

2.2. Benchmark Suites

- real programs ✕ synthetic benchmarks
- ✕ toy programs
- ✕ kernels
- geometric mean
- reference machine

$$\text{std dev} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\text{sample}_i - \text{mean})^2}$$

SPEC,

$$\text{Geom. mean} = \exp\left(\frac{1}{N} \times \sum_{i=1}^N \ln \text{Sample}_i\right)$$

$$\text{geom std dev} = \exp\left(\sqrt{\frac{1}{N} \sum_{i=1}^N (\ln \text{Sample}_i - \ln \text{gmean})^2}\right)$$

2.3. Principles of Computer System Design

① Use parallelism when possible

- CLA

- Pipelining

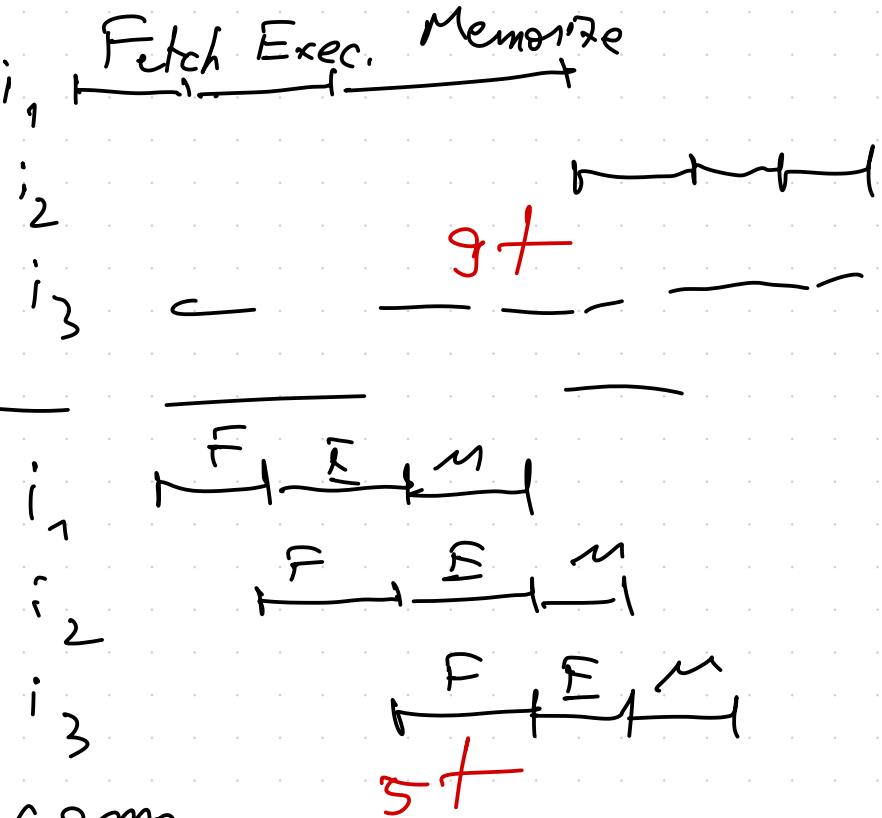
Instruction

Level

Parallelism

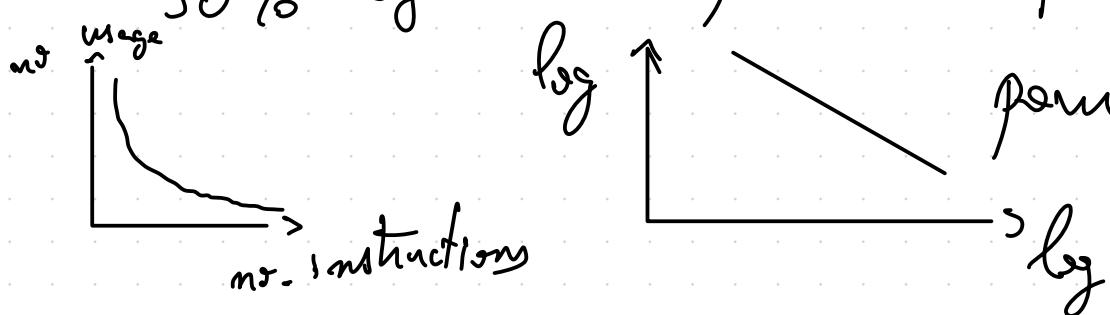
ILP

- multicore comp.

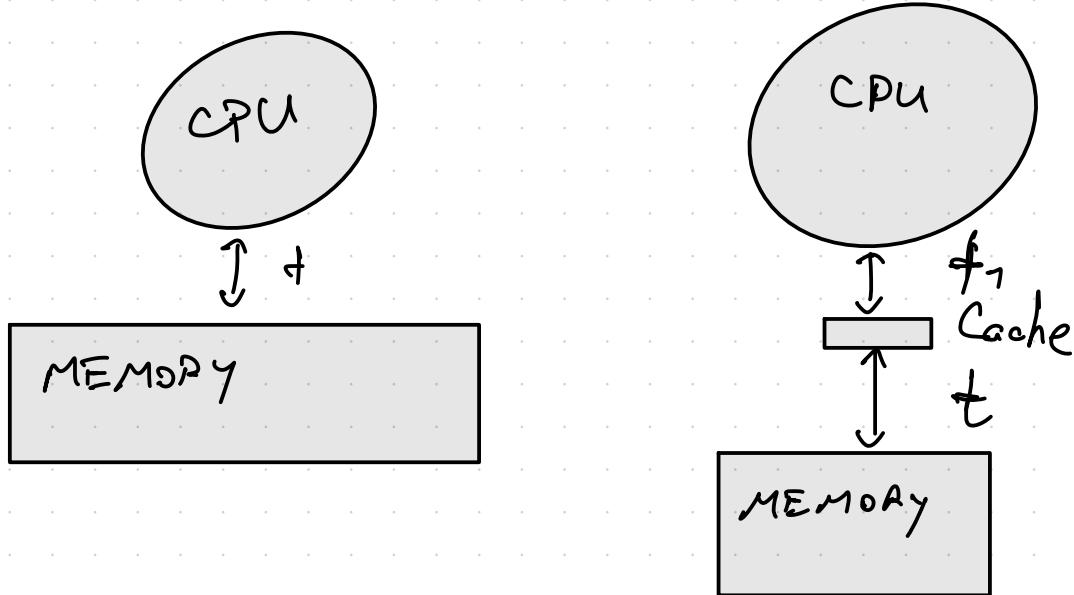


② Locality principle

90% of the time, the comp. runs 10% of the code



Cache Memory \rightarrow pt. code mai utilize operatice

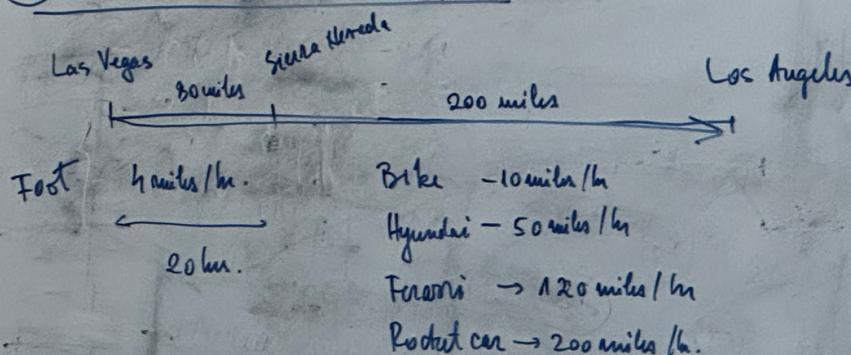


$$\begin{array}{ll} \text{Cache Hit} & H = t_1 \\ \text{Cache Miss} & M = t + t_1 \end{array}$$

③ Make the common case faster

2.3 Principles of computer system design

③ Make the common case faster



Second section vehicle	Time in second section, hr.	Speedup second section	Total Time, hr.	Overall speedup
Foot	(50)	1	70	1
Bike	20	2.5	40	1.75
Hyundai	4	12.5	24	2.92
Ferrari	1.7	29.4	21.7	3.225
Rocket car	1	50	21	3.333

$$\frac{50}{20} = \frac{3}{1.6}$$

$$\frac{50}{4} = \frac{25}{2} = 12.5$$

$$\text{Overall speedup} = \frac{\text{Time without optimization}}{\text{Time with optimization where possible}} =$$

Time without optimization

$$\frac{\text{Time without optimization}}{\text{Time without optimization} (1 - \text{Fraction optimized}) + \frac{\text{Time without optimization}}{\text{Speedup optimized}} \cdot \text{Fraction optimized}}$$

Andahl's Law

$$(1 - \text{Fraction optimized}) + \frac{\text{Fraction optimized}}{\text{Speedup optimized}}$$

$$\lim_{\text{Speedup optimized} \rightarrow \infty} \text{Time Overall speedup} = \frac{1}{1 - \text{Fraction optimized}}$$

2.4. Computer Performance Equation

$$\text{CPU}_{\text{time}} = \frac{\text{Total no. of clock cycles} \times \text{clk cycle time}}{\text{clk frequency}}$$

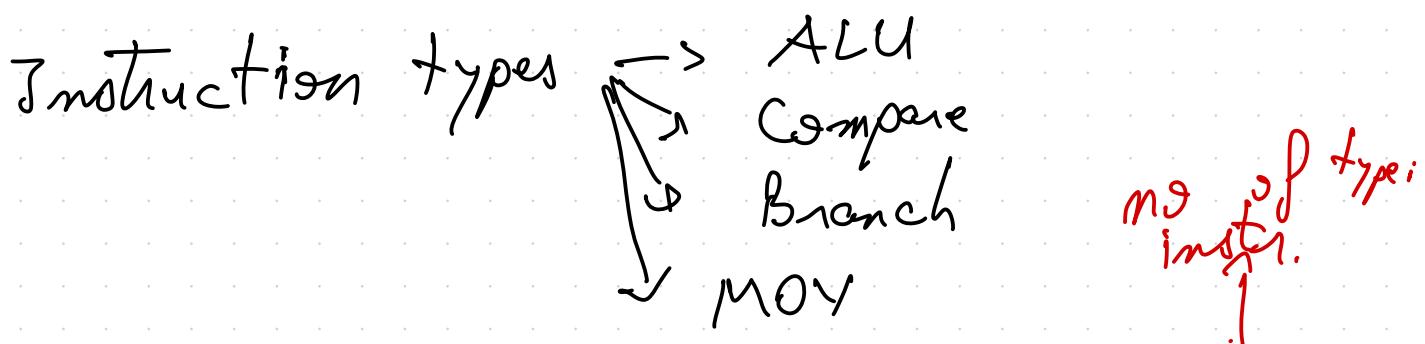
$$\text{clk frequency} = \frac{1}{\text{time}}$$

$$\text{total number of clk cycles} =$$

$$\rightarrow \underbrace{\text{Instruction Count}}_{\text{IC}} \times \underbrace{\text{clk cycles per Inst}}_{\text{CPI}}$$

$$\text{CPU}_{\text{time}} = \text{IC} \times \text{CPI} \times \text{clk cycle time}$$

$$\frac{\text{Instruction}}{\text{Program}} \times \frac{\text{Clk cycles}}{\text{Inst.}} \times \frac{\text{Time}}{\text{Clk cycle}} = \frac{\text{Time}}{\text{Program}}$$



$$IC = IC_1 + IC_2 + IC_3 + \dots + IC_m = \sum_{i=1}^m IC_i$$

$$CPI = \frac{IC_1 \times CPI_1 + IC_2 \times CPI_2 + \dots}{IC}$$

$$= \sum_{i=1}^m \frac{IC_i}{IC} \times CPI_i$$

fraction of type i instructions

$$CPU_{time} = IC \times CPI \times \text{Clk cycle time}$$

RISC \leftrightarrow CISC

Example 1

FLOPS FP instr. freq = 25%

CPI_{FP} = 4.0 c.c.

CPI_{other} = 1.33 c.c.

Freq. of FP SQRT = 2%

CPI_{FP SQRT} = 20 c.c.

(A) Decrease CPI_{FP SQRT} = 2

(B) Decrease CPI_{FP} = 2.5

CPU_{time} = IC₀ × CPI₀ × clk cycle time_{original}

$$CPI_0 = 0.25 \times 4 + 0.75 \times 1.33 = 2 \text{ c.c.}$$

CPU_{time A} = IC₀ × CPI_A × clk cycle time₀

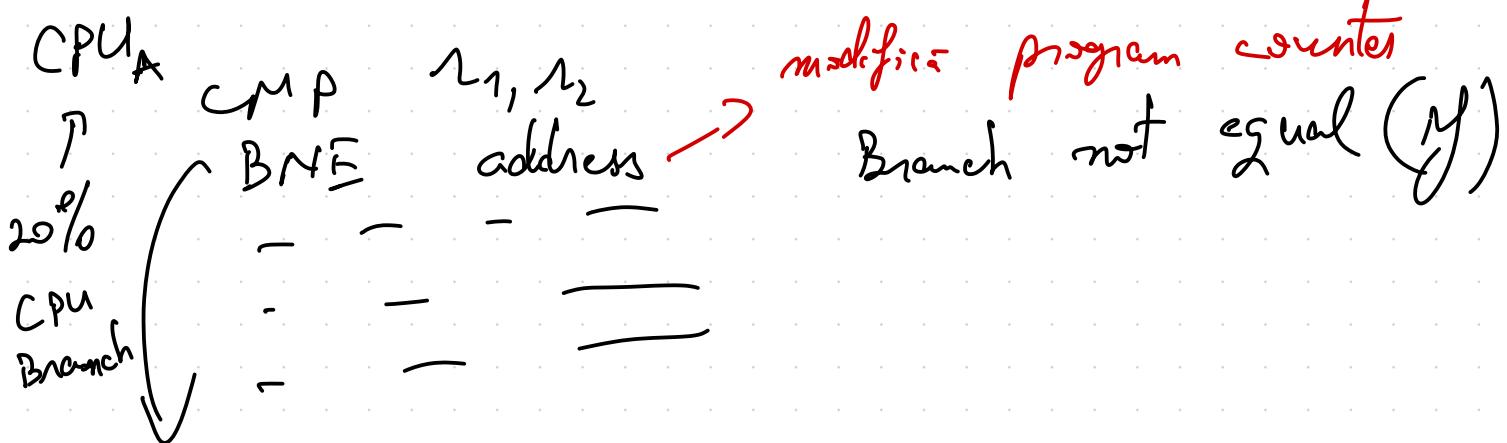
$$CPI_A = 2 - (20 - 2) + 0.02 = \underline{\underline{1.64}} \text{ c.c}$$

$$CPI_B = 0.25 \times 2.5 + 0.75 \times 1.33 = \underline{\underline{1.625}} \text{ c.c.}$$

$$2 - (4 - 2.5) \times 0.25$$

Example 2

Conditional branch



addn. - - - - -

CPU_B **BNE** $r_1, r_2, \text{addr.}$

$$CPI_B = 2C_C$$

$$CPI_{\text{other}} = 1C_C$$

$$\text{clk cycle time } B = 1.25 \times \text{clk cycle time } A$$

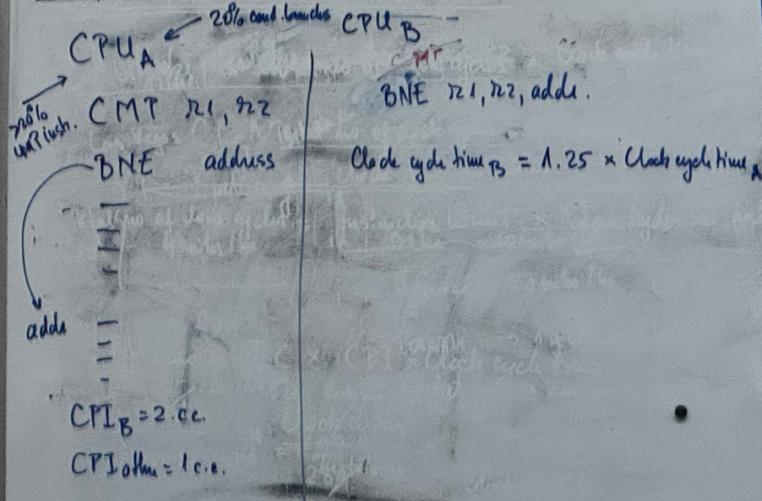
$$\text{CPU}_{\text{time } A} = IC_A \times CPI_A \times CCT_A$$

$$CPI_A = 0.2 \times 2 + 0.8 \times 1 \quad (\text{branch}) \quad (no \text{ branch}) = 1.2$$

$$\text{CPU}_{\text{time } B} = IC_B \times CPI_B \times CCT_B$$

$$IC_B = 0.8 \times IC_A$$

Example 2 Conditional branches



$$\text{CPU}_{\text{time}}_A = ICA \times CPI_A \times CCT_A = ICA \times 1.2 \times CCT_A \quad \checkmark$$

$$CPI_A = 0.2 \times 2 + 0.8 \times 1 = 1.2$$

$$\text{CPU}_{\text{time}}_B = ICB \times CPI_B \times CCT_B$$

$$IC_B = 0.8 \times ICA$$

20 B

20 C

20 B

60 D

60 E

$20 \dots 80$

$2 \dots 100$

$\alpha = 25\%$

$20 \dots 80$

$2 \dots 100$

$\alpha = \frac{20\%}{8} = \frac{10}{4} = 25\%$

$$CPI_B = 0.25 \times 2 + 0.75 \times 1 = 1.25$$

$$\text{CPU}_{\text{time}}_B = ICB \times \underbrace{0.8 \times 1.25}_{= ICA \times 1.25} \times 1.25 \times CCT_A$$

$$= ICA \times 1.25 \times CCT_A$$

Exemplu 3

Load / Store or machine

Insti.	Freq	CPI
ALU	43%	1
Load	21%	2
store	12%	2
Branch	24%	2

25% of all ALU have 1 operand from mem.

LDR $r_1, [r_2]$

ADD r_3, r_4, r_5 ; $r_3 \leftarrow r_3 + r_4$

new ALU 2cc.

new CPU \rightarrow CPI Branch = 3