PSI3441 – Arquitetura de Sistemas Embarcados

- Registradores

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – grehder@lme.usp.br

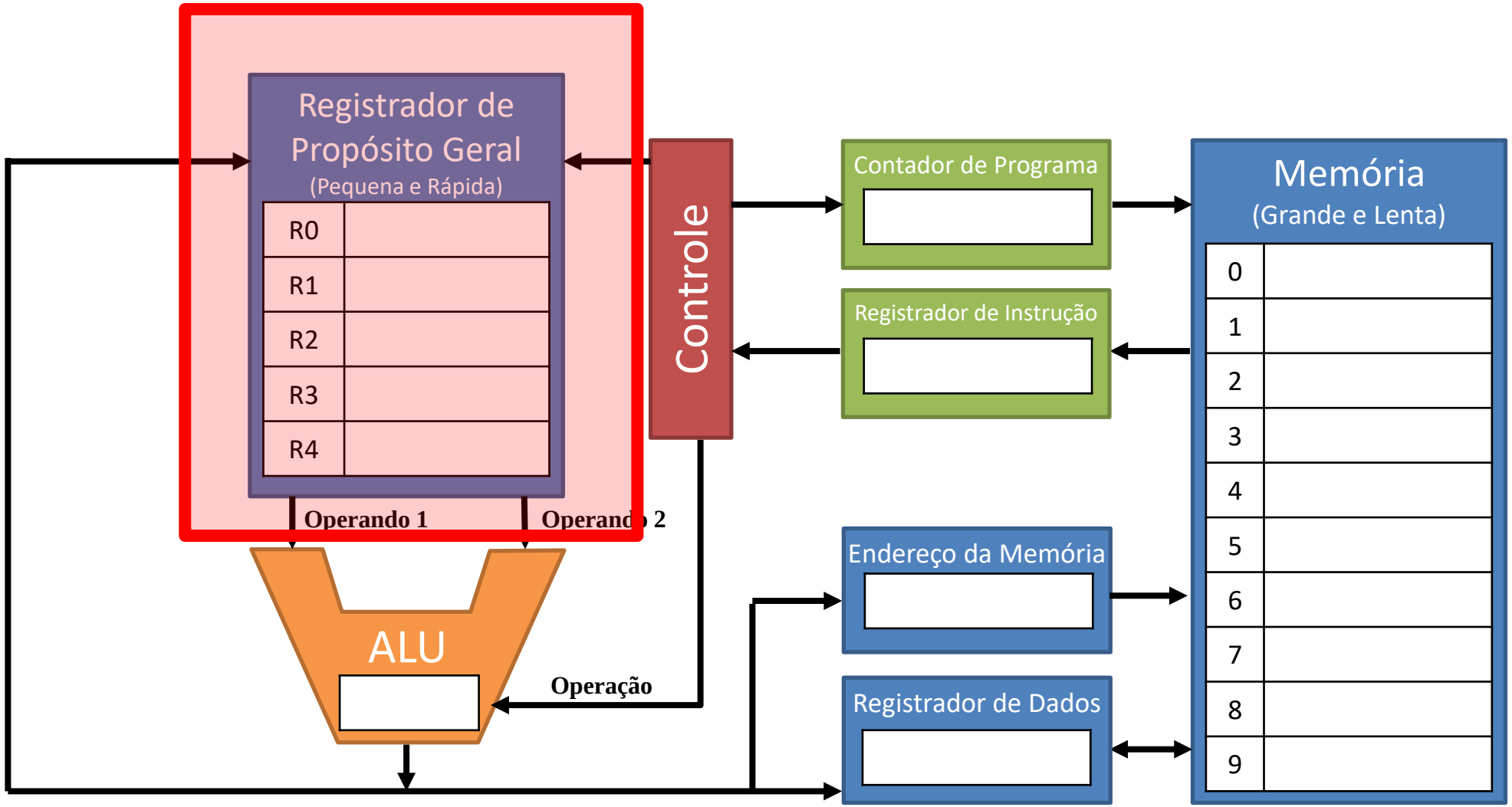
2020

Objetivo do Vídeo

- Registradores de Propósito Geral
- Registradores de Status do Programa
- Instruções com Flags

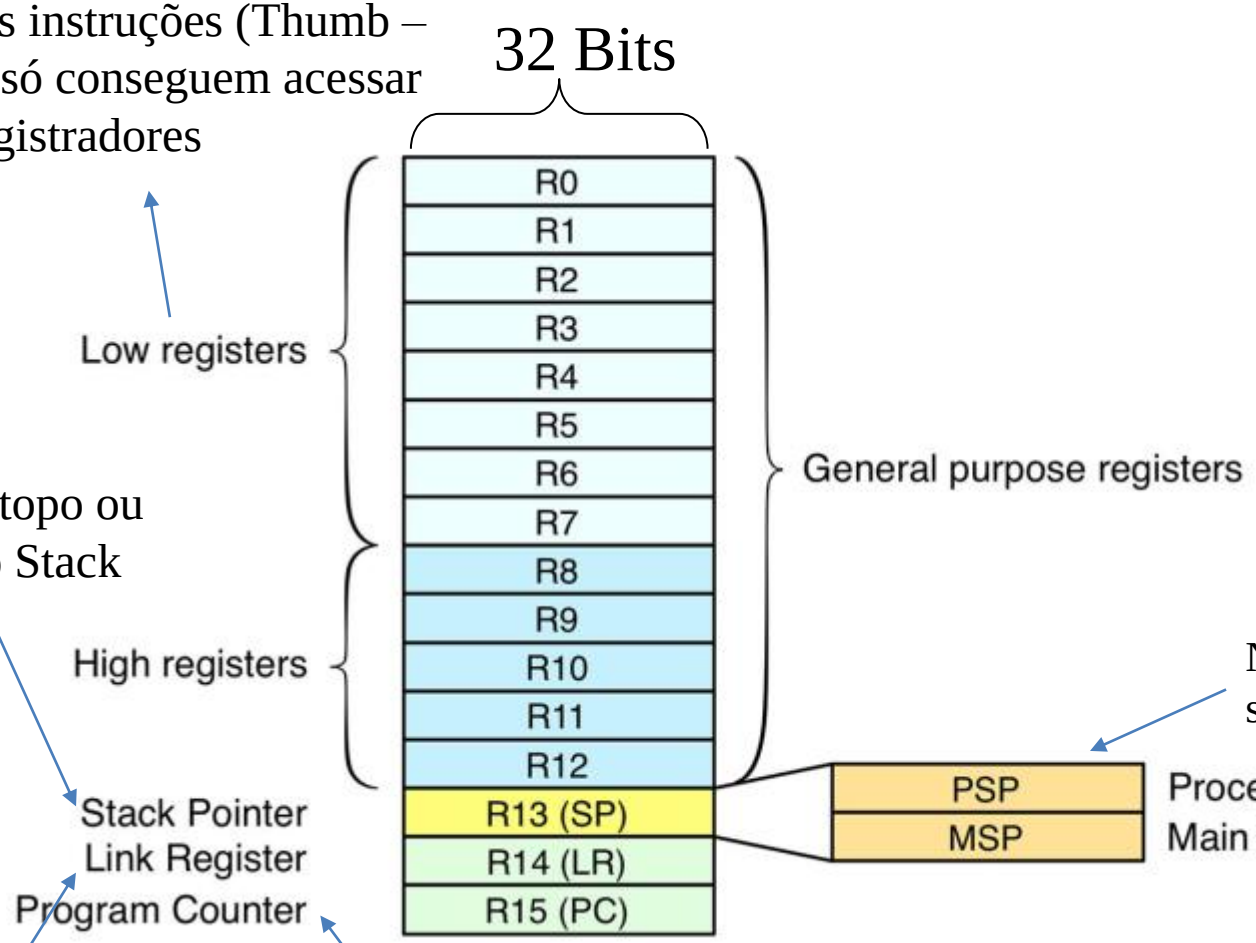
Registradores no ARM Cortex M0+

Arquitetura “Load-Store”



Registradores de Propósito Geral – ARM Cortex M0+

Algumas instruções (Thumb – 16 bits) só conseguem acessar esses registradores



Indica o topo ou fundo do Stack

Normalmente usado com sistemas operacionais

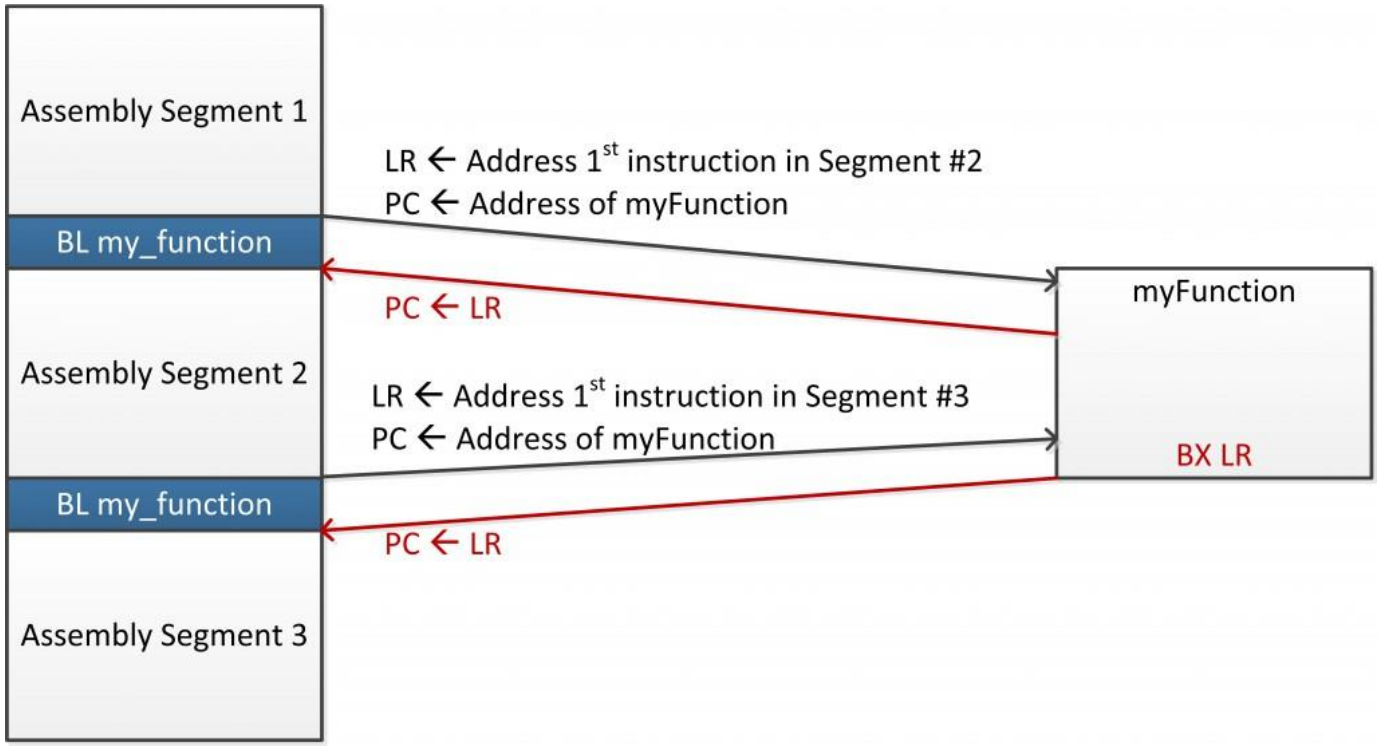
Depende do modo de operação do processador

Guarda o endereço de retorno de uma função

Indica o endereço da instrução atual. Pode ser lido ou escrito. A escrita serve para fazer desvios (loops while, for, if)

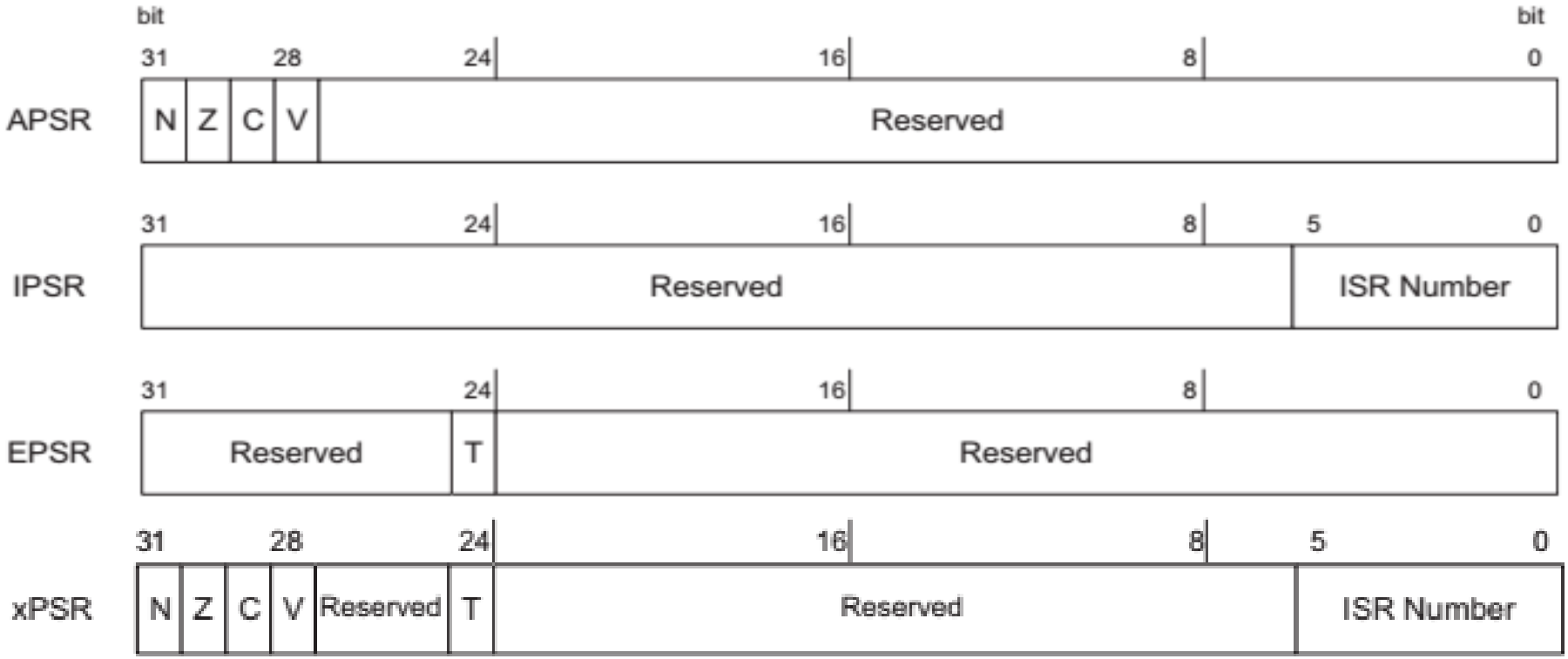
Link Register LR (R14)

- Guarda o valor do *Program Counter* (PC) quando uma função é chamada e permite o retorno da função para a próxima instrução a ser executada



Program Status Register (xPSR)

- Formado por 3 registradores
  - Aplicação (APSR) – Contém flags do ALU (Bits 28-31)
  - Interrupção (IPSR) – Contém o número da ISR (Interrupt Service Routine) em execução (Bits 0-5)
  - Execução (EPSR) – Contém o bit T - estado do Thumb (Bit 24)



APSR

- N (Bit 31) – Flag setado → resultado Negativo
- Z (Bit 30) – Flag setado → resultado Zero ou valor igual em uma comparação
- C (Bit 29) – Flag setado → "Carry" (vai 1) do resultado de soma
- V (Bit 28) – Flag setado → oVerflow do resultado de soma ou subtração

Exemplo: Qual são os valores de N, Z, C, V para:

a) 0x0000009C + 0xFFFFFF64

```
  0x 0000009C –    0000 0000 0000 0000 0000 0000 1001 1100
+ 0x FFFFFF64 –    1111 1111 1111 1111 1111 1111 0110 0100
= 0x1 00000000 – 1 0000 0000 0000 0000 0000 0000 0000 0000
                   ↖ Carry
```

Como existe um Carry, então C=1
O resultado é zero, então Z = 1 e N=0
Não houve overflow, então V=0

APSR

V é setado quando:

- ou existe um Carry do MSB-1 para o MSB (D30 para o D31)
- ou existe um Carry para além do MSB

N é setado quando:

- O MSB é 1

C é setado quando:

- Existe um Carry para além do MSB

MSB = Most Significant Bit

Exemplo:

7+7

7 – 0111
+7 – 0111
14 – 1110

V=1, N=1, C=0, Z=0

Exemplo:

9+9

9 – 1001
+9 – 1001
2 – 1 0010

V=1, N=0, C=1, Z=0

APSR

Signed ou Unsigned ?

Unsigned:

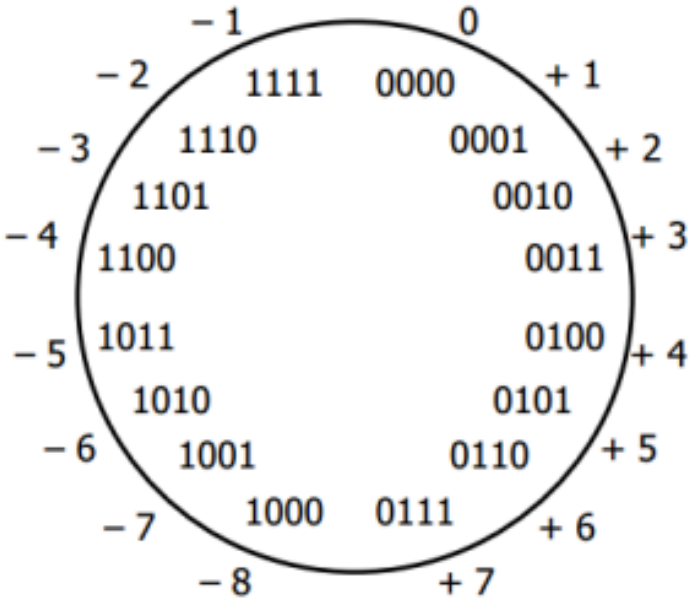
14 – 1110
+14 – 1110
12 – 1100

V=0, N=1, C=1, Z=0

Signed:

-2 – 1110
+-2 – 1110
-4 – 1100

V=0, N=1, C=1, Z=0



Operation	Results, flags
0x70000000 + 0x70000000	Result = 0xE0000000, N = 1, Z = 0, C = 0, V = 1
0x90000000 + 0x90000000	Result = 0x20000000, N = 0, Z = 0, C = 1, V = 1
0x80000000 + 0x80000000	Result = 0x00000000, N = 0, Z = 1, C = 1, V = 1
0x00001234 – 0x00001000	Result = 0x00000234, N = 0, Z = 0, C = 1, V = 0
0x00000004 – 0x00000005	Result = 0xFFFFFFFF, N = 1, Z = 0, C = 0, V = 0
0xFFFFFFFF – 0xFFFFFFFC	Result = 0x00000003, N = 0, Z = 0, C = 1, V = 0
0x80000005 – 0x80000004	Result = 0x00000001, N = 0, Z = 0, C = 1, V = 0
0x70000000 – 0xF0000000	Result = 0x80000000, N = 1, Z = 0, C = 0, V = 1
0xA0000000 – 0xA0000000	Result = 0x00000000, N = 0, Z = 1, C = 1, V = 0

Operações com os Flags do PSR

- ADC Rd, Rn, Op2 ;Rd=Rn+Op2+C Não atualiza flags
- ADCS Rd, Rn, Op2 ; Rd=Rn+Op2+C Atualiza flags
- ADD Rd, Rn, Op2 ;Rd=Rn+Op2 Não atualiza flags
- ADDS Rd, Rn, Op2 ;Rd=Rn+Op2 Atualiza flags
- AND Rd, Rn, Op2 ;Rd=Rn&Op2 Não atualiza flags
- ANDS Rd, Rn, Op2 ;Rd=Rn&Op2 Atualiza flag
- ASR Rd, Rm, Rn ;Rd=Rm>>Rn Não atualiza flags
- ASRS Rd, Rm, Rn ;Rd=Rm>>Rn Atualiza flags

Outros – BIC/BICS, LSL/LSLS, LSR/LSRS, MOV/MOVS, ORN/ORNS, ORR/ORRS, ROR/RORS, RRX/RRXS, RSB/RSBS, SBC/SBCS,

```
// Calculating Z = X + Y, where X, Y and Z are all 64-bit
Z[31:0] = X[31:0] + Y[31:0];     // Calculate lower word addition,
                                 // carry flag get updated
Z[63:32] = X[63:32] + Y[63:32] + Carry;  // Calculate upper word addition.
```