CSE213

Microcontroller Programming

Microprocessor Performance

Performance

- Response time: The time between start and finish of the task (e.g., execution time)

- Throughput: total amount of work done in a given time

Question

- Suppose that we replace the processor in a computer by a faster model
  - Does this improve the response time?
  - How about the throughput?

Question

- Suppose we add an additional processor to a system that uses separate processors for separate tasks.
  - Does this improve the response time?
  - Does this improve the throughput?

Computer Clock Times

- Computers run according to a clock that runs at a steady rate
- The time interval is called a *clock cycle* (eg, 10ns).
- The *clock rate* is the reciprocal of clock cycle - a frequency, how many cycles per sec (eg, 100MHz).
  - 10 ns = 1/100,000,000 (clock cycle), same as:-
  - 1/10ns = 100,000,000 = 100MHz (clock rate).

Purchasing Decision

- Computer A has a 100MHz processor
- Computer B has a 300MHz processor
- So, B is faster, right?
- NOT SO SURE!
- Now, let's get it right…..

CPU Performance

- CPU Execution time for a program

= CPU Clock Cycles for a program x Clock Cycle time

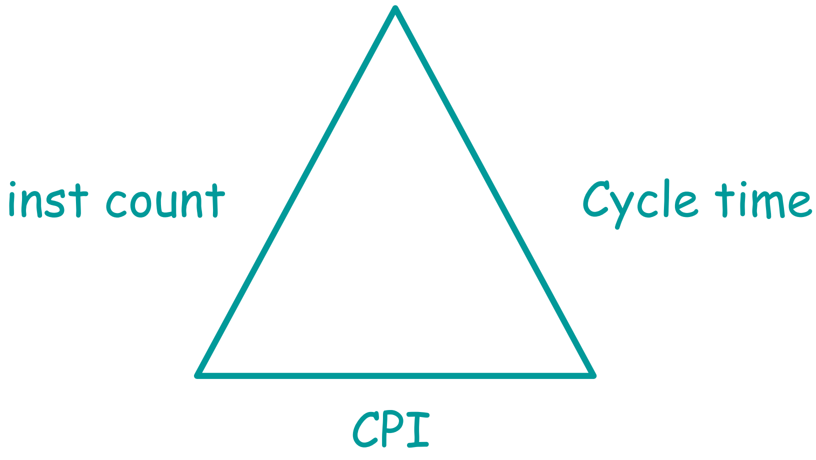
= CPU Clock Cycles for a program / Clock rate

CPU Performance

- CPU clock cycles

  = instruction count x CPU (Cycles per Instruction)
- CPU time

  = instruction count x CPI x Clock cycle time



Cycles Per Instruction (Throughput)

"Average Cycles per Instruction"

CPI = (CPU Time * Clock Rate) / Instruction Count
= Cycles / Instruction Count

$$\text{CPU time} = \text{Cycle Time} \times \sum_{j=1}^{n} CPI_j \times I_j$$

$$CPI = \sum_{j=1}^{n} CPI_j \times F_j \qquad \text{where } F_j = \frac{I_j}{\text{Instruction Count}}$$

"Instruction Frequency"

Example: Calculating CPI

Base Machine (Reg / Reg)

Op	Freq	Cycles	CPI(i)	(% Time)
ALU	50%	1	.5	(33%)
Load	20%	2	.4	(27%)
Store	10%	2	.2	(13%)
Branch	20%	2	.4	(27%)
			1.5	

Typical Mix of instruction types in program

Performance

(Absolute) Performance

$$\text{Performance}_X = \frac{1}{\text{Execution time}_X}$$

Relative Performance

" X is n times faster than Y" means

$$n = \frac{\text{Performance}_X}{\text{Performance}_Y} = \frac{\text{Execution time}_Y}{\text{Execution time}_X}$$

Some performance measures

- MIPS
  - Million Instructions Per Second
- MFLOPS
  - Million Floating Point Operations Per Second
- Benchmarks: SPECs (Standard Performance Evaluation Corporation)
  - Average Performance over a set of example programs

Amdahl's Law

The execution time after making an improvement to the system is given by

Exec time after improvement = I / A + E

I = execution time affected by improvement

A = amount of improvement

E = execution time unaffected

Amdahl’s Law

Suppose that program runs 100 seconds on a machine and multiplication instructions take 80% of the total time. How much do I have to improve the speed of multiplication if I want my program to run 5 times faster?

20 seconds = 80 seconds / n + 20 seconds

=> it is impossible!

References

- Measuring Performance, Chris Clack, B261 Systems Architecture.
- CSCE 350 Computer Architecture, Rabi Mahapatra