Chapter 4

4.1 System attributes to performance

The ideal performance of a computer system demands a perfect match between machine capability and program behavior.

It is impossible to achieve a perfect match between hardware and software by merely improving only a few factors without touching other factors.

The simplest measure of program performance is the turnaround time, which includes disk and memory accesses, input and output activities, compilation time, OS overhead, and CPU time.

Clock Rate and CPI

The CPU (or simply the processor) of today's digital computer is driven by a clock with a constant *cycle time* (τ in microseconds). The inverse of the cycle time is the *clock rate* ($f=1/\tau$ in megahertz). The size of a program is determined by its *instruction count* (I_c), in terms of the number of machine instructions to be executed in the program. Different machine instructions may require different numbers of clock cycles to execute. Therefore, the *cycle per instruction* (CPI) becomes an important parameter for measuring the time needed to execute each instruction. For a given instruction set, we can calculate an *average* CPI over all instruction types, provided we know their frequencies of appearance in the program.

Performance Factors

The CPU time (T in seconds/program) needed to execute the program is estimated by finding the product of three contributing factors:

$$T = I_c \times \mathrm{CPI} \times \tau$$

The execution of an instruction requires going through a cycle of events involving the instruction fetch, decode, operand(s) fetch, execution, and store results. In this cycle, only the instructions decode and execution phases are carried out in the CPU. The remaining three operations may be required to access the memory. We define a *memory cycle* as the time needed to complete one memory reference.

Usually, a memory cycle is k times the processor cycle τ. The value of k depends on the speed of the memory technology and processor-memory interconnection scheme used.

$$\mathrm{T} = \mathrm{I_c} \times (\mathrm{p} + \mathrm{m} \times \mathrm{k}) \times \tau$$

Where p is the number of processor cycles needed for the instruction decode and execution, m is the number of memory references needed; k is the ratio between memory cycle and processor cycle.

MIPS Rate

Let C be the total number of clock cycles needed to execute a given program. Then the CPU time $\mathrm{T} = \mathrm{C} \times \tau$. Furthermore, $\mathrm{CPI} = \mathrm{C}/\mathrm{I}_c$. The processor speed is often measured in terms of *million instructions per second* (MIPS).

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

Throughput Rate

Another important concept is related to how many programs a system can execute per unit time, called the *system throughput* W_s (in programs/second). In a multiprogrammed system, the system throughput is often lower than the *CPU throughput* W_p define by:

$$\mathrm{W}_p = \frac{f}{I_c \times CPI}$$

Note that $\mathrm{W}_p = MIPS \times 10^6 / I_c$. The unit for W_p is programs/second. The CPU throughput is a measure of how many programs can be executed per second, based on the MIPS rate and average program length (I_c). The reason why $W_s < W_p$ is due to the additional system overheads caused by the I/O, compiler, and OS when multiple programs are interleaved for CPU execution by multiprogramming or time-sharing operations. If the CPU is kept busy in a perfect program-interleaving fashion, then $W_s = W_p$. This will probably never happen, since the system overhead often causes an extra delay and the CPU may be left idle for some cycles.

Example 1: A 40 MHz processor was used to execute a benchmark program with the following instruction mix and clock cycle counts:

Instruction type	Instruction count	Clock cycle count
Integer arithmetic	45000	1
Data transfer	32000	2
Floating point	15000	2
Control transfer	8000	2

Determine the effective CPI, MIPS rate, and execution time for this program.

Example 2: Consider the execution of an object code with 200000 instructions on a 40 MHz processor. The program consists of four major types of instructions. The instruction mix and the number of cycles (CPI) needed for each instruction type are given below based on the result of a program trace experiment:

Instruction type	CPI	Instruction mix
Arithmetic and logic	1	60%
Load/store with cache hit	2	18%
Branch	4	12%
Memory reference with cache mix	8	10%

a) Calculate the average CPI when the program is executed on a uniprocessor with the above trace results.
b) Calculate the corresponding MIPS rate based on the CPI obtained in part (a).

Example 3: The execution times (in seconds) of four programs on three computers are given below:

Program	Execution Time (in seconds)		
	Computer A	Computer B	Computer C
Program 1	1	10	20
Program 2	1000	100	20
Program 3	500	1000	50
Program 4	100	800	100

Assume that 100,000,000 instructions were executed in each of the four programs. Calculate the MIPS rate of each program on each of the three machines. Based on these rating, can you draw a clear conclusion regarding the relative performance of the three computers? Give reasons if you find a way to rank them statistically.