

8. Representação de inteiros com e sem sinal (revisão de Sistemas Digitais)

ABF - AC I - MIPS IS_2

1

Inteiros positivos em binário

- Número em binário com n-bits

$$x = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots + x_12^1 + x_02^0$$

- Gama de representação: **0** a **$+2^n - 1$**

- Exemplo

$$\begin{aligned} & \blacksquare 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1011_2 \\ & = 0 + \dots + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ & = 0 + \dots + 8 + 0 + 2 + 1 = 11_{10} \end{aligned}$$

- Gama de representação com 32 bits

$$\blacksquare 0 \text{ a } +4,294,967,295$$

ABF - AC I - MIPS IS_2

2

Representação de números negativos

1. Sinal e módulo

$$-15 \quad \mathbf{1}0001111 = -(2^3+2^2+2^1+2^0)$$

Bit de sinal sem peso associado

- **Inconvenientes da representação em sinal e módulo:**
 - Bit de sinal “especial”
 - Exige somador e subtrator distintos
 - Dupla representação de zero: +0 (00000000) e -0 (10000000)

2. Representação em Complemento

- Subtrair é somar o complemento
- Bit de sinal tratado como os outros bits

ABF - AC I - MIPS IS_2

3

Complemento para 1

Complemento para 1 (complemento falso) obtém-se negando bit a bit, i.e. $\mathbf{-X = (2^{n-1}-1)-|X|}$

$$15 = 00001111$$

$$-15 = \mathbf{1}1110000$$

Bit de sinal com peso $-(2^{n-1}-1)$

- **Vantagens:**
 - Fácil de obter o complemento
 - Bit de sinal processado como os outros bits
 - Lógica de soma e subtração idênticas
- **Inconvenientes**
 - +0 (00000000) e -0 (11111111)
 - *End-around carry*

ABF - AC I - MIPS IS_2

4

Complemento para 1

- Gama de representação: $-(2^{n-1} - 1)$ a $(2^{n-1} - 1)$

- Soma de 2 números negativos:

$$X = (2^{n-1} - 1) - |X| \text{ e } Y = (2^{n-1} - 1) - |Y|$$

$$X + Y = (2^{n-1} - 1) - |X| + (2^{n-1} - 1) - |Y|$$

$$= (2^n - 2) - (|X| + |Y|) \bmod (2^n)$$

$$= (2^{n-1} - 2) - (|X| + |Y|) \quad \text{necessário somar 1 para obter:}$$

$$= (2^{n-1} - 2) - (|X| + |Y|) + 1 = (2^{n-1} - 1) - (|X| + |Y|)$$

Quando é necessário somar 1?

Só quando ambos os números são negativos,

i.e. em ambos o bit de sinal = 1, logo **carry-out = 1**

Solução: somar o carry-out da posição mais significativa

ao resultado da soma – **“end-around carry”**

ABF - AC I - MIPS IS_2

5

Aritmética em 1's complement

$$-8 = 11110111$$

$$+7 = \underline{00000111}$$

$$11111110 \text{ (-1)}$$

$$-8 = 11110111$$

$$-7 = \underline{11111000}$$

$$11110111 \text{ (-16)}$$

end-around carry

$$\underline{1}$$

$$11110000 \text{ (-15)}$$

End-around carry ➡ Adição (subtração) lentas

ABF - AC I - MIPS IS_2

6

Inteiros com sinal: 2s-Complement

- Número com n-bits em 2s-Complement:

$$X = -x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots + x_12^1 + x_02^0$$

- Gama de representação: -2^{n-1} a $+2^{n-1} - 1$

- Exemplo

$$\begin{aligned} & \blacksquare 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1100_2 \\ & = -1 \times 2^{31} + 1 \times 2^{30} + \dots + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ & = -2,147,483,648 + 2,147,483,644 = -4_{10} \end{aligned}$$

- Gama de representação com 32 bits

$$\blacksquare -2,147,483,648 \text{ a } +2,147,483,647$$

ABF - AC I - MIPS IS_2

7

Inteiros com sinal: 2s-Complement (2)

- Bit 31 é o bit de sinal:
 - 1 para numeros negativos
 - 0 para numeros positivos e para zero
- Gama de representação assimétrica: (2^{n-1}) não é representável em n-bits
- Numeros não-negativos têm a mesma representação em *unsigned* e em *2's-complement*
- Exemplos:
 - 0: 0000 0000 ... 0000 (**0 tem representação única**)
 - 1: 1111 1111 ... 1111
 - Mais-negativo: 1000 0000 ... 0000
 - Mais-positivo: 0111 1111 ... 1111

ABF - AC I - MIPS IS_2

8

Obtenção do 2s Complemento

- Negar bit a bit e somar 1

- Exemplo:


- $+2 = 0000\ 0000 \dots 0010$
- $-2 = 1111\ 1111 \dots 1101 + 1$
 $= 1111\ 1111 \dots 1110$

ABF - AC I - MIPS IS_2

9

Aritmética em 2's complement

$$\begin{array}{rcl}
 -8 & = & 11111000 \\
 +7 & = & \underline{00000111} \\
 & & 11111111\ (-1)
 \end{array}
 \qquad
 \begin{array}{rcl}
 -8 & = & 11111000 \\
 -7 & = & \underline{11111001} \\
 & & 111110001\ (-15)
 \end{array}$$


 carry-out ignorado

- **Vantagens 2s complement**

- Representação única para 0
- Soma (e subtração) mais rápidas

➤ **2s complement é a representação usada por todos os computadores atuais**

ABF - AC I - MIPS IS_2

10

2s complement: Extensão do sinal

- Representar um numero usando mais bits
- Replicar o bit de sinal para a esquerda não altera o valor representado
- No instruction set do MIPS:
 - **addi**: valor do imediato estendido a 32-bits
 - **lb, lh**: estende a 32 bits o byte/halfword transferido da memória
 - **beq, bne**: estende o valor do imediato que indica o deslocamento
- Exemplos: 8-bit para 16-bit
 - +2: 0000 0010 => 0000 0000 0000 0010
 - 2: 1111 1110 => 1111 1111 1111 1110

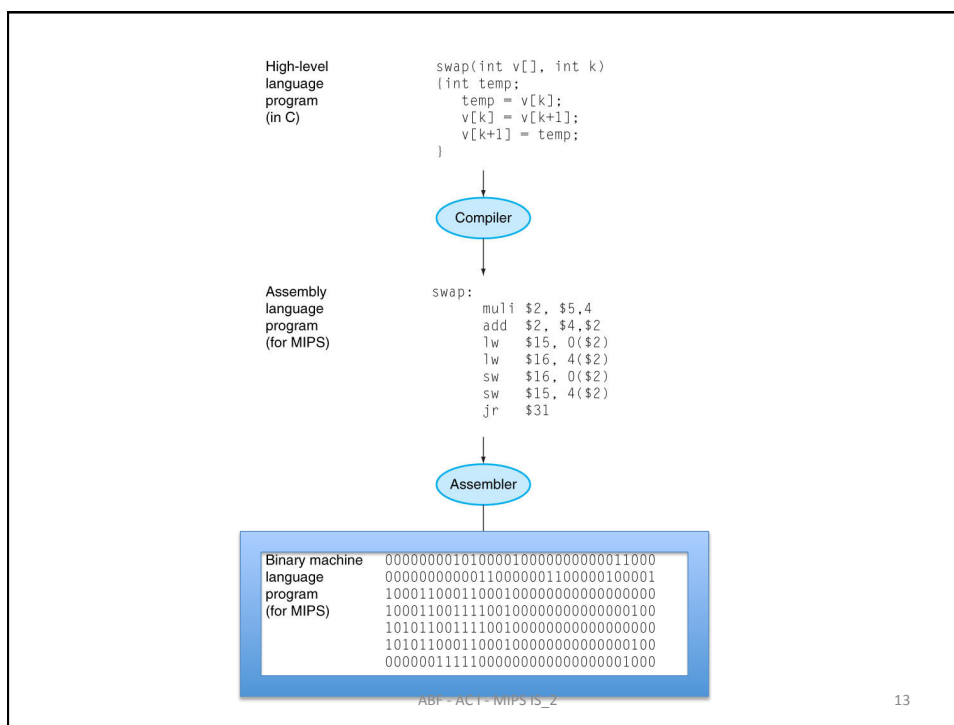
ABF - AC I - MIPS IS_2

11

9. Representação das instruções

ABF - AC I - MIPS IS_2

12



Codificação das instruções

Instruções codificadas em binário: *código máquina*

Instruções MIPS

Codificadas em 32-bits (*instruction words*)

- comprimento fixo
- Número reduzido de formatos de instrução

Codificam: código de operação (*opcode*), número dos registos, ...

Regularidade!

Número dos registos

\$t0 – \$t7 são os registos r8 – r15

\$t8 – \$t9 são os registos r24 – r25

\$s0 – \$s7 são os registos r16 – r23

ABF - AC1 - MIPS IS_2

14

Formatos de instrução: Tipo R

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

- Campos do código de Instrução:
 - **op**: operation code (opcode)
 - **rs**: número do registo do 1º operando
 - **rt**: número do registo do 2º operando
 - **rd**: número do registo de destino
 - **shamt**: shift amount (00000 por agora)
 - **funct**: function code (extende o opcode)

ABF - AC I - MIPS IS_2

15

Formato Tipo R: exemplo

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

add \$t0, \$s1, \$s2

special	\$s1	\$s2	\$t0	0	add
---------	------	------	------	---	-----

0	17	18	8	0	32
---	----	----	---	---	----

000000	10001	10010	01000	00000	100000
--------	-------	-------	-------	-------	--------

$00000010001100100100000000100000_2 = 02324020_{16}$

ABF - AC I - MIPS IS_2

16