

Linguagens de montagem

ARM - Desvios e processamento de dados

Ricardo Anido
Instituto de Computação
Unicamp

Instruções de desvio alteram o valor do registrador pc (que armazena o endereço da próxima instrução que deve ser executada), de forma que podemos controlar o fluxo de execução do programa.

- ▶ Podem ser *incondicionais* ou *condicionais*.

Bits de condição

- ▶ armazenam o estado resultante de algumas instruções do processador.
- ▶ Nem todas as instruções afetam todos os bits de condições; por exemplo, instruções de transferência de dados, como MOV ou LDR, não afetam nenhum bit de condição.
- ▶ No Arm são armazenados em um registrador de estado, chamado *CPSR*.
- ▶ Os bits de condição são usados para controlar os desvios condicionais.

Bits de condição

- ▶ C: vai-um (*carry*). Ligado (ou seja, tem valor 1) se a operação causou vai-um (*carry-out*) ou empresta-um (*carry-in*), desligado caso contrário.
- ▶ Z: zero. Ligado se o resultado foi zero, desligado caso contrário.
- ▶ N: sinal. Cópia do bit mais significativo do resultado; considerando aritmética com sinal, se N igual a zero, o resultado é maior ou igual a zero. Se N igual a 1, resultado é negativo.
- ▶ V: estouro de campo (*overflow*), para operações com números com sinal em complemento de dois. Ligado se ocorreu estouro de campo, desligado caso contrário. Calculado como o ou-exclusivo entre o vai-um do bit mais significativo do resultado e o vai-um do segundo bit mais significativo do resultado.

Condições para os desvios

Código	Sufixo	Condição	Descrição
0000	EQ	$Z = 1$	Igual
0001	NE	$Z = 0$	Diferente
0010	CS	$C = 1$	Maior ou igual, valor sem sinal
0011	CC	$C = 0$	Menor, valor sem sinal
0100	MI	$N = 1$	Negativo
0101	PL	$N = 0$	Positivo ou zero
0110	VS	$V = 1$	Estouro de campo, valor com sinal
0111	VC	$V = 0$	Não estouro de campo, valor com sinal
1000	HI	$C = 1 \wedge Z = 0$	Maior, valor sem sinal
1001	LS	$C = 0 \vee Z = 1$	Menor ou igual, valor sem sinal
1010	GE	$N = V$	Maior ou igual, valor com sinal
1011	LT	$N \neq V$	Menor, valor com sinal
1100	GT	$Z = 0 \wedge (N = V)$	Maior, valor com sinal
1101	LE	$Z = 1 \vee (N \neq V)$	Menor ou igual, valor com sinal
1110	AL	–	Sempre

Desvios com endereço alvo constante

$B\{cond\}$ *endereço*

onde

- ▶ *cond* é um sufixo de condição
- ▶ *endereço* é o endereço alvo do desvio, normalmente dado como um rótulo do programa, que deve estar dentro do intervalo

$$[pc - (2^{25} - 1), pc + 2^{25}]$$

Desvios com alvo constante

Exemplo:

```
    bne  loop           @ desvio condicional
    b    fim            @ desvio incondicional
loop:                          @ um rótulo
    ...
fim:                          @ outro rótulo
    ...
```

Desvios por registrador

Formato geral:

$BX\{cond\} \ Rd$

Exemplo:

1	<code>bxmi r10</code>	@ desvio condicional
2	<code>bx r8</code>	@ desvio incondicional

Algumas Instruções de processamento

COM	Nome	Operação
SUB	Subtração	$Rd \leftarrow Rn - \text{Operando2}$
ADD	Adição	$Rd \leftarrow Rn + \text{Operando2}$
ADC	Adição com vai-um	$Rd \leftarrow Rn + \text{Operando2} + C$
SBC	Subtração com empresta-um	$Rd \leftarrow Rn - \text{Operando2} + C - 1$
RSC	Subtração reversa com empresta-um	$Rd \leftarrow \text{Operando2} - Rn + C - 1$
CMP	Comparação	$Rn - \text{Operando2}$
CMN	Comparação negativa	$Rn + \text{Operando2}$

Instruções de processamento

- ▶ No ARM é possível escolher se a instrução de processamento de dados deve ou não atualizar os bits de condição.
- ▶ o bit S da instrução determina se o processador deve ou não atualizar os bits de condição
- ▶ em linguagem de montagem, usamos o sufixo "s" para indicar que a instrução deve atualizar os bits de condição (como "adds").

Instruções que armazenam o resultado

No ARM as instruções ADD e SUB são exemplos de instruções que armazenam o resultado. Formato geral em linguagem de montagem:

$$instr\{cond\}\{S\} \quad Rd, Rn, Operando2$$

Exemplos:

```
add    r1,r2,r3           @ adição
                        @ r1 <-- r2 + r3
                        @ bits de condição não são atualizados

subs   r14,r14,#2048       @ subtração
                        @ r14 <-- r14 - 0x800
                        @ bits de condição são atualizados
```

Instruções que não armazenam resultado

As instruções CMP (compara), CMN (compara com negação) não armazenam o resultado. Formato geral:

instr{cond} Rn, Operando2

Exemplo:

```
cmp    r1,r2           @ comparação simples
                        @ [C,N,V,Z] <-- r1-r2
cmn    r10,r12          @ comparação com negação
                        @ [C,N,V,Z] <-- r10-(-r12)
```