

Aula 4

Relembrando

Expressões aritméticas operam sobre dois registradores e salvam o resultado em um registrador.

Instruções Imediatas

Operam entre um registrador e uma constante e são instruções presentes na arquitetura. A constante deve ser sempre o terceiro argumento da instrução.

Princípio de Design 3:

Torne o caso comum mais rápido.

Observação: Evite utilizar a constante zero, já que há o registrador \$zero disponível.

Exemplo: inicializar um registrador com zero:

```
add $s0, $zero, $zero
```

`move $s0, $zero` # Lembrando que `move` é uma pseudo instrução que não existe na arquitetura.

Representação de inteiros em base 2

Um registrador contém um binário de 32 bits. Como tratar esse valor como um inteiro?

- A representação binária é *posicional*:

Numa base b qualquer, um número:

$$xyzw = x \cdot b^3 + y \cdot b^2 + z \cdot b^1 + w \cdot b^0$$

Um registrador armazena 32 bits, chamamos bit mais a direita de um número de LSB (least significant bit) e o mais à esquerda de MSB (most significant bit).

O maior número que pode ser armazenado em um registrador de 32 bits é $2^{32} - 1$

Ou seja, o limite de um registrador de 32 bits é 4 bytes.

Representação de números negativos

Dado um inteiro x , seu complemento de 2 é obtido da seguinte maneira:

1. Nega-se o x bit a bit;
2. Soma-se 1;
3. O MSB representa o sinal
 - 0 - positivo

- 1 - negativo

Em uma arquitetura de n bits, tem-se que: $\bar{x} = 2^n - x$, onde \bar{x} representa o complemento de 2 do número x , e por esse motivo chama-se complemento de 2.

Observação: a representação do zero é única em complemento de 2.

Em uma representação com sinal, podemos representar:

- $2^{31} = 2.147.483.647$ números positivos e negativos, pois há 31 bits disponíveis para representação do número, já que o MSB indica o sinal.
- Positivos variam de 0 a 2.147.483.647;
- Negativos variam de -2.147.483.648 a -1.

Observações:

- O zero entra o universo dos positivos, já que seu MSB é 0.
- Em C, utilizamos `int` para representar inteiros com sinal e `unsigned int` para inteiros sem sinal.

Representação decimal:

-1 * 1011 com sinal =

Overflow

Exemplo: em uma arquitetura de 4 bits, somando os números 9 e 7 (decimais) sem sinal:

Extensão de sinal

Para passar um número binário de x bits para uma representação de y bits, sendo $y > x$, temos duas possibilidades:

- Representação sem sinal: completamos os bits à esquerda com 0.
- Representação com sinal: completamos os bits à esquerda com o bit de sinal (MSB) da representação em x bits.