soma (ADD)?**

Sim, com o ADDS o registrador CPSR muda de 0x60000013 para 0x00000013, ou seja 2 bits mudaram de 1 para 0.

Mais especificamente, segundo a imagem fornecida pela apostila:



Figure 1-3 ARM status register format

O nibble mais significativo representa os sinais N, Z, C e V, que foram todos setados para zero, ou seja, a operação:

- N = 0 : Não deu um número negativo
- Z = 0 : Não deu zero
- C = 0 : Não resultou em carry
- V = 0 : Não teve overflow

## Antes do ADDS

```

File Edit View Search Terminal Help
Register group: general
r0      0xf    15      r1      0x14   20      r2      0xffffffff -1
r3      0xa9c8 43464   r4      0x1    1      r5      0x1ffff8 2097144
r6      0x0    0      r7      0x0    0      r8      0x0    0
r9      0x0    0      r10     0x200100 2097408  r11     0x0    0
r12     0x1fffcc 2097100 sp      0x1ffff8 2097144  lr      0x8224 33316
pc      0x8230 33328  fps      0x0    0      cpsr    0x60000013 1610612755

4      MOV     r0, #15
5      MOV     r1, #20
6      BL      firstfunc
7      MOV     r0, #0x18
8      LDR     r1, =0x20026
9      SWI     0x0
b+
10     firstfunc:
B+> 11      ADDS    r0, r0, r1
12      MOV     pc, lr
13

sim process 42 In: firstfunc
Start address 0x8110
Transfer rate: 41824 bits/sec.
(gdb) break 11
Breakpoint 1 at 0x8230: file item-2-2.s, line 11.
(gdb) break 9
Breakpoint 2 at 0x822c: file item-2-2.s, line 9.
(gdb) run
Starting program: /home/student/src/a.out

Breakpoint 1, firstfunc () at item-2-2.s:11
Current language: auto; currently asm
(gdb)

```

## Depois do ADDS

```

Register group: general
r0      0x23      35      r1      0x14      20      r2      0xffffffff -1
r3      0xa9c8    43464   r4      0x1       1      r5      0x1ffff8 2097144
r6      0x0       0       r7      0x0       0      r8      0x0       0
r9      0x0       0       r10     0x200100 2097408 r11     0x0       0
r12     0x1ffffc 2097100  sp      0x1ffff8 2097144 lr      0x8224    33316
pc      0x8234    33332   fps      0x0       0      cpsr    0x13     19

item-2-2.s
8      LDR      r1, =0x20026
9      SWI      0x0
10     firstfunc:
11     ADDS     r0, r0, r1
12     MOV      pc, lr
13
14
15
16
17

sim process 42 In: firstfunc                                     Line: 12   PC: 0x8234
Transfer rate: 41824 bits/sec.
(gdb) break 11
Breakpoint 1 at 0x8230: file item-2-2.s, line 11.
(gdb) break 9
Breakpoint 2 at 0x822c: file item-2-2.s, line 9.
(gdb) run
Starting program: /home/student/src/a.out

Breakpoint 1, firstfunc () at item-2-2.s:11
Current language:  auto; currently asm
(gdb) n
(gdb) █
```

Tudo ocorreu dentro do esperado, já que o código procurava somar 15 e 20, o que não excede a representação de 32 bits e é uma operação que dá resultado positivo.