"Não há razão para qualquer indivíduo ter um computador em casa" (Ken Olsen, 1977).

Ponto Flutuante - Experimentos

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Coprocessadores

Ao adicionar a capacidade de processar pontos flutuantes, é necessário decidir quais registradores usar.

Podemos usar os mesmos registradores de “aritmética convencional”.

Coprocessadores

Ao adicionar a capacidade de processar pontos flutuantes, é necessário decidir quais registradores usar.

Podemos usar os mesmos registradores de “aritmética convencional”.

Muitos projetistas optam por usar um conjunto separado de registradores, específico para ponto flutuante.

Coprocessadores

Ao adicionar a capacidade de processar pontos flutuantes, é necessário decidir quais registradores usar.

Podemos usar os mesmos registradores de “aritmética convencional”.

Muitos projetistas optam por usar um conjunto separado de registradores, específico para ponto flutuante.

Exemplos:

x86-64
ARM
MIPS

Coprocessadores

Conjunto separado de registradores, específico para ponto flutuante.

Vantagens? Desvantagens?

Coprocessadores

Conjunto separado de registradores, específico para ponto flutuante.

+ Mais registradores para trabalhar.
+ Pode liberar a ALU para trabalhar com os registradores “normais” enquanto uma ALU dedicada trabalha com os registradores de ponto flutuante.

- São necessárias instruções específicas para lidar com os novos registradores.
- Muitas vezes é necessário mover os dados entre os bancos de registradores convencionais e de PF.

Coprocessadores

É necessária também uma ALU especializada para tratar os dados em ponto flutuante.

Essa ALU é comumente chamada de Floating-point Unit - **FPU.**

Um pouco de história

Processadores anteriores à década de 90 não tinham uma miniaturização boa o suficiente para comportar a unidade de ponto flutuante.

Essa unidade era comumente vendida como um chip separado, chamado de coprocessador de ponto flutuante.

Um pouco de história

Processadores anteriores à década de 90 não tinham uma miniaturização boa o suficiente para comportar a unidade de ponto flutuante.

Essa unidade era comumente vendida como um chip separado, chamado de coprocessador de ponto flutuante.

Na maioria dos processadores atuais, esse "coprocessador" fica no mesmo chip da CPU, mas o nome persiste.

Exemplo: a instrução do MIPS de mnemônico lwc1 significa "*Load Word to Coprocessor 1*".

Um pouco de história

Na família de processadores x86, o primeiro “coprocessador matemático” era o 8087.

Dessa forma, a FPU do x86 ficou conhecida por x87.

Até nos dias de hoje, é comum encontrar nos manuais as instruções do x86-64 as instruções de ponto flutuante como *conjunto de instruções x87*.

MIPS

No MIPS32, a unidade de ponto flutuante conta com 32 registradores separados.

$f0 até $f31.

Registradores de 32 bits.

MIPS

No MIPS32, a unidade de ponto flutuante conta com 32 registradores separados

$f0 até $f31.

Registradores de 32 bits.

Para operar com precisão dupla, somente os registradores pares podem ser usados.

A união de um registrador par com um ímpar forma o valor de 64 bits.

Exemplo: a união dos registradores $f0 e $f1 formam o registrador de precisão dupla $f0.

Syscalls - Relembrando

Tabela de syscalls do MIPS no MARS:

courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html

Um exemplo de programa

```
.text
.globl main
main:
	li $v0, 6
	syscall
	mov.s $f1, $f0
	li $v0, 6
	syscall
	add.s $f12, $f0, $f1
	li $v0, 2
	syscall

	li $a0, 10
	li $v0, 11
	syscall

	cvt.w.s $f0, $f12
	mfc1 $a0,$f0
	li $v0, 1
	syscall
end:
	li $v0, 10
	syscall
```

Para relembrar: Tabela de syscalls do MIPS no MARS
courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html

Um exemplo de programa

```
.text
.globl main
main:
      li $v0, 6
      syscall
      mov.s $f1, $f0
      li $v0, 6
      syscall
      add.s $f12, $f0, $f1
      li $v0, 2
      syscall

      li $a0, 10
      li $v0, 11
      syscall

      cvt.w.s $f0, $f12
      mfc1 $a0,$f0
      li $v0, 1
      syscall
end:
      li $v0, 10
      syscall
```

Um exemplo de programa

```
.text
.globl main
main:
      li $v0, 6
      syscall
      mov.s $f1, $f0
      li $v0, 6
      syscall
      add.s $f12, $f0, $f1
      li $v0, 2
      syscall

      li $a0, 10
      li $v0, 11
      syscall

      cvt.w.s $f0, $f12
      mfc1 $a0,$f0
      li $v0, 1
      syscall
end:
      li $v0, 10
      syscall
```

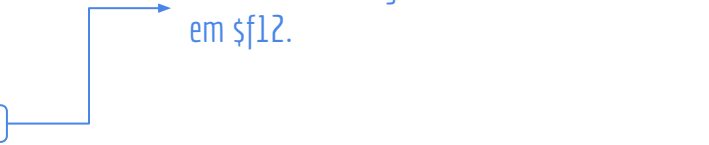
Copie o single do registrador $f0 para $f1.

Um exemplo de programa

```
.text
.globl main
main:
      li $v0, 6
      syscall
      mov.s $f1, $f0
      li $v0, 6
      syscall
      add.s $f12, $f0, $f1
      li $v0, 2
      syscall

      li $a0, 10
      li $v0, 11
      syscall

      cvt.w.s $f0, $f12
      mfc1 $a0,$f0
      li $v0, 1
      syscall
end:
      li $v0, 10
      syscall
```

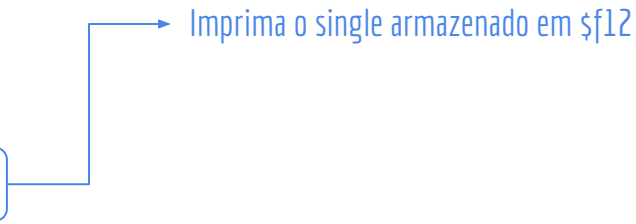


Um exemplo de programa

```
.text
.globl main
main:
      li $v0, 6
      syscall
      mov.s $f1, $f0
      li $v0, 6
      syscall
      add.s $f12, $f0, $f1
      li $v0, 2
      syscall

      li $a0, 10
      li $v0, 11
      syscall

      cvt.w.s $f0, $f12
      mfc1 $a0,$f0
      li $v0, 1
      syscall
end:
      li $v0, 10
      syscall
```



Um exemplo de programa

```
.text
.globl main
main:
      li $v0, 6
      syscall
      mov.s $f1, $f0
      li $v0, 6
      syscall
      add.s $f12, $f0, $f1
      li $v0, 2
      syscall

      li $a0, 10
      li $v0, 11
      syscall

      cvt.w.s $f0, $f12
      mfc1 $a0,$f0
      li $v0, 1
      syscall
end:
      li $v0, 10
      syscall
```

Converta o valor de single para inteiro, e armazene em $f0.

Um exemplo de programa

```
.text
.globl main
main:
	li $v0, 6
	syscall
	mov.s $f1, $f0
	li $v0, 6
	syscall
	add.s $f12, $f0, $f1
	li $v0, 2
	syscall

	li $a0, 10
	li $v0, 11
	syscall

	cvt.w.s $f0, $f12
	mfc1 $a0,$f0
	li $v0, 1
	syscall
end:
	li $v0, 10
	syscall
```

Mova o valor do registrador $f12 do Coprocessador 1 (ponto flutuante) para o registrador $a0.

Aritmética de Ponto Flutuante

Como visto na aula passada, dada a representação, cálculos podem gerar erros de precisão.

Devido ao arredondamento, a quantidade de ciclos necessários para se efetuar o cálculo também pode variar.

Detalhes sobre os erros e como as operações são feitas são dados em.

Patterson e Hennessy (2017).

Null e Lobur (2009).

Em Ruggiero e Lopes (1998) é discutido o erro numérico causado pela aritmética de ponto flutuante.

Tratado ainda na disciplina *Introdução à Computação Científica*.

Aritmética de Ponto Flutuante

Os erros numéricos causados pela aritmética de ponto flutuante não serão tratados nessa disciplina.

Mas vale a pena realizar alguns experimentos e discutir os problemas, mesmo que de maneira informal.

Falha Catastrófica

25 Fev. 1991, Guerra do Golfo.

Sistema antimísseis Patriot.

O sistema falhou ao tentar interceptar um míssil Scud saudita lançado contra uma base americana.

28 Pessoas morreram e 100 ficaram feridas.

Falha Catastrófica

Causa: sistemas de radar dependem fortemente do tempo (de relógio).

O tempo era computado a cada 0.1 segundos, e armazenado utilizando 24 bits.

Aprendemos que isso não pode ser representado perfeitamente em binário.

O erro se acumulou durante 24 horas, até que não foi mais possível calcular com precisão a rota dos mísseis.

https://www-users.cse.umn.edu/~arnold/disasters/patriot.html
http://www.cs.unc.edu/~smp/COMP205/LECTURES/ERROR/lec23/node4.html
https://www-users.cse.umn.edu/~arnold/disasters/Patriot-dharan-skeel-siam.pdf

Especialistas da internet

Com o advento da internet temos agora um boom de “especialistas”.

Dois Exemplos:

www.guj.com.br/t/erro-de-precisao-float-e-double/39142

computingat40s.wordpress.com/java-float-and-double-primitive-types-are-evil-dont-use-them

O que você consegue enxergar de errado nos posts?

Especialistas da internet

Com o advento da internet temos agora um boom de “especialistas”.

Dois Exemplos:

www.guj.com.br/t/erro-de-precisao-float-e-double/39142

computingat40s.wordpress.com/java-float-and-double-primitive-types-are-evil-dont-use-them

O que você consegue enxergar de errado nos posts?

Tudo! (ps.: esses posts estavam nos primeiros resultados de pesquisa quando procurei por “problemas com float”).

Especialistas da internet

Com o advento da internet temos agora um boom de “especialistas”.

Dois Exemplos:

www.guj.com.br/t/erro-de-precisao-float-e-double/39142
computingat40s.wordpress.com/java-float-and-double-primitive-types-are-evil-dont-use-them

O que você consegue enxergar de errado nos posts?

Tudo! (ps.: esses posts estavam nos primeiros resultados de pesquisa quando procurei por “problemas com float”).

Desafio: encontre mais pérolas para postarmos na página do Moodle.

Encontre o erro

```
int main(){
    float valor = 0;
    //... faz algumas operações

    if(valor == 2.0){
      //faz algo
    }

    return 0;
}
```

Ponto fixo

Quando precisamos garantir que os valores são representados de maneira exata, podemos utilizar bibliotecas para ponto fixo, ou criar a nossa própria biblioteca

Ideia: Representar os valores antes de depois da vírgula de maneira separada, utilizando múltiplos bytes se necessário

Ponto fixo

Quando precisamos garantir que os valores são representados de maneira exata, podemos utilizar bibliotecas para ponto fixo, ou criar a nossa própria biblioteca

Ideia: Representar os valores antes de depois da vírgula de maneira separada, utilizando múltiplos bytes se necessário

Pelo menos 1 byte para armazenar o 3

Exemplo: 3,402823466385288598117041834845169254 40

Pelo menos 16 bytes para armazenar a parte decimal como um “inteiro”.

Ponto fixo

Quais são as vantagens e desvantagens do ponto fixo?

Ponto fixo

Quais são as vantagens e desvantagens do ponto fixo?

+ Podemos armazenar valores com qualquer precisão (desde que a memória seja o suficiente);
+ Os resultados obtidos são fiéis aos que um humano (que trabalha no mundo decimal) espera;
- Custo de memória;
- Custo de processamento.

Ponto Fixo E Precisão Arbitrária

Existem bibliotecas prontas nas mais variadas linguagens para lidar com valores em ponto fixo e precisão arbitrária (Você pode especificar quantas casas decimais precisa).

Exemplos:

C++

Boost.Multiprecision -› www.boost.org/doc/libs/1_66_0/libs/multiprecision/doc/html/index.html
GNU Multiple Precision Arithmetic Library -› https://gmplib.org

Java

BigDecimal

C#

Decimal

Veja uma lista -› en.wikipedia.org/wiki/List_of_arbitrary-precision_arithmetic_software

Quando usar

Utilizar ou não o IEEE 754 depende do problema que você tem em mãos.

Na grande maioria dos problemas, os pequenos erros numéricos introduzidos pelo ponto flutuante são irrelevantes.

Em alguns problemas podemos precisar utilizar bibliotecas para ponto fixo ou de precisão arbitrária.

Quando usar

Utilizar ou não o IEEE 754 depende do problema que você tem em mãos.

Na grande maioria dos problemas, os pequenos erros numéricos introduzidos pelo ponto flutuante são irrelevantes.

Em alguns problemas podemos precisar utilizar bibliotecas para ponto fixo ou de precisão arbitrária.

Muitos “especialistas” da internet e afins dizem para sempre usar essas bibliotecas.

Péssima ideia.

Custa muito caro!

Teste você mesmo

Versão com Double

```java
public class MainDouble{
	public static void main(String args[]){
		//teste com doubles
		double teste = 0.0;
		for(int i=0; i < 1000000; i++){//fazendo 1M vezes para medir o tempo
			teste = 0.1;
			for(int j=0; j<9; j++)
				teste+=0.1;
		}
		System.out.println("Resultado com double: " + teste);

		System.out.println("Memória Utilizada (MB): " +
			(Runtime.getRuntime().totalMemory() -
			Runtime.getRuntime().freeMemory())/1024/1024);
	}
}
```

Teste você mesmo

Versão com BigDecimal

```java
import java.math.BigDecimal;

public class MainBigDecimal{
	public static void main(String args[]){
		BigDecimal zeroUm = new BigDecimal("0.1");
		BigDecimal teste = null;
		for(int i=0; i < 10000000; i++){//fazendo 1M vezes para medir o tempo
			teste = zeroUm;
			for(int j=0; j<9; j++)
				teste=teste.add(zeroUm);
		}

		System.out.println("Resultado com BigDecimal: " + teste);
		System.out.println("Memória Utilizada (MB): " +
				(Runtime.getRuntime().totalMemory() -
				Runtime.getRuntime().freeMemory())/1024/1024);
	}
}
```

Teste você mesmo

Atenção: Essa é apenas uma estimativa da memória utilizada. O coletor de lixo pode passar no meio do caminho. A quantidade de memória realmente utilizada pode ser **muito** maior.

```java
import java.math.BigDecimal;

public class MainBigDecimal{
	public static void main(String args[]){
		BigDecimal zeroUm = new BigDecimal("0.1");
		BigDecimal teste = null;
		for(int i=0; i < 10000000; i++){//fazendo 1M vezes para medir o tempo
			teste = zeroUm;
			for(int j=0; j<9; j++)
				teste=teste.add(zeroUm);
		}

		System.out.println("Resultado com BigDecimal: " + teste);
		System.out.println("Memória Utilizada (MB): " +
				(Runtime.getRuntime().totalMemory() -
				Runtime.getRuntime().freeMemory())/1024/1024);
	}
}
```

Para compilar e executar

Compilar

javac MainDouble.java

Será gerado um arquivo de classe jamado MainDouble.class

Executar

time java MainBigDecimal

O comando *time* estima o tempo de execução do programa

Veja a saída “real” do comando

Resultados

Resultados na minha máquina

Resultado com double: 0.9999999999999999
Memória Utilizada (MB): 1

real 0m0,054s
user 0m0,066s
sys 0m0,004s

Resultado com BigDecimal: 1.0
Memória Utilizada (MB): 350 **(350x mais memória)**

real 0m0,448s **(8x mais tempo)**
user 0m0,396s
sys 0m0,097s

Resultados

Resultados na minha máquina

Resultado com double: 0.9999999999999999
Memória Utilizada (MB): 1

real 0m0,054s
user 0m0,066s
sys 0m0,004s

Resultado com BigDecimal: 1.0
Memória Utilizada (MB): 350 **(350x mais memória)**

real 0m0,448s **(8x mais tempo)**
user 0m0,396s
sys 0m0,097s

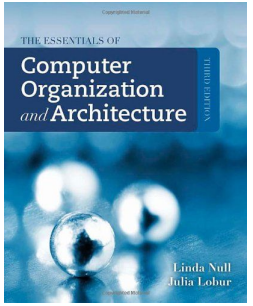
Gasto energético desnecessário (da máquina executando, refrigeração, ...)

Exercícios

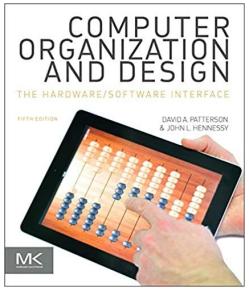
1. Assista a esse vídeo: https://www.youtube.com/watch?v=PZRI1IfStY0
2. Faça um programa em assembly do MIPS que solicita indefinidos valores em ponto flutuante de precisão simples do. O programa deve parar de solicitar valores quando um valor negativo foi digitado. Ao final, você deve exibir o número de valores digitados, e a média dos valores.
3. É comum a utilização de bibliotecas de ponto fixo ou de precisão arbitrária para representar valores monetários. Você consegue pensar numa solução melhor e mais eficiente do que isso para, por exemplo, representar o salário de uma pessoa?

Referências

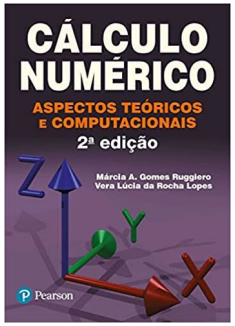
Null, Lobur. The Essentials of Computer Organization and Architecture. 2014.



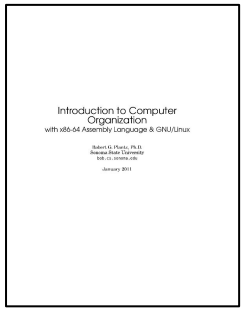
Patterson, Hennessy . Arquitetura e Organização de Computadores: A interface hardware/software. 2014.



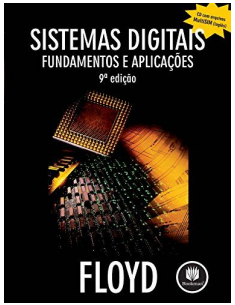
Ruggiero, Lopes. Cálculo numérico: aspectos teóricos e computacionais. 1996.



Plantz. Introduction to Computer Organization with x86-64 Assembly Language & GNU/Linux. 2011.



Floyd. Sistemas Digitais: Fundamentos e Aplicações. 2009.



Licença