

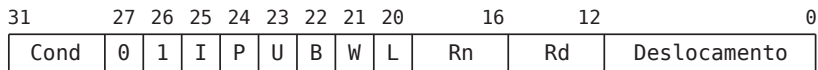
# Linguagens de montagem

## Capítulo 10 - ARM - transferência entre registrador e memória

Ricardo Anido  
Instituto de Computação  
Unicamp

# Carga e Armazenamento

- ▶ Instruções para carga e armazenamento de registradores em memória são respectivamente LDR (do inglês *load register*) e STR (do inglês *store register*).
- ▶ Codificação:



# Carga e armazenamento

Formato básico em linguagem de montagem:

*LDR*    *Rd*, *endereço*

*STR*    *Rd*, *endereço*

onde *endereço* depende do modo de endereçamento.

# Endereçamento direto

$instr\{cond\}\{B\} \quad Rd, \text{endereço}$

onde

- ▶ *instr* especifica a instrução e pode ser LDR ou STR.
- ▶ *cond* é um sufixo de condição.
- ▶ B, se presente, faz com que o montador coloque o valor 1 no bit 22 da instrução (bit B, indicando que a operação deve carregar ou armazenar bytes; o byte menos significativo do registrador Rd é carregado ou armazenado (na instrução LDR os outros 24 bits do registrador são zerados). Se não presente, o bit B da instrução é 0, indicando que a operação deve carregar ou armazenar palavras.
- ▶ Rd é o registrador destino (onde o valor lido da memória é carregado) para instruções LDR e o registrador fonte (contém valor a ser armazenado na memória) para instruções STR.

# Endereçamento direto

- ▶ *endereço* é o endereço do operando na memória (normalmente especificado por um rótulo do programa). O montador monta no campo Deslocamento da instrução a diferença entre *endereço* e o ponto corrente de montagem. Em outras palavras, o endereçamento é relativo ao valor do registrador contador de programa pc. No momento da execução, o processador calcula o endereço efetivo do operando adicionando o valor do registrador pc ao valor do campo Deslocamento com o sinal estendido para 32 bits. Um erro é gerado pelo montador se a diferença entre o endereço do operando e o ponto corrente de montagem não puder ser codificado em 12 bits.

# Endereçamento direto

```
.org 0x1000
ldr    r1, var1  @ carrega palavra em r1, incondicional
ldrgt  r2, var1  @ carrega palavra em r2, condicional
ldreqb r3, var2  @ carrega byte em r3, condicional

str     r4, var1  @ armazena r4 em palavra, incondicional
strcc   r5, var1  @ armazena r5 em palavra, condicional
strleb  r6, var2  @ armazena r6 em byte, condicional

ldr     r7, var_longe @ gera erro, endereço muito distante

.org 0x2000
var1:
.word 0x12341234
var2:
.byte 0xff

.org 0x20000
var_longe:
.word 0x0
```

# Endereçamento indireto por registrador

Formato geral em linguagem de montagem:

$$instr\{cond\}\{B\} \quad Rd, [Rn]$$

Exemplos:

```
ldr    r1,[r0]  @ carrega palavra em r1, incondicional
ldrmi  r2,[r8]  @ carrega palavra em r2, condicional
ldrplb r3,[r9]  @ carrega byte em r3, condicional

str    r4,[r10] @ armazena r4 em palavra, incondicional
strlt  r5,[r11] @ armazena r5 em palavra, condicional
strgtb r6,[r12] @ armazena r6 em byte, condicional
```

# Modos de endereçamento pré-indexados

- ▶ Nos modos de endereçamento pré-indexados, o endereço efetivo do operando em memória é formado pelo valor do registrador base mais (ou menos) um *deslocamento*, codificado no campo Deslocamento da instrução.
- ▶ A forma do cálculo desse *deslocamento* dá origem a diferentes modos de endereçamento.



# Modos de endereçamento pré-indexados

O *deslocamento* pode ser dado por:

- ▶ um valor imediato, constante, dando origem ao modo de endereçamento *indireto por registrador mais constante*
- ▶ o valor de um registrador, dando origem ao modo de endereçamento *indireto por registrador base e registrador índice*
- ▶ o valor de um registrador deslocado com o auxílio da unidade de deslocamento (*barrel shifter*), dando origem ao modo de endereçamento *indireto por registrador e registrador índice escalado*.

- ▶ Há ainda a possibilidade de atualizar o valor do registrador base com o valor do endereço efetivo, dessa forma fazendo com que o registrador “avance” (ou “recue”) automaticamente por um valor controlado, dentro do mesmo ciclo de instrução.
- ▶ usado na implementação de comandos de repetição e na varredura de estruturas de dados sequenciais (como vetores, por exemplo).

# Endereçamento pré-indexado indireto por registrador mais constante

Formato geral:

$$instr\{cond\}\{B\} \quad Rd, [Rn, \#expr12] \{!\}$$

Exemplos:

```
ldr    r1, [r0,#1]  @ carrega palavra em r1, incondicional
ldrmi  r2, [r8,#4]! @ carrega palavra em r2, condicional
                    @ r2 é incrementado de 4
ldrplb r3, [r9,#2]  @ carrega byte em r3, condicional

str     r4, [r10,#-2]! @ armazena r4 em palavra, incondicional
                    @ r10 é decrementado de 2
strlt  r5, [r11,#4]  @ armazena r5 em palavra, condicional
strgtb r6, [r12,#1]! @ armazena r6 em byte, condicional
                    @ r12 é incrementado de 1
```