Universidade de Brasília — Faculdade UnB Gama

Fundamentos de Arquitetura de Computadores - Turma B

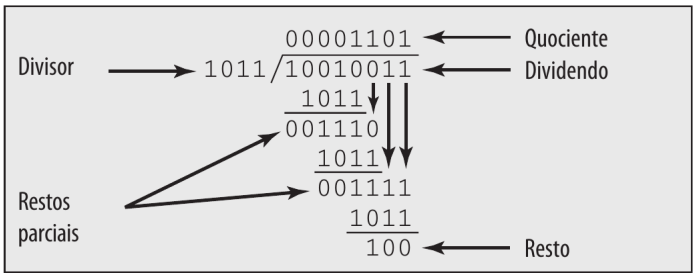
Trabalho 2

Algoritmo para divisão de inteiros

As operações de multiplicação e divisão com números binários são não triviais e requerem algoritmos bem pensados para aproveitar recursos de tempo e de *hardware* da melhor maneira possível.

Como vimos em aula, há dois bons algoritmos que exploram tais características, um para multiplicação, o algoritmo de Booth, e outro para divisão. Neste trabalho, exploraremos o algoritmo de divisão.

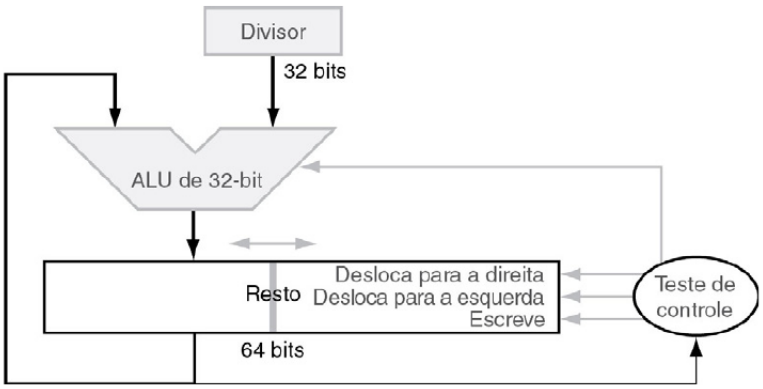
A forma como dividimos no papel é ilustrada na Figura 1.



Fonte: [2, Figura 9.15].

Figura 1: Exemplo de divisão de inteiros sem sinal.

O algoritmo mais ingênuo, o primeiro para divisão que vimos em sala de aula, usa-se de dois registradores de 64 bits e um de 32 bits. Após alguns refinamentos, chegamos à arquitetura ilustrada na Figura 2, que usa-se de apenas um registrador de 64 bits e outro de 32 bits.



Fonte: [1, Figura 3.13].

Figura 2: Versão refinada do *hardware* para divisão.

Utilizando-se do *hardware* proposto, para divisão de inteiros positivos sem sinal, aplicamos o Algoritmo 1.

Algoritmo 1: Algoritmo refinado para divisão.

Dados: o dividendo e o divisor, inteiros positivos, em representação sem sinal.

Passo 1. Salve o dividendo no registrador RESTO, o divisor no registrador DIVISOR e defina contador = 1.

Passo 2. Faça um deslocamento de 1 bit à esquerda no RESTO.

Passo 3. Subtraia o DIVISOR dos 32 bits mais significativos do RESTO, e salve o resultado nos 32 bits mais significativos do RESTO.

Passo 4. Faça os seguintes passos.

Passo 4.1. Se RESTO ≥ 0, faça um deslocamento de 1 bit à esquerda no RESTO e defina o bit menos significativo do RESTO como sendo 1.

Passo 4.2. Se RESTO < 0, restaure o valor original dos 32 bits mais significativos do RESTO (adicionando ao valor atual o DIVISOR) e faça um deslocamento de 1 bit à esquerda no RESTO.

Passo 5. Se contador ≤ 32, faça contador=contador+1 e volte ao Passo 3.

Passo 6. Faça um deslocamento de 1 bit à direita apenas nos 32 bits mais significativos do RESTO.

O presente trabalho tem por objetivos

- o exercício de conceitos fundamentais de assembly MIPS e
- o exercício de entendimento e fixação do algoritmo refinado de divisão.

Para tanto, faremos uso do simulador MARS[1].

Este trabalho compõe a média semestral de trabalhos e é individual!

Sua tarefa neste trabalho é **implementar um programa** em assembly MIPS que leia dois inteiros a e b e devolva a divisão inteira de a por b, bem como o resto dessa divisão. Os **requisitos mínimos** que seu programa **deve** cumprir são:

1. Você deve implementar o Algoritmo 1.
2. Seu programa deve ser implementado em assembly MIPS e funcionar no simulador MARS versão 4.5.
3. Seu programa deve ser capaz de efetuar corretamente a divisão tanto de inteiros sem sinal quanto de inteiros com sinal.
4. Seu programa deve conter uma função **divfac** que recebe como argumentos a e b e retorna o resto no registrador *Hi* e o quociente no registrador *Lo*. Essa função deve retornar 0 se a divisão foi bem sucedida ou 1 caso contrário.

[1] Disponível em http://courses.missouristate.edu/KenVollmar/mars/.

5. Seu programa deve conter uma função **main** que faça a leitura dos números, chame a função **divfac** e imprima o resultado da divisão na tela ou uma mensagem dizendo que não foi possível efetuar a divisão.

   (a) Capriche na leitura dos números! Coloque mensagens como, por exemplo,

   Digite o dividendo: 8
   Digite o divisor: 3

   (b) Capriche na saída! É desejável que seu programa imprima algo do tipo (para a entrada acima, por exemplo)

   $8/3 = 2$
   $8\%3 = 2$

Depois de ter cumprido todos os requisitos, **teste** seu código. Invente casos de teste, isso também é sua tarefa neste trabalho.

Depois de ter cumprido todos os requisitos e testado seu código, você deve submeter apenas seu código fonte, com a extensão `asm` e nomeado da seguinte forma

nome_sobrenome_matricula_**trab02**.asm

no link Entrega do Trabalho 2 na página da nossa disciplina no Moodle até o dia **10 de maio de 2019** às **23:55**.

Será avaliado no seu trabalho

1. a corretude do seu código em assembly MIPS, ou seja
   - seu código roda sem erros,
   - não contém uso indevido de funções e estruturas e
   - funciona,
2. o capricho no código e na indentação,
3. o cumprimento de todos os requisitos pelo seu programa,
4. o envio correto na plataforma Aprender e
5. a pontualidade na entrega do trabalho.

Farei uma correção manual, então seu código não vai rodar em juiz eletrônico. Se for detectado algum plágio, a nota atribuída será ZERO a **todos** os envolvidos.

Bons estudos!

Prof. John Lenon Gardenghi
john.gardenghi@unb.br
Sala 22-UED

Referências

[1] PATTERSON, D. A.; HENNESSY, J. L. **Organização e projeto de computadores**. 3 ed. Elsevier, 2005.

[2] STALLINGS, W. **Arquitetura e organização de computadores**. 8 ed. Prentice Hall. 2010.