Projeto - MC404

Rian Radeck Santos Costa - 187793
Cirilo Max Macêdo de Morais Filho - 168838

30 de Junho de 2022

Sumário

1 Esclarecimentos

Todas as operações e conversões foram preparadas para inteiros sinalizados de 32 bits.

Nós decidimos adicionar uma operação de exponenciação pios achamos que poderia agregar positivamente no conjunto do trabalho.

O projeto foi dividido em duas partes:

- Operações (projeto1.s)
- Conversões (projeto2.s)

essa divisão foi feita pois o simulador RISC-V utilizado não aceita códigos muito extensos.

Definições:

- n é a quantidade de bits de a_0.
- m é a quantidade de bits de a_1.

2 Operações

Estas operações estão no arquivo projeto1.s, elas foram feitas baseadas com o conhecimento adiquirido dentro de sala e com base no material fornecido pelo professor Ricardo Pannain no classroom.

2.1 Soma

A operação feita no progrma é $a_0 \leftarrow a_0 + a_1$.

O overflow foi verificado somando os valores absolutos dos registradores e caso os seus sinais fossem iguais, o resultado não poderia ser negativo.

Complexidade: $O(1)$

2.2 Subtração

A operação feita no progrma é $a_0 \leftarrow a_0 - a_1$.

A execução dessa operação é feita invertendo o sinal do registrador a_1 e chamando a função soma.

Complexidade: $O(1)$

2.3 Multiplicação

A operação feita no programa é $a_0 \leftarrow a_0 \times a_1$.

O algoritmo utilizado foi o algoritmo de Booth, sendo assim o tratamento de sinal era feito no fim do algoritmo, após a multiplicação dos fatores. O overflow foi verificado em cada parte da soma das parcelas da multiplicação, assim se em algum momento o produto se tornasse negativo havia ocorrido o overflow.

Complexidade: $O(n)$.

2.4 Divisão

A operação feita no programa é $a_0 \leftarrow \left\lfloor \frac{a_0}{a_1} \right\rfloor$ e $a_1 \leftarrow a_0 \mod a_1$

O algoritmo utilizado foi o seguinte:

- Verifica se o divisor é 0.
- Se o divisor for menor ou igual ou dividendo.

  - Acrescenta 1 no quociente.
  - Subtrai o divisor dos bits considerados até o memento do dividendo.

- Caso contrário
  - Acrescenta 0 no quociente.
  - É considerado o próximo bit mais significativo do dividendo

- Quando não houver mais bits para se considerar do dividendo, o que sobrou do dividendo é o resto

Complexidade: $O(n)$

2.5 Exponenciação

A operação feita no programa é $a_0 \leftarrow {a_0}^{a_1}$

O algoritmo utilizado foi o de exponenciação rápida. Ele funciona da seguinte maneira:

- Caso o expoente seja 0, o resultado é 1
- Caso o expoente seja par, dividimos por 2 e calculamos o seu quadrado com uma multiplicação
- Caso o expoente seja ímpar, subtraimos 1 do expoente e calculamos o caso par, depois multiplicamos pela base.

Complexidade: $O(n \log_2 m)$

3 Conversões

Algoritmo Sabemos que um número é representado em certa base da seguinte maneira:

$$N = c_0 \times b^0 + c_1 \times b^1 + c_2 \times b^2 + \cdots$$

onde b é a base e c_0 é seu algarismo menos significativo e $0 \leq c_k < b \; \forall \; k \in Z$.

O algoritmo usado na conversão de bases foi o seguinte:

- Tome um expoente e grande o suficiente ($N \leq b^e$).
- Tome o seu algarismo c como $b - 1$.
- Diminua seu algarismo até $c \times b^e \leq N$.
- Acrescente esse algarismo na resposta.
- Diminua N de $c \times b^e$.
- Subtraia 1 do expoente
- Se o expoente é negativo, o algoritmo para.

Todas as conversões foram feitas utilizando esse algoritmo.

3.1 Binário

- Leitura: string.
- Escrita: string.

3.2 Hexadecimal

- Leitura: string.
- Escrita: cadeia de caracteres.

3.3 Decimal

- Leitura: RISC-V.
- Escrita: RISC-V.