Performance

Lucas

13 de outubro de 2015

pg 26 O transistor é um interruptor simples de ligar/desligar, controlado por eletricidade. O circuito integrado (CI) combina dezenas de centenas de transistores em um único chip. VLSI (milhares de transistores), chamamos de circuito integrado de larguíssima escala (very large-scale integrated circuit).

pg 28 Tempo de Resposta (Responde Time) também chamado Tempo de Execução. O tempo total requerido para o computador completar uma tarefa qualquer, incluindo acessos ao disco, memória, I/O, overhead do sistema operacional, tempo de execução da CPU, etc.

Vazão (Throughput) também chamado de largura de banda. É outra medida de desempenho, é o número de tarefas completadas por unidade de processamento.

pg 29 Na página 29 temos uma equação simples, comparativa: $Performance_x = \frac{1}{Execution\ Time_x}$ Se o desempenho de x é melhor que o de y então $Performance_x > Performance_y$ ou ainda $\frac{1}{Execution\ Time_x} > \frac{1}{Execution\ Time_y}$ ou também $Execution\ Time_x < Execution\ Time_y$. Isso significa que o tempo de execução em y é mais longo que o tempo de execução em x, bastante simples.

$$\frac{Performance_x}{Performance_y} = \frac{Execution\ Time_y}{Execution\ Time_x} = n$$

pg 30 O conceito de medir desempenho de computador se baseia em: entender qual montante de (trabalho) ele estará processando, em um dado tempo. O processador que realizar o mesmo montante no menor tempo, teoricamente é mais rápido. Tempo de execução do programa é medido em segundos por programa. A definição mais direta é chamada de 'wall clock time', 'response time', ou 'elapsed time'. Esses termos significam o total para uma tarefa ser completada.

pg 31 Por consistência, vamos manter a distinção entre desempenho, baseado em 'elapsed time' e por sua vez baseada em CPU 'execution time'. Clock cycle (Ciclos de Relógio) também chamado de 'clock tick', 'period', 'cycle'. É o tempo para um período de relógio (geralmente em picosegundos, microsegundos, nanosegundos, ou segundos, depende do problema), usualmente o relógio do processador, o qual está rodando à uma certa taxa constante (por exemplo: 4 GHz)

pg 32 Uma fórmula simples que relaciona fatores simples é:

$$CPU\ execution\ time\ for\ a\ program\ = CPU\ clock\ cycles\ for\ a\ program \times Clock\ cycle\ time$$

Um exemplo genérico de melhoramento de performance usando essa fórmula acima é:

$$CPU\ time_A = \frac{CPU\ clock\ cycles_A}{Clock\ rate_A}$$

$$\frac{Seconds}{Program} = \frac{CPU\ clock\ cycles_A}{f \times 10^n\ Hz}$$

$$CPU\ clock\ cycles_A = (t\ [\mathbf{s}]) \times (f \times 10^n\ [\mathbf{Hz}])$$

Onde t é por exemplo: 12 segundos, ou 3 nanosegundos. E $f \times 10^n$ é por exemplo: 9×10^9 hertz, ou 9 Gigahertz. E assim por diante.

$$\frac{Seconds}{Program} = \frac{(CPU\ clock\ cycles_B = CPU\ clock\ cycles_A \times scalar\ cost)}{Clock\ Rate_B}$$

assim, podemos simplificar para:

$$\frac{Seconds}{Program} = \frac{(CPU\ clock\ cycles_A \times scalar\ cost)}{Clock\ Rate_B}$$

Ou seja se eu tenho um computador A que processa um programa qualquer em 13 segundos, sendo que esse tem 3 GHz. Para que eu gere um computador B que processe o mesmo 30% mais rápido que o computador A. Eu precisaria que ele tivesse 4,68 GHz se quisemos que o CPU B processasse a mesma informação em 10 segundos (3 segundos a menos, melhoria de 30%), pagando um custo de 1,4x ciclos a mais, aí vai do seu critério, quanto você quer pagar a mais?

Changes	$\frac{Seconds}{Program} = \frac{(CPU\ clock\ cycles_A \times scalar\ cost)}{Clock\ Rate_B}$
$\uparrow \frac{Seconds}{Program_B}$ (e.g: $3\ s \rightarrow 4\ s$)	Decresce $Clock\ rate_B$
$\uparrow (CPU\ clock\ cycles_A \times improvement)$	Aumenta $Clock\ rate_B$

pg 33 O termo clock cycles per intruction CPI, o qual é a média numérica de ciclos de clock para cada instruções executada, varia bastante de classes de instruções para classes de instruções, exemplos o CPI de instruções do tipo R não vai ser o mesmo CPI para instruções do tipo I, de fato instruções do tipo I contém em seu corpo branch, loads, stores, essas instruções contém veneno para o datapath, da forma que ele leva mais ciclos para processar.

$$CPU\ clock\ cycles = Instructions\ for\ a\ program \times Average\ clock\ cycles\ per\ instruction$$

Suponha você agora que temos um processador X com tempo para cada ciclo de relógio de N pico segundos e em média faz s ciclos por instrução (CPI), e um processador Y com tempo de clock cycle de K pico segundos e em média faz p ciclos por instrução (CPI). Quem é mais rápido?

$CPU\ clock\ cycles_A = (I\ \textbf{instr.} \times s\ \textbf{CPI})\ \textbf{ciclos}$

$CPU\ clock\ cycles_B = (I\ \textbf{instr.} \times p\ \textbf{CPI})\ \textbf{ciclos}$

$CPU\ time_A = CPU\ clock\ cycles_A \times Clock\ cycle\ time$[a]

$CPU\ time_A = I \times s \times N\ \textbf{pico segundos}$

$CPU\ time_B = CPU\ clock\ cycles_B \times Clock\ cycle\ time$

$CPU\ time_B = I \times p \times K\ \textbf{pico segundos}$

Basta comparar ambos os resultados de tempo de CPU e ver qual se desempenhou melhor.

$$\frac{CPU\ time_B}{CPU\ time_A} = \frac{I \times p \times K}{I \times s \times N} = \frac{p\ \textbf{CPI} \times K\ \textbf{pico segundos}}{s\ \textbf{CPI} \times N\ \textbf{pico segundos}}$$

Se por acaso, $p \times K > s \times N$ então CPU_B se desempenha melhor no conjunto de instruções que você passou como parâmetro. Caso contrário CPU_A se desempenha melhor.

$1,7\ \textbf{CPI} \times 190\ \textbf{pico segundos} < 2\ \textbf{CPI} \times 180\ \textbf{pico segundos}$ pois $323\ \textbf{pico segundos} < 360\ \textbf{pico segundos}$ então CPU_A tem 11% menos desempenho que CPU_B.

pg 35 Equação de Desempenho Clássica

$$CPU\ time = Instruction\ count \times CPI \times Clock\ cycle\ time$$

$$CPU\ time = \frac{Instruction\ count\ \times CPI}{Clock\ rate}$$

Essas fórmulas são bastante úteis pois elas separam os três fatores que afetam o desempenho. Nós podemos usar essas fórmulas para comparar duas implementações diferentes ou também para avaliar uma alternativa se nós sabemos os impactos nesses três parâmetros.

$CPU\ clock\ cycles = \sum_{i=1}^{n}(CPI_i \times C_i)$[b]

Por exemplo: $CPU\ clock\ cycles = 2 \times 3 + 1,7 \times 9 + 1 \times 5 = 26$ ciclos de clock. Veja que poderíamos considerar as contagens de cada classe como sendo $\times 10^6$ instruções $CPU\ clock\ cycles = 2 \times 3\ \times 10^6 + 1,7 \times 9\ \times 10^6 + 1 \times 5\ \times 10^6 = 26\ \times 10^6$ ciclos de clock. Não tem problema nenhum. Para calcular o CPI geral, temos a seguinte fórmula $CPI_{total} = \frac{CPU\ clock\ cycles}{Instruction\ count}$. Ora, então temos: Contagem 1 $3 + 9 + 5 = 17$ e aplicando a fórmula temos o seguinte: $CPI_{total} = \frac{26}{17} = 1,5$

$$Time = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Clock\ cycles}{Instruction} \times \frac{Seconds}{Clock\ cycle}$$

Sempre tenha em mente que a medida confiável de medição do desempenho (performance) é Tempo. Por exemplo: mudar o conjunto de instruções para uma contagem menor, pode liderar para tempo do ciclo de relógio maior (ou

[a] O fator tempo de ciclo de relógio, é o tempo de 1 ciclo de relógio. O montante total de ciclos para executar um programa completamente, executar uma tarefa qualquer, vai levar (pensando logicamente) n **cycles** $\times$ *clock cycle time*.

[b] CPI é uma média de ciclos por instruções.

seja, mais lento); ou também pode liderar para um maior CPI (média de ciclos por instruções) que infere mudanças de melhoria na contagem de instrução. Similarmente, por causa que CPI depende do tipo de instrução executada, **o código que executa o menor número de instruções pode não ser o mais rápido**.

1 Questões de performance

1.3.1 Qual processador tem o melhor desempenho expresso em instruções por segundo?

Você pode decidir se quer contar como Milhões de Instruções por Segundo, ou Instruções por Segundo. Basta multiplicar por 10^6 todos os resultados abaixo.

$MIPS_{a.P_1} = \frac{Clock\ Rate}{CPI \times 10^6} = \frac{3\ GHz}{1{,}5 \times 10^6} = 2 \times 10^3$ milhões de instruções por segundo.

$MIPS_{a.P_2} = \frac{Clock\ Rate}{CPI \times 10^6} = \frac{2{,}5\ GHz}{1{,}0 \times 10^6} = 2{,}5 \times 10^3$ milhões de instruções por segundo.

$MIPS_{a.P_3} = \frac{Clock\ Rate}{CPI \times 10^6} = \frac{4\ GHz}{2{,}2 \times 10^6} = 1{,}8 \times 10^3$ milhões de instruções por segundo.

$MIPS_{b.P_1} = \frac{Clock\ Rate}{CPI \times 10^6} = \frac{2\ GHz}{1{,}2 \times 10^6} = 1{,}6\ \times 10^3$ milhões de instruções por segundo.

$MIPS_{b.P_2} = \frac{Clock\ Rate}{CPI \times 10^6} = \frac{3\ GHz}{0{,}8 \times 10^6} = 3{,}75\ \times 10^3$ milhões de instruções por segundo.

$MIPS_{b.P_2} = \frac{Clock\ Rate}{CPI \times 10^6} = \frac{4\ GHz}{2{,}0 \times 10^6} = 2 \times 10^3$ milhões de instruções por segundo.

CPI e Frequência trabalham juntas na vazão de instruções.

1.3.2 Achar o número de ciclos e o número de instruções para 10 segundos de execução de uma tarefa por esses processadores.

$Clock\ Cycle\ Time_{a.P_1} = \frac{1}{Clock\ Rate} = \frac{1}{3\ GHz} = 0{,}33\ nanosegundos$

$Clock\ Cycle\ Time_{a.P_2} = \frac{1}{Clock\ Rate} = \frac{1}{2{,}5\ GHz} = 0{,}4\ nanosegundos$

$Clock\ Cycle\ Time_{a.P_3} = \frac{1}{Clock\ Rate} = \frac{1}{4\ GHz} = 0{,}25\ nanosegundos$

$Clock\ Cycle\ Time_{b.P_1} = \frac{1}{Clock\ Rate} = \frac{1}{2\ GHz} = 0{,}5\ nanosegundos$

$Clock\ Cycle\ Time_{b.P_2} = \frac{1}{Clock\ Rate} = \frac{1}{3\ GHz} = 0{,}33\ nanosegundos$

$Clock\ Cycle\ Time_{b.P_3} = \frac{1}{Clock\ Rate} = \frac{1}{4\ GHz} = 0{,}25\ nanosegundos$

$10\ s = I \times 1{,}5 \times 0{,}33 \times 10^{-9} \rightarrow I = 20$ bilhões de instruções.

$10\ s = I \times 1{,}0 \times 0{,}4 \times 10^{-9} \rightarrow I = 25$ bilhões de instruções.

$10\ s = I \times 2,2 \times 0,25 \times 10^{-9} \rightarrow I = 18$ bilhões de instruções.

$10\ s = I \times 1,2 \times 0,5 \times 10^{-9} \rightarrow I = 16$ bilhões de instruções.

$10\ s = I \times 0,8 \times 0,33 \times 10^{-9} \rightarrow I = 37$ bilhões de instruções.

$10\ s = I \times 2,0 \times 0,25 \times 10^{-9} \rightarrow I = 20$ bilhões de instruções.

1.3.4 O ICP é o inverso do CPI então, lembrar que se CPI é uma média de ciclos por instrução, ICP também é uma média de instruções por ciclo.

$Tempo = 7 = Nro.\ Instr \times CPI \times \frac{1}{Freq.\ Relog.} \rightarrow \frac{7 \times 3 \times 10^9}{20 \times 10^9} = CPI = 1,0 \rightarrow ICP = \frac{1}{CPI} = \frac{1}{1} = 1$

$Tempo = 10 = Nro.\ Instr \times CPI \times \frac{1}{Freq.\ Relog.} \rightarrow \frac{10 \times 2,5 \times 10^9}{30 \times 10^9} = CPI = 0,8 \rightarrow ICP = \frac{1}{CPI} = \frac{1}{0,8} = 1,25$

$Tempo = 9 = Nro.\ Instr \times CPI \times \frac{1}{Freq.\ Relog.} \rightarrow \frac{9 \times 4 \times 10^9}{90 \times 10^9} = CPI = 0,4 \rightarrow ICP = \frac{1}{CPI} = \frac{1}{0,4} = 2,5$

$Tempo = 5 = Nro.\ Instr \times CPI \times \frac{1}{Freq.\ Relog.} \rightarrow \frac{5 \times 2 \times 10^9}{20 \times 10^9} = CPI = 0,5 \rightarrow ICP = \frac{1}{CPI} = \frac{1}{0,5} = 2$

$Tempo = 8 = Nro.\ Instr \times CPI \times \frac{1}{Freq.\ Relog.} \rightarrow \frac{8 \times 3 \times 10^9}{30 \times 10^9} = CPI = 0,8 \rightarrow ICP = \frac{1}{CPI} = \frac{1}{0,8} = 1,25$

$Tempo = 7 = Nro.\ Instr \times CPI \times \frac{1}{Freq.\ Relog.} \rightarrow \frac{7 \times 4 \times 10^9}{25 \times 10^9} = CPI = 1,1 \rightarrow ICP = \frac{1}{CPI} = \frac{1}{1,1} = 0,9$

1.4.1a Dado um programa de 1 milhão de instruções divididas em classes de 10% classe A, 20% classe B, 50% classe C, e 20% classe D, qual implementação é mais rápida?[c]

```
        Clock    CPI Class
        Rate     A  B   C  D
a.
   P1 | 2,5 GHz  1  2   3  3
   ---+--------------------
   P2 | 3,0 GHz  2  2   2  2
b.
   P1 | 2,5 GHz  2  1,5 2  1
   ---+--------------------
   P2 | 3,0 GHz  1  2,0 1  1
```

$$Tempo = No.\ Instr. \times CPI \times \frac{1}{Clock\ Rate}$$

Processador 1.

Classe A	$10 \times 10^4 = 1 \times 10^5$	$Tempo = 1 \times 10^5 \times 1 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 0,4 \times 10^{-4}$ segundos
Classe B	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 2 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 1,6 \times 10^{-4}$ segundos
Classe C	$50 \times 10^4 = 5 \times 10^5$	$Tempo = 5 \times 10^5 \times 3 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 6,0 \times 10^{-4}$ segundos
Classe D	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 3 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 2,4 \times 10^{-4}$ segundos
		Tempo Total $10,4 \times 10^{-4}$ segundos

Processador 2.

Classe A	$10 \times 10^4 = 1 \times 10^5$	$Tempo = 1 \times 10^5 \times 2 \times \frac{1}{3\ \mathbf{GHz}} = 0,6 \times 10^{-4}$ segundos
Classe B	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 2 \times \frac{1}{3\ \mathbf{GHz}} = 1,3 \times 10^{-4}$ segundos
Classe C	$50 \times 10^4 = 5 \times 10^5$	$Tempo = 5 \times 10^5 \times 2 \times \frac{1}{3\ \mathbf{GHz}} = 3,3 \times 10^{-4}$ segundos
Classe D	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 2 \times \frac{1}{3\ \mathbf{GHz}} = 1,3 \times 10^{-4}$ segundos
		Tempo Total $6,5 \times 10^{-4}$ segundos

Nós constatamos que o Processador 1 leva mais tempo para completar a execução das instruções (separadas em classes).

[c] Tempo é a medida mais honesta que você pode ter, em performance.

1.4.1b Dado um programa de 1 milhão de instruções divididas em classes de 10% classe A, 20% classe B, 50% classe C, e 20% classe D, qual implementação é mais rápida?[d]

$$Tempo = No.\ Instr. \times CPI \times \frac{1}{Clock\ Rate}$$

Processador 1.

Classe A	$10 \times 10^4 = 1 \times 10^5$	$Tempo = 1 \times 10^5 \times 2 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 0,8 \times 10^{-4}$ segundos
Classe B	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 1,5 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 1,2 \times 10^{-4}$ segundos
Classe C	$50 \times 10^4 = 5 \times 10^5$	$Tempo = 5 \times 10^5 \times 2 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 4,0 \times 10^{-4}$ segundos
Classe D	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 1 \times \frac{1}{2{,}5\ \mathbf{GHz}} = 0,8 \times 10^{-4}$ segundos
		Tempo Total $6,8 \times 10^{-4}$ segundos

Processador 2.

Classe A	$10 \times 10^4 = 1 \times 10^5$	$Tempo = 1 \times 10^5 \times 1 \times \frac{1}{3\ \mathbf{GHz}} = 0,3 \times 10^{-4}$ segundos
Classe B	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 2 \times \frac{1}{3\ \mathbf{GHz}} = 1,3 \times 10^{-4}$ segundos
Classe C	$50 \times 10^4 = 5 \times 10^5$	$Tempo = 5 \times 10^5 \times 1 \times \frac{1}{3\ \mathbf{GHz}} = 1,6 \times 10^{-4}$ segundos
Classe D	$20 \times 10^4 = 2 \times 10^5$	$Tempo = 2 \times 10^5 \times 1 \times \frac{1}{3\ \mathbf{GHz}} = 0,6 \times 10^{-4}$ segundos
		Tempo Total $3,8 \times 10^{-4}$ segundos

1.4.2a What is the global CPI for each implementation?

$CPI_{a.P_1} = \frac{\sum_{i=0}^{3}(CPI_i \times Contagem\ Instr_i)}{Total\ Instr} = \frac{(1\times10^4\times10+2\times10^4\times20+3\times10^4\times50+3\times10^4\times20)}{10^6} = \frac{260\times10^4}{10^6} = 2,6$ ciclos por instrução (em média e considerando separação por classes).

$CPI_{a.P_2} = \frac{\sum_{i=0}^{3}(CPI_i \times Contagem\ Instr_i)}{Total\ Instr} = \frac{(2\times10^4\times10+2\times10^4\times20+2\times10^4\times50+2\times10^4\times20)}{10^6} = \frac{200\times10^4}{10^6} = 2$ ciclos por instrução (em média e considerando separação por classes).

O $CPI_{a.P_1} > CPI_{a.P_2}$ possivelmente terá melhor desempenho, porém. Essa análise confirma para essas configurações de conjuntos de instruções separados por classe. Outros casos podem alterar o resultado. Infelizmente $a.P_1$ obteve 30% melhor desempenho, para essa configuração da tabela, do que $a.P_2$, o que é uma pena, pois eu estava confiante que $a.P_2$ iria vencer.

[d] Tempo é a medida mais honesta que você pode ter, em performance.

1.4.2b What is the global CPI for each implementation?

$CPI_{b.P_1} = \frac{\sum_{i=0}^{3}(CPI_i \times Contagem\ Instr_i)}{Total\ Instr} = \frac{(2\times10^4\times10+1{,}5\times10^4\times20+2\times10^4\times50+1\times10^4\times20)}{10^6} = \frac{170\times10^4}{10^6} = 1,7$ ciclos por instrução (em média e considerando separação por classes).

$CPI_{b.P_2} = \frac{\sum_{i=0}^{3}(CPI_i \times Contagem\ Instr_i)}{Total\ Instr} = \frac{(1\times10^4\times10+2\times10^4\times20+1\times10^4\times50+1\times10^4\times20)}{10^6} = \frac{120\times10^4}{10^6} = 1,2$ ciclos por instrução (em média e considerando separação por classes).

A primeira implementação obteve melhor resultado, interessante.

1.5.1a Considere duas implementações diferentes, P1 e P2, do mesmo conjunto de instruções (mesma arquitetura). Existem 5 classes de instruções (A, B, C, D, e E) nesse conjunto de instruções. A frequência de relógio e o CPI de cada classe é dado abaixo:

		Clock Rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D	CPI Class E
a.	P1	2,0 GHz	1	2	3	4	3
	P2	4,0 GHz	2	2	2	4	4

Assuma que o desempenho **pico** é definido como a taxa mais rápida que o computador pode executar qualquer sequência de instruções (rajada). Quais são os picos de desempenhos para P1 e P2 individualmente medido em instruções por segundos?

$MIPS_{P_1} = \frac{Clock\ Rate}{CPI\times10^6} = \frac{2\ \text{GHz}}{1\times10^6} = 2000$ milhões de instruções por segundo.

$MIPS_{P_2} = \frac{Clock\ Rate}{CPI\times10^6} = \frac{2\ \text{GHz}}{2\times10^6} = 1000$ milhões de instruções por segundo.

Nessa questão precisamos notar que, o menos CPI é o maior IPC. Se você tem um CPI igual à 4, terá um IPC = 1/4 que é menor que se você tivesse um CPI igual à 2 e então um IPC = 1/2. Ou seja, $1/4 < 1/2$. E IPC é uma média de instruções por ciclo. Note que na fórmula do MIPS você está trabalhando com IPC.

$$MIPS = Clock\ Rate \times \frac{1}{CPI\downarrow} \times \frac{1}{10^6}$$

1.6.1a Compiladores diferentes tem impactos em desempenho nos processadores. Para diferentes compiladores em um mesmo processador, ache o CPI se o clock cycle time é de 1 nanosegundo.

	Compiler A		Compiler B	
a.	No. Instr.	Exec. Time	No. Instr.	Exec. Time
	1,00E+09	1,8 s	1,20E+09	1,8s

$Tempo_{copmilador_A} = 1,8s = \frac{CPI \times No.\ Instr.}{\frac{1}{1\times10^{-9}\,\mathbf{GHz}}} \rightarrow \frac{1,8\times10^9}{1,0\times10^9} = 1,8$ ciclos por instrução.

$Tempo_{copmilador_B} = 1,8s = \frac{CPI \times No.\ Instr.}{\frac{1}{1\times10^{-9}\,\mathbf{GHz}}} \rightarrow \frac{1,8\times10^9}{1,2\times10^9} = 1,5$ ciclos por instrução.

Para um mesmo processador, mesma frequência de relógio, temos que um compilador gerou mais instruções que o outro. Esse que gerou mais instruções obteve menor CPI, e portanto melhor desempenho. Tempo vezes frequência de relógio, dá em ciclos do relógio naquele comprimento de tempo, isso dividido pelo número de instruções dá o CPI ciclos totais naquela largura de tempo dividido pelo total de instruções geradas pelo compilador. Quanto maior o número de instruções e mantendo fixo $tempo \times freq$ menor irá ser o CPI médio. O que é bom, termos poucos ciclos médios por instrução, isso aumenta o IPC que é o inverso do CPI, aumentando IPC você tem mais instruções dentro de um ciclo de relógio. Portanto o compilador que gerar mais instruções nesse ensaio, fará com que o processador tenha mais desempenho. Podemos pensar que o compilador que gerou menos instruções, foi mais eficiente. O compilador B obteve 16,6% mais eficiência do que o compilador A. É mito dizer que o compilador que gerar menos instruções será melhor sempre, e hoje em dia com multinucleado multicores e pipeline você pode ter um compilador gerando mais instruções que outro porém ele sabe muito bem que as instruções estão organizadas para atingir bom desempenho utilizando os vários cores disponíveis, ou thread, etc.

~~**1.7.2**~~, ~~**1.7.3**~~, ~~**1.7.4**~~, ~~**1.8.1**~~, ~~**1.8.2**~~ cancelei essas questões, na quarta edição revisada, elas me parecem não ter relevância.

1.9.3a Determine a razão da potência-estática e potência-dinâmica para cada tecnologia.

Frequência de troca é uma função da frequência de relógio. A carga capacitiva por transistor é uma função de ambos o número de transistores conectados a uma saída (chamada de fanout) e a tecnologia. As quais ambas determinam a capacitância de ambos wires e transistores.

	Technology	Dynamic Power (W)	Static Power (W)	Voltage (V)
a.	180 nm	50	10	1.2
b.	70 nm	90	60	0.9

Dissipação estática de potência[e] $\int_0^1 V_{DD} I_{leak} dt$ Você pode minimizar I_{leak} fazendo diminuição da tensão, ou também colocando

[e] A potência consumida pelo dispositivo. potência estática vai ser consumida sempre, e tem contribuição de vazamento nos transistores.

menos transistores I_{leak}.

Dissipação dinâmica de potência[f] $\int_0^1 VC_{DD}^2 f_c dt$, podemos diminuir I_{switch} fazendo diminuição da tensão, menos trocas capacitivas, diminuição da frequência de trocas.

Ambos somados dá na potência total $E = \int_0^t (V_{DD} I_{leak} + VC_{DD}^2 f_c) dt$

$Power = Capacitive\ load \times Voltage^2 \times Frequency_{switched}$

$Raz = \frac{10}{50} = \frac{1}{5}$

	Processadores		No. Instruções por Processador			CPI	
		Aritméticas	LOAD/STORE	Branch	Aritméticas	LW/SW	Branch
a.	1	2560	1280	256	1	4	2
	2	1280	640	128	1	5	2
	4	640	320	64	1	7	2
	8	320	160	32	1	12	2

1.10.2a Dado os valores de CPI, ache o tempo total de execução para esses programas em números de processadores: 1,2,4, e 8. Assumir que cada processador tem frequência de relógio 2 GHz.

$Tempo = CPI \times Instr. \times Clock\ Rate$

$CPI_{p_1} = \sum_1^3 (CPI_i \times Instr_i) = 1 \times 2560 + 4 \times 1280 + 2 \times 256 = 8192$

$Tempo_{p_1} = 8192 \times \frac{1}{2 \times 10^9} = \frac{8192}{2 \times 10^9} = 4096\ nanosegundos$

$CPI_{p_2} = \sum_1^3 (CPI_i \times Instr_i) = 1 \times 1280 + 5 \times 640 + 2 \times 128 = 4736$

$\frac{CPI}{Nro\ Processadores} = \frac{4736}{2} = 2368$

$Tempo_{p_2} = 2368 \times \frac{1}{2 \times 10^9} = \frac{2368}{2 \times 10^9} = 1184\ nanosegundos$, mas com uma vazão duas vezes maior que o primeiro ensaio.

$CPI_{p_3} = \sum_1^3 (CPI_i \times Instr_i) = 1 \times 640 + 7 \times 320 + 2 \times 64 = 3008$

$\frac{CPI}{Nro\ Processadores} = \frac{3008}{4} = 752$

$Tempo_{p_3} = 752 \times \frac{1}{2 \times 10^9} = \frac{752}{2 \times 10^9} = 376\ nanosegundos$, mas com uma vazão duas vezes maior que o segundo ensaio.

[f] Potência consumida quando o dispositivo está em operação. Esta associado com a frequência de troca dos transistores.

$CPI_{p_4} = \sum_1^3 (CPI_i \times Instr_i) = 1 \times 320 + 12 \times 320 + 2 \times 32 = 2304$

$\frac{CPI}{Nro\ Processadores} = \frac{2304}{8} = 288$

$Tempo_{p_4} = 288 \times \frac{1}{2 \times 10^9} = \frac{288}{2 \times 10^9} = 36$ *nanosegundos*, mas com uma vazão duas vezes maior que o terceiro ensaio e quatro vezes maior que o segundo ensaio.

Na melhor das hipóteses os tempos serão esses, mas nunca é garantido que os processadores conseguirão dividir o CPI médio. Existe uma dificuldade em relacionar instruções e sincronia de threads.

1.10.4a Assuma frequência de relógio 3 GHz. Qual são os tempos de execuções?

	Cores per Processor	Instructions per Core	Average CPI
a.	1 cores	1.00E+10	1.2
	2 cores	5.00E+09	1.4
	4 cores	2.50E+09	1.8
	8 cores	1.25E+09	2.6

$CPI = 1.2 \times \frac{\mathbf{1.00E+10}}{1\ core} \times \frac{1}{3 \times 10^9} = 4,0$

$CPI = 1.4 \times \frac{\mathbf{5.00E+09}}{2\ core} \times \frac{1}{3 \times 10^9} = 1,16$

$CPI = 1.8 \times \frac{\mathbf{2.50E+09}}{4\ core} \times \frac{1}{3 \times 10^9} = 0,375$

$CPI = 2.6 \times \frac{\mathbf{1.25E+09}}{8\ core} \times \frac{1}{3 \times 10^9} = 0,13$

~~**1.10.5a**~~ cancelei essa questão, na quarta edição revisada. essa não me parece ter relevância.

1.10.6a Usando um core único, ache o CPI requerido nesse core para obter um tempo de execução igual aos tempos obtidos na questão 1.10.4. Note que o número de instruções deve ser o número de instruções agregado ao tempo de execução através dos cores.