



ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Desempenho computacional: Medições.

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Latência x Vazão

- Tempo de resposta (latência)
 - Tempo entre o início e o fim de uma tarefa
 - Diminuir o tempo de resposta quase sempre melhora a vazão

- Vazão (throughput)
 - Quantas tarefas simultâneas um sistema pode executar?
 - Qual é a taxa média de execução de tarefas?
 - Quanto trabalho está sendo feito?

- Latência vs vazão
 - O que melhora se?
 - Se atualizarmos um computador com um novo processador?
 - Se um novo computador for acrescentado a um laboratório?

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Comparativo

- Dizer que um Computador A é n vezes mais rápido que B significa que:

$$n = \frac{\textit{tempo de execução}B}{\textit{tempo de execução}A}$$

Métrica

- Desempenho: definido como o inverso do tempo de execução

$$desempenho_A = \frac{1}{tempo\ de\ execução_A}$$

- Dizer que um Computador A é n vezes mais rápido que B também significa que (desempenho relativo):

$$n = \frac{tempo\ de\ execução_B}{tempo\ de\ execução_A} = \frac{\frac{1}{desempenho_B}}{\frac{1}{desempenho_A}} = \frac{desempenho_A}{desempenho_B} = 1 + \frac{n}{100}$$

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Aperfeiçoamento (improvement)

- Melhora do desempenho significa incremento:
→ Mais é melhor
- Tempo de execução (ou tempo de resposta) significa decremento:
→ Menos é melhor

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Exemplo 1

➤ Se um computador A executa o programa X em 10 segundos e um computador B executa o mesmo programa em 15 segundos:

→ A é 50% mais rápido que B ou A é 33% mais rápido que B?

Solução exemplo 1

➤ A afirmação que diz que A é $n\%$ mais rápida que B pode ser expressa como:

$$\Rightarrow \frac{\text{tempo de execução } B}{\text{tempo de execução } A} = 1 + \frac{n}{100}$$

$$n = \frac{(\text{tempo de execução } B - \text{tempo de execução } A) * 100}{\text{tempo de execução } A} = \frac{(15 - 10) * 100}{10} \\ = 50$$

→ Computador A é 50% mais rápido que B

Tempo de execução

- Tempo decorrido (real)
 - Contagem total do tempo (acesso a disco e memória, E/S e etc.)
 - Normalmente não é ideal para fins de comparação
 - Interrupções do processo pelo escalonador e tempos imprevisíveis de acesso a E/S
- Tempo de CPU (user + system)
 - Não computa E/S e tempo de execução de outros programas (tempo do sistema + usuário)
 - Foco deste curso: tempo consumido em linhas de código do “nosso programa”
- Exemplo: comando time do Linux

```
real    0m25.042s
user    0m1.196s
sys     0m7.972s
```


Ciclos de clock

- Ao invés de relatar o tempo de execução em segundos, geralmente é utilizado ciclos:

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

"ticks" de clock indicam quando uma atividade inicia



Ciclo de clock: tempo entre ticks = segundos por ciclo

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

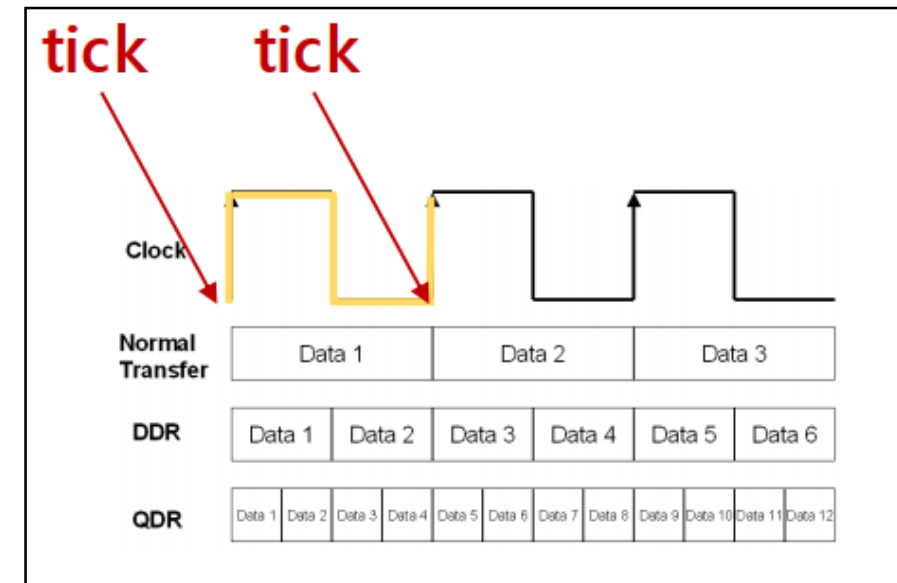
07h40-09h20

Sala: H204

AULA 18

Ciclos de clock

- **TCLK** = Período ou ciclo de clock
 - Tempo entre dois pulsos de clock
 - Segundos por ciclo
- **fCLK** = Frequência de clock
- Ciclos de clock por segundo = $1/T$
- $1 \text{ Hz} = 1 / \text{segundo}$

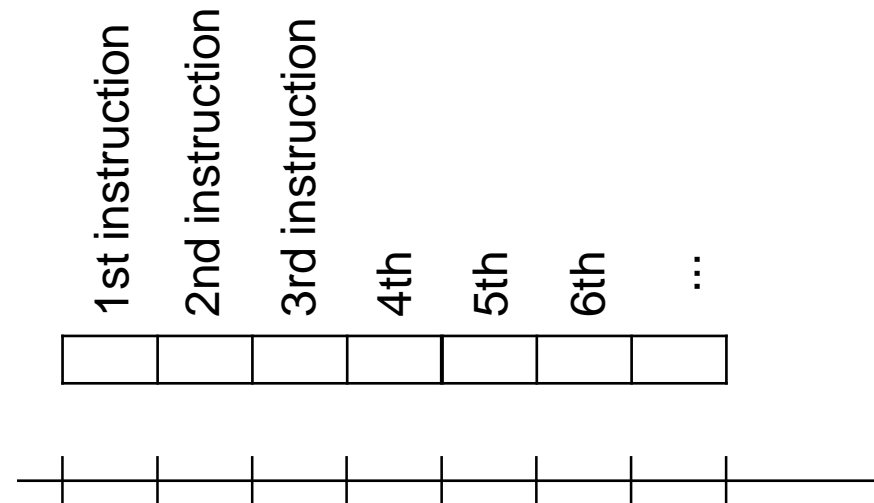


Exemplo 2

- A frequência de clock de 500 MHz corresponde ao tempo de ciclo de clock: $1 / (500 * 10^6) = 2 * 10^{-9}$
 $= 2 \text{ nseg}$
- A frequência de clock de 1 GHz corresponde ao tempo de ciclo de clock: $1 / (10^9) = 1 * 10^{-9} = 1 \text{ nseg}$

Quantos ciclos de clock são necessários para executar um programa?

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



Essa afirmação é **incorreta**: Diferentes instruções levam diferentes quantidades de tempo em diferentes arquiteturas

ECM 245

Arquitetura e
Organização de
Computadores

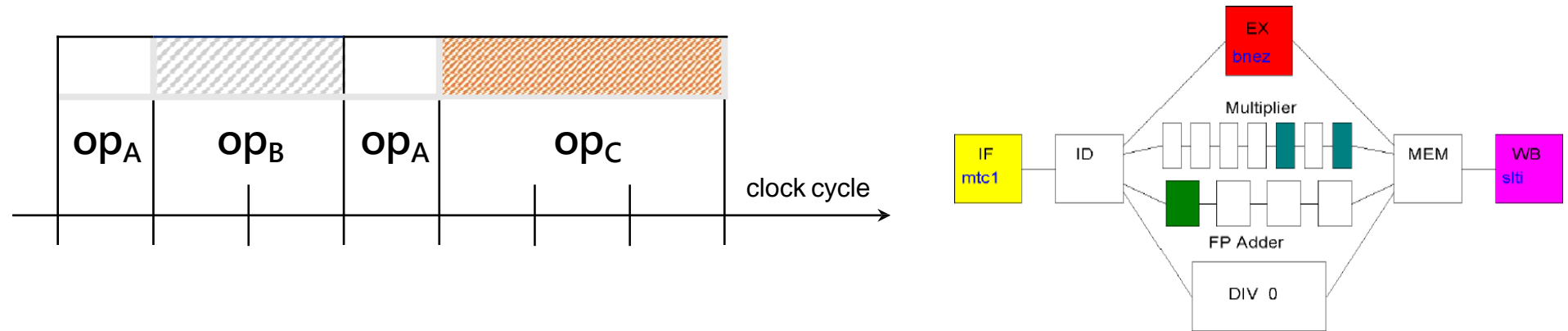
5ª-feira

07h40-09h20

Sala: H204

AULA 18

Números de ciclos variam para diferentes instruções



- Multiplicação leva mais tempo que adição;
- Operações de ponto flutuante levam mais tempo que as de inteiro;
- Acessar a memória leva mais tempo que acessar registradores.

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Perspectiva do programa

- Sua execução envolve:
 - Um determinado número de instruções;
 - Um determinado número de ciclos;
 - Um determinado número de segundos.

- Vocabulário relacionado:
 - CPI (Ciclos por Instrução)
 - Específico para cada instrução de uma dada arquitetura;
 - Pode ser utilizada como medida de desempenho;
 - Aplicações com elevada computação de ponto flutuante geralmente tem maior CPI.

- MIPS (Milhões de instruções por segundo)
 - Pode ser utilizada como medida para vazão de dados.

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Tempo de execução ou Tempo de CPU

$$CPU_{time} = Clock\ Cycles \times T_{CLK} = \frac{Clock\ Cycles}{f_{CLK}}$$

- Otimizar o desempenho significa reduzir o tempo de execução (ou CPUtime):
 - Reduzir o número de ciclos de clock por programa;
 - Reduzir o período de clock;
 - Aumentar a frequência de clock. Factível hoje?

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

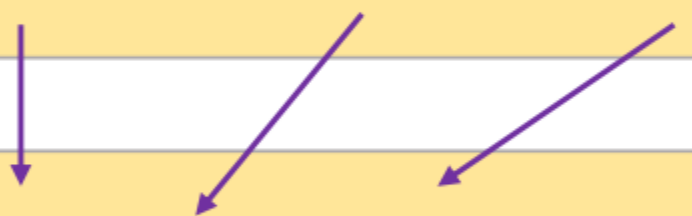
07h40-09h20

Sala: H204

AULA 18

Tempo de execução

$$CPU_{time} = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$


$$CPU_{time} = \frac{Seconds}{Program} = IC \times CPI \times T_{CLK} = \frac{IC \times CPI}{f_{CLK}}$$

➤ Ciclos por Instrução

➤ CPI = Clock Cycles / Instruction Count

➤ Instruções por Ciclos

➤ IPC = 1 / CPI

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Tempo de execução

$$CPU_{time} = Clock\ Cycles \times T_{CLK} = \frac{Clock\ Cycles}{f_{CLK}}$$

Onde: $Clock\ cycles = \sum_{i=1}^n (CPI_i \times I_i)$

$$\Rightarrow CPU\ time = \sum_{i=1}^n (CPI_i \times I_i) \times TC_{LK}$$

$$CPI = \sum_{i=1}^n (CPI_i \times F_i) \quad \text{Onde } F_i = \frac{I_i}{IC}, \text{ "frequência da instrução"}$$

$$\Rightarrow CPU\ time = IC \times CPI \times T_{CLK} = IC \times \sum (CPI_i \times F_i) \times TC_{LK}$$

Exemplo 3

Calcule os CPI e o tempo da CPU para executar um programa composto por 100 instruções nas proporções descritas pela tabela. A CPU tem clock de 500 MHz.

	Frequência	Ciclos de clock
ALU	43%	1
LOAD	21 %	4
STORE	12 %	4
BRANCH	12 %	2
JUMP	12%	2

$$\text{CPU time} = \text{IC} \times \text{CPI} \times T_{\text{CLK}}$$

$$\text{CPI} = \sum_{i=1}^n (\text{CPI}_i \times F_i)$$



ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Solução exemplo 3

$$\text{➤ CPI} = 0.43 * 1 + 0.21 * 4 + 0.12 * 4 + 0.12 * 2 + 0.12 * 2 = 2,23$$

$$\text{➤ CPU time} = IC * CPI + T_{\text{clock}} = 100 * 2,23 * 2 \text{ ns} = 446 \text{ ns}$$

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

MIPS – Millions of instructions per second

➤ Milhões de instruções por segundo

$$MIPS = \frac{Instruction\ Count}{Execution\ time \times 10^6}$$

➤ Onde:

$$Execution\ time = \frac{IC \times CPI}{f_{CLK}}$$

$$MIPS = \frac{f_{CLK}}{CPI \times 10^6}$$

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

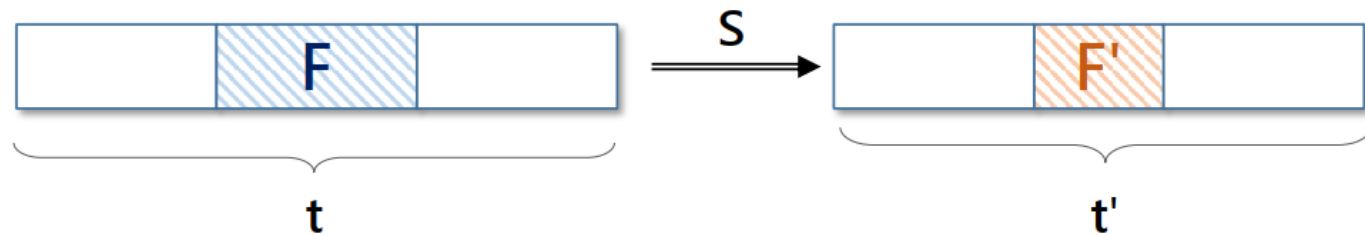
Sala: H204

AULA 18

Lei de Amdahl

- Como calcular o aumento de desempenho ou $Speedup(E)$?

$$Speedup(E) = \frac{Exec_{time\ without\ E}}{Exec_{time\ with\ E}} = \frac{Performance_{time\ with\ E}}{Performance_{time\ without\ E}}$$



$$t' = 1 - F + F' \Rightarrow t' = 1 - F + F/S$$

Lei de Amdahl

- Suponha que E acelera uma fração F da tarefa por um fator S e o restante da tarefa não é afetado. Assim:

$$Exec_{time}(E) = \left((1 - F) + \frac{F}{S} \right) \times Exec_{time\ without\ E}$$

↓ $Exec_{time}(E) / Exec_{time}(wo\ E) = 1 / Speedup(E)$

$$Speedup(with\ E) = \frac{1}{(1 - F) + \frac{F}{S}}$$

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Lei de Amdahl

- Ideia básica: torne o caso comum mais rápido
- 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 durante a sua operação.
 - Fração E = a fração do tempo de computação na máquina original que pode ser convertida para tirar vantagem de alguma melhoria
 - Speedup E = a melhoria obtida pela execução "acelerada"
- O speedup total é dado por:

$$Speedup_{total} = \frac{Exec_{time}(wo\ E)}{Exec_{time}(E)} = \frac{1}{(1 - FractionE) + \frac{FractionE}{Speedup_E}}$$

ECM 245

Arquitetura e
Organização de
Computadores

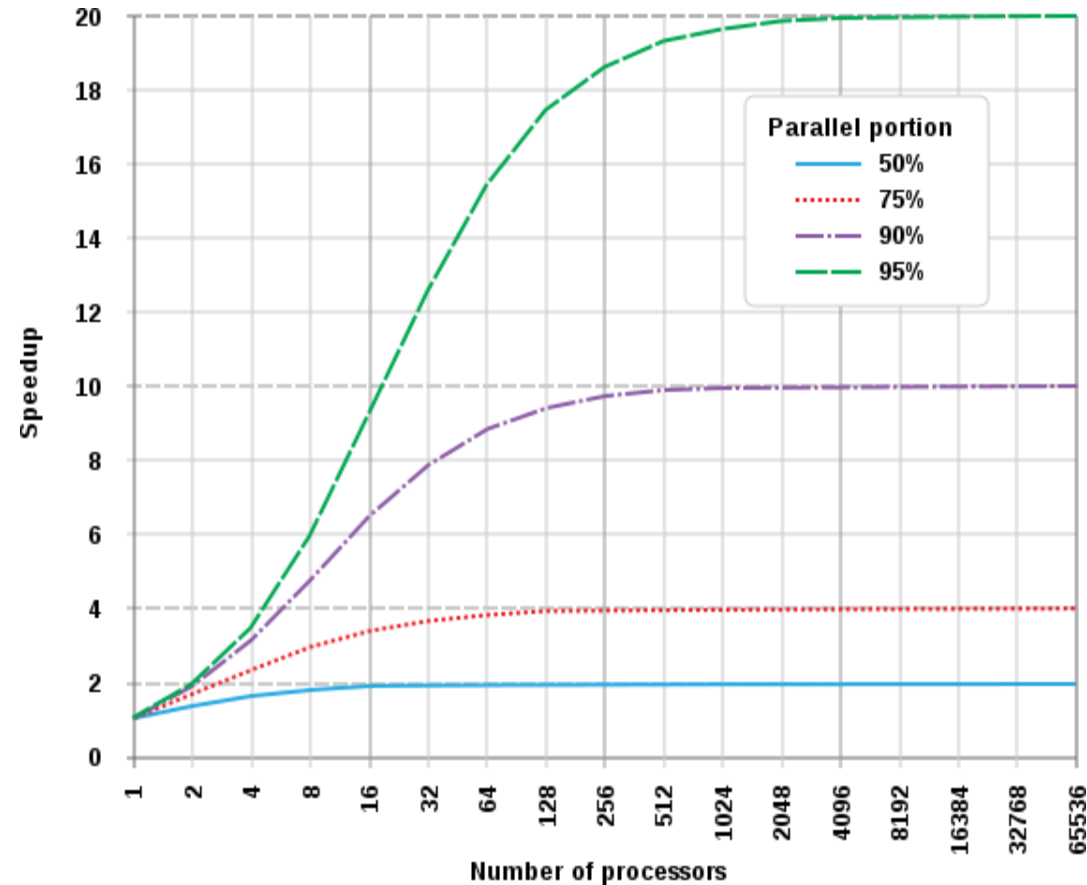
5ª-feira

07h40-09h20

Sala: H204

AULA 18

Lei de Amdahl



- Speedup limitado pela parte serial do programa. Se 95% for paralelizável, teoricamente o máximo de speedup possível é de 20x.

Exemplo 4

➤ Considere o aperfeiçoamento de uma CPU resultando em computação dez vezes mais rápida em relação a CPU original. Esta CPU é ocupada com computação apenas 40% de todo tempo. Qual é o speedup total obtido pela introdução da melhoria?

➤ Solução:

- Caso comum: 40% da computação
- Fração $E = 0.4$
- Speedup $E = 10$

$$Speedup_{total} = \frac{1}{(1 - FractionE) + \frac{FractionE}{Speedup_E}} = \frac{1}{(1 - 0.4) + \frac{0.4}{10}} = \mathbf{1.56}$$

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Base para avaliação

PRÓS		CONTRAS
representativos	Workload da aplicação alvo	Muito específico Não portátil Difícil de executar ou medir
Portáteis Amplamente usados Maior variação de workloads	Conjuntos completos de benchmarks	Menos representativos
Fácil de ser executado. Até mesmo no ciclo de desenvolvimento	Benchmarks com kernels pequenos	Fácil de "enganar"
Identifica máximo desempenho (pico) e possíveis gargalos	Microbenchmarks	"Pico" pode estar muito longe do desempenho da aplicação

ECM 245

Arquitetura e
Organização de
Computadores

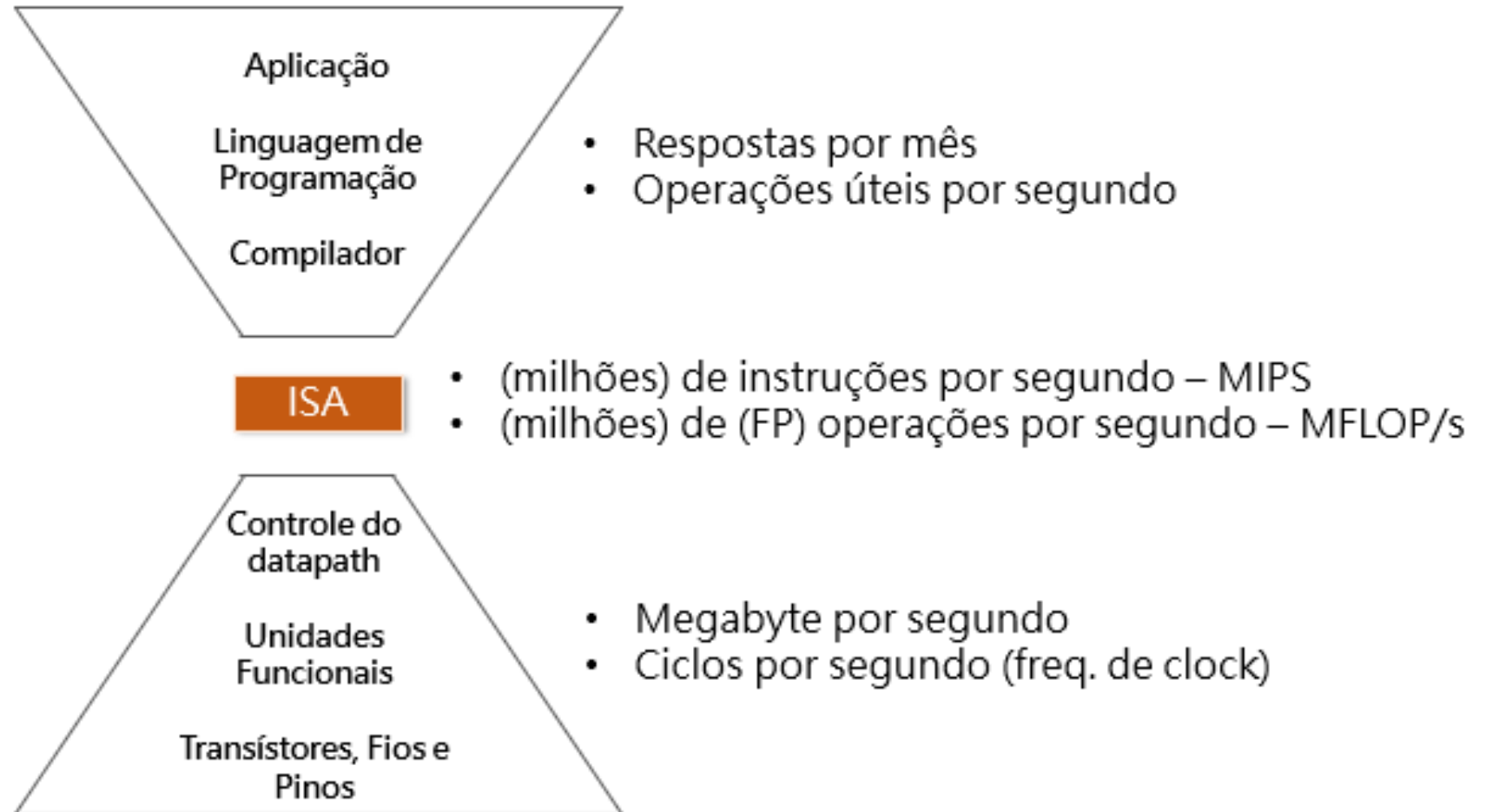
5ª-feira

07h40-09h20

Sala: H204

AULA 18

Métricas de desempenho



Cada métrica tem um lugar e um propósito, e cada uma pode ser mal utilizada.

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

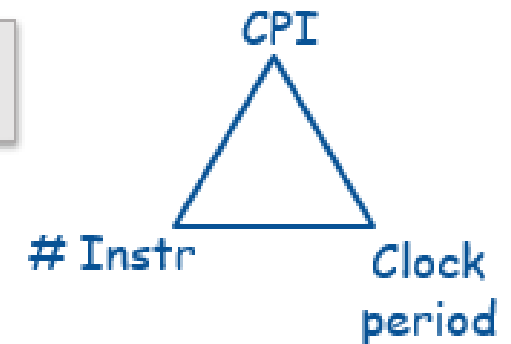
Sala: H204

AULA 18

Aspectos do desempenho da CPU

$$CPU_{time} = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

$$CPU_{time} = IC \times CPI \times T_{CLK}$$



	IC (# Instruções)	CPI	Clock Period
Programa	X		
Compilador	X	(X)	
Conj. de Instruções	X	X	X
Organização		X	X
Tecnologia			X

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Questões de desempenho com pipelining

- O pipelining aumenta a vazão de instruções (número de instruções concluídas por unidade de tempo), mas não reduz o tempo de execução (latência) de uma única instrução.

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Questões de desempenho com pipelining

- Pipelining geralmente aumenta ligeiramente a latência de cada instrução devido ao desequilíbrio entre os estágios do pipeline e overhead no seu controle.
 - O desequilíbrio entre os estágios de pipeline reduz o desempenho, pois a frequência de clock não pode ser maior do que o tempo necessário para o estágio mais lento.
 - O overhead do pipeline surge pelo atraso no uso dos registradores do pipeline e do clock skew (atraso no sinal do clock nos latches). Assim: $\text{ciclo de clock} > \text{atraso dos registradores} + \text{clock skew}$

ECM 245

Arquitetura e
Organização de
Computadores

5ª-feira

07h40-09h20

Sala: H204

AULA 18

Métricas com pipeline

- IC = Instruction Count
- $\# \text{ Clock Cycles} = \text{IC} + \# \text{ Stall Cycles} + L_L$
- $L_L = \text{Latência dos Latches de Pipeline} = n_s - 1$
- $n_s = \text{Número de estágios do pipeline}$
- $\text{CPI} = \text{Clock por Instrução} = \# \text{ Clock Cycles} / \text{IC}$
- $\text{CPI} = (\text{IC} + \# \text{ Stall Cycles} + L_L) / \text{IC}$
- $\text{MIPS} = \text{fclock} / (\text{CPI} * 10^6)$

Exemplo 5

- IC = Instruction Count = 5
- # Clock Cycles = IC + # Stall Cycles + L_L
- # Clock Cycles = 5 + 3 + 4 = 12
- CPI = (IC + #Stall Cycles + L_L) / IC = 12 / 5 = 2.5
- MIPS = $f_{\text{clock}} / (\text{CPI} * 10^6) = 500 \text{ Mhz} / 2.4 * 10^6 = 208.3$

