CNT	OVR	A	Q[8]	Q	F	M
000		00000000	1	10111011	0	11011011
	+	00100101				00100101 = $\bar{M}$
	0	00100101				
		00010010	1	11011101	1	
001		00001001	0	11101110	1	
010	+	00100101				
	0	00101110				
		00010111	0	01110111	1	
011		00001011	1	00111011	1	
100		00000101	1	10011101	1	
101		00000010	1	11001110	1	
110	+	00100101				
		00100111				
		00010011	1	11100111	1	
111		00001001	1	11110011		

P

CURS 8

16.04.2019

2.3. Other metrics

- **MIPS – Million Instructions Per Second**

$$\boxed{MIPS = \frac{\text{Instruction Count}}{\underbrace{\text{Execution Time}}_{CPU_{time}} \times 10^6} = \frac{\cancel{IC}}{\cancel{IC} \times CPI \times CCT \times 10^6} = \frac{\text{Clock frequency}}{CPI \times 10^6}}$$

$\boxed{10^0}$ 10^3 kilo, 10^6 mega, 10^9 giga, 10^{12} tera, 10^{15} peta, 10^{18} exa, 10^{21} zetta, 10^{24} yotta

10^{-24} yocto, 10^{-21} zepto, 10^{-18} atto, 10^{-15} femto, 10^{-12} pico, 10^{-9} nano, 10^{-6} micro, 10^{-3} mili $\boxed{10^0}$

◎ Example :

$$\text{Clock frequency} = \text{Clock rate} = \frac{1}{20ns} = \frac{1}{20 \cdot 10^{-9}} = 50\ MHz$$

$$MIPS_{original} = \frac{50 \cdot 10^6}{CPI_{original} \cdot 10^6} = \frac{50}{CPI_{original}} = \frac{50}{1,57} = \boxed{31,85}$$

$$CPI_{original} = 0,43 \cdot 1 + 0,57 \cdot 2 = \boxed{1,57}$$

$$CPU_{time\ original} = IC_{original} \cdot 1,57 \times 20ns = \boxed{IC_{original} \times 31,4\ ns}$$

$$MIPS_{optimizat} = \frac{50}{CPI_{optimizat}} = \frac{50}{1,73} = \boxed{28,9}$$

50% = 0,5 (scad 50% din ALU)

$$CPI_{optimizat} = \frac{\left(0,43 \cdot \frac{50}{100}\right) \cdot 1 + 0,57 \times 2}{1 - 0,43 \cdot 0,5} = \boxed{1,73}$$

$$CPU_{time\ optimizat} = IC_{original} \times (1 - 0,43 \cdot 0,5) \times CPI_{optimizat} \times 20ns$$

$$= IC_{original} \times \underbrace{0,785 \times 1,73}_{1,355} \times 20ns =$$

$$= \boxed{IC_{original} \times 27,1\ ns}$$

Concluzie : Uneori MISP variază invers proporțional cu CPU_{time}

- MFLOPS – Million Floating Point Operations Per Second
- GFLOPS – Giga Floating Point Operations Per Second

$$GFLOPS = \frac{\text{Instruction Count}_{FP}}{CPU_{time} \times 10^9}$$

$$\text{Relative } MIPS_x = \frac{\text{Execution time}_{Reference}}{\text{Execution time}_x} \times MIPS_{Reference}$$

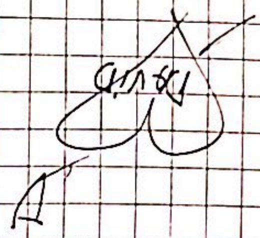
VAX 11/780 → 1 MIPS Machine

$$\text{Normalised } GFLOPS = \frac{\text{Normalised } IC_{FP}}{CPU_{time} \times 10^9}$$

Mc Mahon – Laurence Livermore National Laboratories

Livermore Loops:

Native FP Operation	Normalised FP Operation
+, −, *, COMP	1
DIVIDE, SQRT	4
EXP, SIN, ...	8



2.4. Quantitative Principles of Computer Design

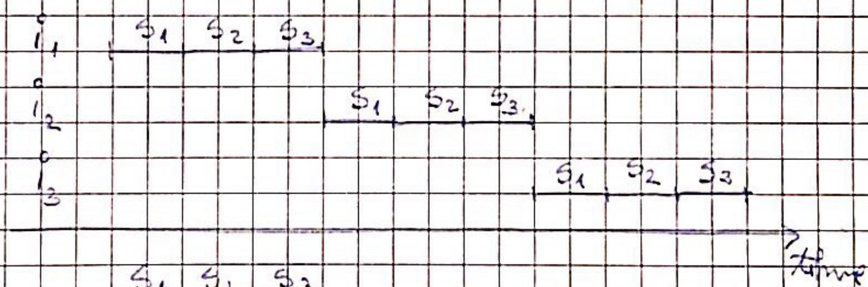
(1.) Use parallelism when possible:

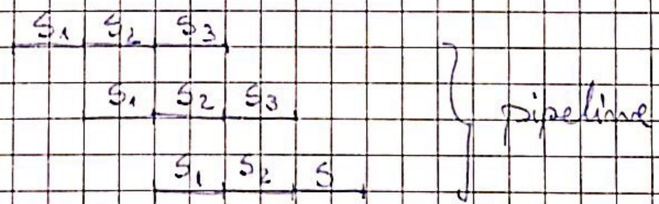
↳ task-level parallelism

↳ data-level parallelism

↳ logical-level parallelism (CLA)

↳ instruction-level pipeline



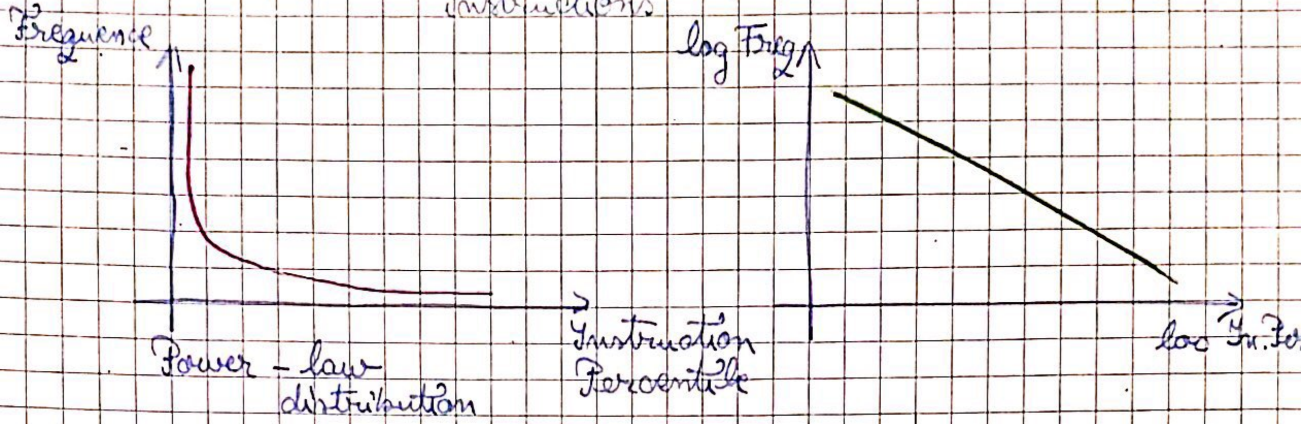


↳ system-level parallelism

Flynn's Taxonomy → SISD
→ SIMD
→ MISD
→ MIMD → NOC (Network On Chip)

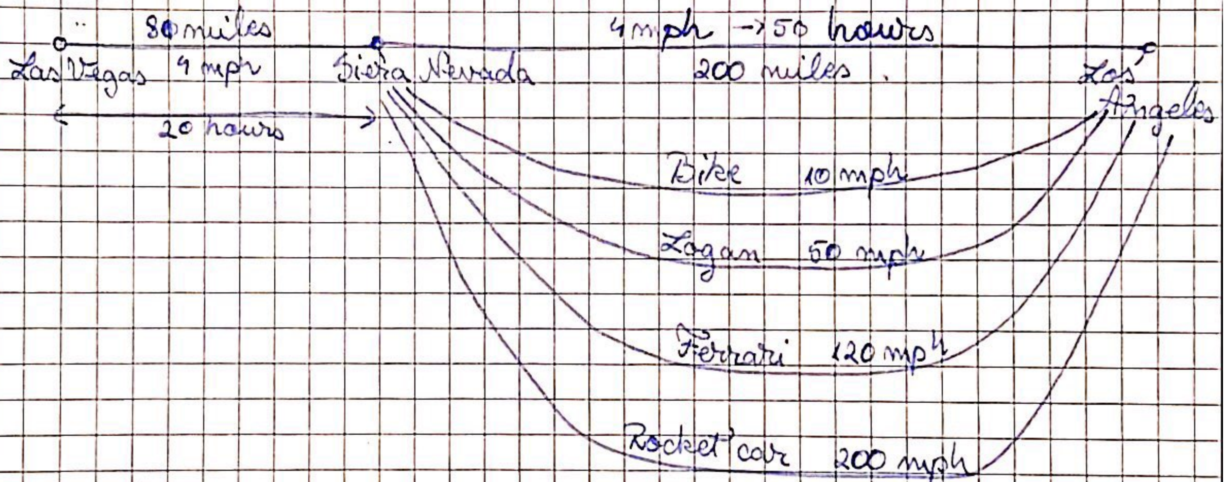
2.) Locality principle:

Rule of thumb: 90% of the time the computer uses 10% of the instructions



3.) Always optimise the common case:

Amdahl's Law:



Vehicle for 2nd part of the trip	Time for 2nd part	Speedup 2nd part	Total time	Overall Speedup
Feet	50	1	70	1
Bike	20	2,5	40	1,75
Logan	4	12,5	24	2,9
Ferrari	1,6	30	21,6	3,23
Rocket car	1	50	21	$\frac{70}{21} = 3,33$

$$\text{Speedup}_{\text{Overall}} = \frac{\text{Time}_{\text{unoptimised}}}{\text{Time}_{\text{optimised}}} = \frac{\text{Time}_{\text{unop.}}}{\text{Time}_{\text{unop.}} \cdot \left(1 - \text{Fracture}_{\text{optimisation}}\right) + \text{Fracture}_{\text{optimisation}}}$$

Scanned with CamScanner

$$\Rightarrow \text{Speedup}_{\text{Overall}} = \frac{1}{\left(1 - \text{Fraction}_{\text{Optimised}}\right) + \frac{\text{Fraction optimised}}{\text{Speedup}}}$$

In the example:

$$\text{Fraction}_{\text{optimised}} = \frac{200}{280} = \frac{20}{28} = \frac{5}{7}$$

$$\text{Speedup}_f = 50$$

$$\text{Speedup overall} = \frac{1}{\frac{2}{7} + \frac{5}{7 \cdot 50}} = \frac{70}{21} = 3.3$$

$$\boxed{\text{Când } \text{Speedup}_f \to \infty \Rightarrow \text{Speedup}_{\text{Overall}} = \frac{1}{1 - \text{Fraction Optimised}}}$$

Cap III Memory hierarchy