

Desempenho do computador

- TEMPO, TEMPO, TEMPO!!!!
- Tempo de resposta (latência)
 - Quanto tempo leva para meu trabalho ser realizado?
 - Quanto tempo leva para realizar um trabalho específico?
 - Quanto tempo preciso esperar para finalizar minha simulação?
- *Vazão (throughput)*
 - Quantos trabalhos a máquina pode realizar em um intervalo de tempo?
 - Qual é a velocidade média de execução ?
 - Quanto trabalho está sendo feito?
- Tempo de resposta vs Vazão
 - Se atualizarmos uma máquina com um novo processador, em que melhoramos?

- Se acrescentarmos uma máquina ao laboratório, em que melhoramos?

-

Tempo de execução

- Tempo decorrido (*real time*)
 - Conta tudo (*acessos a disco e a memória, E/S etc.*)
 - Um número útil, mas normalmente não é ideal para fins de comparação
- Tempo de CPU (*user time + system time*)
 - Não conta E/S ou tempo gasto executando outros programas
pode ser dividido em tempo de sistema e tempo de usuário
- Nosso foco: tempo de CPU do usuário (*user time*)
 - Tempo gasto executando as linhas de código que estão em
nosso programa desconsiderando chamadas de sistema e
tratamento por parte do SO

- Exemplo: Comando time do linux

Ciclos de relógio (clock)

- Em vez de informar o tempo de execução em segundos, normalmente usamos ciclos

$$\underline{\text{segundos}} = \text{ciclos} \quad \square \quad \underline{\text{segundos}}_{\text{programa}} \text{ programa } \text{ciclos}$$

- Os "tiques" de clock indicam quando iniciar as atividades (uma abstração):



- Tempo de ciclo = tempo entre os tiques = (fração de) segundos por ciclo
- Velocidade de clock (frequência) = número de ciclos por segundo

Um clock de 4Ghz possui um período de ciclo de(ps)

Ciclos de relógio (clock)

- Em vez de informar o tempo de execução em segundos, normalmente usamos ciclos

$$\underline{\text{segundos}} = \text{ciclos} \quad \square \quad \underline{\text{segundos}} \text{ programa} \text{ programa} \text{ ciclos}$$

- Os "tiques" de clock indicam quando iniciar as atividades (uma abstração):



- Tempo de ciclo = tempo entre os tiques = segundos por ciclo
- Velocidade de clock (frequência) = ciclos por segundo

$$\frac{1}{4 \times 10^9} \times 10^{12} = 250 \text{ picoseconds (ps)}$$

Um clock de 4Ghz possui um período de ciclo de

$$\mathbf{1 \text{ Kilo} = 10^3; 1 \text{ Mega} = 10^6; \text{Giga} = 10^9; \text{Tera} = 10^{12}}$$

$$1 \text{ seg.} = 10^3 \text{ ms} = 10^6 \text{ us} = 10^9 \text{ ns} = 10^{12} \text{ ps}$$

Como melhorar o desempenho

$$\text{Tempo de Exec. de programa} = \frac{\text{Número de ciclos de programa}}{\text{Tempo de um ciclo}}$$

- Portanto, para melhorar a quantidade de segundos para a execução de um programa (tudo mais sendo igual), você pode (aumentar ou diminuir?)

_____ o número de ciclos necessários para um programa, ou

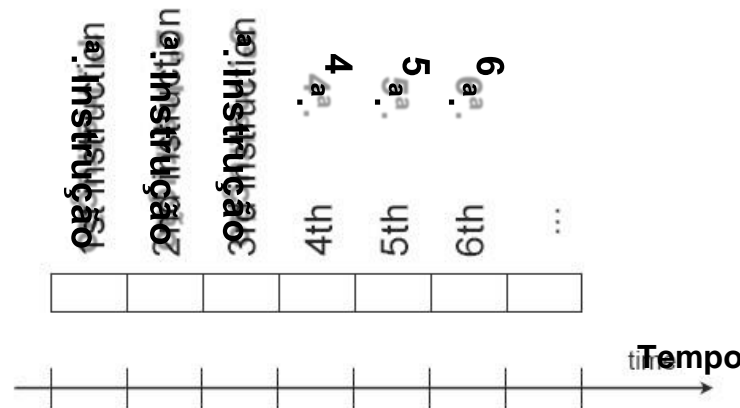
_____ o período de duração de um ciclo de clock ou, dito de outra maneira,

_____ a velocidade de clock.

- Desta forma tem-se:
- Tempo de execução = nro de ciclos de clock * período do clock ou nro de ciclos de clock / frequência do clock

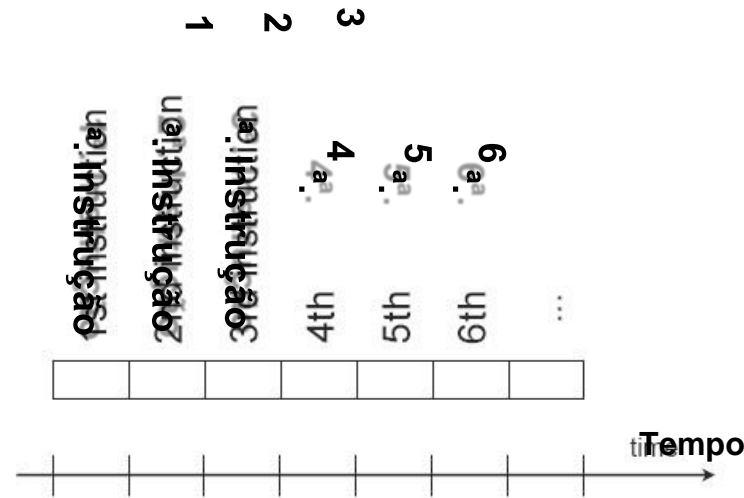
Ciclos necessários por programa

- Poderíamos considerar que o número de ciclos é igual ao número de instruções 😊 ?



Ciclos necessários por programa

- Poderíamos considerar que o número de ciclos é igual ao número de instruções ☹ !



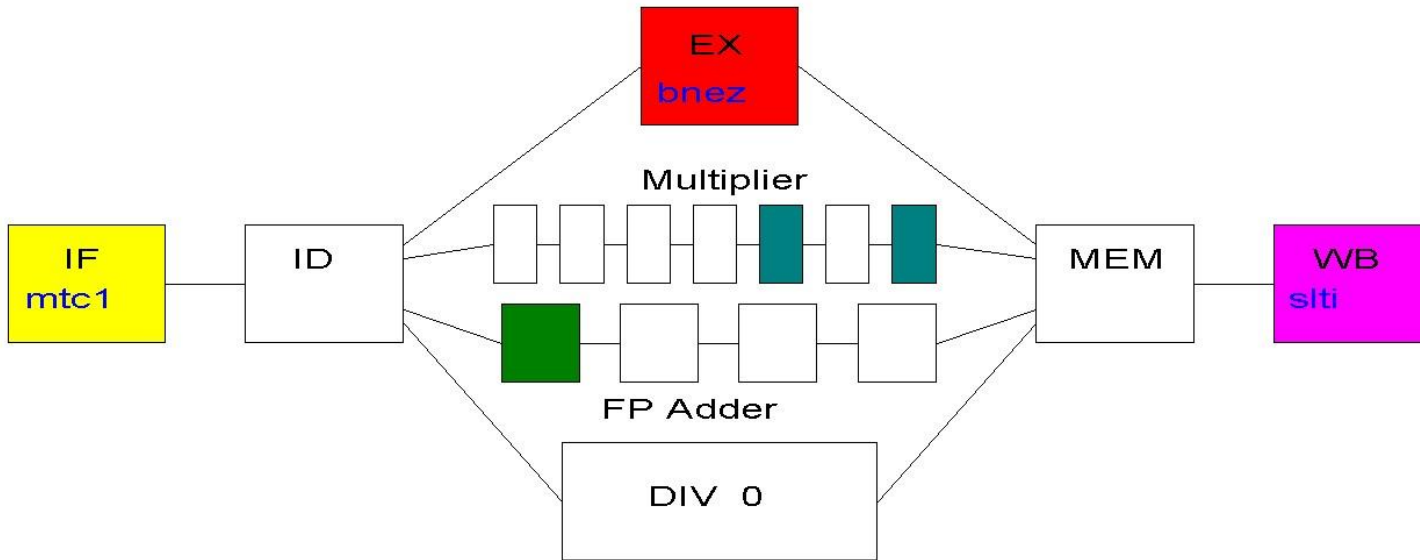
Essa suposição é incorreta, pois:

Diferentes instruções levam a diferentes períodos em diferentes máquinas

Diversidade arquitetural

- Por exemplo:
- Multiplicação vs adição
- Operações de ponto flutuante vs operações de inteiros

- Acessar a memória *vs* acesso a registradores



Exemplo

- Nosso programa favorito é executado em 10 segundos no computador A, que possui uma frequência de relógio de 4MHz.
- Estamos tentando ajudar um projetista de computador a construir uma nova máquina B, que execute esse programa em 6 segundos.
- O projetista determinou que um aumento substancial na velocidade de clock é possível, mas esse aumento afetará o restante do projeto da CPU, fazendo com que

o computador B exija 1,2 vez mais ciclos de relógio do que o computador A para esse programa.

- Que frequência de relógio/velocidade de clock devemos pedir para que o projetista almeje?
- Lembrando:
- Tempo de execução = nro de ciclos de clock / frequência do clock
- 1 seg. = 10^3 ms = 10^6 us = 10^9 ns = 10^{12} ps
- 1 Hz = 1 ciclo/seg., 1KHz = 10^3 ciclo/seg., 1MHz = 10^6 ciclo/seg.

Exemplo - Solução

- $T_{execA} = 10\text{seg}$; $NroCiclosA = ?$; $FreqA = 4 * 10^6$
- $T_{execB} = 6\text{seg}$; $NroCiclosB = 1,2 * NroCiclosA$; $FreqA = ?$

$$T_{execA} = NroCiclosA / FreqA$$

$$10 = NroCiclosA / 4 * 10^6$$

$$NroCiclosA = 40 * 10^6$$

$$T_{\text{ExecB}} = \text{NroCiclosB} / \text{FreqB}$$

$$6 = (1,2 * \text{NroCiclosA}) / \text{FreqB}$$

$$\text{FreqB} = 8 * 10^6 \text{ ou seja } 8\text{MHz}$$

- Lembrando:
- Tempo de execução = nro de ciclos de clock / frequência do clock
- 1 seg. = 10^3 ms = 10^6 us = 10^9 ns = 10^{12} ps
- 1 Hz = 1 ciclo/seg., 1KHz = 10^3 ciclo/seg., 1MHz = 10^6 ciclo/seg.

Tendo sido entendido os ciclos

- Um dado programa vai exigir
 - Um dado número de instruções
 - Um dado número de ciclos
 - Um dado número de segundos
-
- Tem-se agora um vocabulário que relaciona estas quantidades

- Período de ciclo (segundos por ciclo)
- Frequência de relógio (ciclos por segundo) ● CPI (ciclos por instrução)
- Aplicações com intensas operações de pto flutuante acarretam em um maior CPI
 - MIPS (milhões de instruções por segundo)
- Pode ser grande para programas que usam instruções simples

Desempenho

- Deteminado pelo tempo de execução
- Alguma das variáveis pode ser utilizada para determinar desempenho?
- # de ciclos de execução de um programa?
- # de instruções em um programa?
- # de ciclos por segundo?
- # médio de ciclos por instrução?

- # médio de instruções por segundo?

Desempenho

- Para um programa sendo executado na máquina X,
 $\text{DesempenhoX} = 1 / \text{Tempo_execuçãoX}$
- “X é n vezes mais rápido do que Y” $\text{DesempenhoX} / \text{DesempenhoY} = n$
- Problema:
 - se a máquina A executa um programa em 20 segundos
 - e a máquina B executa o mesmo programa em 25 segundos
 - defina quantas vezes a máquina A é mais rápida que a máquina B.

CPI

- Número de ciclos de clock por instrução
- Específico para cada instrução em um dada arquitetura
- Pode-se contabilizar o número de ciclos médio por instrução como medida de desempenho

- Considerando o número de instruções no cálculo de desempenho:

- $\text{Nro de ciclos de relógio} = \text{nro de instruções de um dado programa} * \text{CPI}$

- Como diferentes instruções executam em um número de ciclos de relógio diferentes, a CPI é dada por uma média ponderada das instruções executadas pelo programa.

Exemplo de CPI

- Suponha que tenhamos duas implementações da mesma arquitetura do conjunto de instruções (ISA)

Para um determinado programa,

A máquina A tem tempo de ciclo de clock de 250 ps e CPI de 2,0

A máquina B tem tempo de ciclo de clock de 500 ps e CPI de 1,2

Que máquina é mais rápida para esse programa e o quanto?

- Lembrando:

Nro de ciclos de clock = nro de instruções * CPI

Tempo de execução = nro de ciclos de clock * período de um clock

Desempenho = 1 / Tempo de execução

Exemplo de CPI

- Suponha que tenhamos duas implementações da mesma arquitetura do conjunto de instruções (ISA)

Para um determinado programa,

A máquina A tem tempo de ciclo de clock de 250 ps e CPI de 2,0

A máquina B tem tempo de ciclo de clock de 500 ps e CPI de 1,2

Que máquina é mais rápida para esse programa e o quanto?

Programa tem $nroInstrucoesX$

$NroCiclosA = nroInstrucoesX * 2; NroCiclosB = nroInstrucoesX * 1,2$

$TExecA = NroCiclosA * PeríodoClockA = nroInstrucoesX * 2 * 250 = 500 * nroInstrucoesX$

$TExecB = NroCiclosB * PeríodoClockB = nroInstrucoesX * 1,2 * 500 = 600 * nroInstrucoesX$

$DesempenhoA / DesempenhoB = TExecB / TExecA = 600 * nroInstrucoesX / 500 * nroInstrucoesX$

Logo a máquina A é 1,2X mais rápida do que a máquina B

Equação de desempenho

- A partir do exemplo anterior, podemos escrever a equação básica de desempenho:
- Tempo de execução = nro de instruções * CPI * período de clock ou
(nro de instruções * CPI) / frequência

Equação de desempenho

- A partir do conhecimento do funcionamento das diferentes instruções que compõem uma dada arquitetura e que é empregado em um dado programa, o número de ciclos de clock pode ser calculado a partir da seguinte equação

$$\text{Ciclos de clock da CPU} = \sum_{i=0}^n (\text{CPI}_i \times C_i)$$

Exemplo de CPI

O projetista de um compilador deseja decidir entre duas possíveis seqüências de código para a resolução de um problema. Dado os tipos de instrução é o número de ciclos por instrução (CPI) de cada tipo defina:

- 1) Qual o código mais rápida?
- 2) Qual a CPI de cada um dos programas?

tipo de instrução	CPI
A	1
B	2
C	3

código	nº de instruções (x N)		
	tipo A	tipo B	tipo C
1	2	1	2
2	4	1	1

Exemplo de CPI

O projetista de um compilador deseja decidir entre duas possíveis seqüências de código para a resolução de um problema. Dado os tipos de instrução é o número de ciclos por instrução de cada tipo, qual seqüência é mais rápida?

tipo de instrução	CPI
A	1
B	2
C	3

código	nº de instruções (x N)		
	tipo A	tipo B	tipo C
1	2	1	2
2	4	1	1

O código 1 executa $2+1+2=5$ Instruções.

O código 2 executa $4+1+1=6$ Instruções.

O número de ciclos de clock para o código 1 $= (2 \times 1) + (1 \times 2) + (2 \times 3) = 10$ ciclos.

O número de ciclos de clock para o código 2 $= (4 \times 1) + (1 \times 2) + (1 \times 3) = 9$ ciclos.

$$\text{CPI} = \frac{\text{ciclos de clock}}{\text{n}^\circ \text{ de instruções}}$$

CPI código 1 $= 10/5 = 2,0$

CPI código 2 $= 9/6 = 1,5$

O código 2 é mais rápido, mesmo que execute uma instrução a mais pois tem CPI mais baixo.

Lei de Amdahl

- *O ganho de desempenho que pode ser obtido melhorando uma determinada parte do sistema é limitado pela fração de tempo que essa parte é utilizada pelo sistema durante a sua operação.*

Tempo de execução após melhoria =

Tempo de execução não afetado +

(Tempo de execução afetado / Quantidade de melhoria)

- Exemplo:

“Suponha que um programa seja executado em 100 segundos em uma máquina, com multiplicação responsável por 80 segundos desse tempo. O quanto precisamos melhorar a velocidade da multiplicação se queremos que o programa seja executado 4 vezes mais rápido?”

Lei de Amdahl

- Exemplo:

“Suponha que um programa seja executado em 100 segundos em uma máquina, com multiplicação responsável por 80 segundos desse tempo. O quanto precisamos melhorar a velocidade da multiplicação se queremos que o programa seja executado 4 vezes mais rápido?”

Tempo de execução após melhoria =

Tempo de execução não afetado +

(Tempo de execução afetado / Quantidade de melhoria)

$$100/4 = (100-80) + (80 / n)$$

$(25 - 20) * n = 80$ $n = 16 \rightarrow$ Quantidade de melhoria a ser aplicada

sobre a parte “melhorável”

Bibliografia

- Hennessy, J. e Patterson, D. “Organização e Projeto de Computadores: A interface Hardware/Software”. 3ª Edição, Capítulo 4.