# ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO Graduação em Engenharia Computação

### PCS3732 - Laboratório de Processadores

## Professor Jorge Kinoshita



Grupo 10 - Planejamento 1

NUSP: 10773096

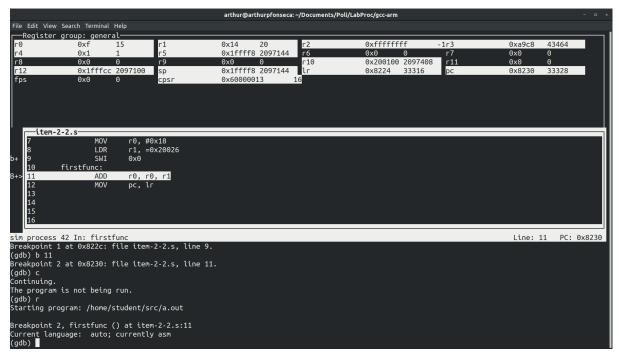
Arthur Pires da Fonseca

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

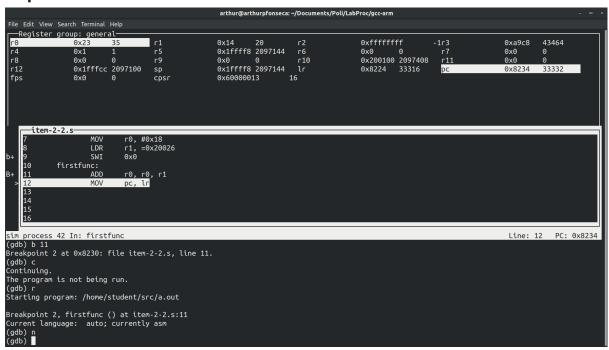
# Observe as flags do CSPR (apostila cap 1) ao fazer a soma. As flags são atualizadas?

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



### **Depois do ADD**



# 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

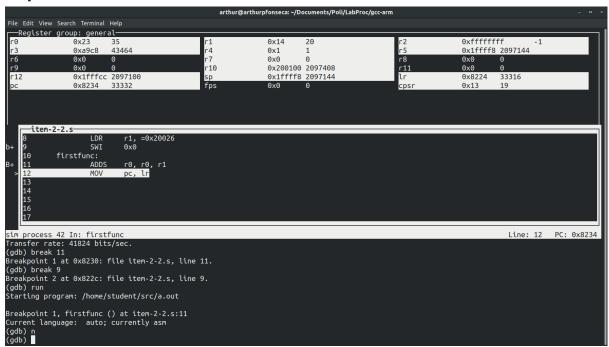
O nibble mais significativo representa os sinais N, Z, C e V, que foram todos setados para zero, ou seja, a operação:

- 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**

```
arthur@arthurpfonseca: ~/Documents/Poli/LabProc/gcc-arm
     Register group: general
0 0xf 1
                            0x1 15
0xa9c8 43464
0x0 0
                                                                                                                                                                                                  0xffffffff
0x1fffff8 2097144
0x0 0
                                                                                    Γ1
Γ4
Γ7
Γ10
                                                                                                               0x1
0x0
                                                                                                               0x200100 2097408
0x1fffff8 2097144
0x0 0
                                                                                                                                                                                                  0x0 0
0x8224 33316
0x60000013
                            0x1fffcc 2097100
0x8230 33328
                                                                                                                                                                                                                                 1610612755
                                                 r0, #15
r1, #20
firstfunc
r0, #0x18
r1, =0x20026
0x0
                                                  г0, г0, г1
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.
                                                                                                                                                                                                                      Line: 11 PC: 0x8230
 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
```

### **Depois do ADDS**



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.