COMPUTER ORGANIZATION AND ARCHITECTURE (COA)

EET 2211

4TH SEMESTER – CSE & CSIT

CHAPTER 2, LECTURE 6

CHAPTER 2 – PERFORMANCE ISSUES

TOPICS TO BE COVERED

- Designing for performance
- Multicore, MICs and GPGPUs
- Amdahl’s & Little’s Law
- Basic measures of Computer performance
- Calculating the mean

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Understand the key performance issues that relate to computer design.
- Explain the reasons for the move to multicore organization, and understand the trade-off between cache and processor resources on a single chip.
- Distinguish among multicore, MIC and GPGPU organizations.
- Summarize some of the issues in computer performance assessment.
- Explain the differences among arithmetic, harmonic and geometric means.

Overview of Previous Lecture

Designing for Performance:

<u>Microprocessor Speed :</u>

Pipelining:

Branch prediction:

Superscalar execution:

Data flow analysis:

Speculative execution:

<u>Performance Balance:</u>

<u>Improvements in Chip Organization and Architecture:</u>

MULTICORE,MICS,GPGPUS:

Amdhal's Law & Little's Law:

Contd.

- **AMDAHL'S LAW:**

$$\text{Speedup} = \frac{\text{Time to execute program on a single processor}}{\text{Time to execute program on N parallel processors}}$$

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

- **LITTLE'S LAW:**

We have a steady state system to which items arrive at an average rate of λ items per unit time. The items stay in the system an average of W units of time. Finally, there is an average of L units in the system at any one time. Little's Law relates these three variables as $L = \lambda\ W$

Basic Measures of a Computer Performance

- In evaluating processor hardware and setting requirements for new systems, performance is one of the key parameters to consider, along with cost, size, security, reliability and in some cases, power consumption.

- In this section, we look at some traditional measures of processor speed. In the next section, we examine benchmarking, which is the most common approach to assessing processor and computer system performance.

1. Clock Speed

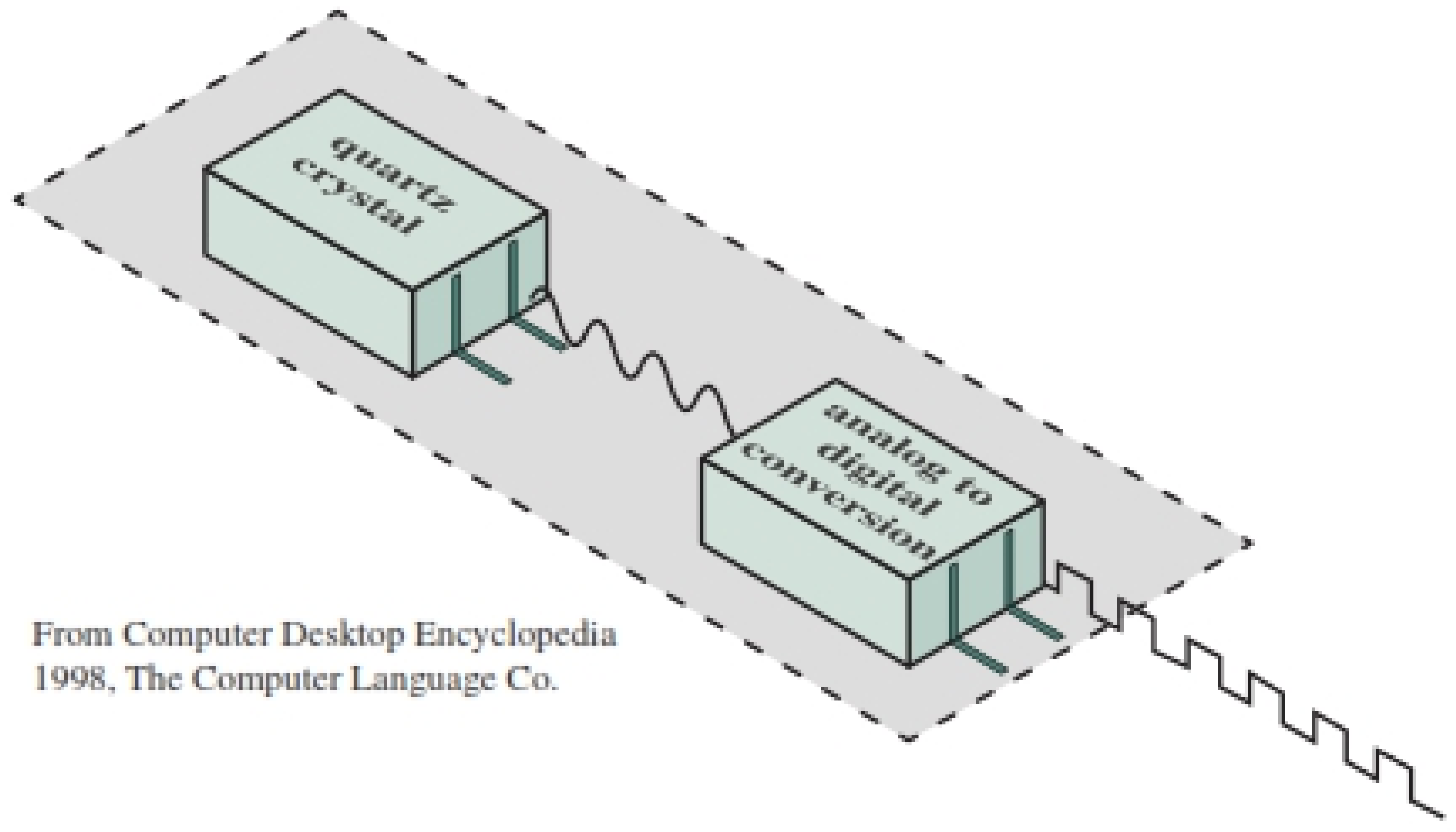
- ✓ Operations performed by a processor, such as fetching an instruction, decoding the instruction, performing an arithmetic operation, and so on, are governed by a system clock.
- ✓ the speed of a processor is dictated by the pulse frequency produced by the clock, measured in cycles per second, or Hertz (Hz).
- ✓ The rate of pulses is known as the **clock rate, or clock speed.** One increment, or pulse, of the clock is referred to as a **clock cycle, or a clock tick.** The time between pulses is the **cycle time.**

- ✓ The clock rate is not arbitrary, but must be appropriate for the physical layout of the processor. Actions in the processor require signals to be sent from one processor element to another.

- ✓ Most instructions on most processors require multiple clock cycles to complete. Some instructions may take only a few cycles, while others require dozens.

- ✓ In addition, when pipelining is used, multiple instructions are being executed simultaneously.

- ✓ Thus, a straight comparison of clock speeds on different processors does not tell the whole story about performance.



From Computer Desktop Encyclopedia
1998, The Computer Language Co.

System Clock

2. Instruction Execution Rate

- A processor is driven by a clock with a constant frequency f or, equivalently, a constant cycle time τ, where **τ = 1/ f**.

- The instruction count, I_c , for a program is the number of machine instructions executed for that program until it runs to completion or for some defined time interval.

- An important parameter is the average cycles per instruction (CPI) for a program.

- The overall CPI:

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

- The processor time T needed to execute a given program can be expressed as:

$$T = I_c \times CPI \times \tau$$

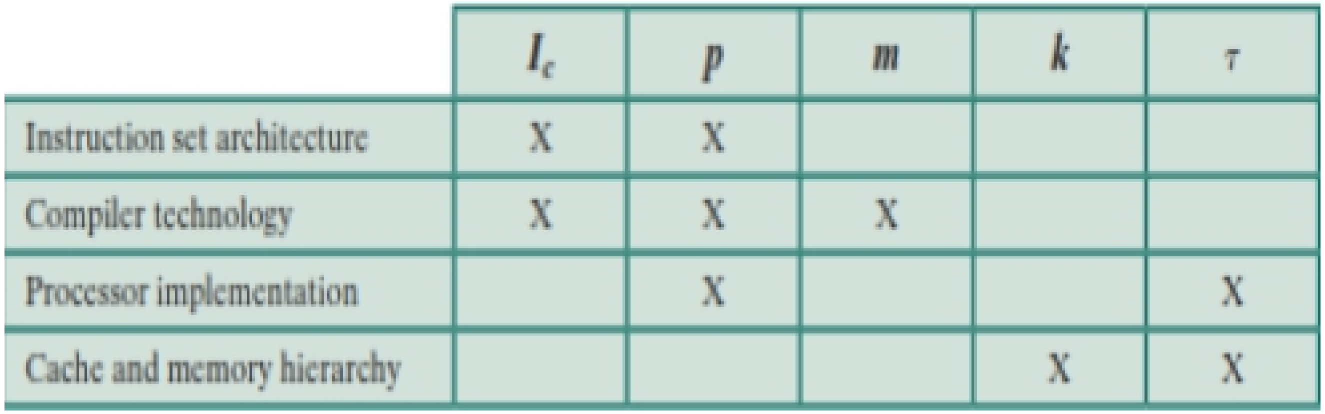
- The above formula can be refined by recognizing that during the execution of an instruction, part of work done by the processor, and part of the time a word is being transferred to or from memory. The time to transfer depends on the memory cycle time which may be greater than the processor cycle time.

$$T = I_c \times [p + (m \times k)] \times \tau$$

- Here p = number of processor cycles needed to decode and execute the instruction, m= number of memory references needed and k = ratio between memory cycle time and processor cycle time.

- The five performance factors in the preceding equation (I_c, p, m, k, τ) are influenced by four system attributes: the design of the instruction set (known as *instruction set architecture);* compiler technology (how effective the compiler is in producing an efficient machine language program from a high-level language program); processor implementation; and cache and memory hierarchy.

TABLE: Performance Factors and System Attributes



	I_c	p	m	k	τ
Instruction set architecture	X	X			
Compiler technology	X	X	X		
Processor implementation		X			X
Cache and memory hierarchy				X	X

- A common measure of performance for a processor is the rate at which instructions are executed, expressed as millions of instructions per second (MIPS), referred to as the **MIPS rate.** We can express the MIPS rate in terms of the clock rate and *CPI as follows:*

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

- Another common performance measure deals only with floating-point instructions. These are common in many scientific and game applications. Floating-point performance is expressed as millions of floating-point operations per second(MFLOPS),defined as follows:

$$\text{MFLOPS rate} = \frac{\textit{Number of executed floating-point operations in a program}}{\textit{Execution time} \times 10^6}$$

Equation summary

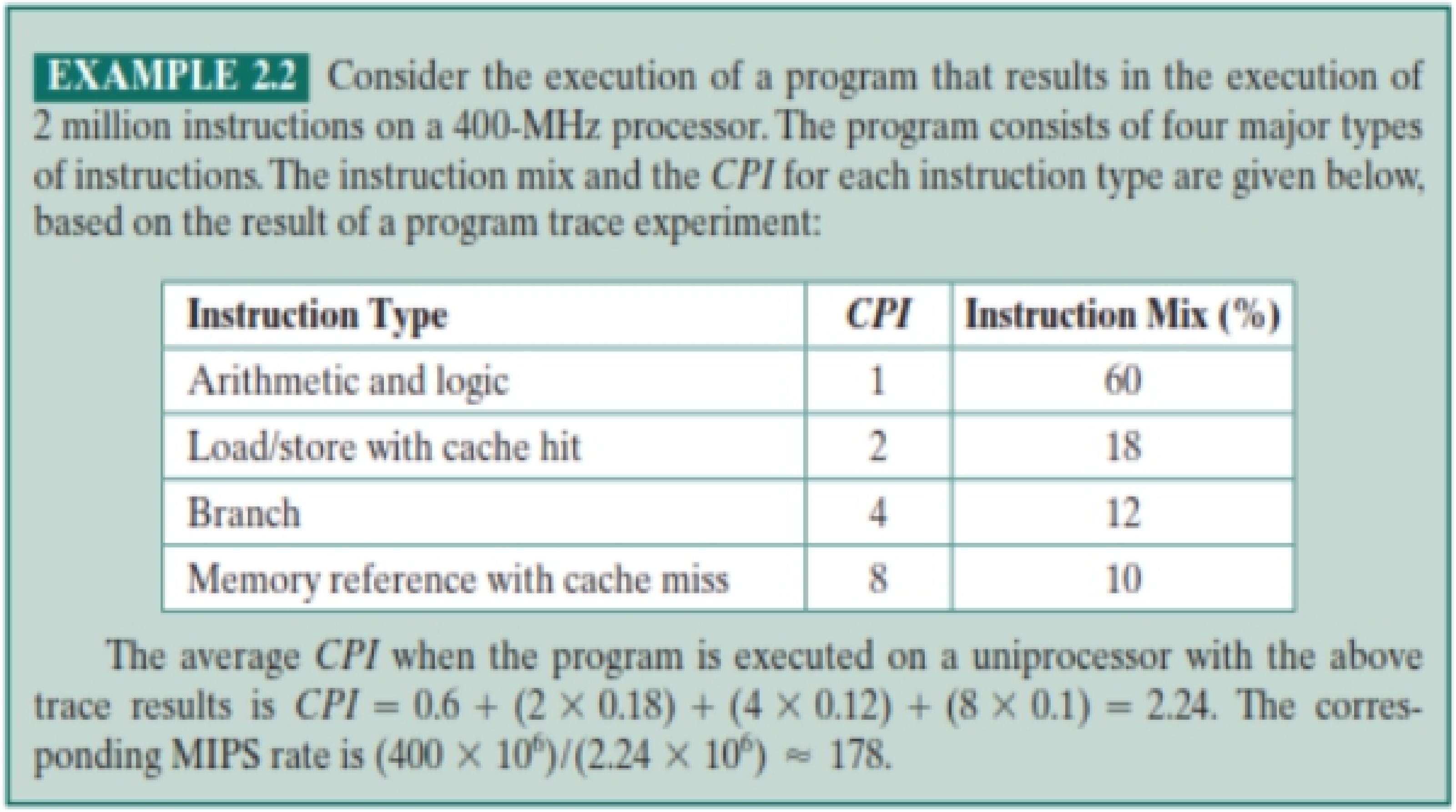
1. $\text{CPI} = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$

2. $\text{T} = \text{I}_\text{c} \times \text{CPI} \times \tau$

3. $\text{T} = \text{I}_\text{c} \times [\text{p} + (\text{m} \times \text{k})] \times \tau$

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

5. $\text{MIPS rate} = \frac{Number\ of\ ececuted\ floating-point\ operations\ in\ a\ program}{Execution\ time \times 10^6}$



EXAMPLE 2.2 Consider the execution of a program that results in the execution of 2 million instructions on a 400-MHz processor. The program consists of four major types of instructions. The instruction mix and the *CPI* 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 miss	8	10

The average *CPI* when the program is executed on a uniprocessor with the above trace results is $CPI = 0.6 + (2 \times 0.18) + (4 \times 0.12) + (8 \times 0.1) = 2.24$. The corresponding MIPS rate is $(400 \times 10^6)/(2.24 \times 10^6) \approx 178$.

THANK YOU