* $Performance=1/(Execution time)$
* $CPU\_{time}=CPU Clock Cycles×Clock Cycle Time (T)=Instruction Count (IC)×Cycles per Instruction×T$

$ =\frac{Instructions}{Program}×\frac{Clock Cycles}{Instructuin}×\frac{Seconds}{Clock Cycle}$ $CPI：Cycles per Instruction$

* + Average CPI 可能會< 1
	+ CPI會受Computer architecture 影響，因為pipeline可以影響CPI
* Amdahl’s Law：$T\_{improved}=\frac{T\_{affected}}{Improvement factor}+T\_{unaffected}$, $Speedup=\frac{1}{\frac{f}{n}+(1-f)}$

f：% that can be improved ($T\_{aff}/T\_{orig}$), n：improvement factor



* 【R-format】For srl, shift right and fill with 0 bits. For sra, shift right and fill with sign bits.
* 【I-format】immediate 🡪 constant operand,$ -2^{11}(-2048) to 2^{11}-1(2047)$

For ”slli”, only use imm[4:0] cuz 32-bit data can only be shifted 32 bit positions

For “ld”, ex. ld x9, 120(x22) 🡪 immediate = 120, rs1 = 22, rd = 9

* 【S-format】For “sd”, ex. sd x9, 120(x22) 🡪 immediate = 120, rs1 = 22, rs2 = 9
* 【SB-format】imm只放13 bits中的12 bits，最右邊的”0”被丟掉。

bne rs1, rs2, +8 = bne rs1, rs2, 4 🡪 “+8”表示”PC+8”, “4”表示”immediate的值”

ex.【beq rs1, rs2, imm】”imm” is “Branch offset” = n×32-bit instructions = 4n bytes

 target address = PC + immediate

* 【U-format】lui x19, 26138 (107062541 / 4096取整數) 【lui rd, constant】

 addi x19, x19, 1293 (107062541 % 4096)

* 【UJ-format】jal x1, ProcedureLabel (jumps to address “ProcedureLabel”)

Jalr x0, 0(x1) … return to caller

* ld：memory 🡪 reg。sd：reg 🡪 memory。
* ex. ld x9, 120(x22) 🡪 x22裡面本應要放memory, 此是visit x22(register) get value 🡪 use this value (address of memory) to find the value in the memory.



* Design Principle：Simplicity favors regularity, Smaller is faster, Make the common case fast