

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

Qualificadores

prof. Dr. César Augusto M. Marcon
prof. Dr. Edson Ifarraguirre Moreno

Introdução

- Como operações aritméticas podem informar o fluxo de controle sobre o resultado de suas operações?
 - Com **qualificadores**! Mas o que é isto?
- **Note:** Comandos de alto nível são compilados para operações suportadas pela arquitetura alvo. Estas operações são executadas por unidades tais como os somadores descritos aqui. Exemplo:

```
if (X != Y)
    X++;
Y++;
```

Em uma linguagem de montagem do processador MIPS poderia ser implementado como

```
beq X, Y, Salta
inc X
Salta:
inc Y
```

Introdução

- **Como a instrução de montagem é implementada em hardware?**

Uma comparação, tal como descrito abaixo, pode ser implementada com uma subtração!

Caso subtração resulte em 0, então os valores subtraídos são iguais

beq X, Y, Salta

- **Como implementar uma subtração em hardware?**
 - Utilizando circuitos somadores e considerando operandos **com** sinal
- **Voltando e reformulando a pergunta inicial. Como, neste caso, os somadores podem informar o fluxo de controle sobre o resultado de suas operações?**
 - Utilizando uma porta de um bit, que informe se a operação resultou, ou não, em 0
 - Este bit é um qualificador, também chamado de **bit de status**, ou **flag**. Neste caso, a **flag de ZERO**

Introdução

- Com qualificadores poderíamos reler a descrição C/Java, descrita a seguir, com uma de linguagem de montagem equivalente no MIPS

```
if (X != Y)
    X++;
Y++;
```

```
beq X, Y, Salto
inc X
Salto:
inc Y
```

Da seguinte forma

```
Atribui 1 para Z se  $X + (-Y)$  der zero, senão atribui 0
Se Z for igual a 1 vai para Salto, senão segue o código
X recebe X mais 1
Salto:
Y recebe Y mais 1
```

- Neste caso Z é um qualificador que indica que a operação deu 0

Resumo

- **Existem mais qualificadores possíveis. Aqui analisaremos apenas 4, com suas abreviaturas**
 - Resultado da operação é zero (**Z**)
 - Qualificador **Z** recebe 1 quando todos os bits do resultado de uma operação forem 0, senão recebe 0
 - Resultado da operação é um número negativo (**N**)
 - Qualificador **N** recebe 1 quando o bit mais significativo do resultado de uma operação for 1, (representação de um número negativo em complemento de 2) senão recebe 0
 - Não foi possível representar o resultado da operação com números **sem** sinal (**carry**, ou apenas **C**)
 - Qualificador **C** recebe 1 quando o último estágio do somador gerar um “vai um”
 - Não foi possível representar o resultado da operação com números **com** sinal (**overflow**, ou apenas **Ov**, ou também **V**)
 - Qualificador **V** recebe 1 quando a soma de dois positivos der negativa, a soma de dois negativos der positiva, um positivo subtraído de um negativo der negativo, um negativo subtraído de um positivo der positivo, senão recebe 0
- **Conforme os tipos de dados que estamos trabalhando em alto nível, a informação dos qualificadores pode ser utilizada pelos compiladores para gerar linguagem de montagem adequada**
 - Qualificadores Z e C são usados em operações aritméticas de números sem sinal
 - Qualificadores Z e N são usados em operações lógicas
 - Qualificadores Z, N e V são usados em operações aritméticas de números com sinal

Revendo Representações Numéricas

- **Representação de números **sem** sinal**

- Notação posicional
- Cada bit tem exatamente o peso de sua posição
- O valor do bit (**b** = 0, 1) é multiplicado pela base (**2**) elevada na posição (**p**)
$$b * 2^p$$
- O valor do número é obtido com a soma de todos os bits com seus pesos parciais
- **Exercício:**
 - Calcule o intervalo de representação numérico de um inteiro de 16 bits
 - Verifique se é possível armazenar o número 1034 em 10 bits
 - Realize as seguintes somas em 8 bits (primeiro converta para número binário):
 - a. 129 + 76
 - b. 150 + 200
- Como saber se uma operação de número sem sinal teve um resultado correto?
 - Analisando o qualificador **carry**

Revendo Representações Numéricas

- **Representação de números com sinal**

- Em complemento de 2, temos a mesma notação posicional, porém o bit mais significativo vale menos, enquanto os demais valem mais

$-b * 2^p$, para o primeiro bit;

$b * 2^p$, para demais bits

- **Exercício:**

- Calcule o intervalo de representação numérico de um inteiro de 16 bits
- Verifique se é possível armazenar os seguintes números em 10 bits
 - a. -512
 - b. 345
- Realize as seguintes somas em 8 bits (primeiro converta para número binário):
 - a. -129 - 76
 - b. 150 - 200

- Como saber se uma operação de número com sinal teve um resultado correto?

- Analisando o qualificador **overflow**

Planejando Construir um Somador com Sinal

- **O que deve ser alterado no hardware de um somador, quando é considerado o sinal?**
 - Deve ser inserido o cálculo do qualificador de overflow (V)
- **Entendendo melhor**
 - Overflow significa que uma operação aritmética com sinal não teve o resultado correto, pois não foi possível representar todo o número.
 - Situações em que isto pode acontecer:
 - Soma de dois inteiros positivos e resultado um inteiro negativo
 - Soma de dois inteiros negativos e resultado um inteiro positivo
 - Subtrair um inteiro positivo de um inteiro negativo e o resultado der negativo
 - Subtrair um inteiro negativo de um inteiro positivo e o resultado der positivo
- **Como obter o cálculo deste qualificador?**
 - Com ou exclusivo entre o carry do último estágio e do penúltimo estágio

Exercício

1. Implementar um somador de 4 bits com sinal, levando em consideração os qualificadores necessários
2. Implementar um somador de 4 bits com e sem sinal, levando em consideração os qualificadores necessários