Computer Architecture
Homework 2
Theme: Basics and Performance
All questions carry equal weight. Show your work to receive credit.

---

1. **A common measure of performance for a processor is the rate at which instructions are executed which is often expressed in MIPS. Express MIPS rate in terms of clock rate and CPI.**

Say,

- Clock frequency of a processor is f. Then, clock cycle time is 1/f
- CPI is the average processor cycles per instruction for a program
- CPI_i is the number of processor cycle required for an instruction of type i
- I_i is the number of instructions of type i executed by the program
- N is the number of type of instructions executed by program
- I_c is the total number of instructions executed by the program

Then, average number of processor cycles per instruction for the program can be calculated as

$$CPI = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$$

Total processor time required to execute program (i.e. I_c instructions) is

$$T = I_c \times CPI \times \frac{1}{f}$$

Instructions that can be executed in one second = ${}^{I_c}/_{T} = {}^{f}/_{CPI}$ (from above equation)

Number of Millions of Instructions that can be executed by a processor in one second is

$$f \,/\, (CPI \times 10^6)$$

Hence, we can express the MIPS rate in terms of the clock rate and CPI as fellows

$$MIPS\ rate = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

2. A benchmark program is run on a 6 GHz processor. The object code consists of the following instruction mix and clock cycle counts. Determine the effective CPI, MIPS rate and execution time for this program.

Instruction Type	Instruction Count	Clock Count
Integer Arithmetic	**40,000**	**1**
Data Transfer	**35, 000**	**2**
Floating Point	**18,000**	**3**
Control transfer	**11,000**	**2**

Given,

- The speed of processor is 6GHz.
- The instructions in program are 40000+35000+18000+11000 = 104000

Average cycles per instruction (CPI) = $\frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$

= [(40000 × 1) + (35000 × 2) + (18000 × 3) + (11000 × 2)] / 104000

= [(40 × 1) + (35 × 2) + (18 × 3) + (11 × 2)] / 104

= (40 + 70 + 54 + 22) / 104

= 1.788 (5 points)

Rate at which instructions are executed MIPS = $f\ /\ (CPI\ \times\ 10^6)$

= 6×10^9 / (1.788×10^6) = 6000 / 1. 788 = 3355.70 (5 points)

Total execution time for this program T

= Total Instructions in program * Average cycles per instruction / processor speed

= 104000 × 1. 788 / (6×10^9) sec

= 0.000031 seconds (or 0.031 milliseconds(ms) or 31 microseconds(μs)) (4 points)

3. In the Intel architecture, the addresses are staggered into two separate units (e.g. all even addressed words in one unit and odd ones in another). What might be the purpose of this technique?

The purpose of this technique is to increase the performance. Traditionally, CPU instruction execution speed is much higher than memory access speed. Therefore the instructions which involve a read / write to memory executed much slower as the processor has to wait for memory access operation to complete. The delay could not be avoided as memory operates as lower speed compared to processor.

But, by separating memory into two modules – odd and even – processor could access even and odd addresses in parallel. This helps in improving overall performance of system when the program execution involves a lot of memory access operations.

4. What is Amdahl's Law? Explain the Speedup equation.

$$Speedup = \frac{1}{(1-f) + \frac{f}{N}}$$

Where f is the fraction of code which can be infinitely parallelizable with no scheduling overhead and N is the number of parallel processors.

Amdahl's law:

- When f is small, the use of parallel processors has little effect.
- As N approaches infinity, speedup is bound by 1/(1-f), so that there are diminishing returns for using more processors.

5. What are SPEC Benchmarks? Discuss any three component SPEC benchmarks. Discuss the use of AM, GM and HM for performance evaluation. Which means should you use for which metrics?

Benchmarks are programs that are compiled and run on a computer, with their execution timed, to test various performance aspects of a computer. The SPEC benchmarks happen to be the most widely-used and well-recognized benchmark programs.

Examples:

- 400.perlbench - a C-language program that interprets PERL code, applied to a set of three programs.
- 453.povray - a C++ program that renders 3D images using ray-tracing.
- 485.wf - a program written in C and Fortran for weather forecasting.

AG, GM and HM

- Arithmetic mean is used to compare the execution time performance of several systems
- Geometric mean, with takes the nth root of the product of n numbers, is most appropriate for normalized ratios. It should be used for the base metric, the speed metric and rate metric.
- Harmonic mean is appropriate to compute the mean of the rates of execution.

6. List and briefly define some of the techniques used in contemporary processors to increase performance.

1. Pipelining
   Pipelining enables a processor to work simultaneously on multiple instructions by performing a different phase for each of the multiple instructions at the same time. The processor overlaps operations by moving data and instructions into a conceptual pipe with all stages of the pipe processing simultaneously.
2. Branch prediction
   Processor looks ahead in the instruction code fetched from memory and predicts which branches or groups of instructions are likely to be processed next, pre-fetches and buffers them so that the processor is kept busy.
3. Super scalar execution
   Ability to issue more than one instruction in every processor clock cycle. i.e., in effect, multiple parallel pipelines are used.
4. Data flow analysis
   Processor analyzes the dependency of instructions on data, to create an optimized schedule of instructions so that they can be executed when ready, independent of the original program order.
5. Speculative execution
   Processor speculatively executes instructions ahead of their actual appearance in the program execution, holding the results in temporary locations

7. **Suppose a program P makes extensive use of floating-point operations, so that 35% of the time gets consumed in floats. You designed a new hardware that doubles the speed of float executions. What is the effective speedup achieved?**

$$Speedup = \frac{1}{(1-f)+\frac{f}{N}}$$

$$= \frac{1}{(1-0.35)+\frac{0.35}{2}}$$

$$= 1.21$$