

Avaliação do Desempenho 1: Métricas

Sistemas de Computação

Licenciatura em Ciências da Computação

João Luís Sobral, 2023

Desempenho do [C]PU

- Para **prever o desempenho** (T_{EXEC} – tempo de execução) de um dado programa num determinado processador é necessário um **modelo** que relacione o desempenho com as características do sistema de computação (*hw+sw*)

Um programa numa máquina
executa num determinado número
médio de ciclos de relógio (#CC):
clock cycles

A duração de cada ciclo de relógio
(período do relógio)
do CPU é constante:
 $T_{cc} = 1 / \text{frequência do relógio}$

$$T_{EXEC} = \# \text{ clock cycles} * T_{cc}$$

ou

$$T_{EXEC} = \# \text{ clock cycles} / f$$

Desempenho do CPU

- De que depende o número médio de ciclos necessários para executar um programa?

A execução de um programa consiste na execução de um determinado número de instruções:
#I

Através de medições sucessivas é possível estimar o número médio de ciclos necessário para executar uma instrução:
CPI

$$\# \text{ clock cycles} = \#I * CPI$$

$$T_{EXEC} = \# \text{ clock cycles} * T_{cc} = \#I * CPI * T_{cc} = \#I * CPI / f$$

Desempenho do CPU

- Um programador quer escolher entre dois segmentos de código diferentes para um mesmo algoritmo. Qual o mais rápido?

Tipo de Instrução	CPI
A	1
B	2
C	3

Código	Número de Instruções		
	A	B	C
1	2000	1000	100
2	100	1000	1000

$$T_{EXEC1} = (1 * 2000 + 2 * 1000 + 3 * 100) / f = 4300 / f$$

$$T_{EXEC2} = (1 * 100 + 2 * 1000 + 3 * 1000) / f = 5100 / f$$

Desempenho do CPU

- Calcule o tempo de execução do programa abaixo numa máquina com um relógio de 2 GHz e CPI (médio) de 1.5

```
    movl 10, %ecx  
    movl 0, %eax  
ciclo:  
    addl %ecx, %eax  
    decl %ecx  
    jnz ciclo
```

#l = 32

NOTA: O número de instruções a considerar é o número de instruções **executadas**.

$$T_{\text{exec}} = 32 * 1.5 / 2\text{E}9 = 24\text{E-}9 \text{ s} = 24 \text{ ns}$$

Desempenho do CPU

- **Como determinar o CPI?**

- Uma aproximação grosseira será dizer que uma máquina apresenta um determinado CPI, independentemente do tipo de instruções.
 - Se tivermos a possibilidade de medir #CC e #I então $CPI = \#CC / \#I$
- Diferentes tipos de instruções exibem valores de CPI diferentes:
 - Divisões exigem mais ciclos do que adições ou multiplicações
 - Acessos à memória exigem mais ciclos do que acessos a registos
 - Operações em vírgula flutuante podem exigir mais ciclos do que operações com inteiros
- O CPI pode ser determinado para cada classe de instruções, sendo o CPI médio (ou CPI_{global}) calculado multiplicando a frequência de cada instrução pelo seu CPI:

$$CPI_{global} = \sum_{i=1}^n CPI_i * F_i \quad \text{onde } F_i = \frac{\#I_i}{\#I}$$

- Exemplo:

Operação	Freq (Fi)	Ciclos por Instr. (CPIi)	CPIi x Fi	(% tempo)
ALU	50%	1	0,5	(33%)
Load	20%	2	0,4	(27%)
Store	10%	2	0,2	(13%)
Branch	20%	2	0,4	(27%)

Combinção típica (mix)

$\Sigma =$

1,5

CPI_{global} para este exemplo

Atenção:

- Desempenho de pico usa o menor CPI

Relação entre as métricas

Exemplo 1 : Aumentar a frequência do relógio (diminuir Tcc) implica frequentemente um aumento do CPI!

Explicação: Entre outros factores, deve-se considerar o tempo de acesso à memória (Tmem). Se Tcc diminui e Tmem se mantiver, então serão necessários mais ciclos para aceder à memória.

$$f_1 = 1GHz$$

$$T_{cc1} = 1ns$$

$$T_{mem} = 40ns$$

$$Ciclos_{mem1} = 40$$

$$f_2 = 2GHz$$

$$T_{cc2} = 0.5ns$$

$$T_{mem} = 40ns$$

$$Ciclos_{mem2} = 80$$

Conclusão: Apesar de Tcc diminuir para metade, Texec não diminui para metade, pois o número de ciclos de acesso à memória aumenta.

Relação entre as métricas

Exemplo 2 : Diminuir o número de instruções (#I) recorrendo a instruções mais complexas resulta num aumento do CPI!

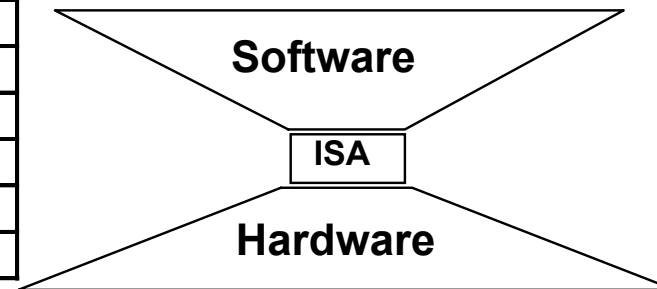
Explicação: As instruções mais complexas realizam o trabalho de várias instruções simples, mas podem necessitar de mais ciclos para o completar, resultando num aumento do CPI. Este é um dos argumentos dos defensores de arquitecturas RISC.

Conclusão: O número de instruções diminui, mas o ganho em tempo de execução não diminui na mesma proporção, devido ao aumento do CPI.

Relação entre as métricas

	Tem impacto em		
	#I	CPI	Tcc
Algoritmo	S	(S)	
Linguagem	S	(S)	
Compilador	S	(S)	
Conj. Instruções (ISA)	S	S	
Organização		S	S
Tecnologia			S

$$\text{Texe} = \#I \times \text{CPI} \times T_{cc}$$



#I – depende do algoritmo, da linguagem de programação, do compilador e da arquitectura (ISA)

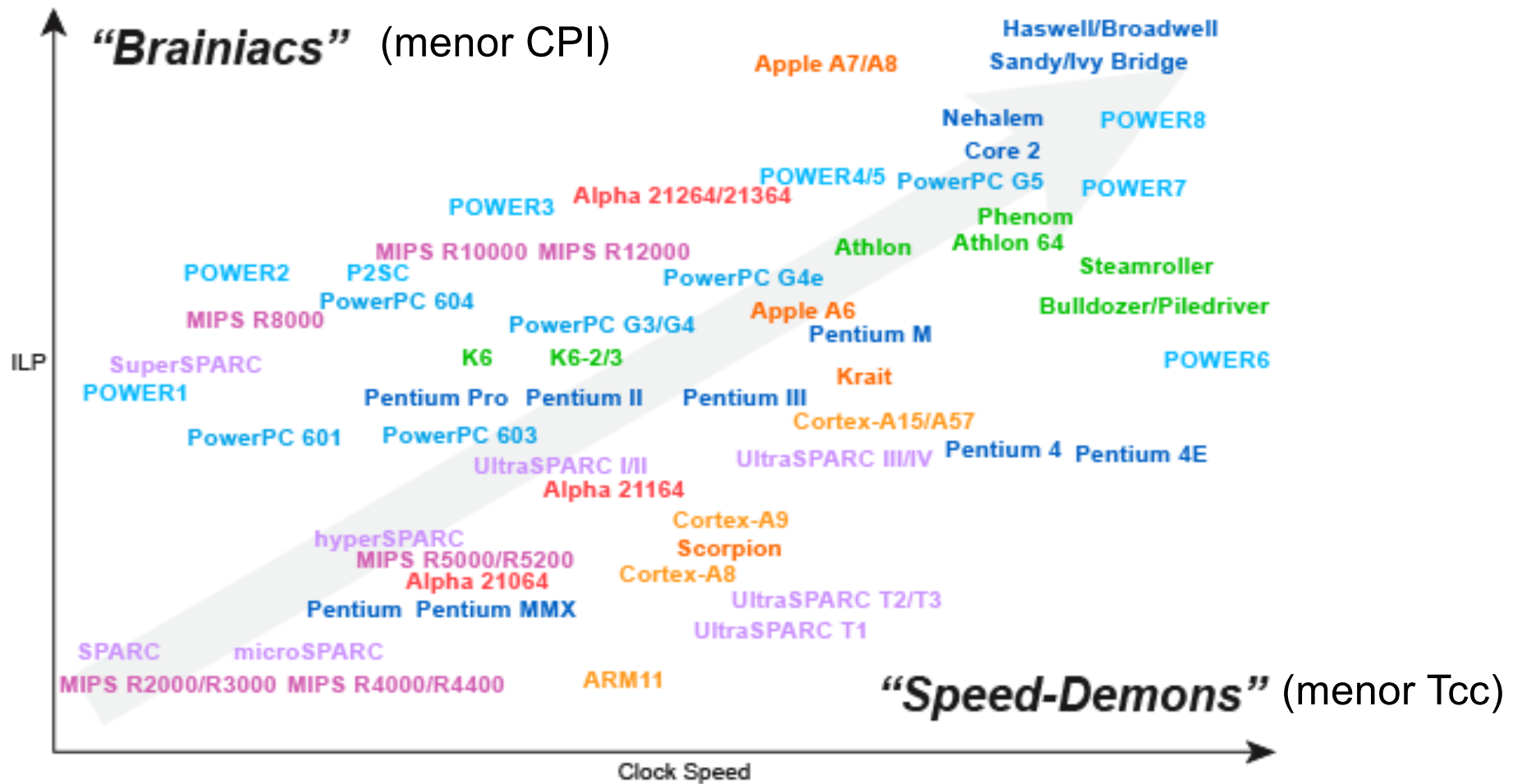
CPI – depende da arquitectura (ISA), da mistura de instruções efectivamente utilizadas, da organização do processador e da organização dos restantes componentes do sistema (ex., memória)

f – depende da organização do processador e da tecnologia utilizada

“A única métrica completa e fiável para avaliar o desempenho de um computador é o tempo de execução”

As métricas CPI, f e #I não podem ser avaliadas isoladamente, devendo ser sempre consideradas em conjunto, pois dependem umas das outras.

Relação entre as métricas



O diagrama ilustra a evolução da arquitetura de computadores ao longo do tempo, representada por uma barra horizontal dividida em segmentos coloridos. Acima da barra, setas apontam para os paradigmas dominantes em cada fase:

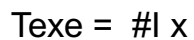
- Microprogramação** (preto)
- Pipelining** (preto)
- Superescalaridade** (preto)
- Caches** (preto)
- Timeshared** (preto)
- Memória Virtual** (cinza)
- RISC** (laranja claro)
- VLIW (EPIC)** (laranja claro)
- CC-UMA** (laranja claro)
- CC-NUMA** (laranja claro)
- Not-CC-NUMA** (laranja claro)
- Passagem de Mensagem** (laranja claro)
- GRID/Internet** (laranja escuro)
- SIMD** (laranja escuro)

Abaixo da barra, setas apontam para as fases de evolução:

- Evolução** (seta azul apontando para o segmento preto)
- ILP** (seta azul apontando para o segmento preto)
- Multi-threading** (seta verde apontando para o segmento de RISC a CC-NUMA)
- Processos comunicantes** (seta vermelha apontando para o segmento de Passagem de Mensagem a GRID/Internet)
- Revolução** (seta vermelha apontando para o segmento de SIMD)



- Processador: Intel 8048; clock: 1.79 MHz. (4 bits)
- Memória **RAM**: 256 bytes; 64 bytes no processador
- Memória de vídeo(**VRAM**): 256 bytes
- Gráficos: Intel 8244 custom Audio/Video IC; 16 cores
- Resolution: 280x192
- Game Media: 2K programmable game card



$1/\#\text{cores} \times$

$1/\text{SIMD} \times$

CPI x

Tcc

(CPI = IPC)

$(T_{cc} = 1/f)$

Peak FLOPs =

Number of cores ×

FLOPS per instruction ×

Instruction per cycle ×

Cycles per second

(Task-Level Parallelism)

(SIMD and FMA)

(Instruction Level Parallelism)

(Frequency)

Relação entre as métricas

CPU	80286	80386	80486	Pentium	Pentium Pro	Pentium 4	Core i7
Ano	1982	1985	1989	1993	1997	2001	2010
Tipo	16 bits	32 bits	pipelined 5-stage Cache L1	Super- escalar 2-way	OOO 3-way	OOO super- pipelined Cache L2	OOO 4-way Cache L3
Freq.	12,5 MHz	16 MHz	25 MHz	66 MHz	200 MHz	1,5 GHz	3,3 GHz
Core/ chip	1	1	1	1	1	1	4
MIPS	2	6	25	132	600	4500	50000

Latência 6
(#CC)

5

5

5

10

22

14