- $CPU_{time} = IC \times CPI \times$ Clock cycle time
  ↳ CCT
- $CPI = \sum_{i=1}^{n} \frac{IC_i}{IC} \times CPI_i$ unde IC_i - no. of instructions in class i
  CPI_i - CPI of class i.
- Performance of $X = \frac{1}{\text{Execution time } X}$
- Computer A is n times faster than computer B =>

=> => $n = \frac{\text{Performance A}}{\text{Performance B}} = \frac{\text{Execution time B}}{\text{Execution time A}}$

Example: Pp. că avem 2 implementări ale aceleași arhitecturi: Computer A: CCT = 250 ps și CPI = 2,0 } aceleași program (UC)
Computer B: CCT = 500 ps și CPI = 1,2 }

Care comp. e mai rapid și cu cât

$CPU_{time\,A} = IC \times 2,0 \times 250 = IC \times 500$ ps
$CPU_{time\,B} = IC \times 1,2 \times 500 = IC \times 600$ ps } =>

=> Comp. A is $\frac{IC \times 600}{IC \times 500} = 1,2$ times faster than comp. B.

CURS 7 9.04.2019

2.1 Performance Equation

CPU_{time} = Instruction Count x Clock cycles per instruction * Clock cycle time
= $IC \times CPI \times$ Clock cycle time

Example: 2 segmente de cod pt. care:

	CPI for each instruction class		
	A	B	C
CPI	1	2	3

Code Sequence	Instruction Counts for each instruction class		
	A	B	C
1	2	1	2
2	4	1	1

$$CPU_{time_1} = IC_1 \times CPI_1 \times \text{Clock cycle time} = \underbrace{5}_{IC_1} \times \underbrace{\frac{2\times1+1\times2+2\times3}{5}}_{CPI_1} \times CCT$$

$$= 10 \text{ clock cycles} \times CCT$$

$$CPU_{time_2} = IC_2 \times CPI_2 \times CCT = \underbrace{6}_{IC_2} \times \frac{4\times1+1\times2+1\times3}{6} \times CCT =$$

$$= 9 \text{ clock cycles} \times CCT$$

← compilatorul mai bun

$$\frac{\text{Performance}_2}{\text{Performance}_1} = \frac{CPU_{time_1}}{CPU_{time_2}} = \frac{10}{9} \cong 1,1$$

Example: Fie un compilator vechi (CV) și unul nou (CN)

CV: $CPU_{time_{CV}} = IC_{old} \times CPI_{old} \times CCT$

CN: $CPU_{time_{CN}} = IC_{new} \times CPI_{new} \times CCT$

! CCT nu se schimbă pt. că se rulează pe același PC

$IC_{new} = 0,6 \times IC_{old}$

$CPI_{new} = 1,1 \times CPI_{old}$

$CPU_{time_{CV}} = 15$ sec

Deci $CPU_{time_{CN}} = 0,6 \times IC_{old} \times 1,1 \times CPI_{old} \times CCT = CPU_{time_{CV}} \times 0,6 \times 1,1$

$= 15 \times 0,6 \times 1,1 = 9,9$ sec

Example:
- Frequency of FP (floating point operations) op = 25%
- Average CPI of FP op = 4,0
- Avg. CPI of (#) other op = 1,33
- Frequency of FPSQR = 2%
- CPI of FPSQR = 20.

Two options: – to decrease CPI of FPSQR to 2

– to decrease the avg. CPI of all FP op. to 2.5

Which option is the best? (CPU_{time_1} vs. CPU_{time_2} (→) (→) CPI°_1 vs. CPI_2)

75%

$CPI_{original} = 0,25 \times 4,0 + 0,75 \times 1,33 \simeq 2$ clock cycles (c.c)

$CPI_1 = CPI_{original} - 2\% \times (20-2) = 2 - 0,02 \times 18 \simeq 1,64$ c.c.

$CPI_2 = 0,25 \times 2,5 + 0,75 \times 1,33 \simeq 1,625$ c.c.

=> Second option is better.

⊙ Example:

CPU_A

```
CMP  r1, r2
BEQ  F00
 .
 .
 .
F00: ADD  r6, r5, r6
 .
 .
 .
```

CPU_B

```
BEQ  r1, r2, F00
 .
 .
 .
F00 .
```

$CPU_{time\,A} = CCT_A \times IC_A \times CPI^{\circ}_A$

$CPI_A = 20\% \times 2 + 80\% \times 1 \simeq 1,2$ c.c } =>

2 c.c. – pt. branchuri condiționate

$=> CPU_{time\,A} = \underbrace{1,2}_{\simeq 1} \times IC_A \times CCT_A$

$CPU_{time\,B} = \underbrace{0,8 \times IC_A}_{IC_B} \times CPI_B \times \underbrace{1,25}_{CCT_B} \times CCT_A$

↳ 80% pt. că au dispărut cele 20% de la instr. de comparare

$CPI^{\circ}_B = 0,25 \times 2 + 0,75 \times 1 = 1,25$ c.c.

$CPU_{time\,B} = IC_A \times 1,25 \times CCT_A$

=> Nu se merită optimizarea B.

Example: (A)

CPI = 2 (ADD r_1, r_2, # mem. loc) (memory location) → (B)

LDR r_1, # mem. loc

CPI = 1 ADD r_2, r_2, r_1

STR r_2, # mem. addr_2

$$\text{CPU time}_A = IC_A \times CPI_A \times CCT_A = IC_A \times \underline{1{,}57} \times CCT_A$$

↳ neoptimizat

$$CPI_A = 0{,}43 \times 1 + 0{,}57 \times 2 = 1{,}57 \text{ c.c.}$$

↳ 43% durează un clock ↳ 57% durează 2 clocki

instrucțiunile care nu se mai execută

$$\text{CPU time}_B = \underbrace{IC_A (1 - 0{,}43 \times 0{,}25)}_{IC_B} \times CPI_B \times CCT_A$$

↳ care nu se schimbă

instrucțiuni ce rămân să se execute pt. 1 c.c — loadurile

$$CPI_B = \frac{(0{,}75 \times 0{,}43) \times 1 + (0{,}25 \times 0{,}43) \times 2 + (0{,}21 - 0{,}25 \times 0{,}43) \times 2}{1 - 0{,}43 \times 0{,}5} +$$

store-urile, branchuri

$$+ \frac{(0{,}12 \times 2) + (0{,}24 \times 3)}{1 - 0{,}43 \times 0{,}5} \simeq 1{,}7$$

$1 - 0{,}43 \times 0{,}5$ → nr. nou de instrucțiuni

Deci $\text{CPU time}_B = IC_A \times \underline{1{,}7} \times CCT_A$

2.2 Real-world benchmarks

MOR DE PLICTISEALĂ KK

KK, nu bun facultaa!!!

buCKK KK KK

KK

KK

PAPA plec !! :)

A. Synthetic benchmarks
B. Toy programs
C. Kernels
D. Real programs mix

SPEC (Standard Performance Evaluation Corporation)
— Benchmarks suites

$$SPEC_A = \sqrt[n]{\prod_{i=1}^{n} SPEC_{A_i}}$$ — SPEC native

unde $SPEC_{A_i}$ = execution time on machine A of program i from the SPEC Benchmark Suite

$$\frac{SPEC_{reference}}{SPEC_A} = \frac{\sqrt[n]{\prod_{i=1}^{n} SPEC_{reference_i}}}{\sqrt[n]{\prod_{i=1}^{n} SPEC_{A_i}}} = \sqrt[n]{\prod_{i=1}^{n} \frac{SPEC_{reference_i}}{SPEC_{A_i}}}$$

$$\text{SPEC relativ} = \frac{\text{Performance A}}{\text{Performance}_{reference}} = \text{SPEC Ratio}_A$$

$$\frac{\text{Performance A}}{\text{Performance B}} = \frac{\text{SPEC Ratio A}}{\text{SPEC Ratio B}} = \frac{\sqrt[n]{\prod_{i=1}^{n} SPEC_{B_i}}}{\sqrt[n]{\prod_{i=1}^{n} SPEC_{A_i}}}$$