Example

- 20% of time doing integer instructions
- 35% percent of time doing I/O

–Which is the better tradeoff?

•Compiler optimization that reduces number of integer instructions by 25% (assume each integer instruction takes the same amount of time)

•Hardware optimization that reduces the latency of each IO operations from 6us to 5us.

I/O optimization is better, 94% cpu_time, to the
Integer manipulation of 95% time.

Example

- Memory operations currently take 30% of execution time.
- A new widget called a “cache” speeds up 80% of memory operations by a factor of 4
- A second new widget called a “L2 cache” speeds up 1/2 the remaining 20% by a factor or 2.
- What is the total speed up?

Speed up is 1/.805 = 1.242. Time = 70% + 24%/4 + 3%/2 + 3% = 80.5% of original time

Example

- A Program is running on a specific machine with the following parameters:
  - Total instruction count: 10,000,000 instructions
  - Average CPI for the program: 2.5 cycles/instruction.
  - CPU clock rate: 200 MHz.
- What is the execution time for this program:

Time = 10 000 000 x 2.5/(200 000 000) = .125 second

Example

- There are four classes of instructions (A, B, C and D) in a certain instruction set.

Consider two different implementations, M1 and M2, of the same instruction set.
M1 has a clock rate of 500MHz.
The average number of cycles for each instruction class on M1 is as follows:

Class	CPI for this class
A	1
B	2
C	3
D	4

M2 has a clock rate of 750MHz. The average number of cycles for each instruction class on M2 is as follows:

Class	CPI for this class
A	2
B	2
C	4
D	4

Question: If the number of instructions executed in a certain program is divided equally among the classes of instructions in question, how much faster is M2 than M1?

M2 is faster By a factor of 1.25