University of Nottingham
UK | CHINA | MALAYSIA

Computer Performance

Ying Weng

2023 Spring Semester

COMP1047 Systems & Architecture

Outline

- **Timing Performance**
- **Other Performance Metrics**

Response Time and Throughput

- Response time (execution time)
  - The total time for the computer to complete a task
- Throughput (bandwidth)
  - The number of tasks completed per unit time
- How are response time and throughput affected?
  - Replacing the processor with a faster version?
    - Improves both response time and throughput
  - Adding more processors?
    - Improves throughput
- Our focus will be on response time

Measuring Performance

- Response time (Elapsed Time)
  - Total time to complete a task, including all aspects:
  - Disk accesses, memory accesses, I/O activities, CPU time, etc.
- CPU execution time (CPU time)
  - The actual time the CPU spends computing a specific task
  - Does not include time spent waiting for I/O or running other programs
  - Can be further divided into
    - User CPU time: the CPU time spent in a program itself
    - System CPU time: the CPU time spent in the OS performing tasks on behalf of the program

Relative Performance

- To maximize performance, we want to minimize execution time, then we can relate performance and execution time for a computer X as

  Performance_X = 1 / Execution_Time_X

- Computer X is n times faster than computer Y, then their relative performance *n* is

  n = Performance_X / Performance_Y

  = Execution_Time_Y / Execution_Time_X

- **Question: If Computer A runs a program in 10 secs and Computer B runs the same program in 15 secs, which one is faster? And by how much?**

Relative Performance

- To maximize performance, we want to minimize execution time, then we can relate performance and execution time for a computer X as

Performance_X = 1 / Execution_Time_X

- Computer X is n times faster than computer Y, then their relative performance *n* is

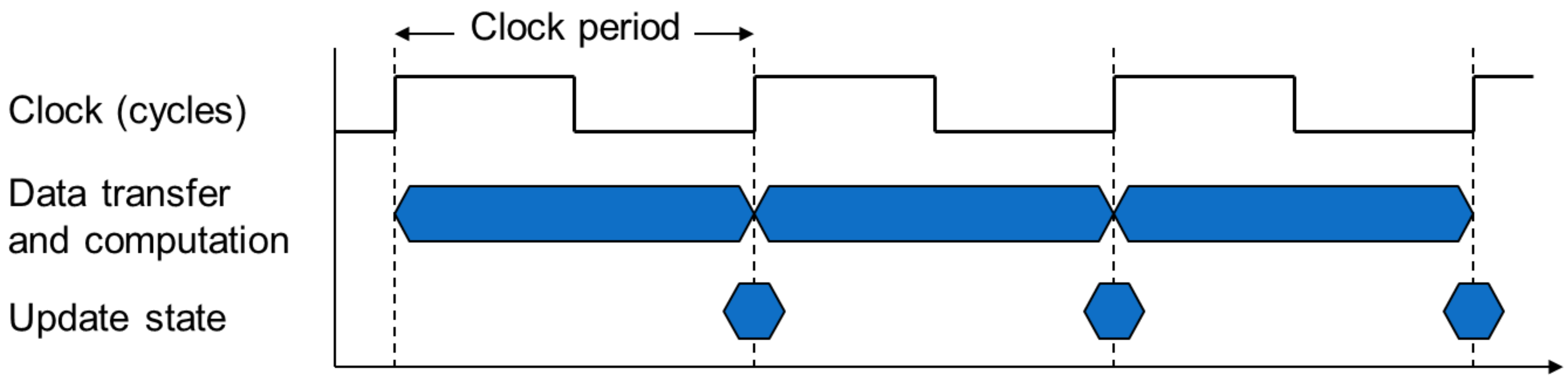
n = Performance_X / Performance_Y

= Execution_Time_Y / Execution_Time_X

- **Question: If Computer A runs a program in 10 secs and Computer B runs the same program in 15 secs, which one is faster? And by how much?**
- **Computer A is 1.5 times as fast as Computer B**

CPU Clocking

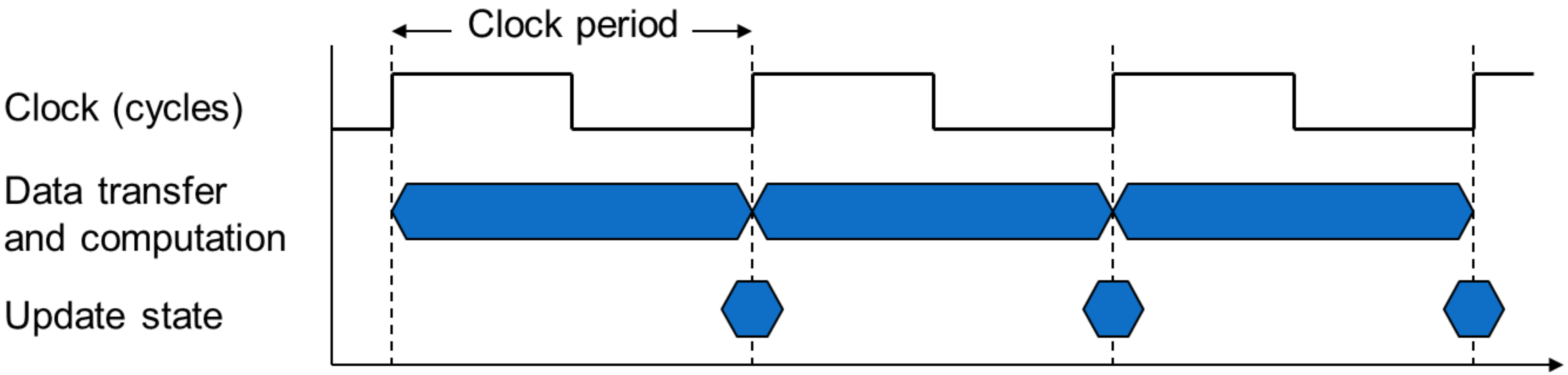
- Computers are constructed using a clock that determines when events take place in hardware
- The clock signal is produced by an external oscillator circuit that generates a consistent number of pulses each second in the form of a periodic square wave
- One clock cycle (clock tick): the unit of the CPU clock. It must be constant as CPU clock runs.
- Clock period: the time spend to run one clock cycle.



University of Nottingham
UK | CHINA | MALAYSIA

CPU Clocking

- One clock cycle (clock tick): the unit of the CPU clock. It must be constant as CPU clock runs.
  - e.g. 250ps = 0.25ns = 250×10^{-12}s (ps: picosecond)
- Clock frequency (clock rate): cycles per second, which is the inverse of the clock period
  - e.g. 4.0GHz = 4000MHz = 4.0×10^{9}Hz
  - **Clock frequency = 1/Clock period**



CPU Time

- A simple formula relates CPU clock cycles and CPU clock period to CPU time

$$\begin{aligned}\text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Period} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}\end{aligned}$$

- The performance can be improved by
  - Reducing number of clock cycles numerator
  - Increasing clock rate denominator
  - But they can not be altered arbitrarily – needs to adhere to circuit limitations.

CPU Time Example

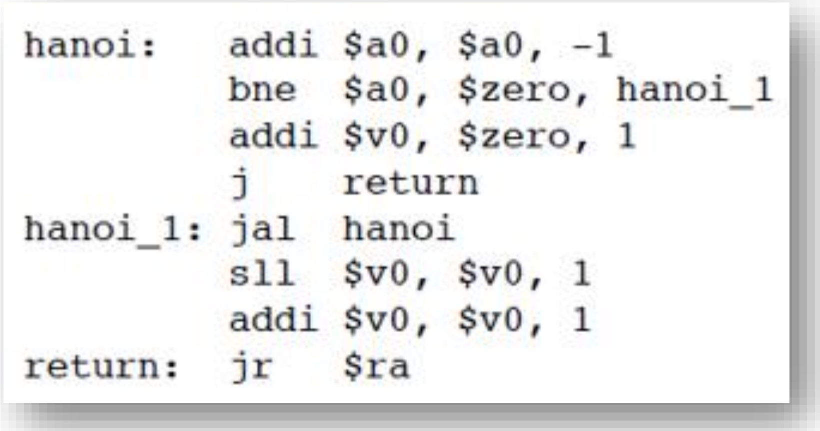
- Computer A: 2GHz clock, 10s CPU time to run a program.
- Build Computer B
  - Aim for 6s CPU time to run the same program
  - Can do a faster clock, but it requires 1.2 times as many clock cycles as computer A (Clock Cycles_B = 1.2 x Clock Cycles_A)
- **Question: What clock rate should we tell the designer to target on Computer B?**

Instruction Performance

- The computer need to execute the instructions to run the program, and the execution time should depend on the number of instructions in a program
- Clock cycles per instruction (CPI): the average number of clock cycles each instruction takes to execute
- The number of clock cycles required for a program is

$$\text{CPU Clock Cycles} = \text{Instruction Count} \times \text{CPI}$$

- The instruction count for a program
  - Determined by program, ISA and compiler
- CPI
  - Determined by how you design the CPU



```
hanoi:   addi $a0, $a0, -1
         bne  $a0, $zero, hanoi_1
         addi $v0, $zero, 1
         j    return
hanoi_1: jal  hanoi
         sll  $v0, $v0, 1
         addi $v0, $v0, 1
return:  jr   $ra
```

The Classic CPU Performance Equation

$$\begin{aligned}\text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Period} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}\end{aligned}$$

$$\text{CPU Clock Cycles} = \text{Instruction Count} \times \text{CPI}$$

Use Instruction Count, CPI, Clock Period or Clock Rate to describe the CPU Time

$$\begin{aligned}\text{CPU Time} &= \text{Instruction Count} \times \text{CPI} \times \text{Clock Period} \\ &= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}\end{aligned}$$

CPI Example

- Two computers
  - Computer A: Clock Period = 250ps, CPI = 2.0
  - Computer B: Clock Period = 500ps, CPI = 1.2
  - Same ISA
- **Which computer is faster, and by how much?**

$$\begin{aligned}\text{CPU Time}_A &= \text{Instruct. Count} \times \text{CPI}_A \times \text{Clock Period}_A \\ &= C \times 2.0 \times 250\text{ps} = C \times 500\text{ps}\end{aligned}$$

A is faster

$$\begin{aligned}\text{CPU Time}_B &= \text{Instruct. Count} \times \text{CPI}_B \times \text{Clock Period}_B \\ &= C \times 1.2 \times 500\text{ps} = C \times 600\text{ps}\end{aligned}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{C \times 600\text{ps}}{C \times 500\text{ps}} = 1.2$$

A is 1.2 times as fast as B

CPI In More Details

- Typically different instruction classes take different numbers of cycles

$$\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)$$

- Average CPI

weighted average of CPI

ICi is weighted SUM of instructions
IC is just the SUM of instructions

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^{n} \left(\text{CPI}_i \times \underbrace{\frac{\text{Instruction Count}_i}{\text{Instruction Count}}}_{\boxed{\text{Relative Frequency}}} \right)$$

Exercise

- A compiler designer is trying to decide between two code sequences for a particular computer. The CPI for each instruction class and the instruction counts for each instruction class are given as

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- **Question: What is the average CPI for each sequence?**

Exercise

- A compiler designer is trying to decide between two code sequences for a particular computer. The CPI for each instruction class and the instruction counts for each instruction class are given as

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- **Question: What is the average CPI for each sequence?**

- Sequence 1: instruction count = 5
  - Clock Cycles
    = 1×2 + 2×1 + 3×2
    = 10
  - Avg. CPI = 10/5 = 2.0

- Sequence 2: instruction count = 6
  - Clock Cycles
    = 1×4 + 2×1 + 3×1
    = 9
  - Avg. CPI = 9/6 = 1.5

Performance Summary

- The basic components of performance and how each is measured

Components of performance	Units of measure
CPU execution time for a program	Seconds
Instruction count	Instructions executed for the program
Clock cycles per instruction (CPI)	Average number of clock cycles per instruction
Clock cycle time (period)	Seconds per clock cycle

- The big picture

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Period}$$

Factors Affecting Performance

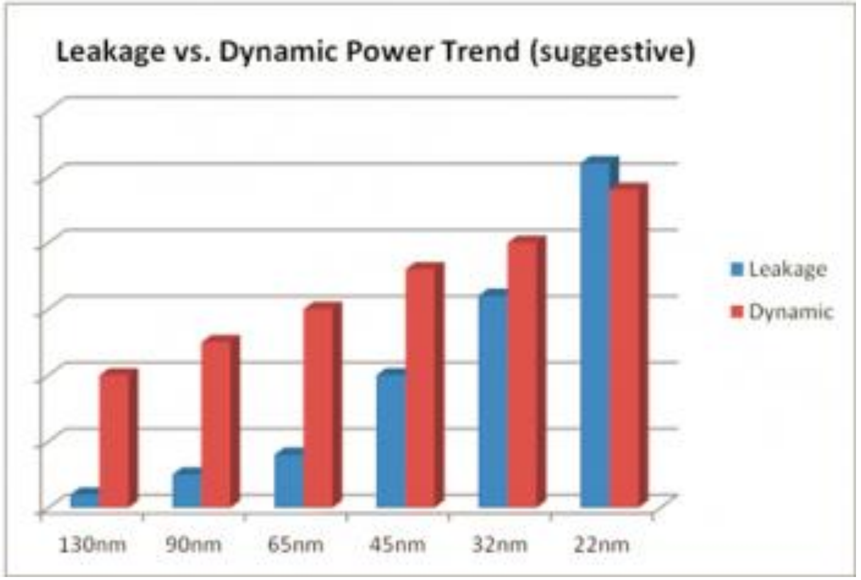
The performance of a program may depend on a series of hardware or software components

- Program
  - Affects instruction count
  - Possibly affects CPI – by favoring slower or faster instructions
- Programming language
  - Affects instruction count
  - Affects CPI – because of its own features
- Compiler
  - Affects instruction count
  - Affects CPI
- ISA
  - Affects instruction count
  - Affects CPI, clock rate

Other Important Metrics

Power & Energy

- Dynamic power: P=0.5*C * V^2 * f
  - C: effective capacitance
  - V: voltage
  - f: frequency, usually linear with V
- Doubling the clock frequency consumes more power than a quadcore processor!
- Static/Leakage power becomes the dominant factor in the most advanced process technologies.
- Power is the direct contributor of "heat"
  - Packaging of the chip
  - Heat dissipation cost
- Dynamic energy = P * t
  - Battery life is related to energy
  - Lower power does not necessarily mean better battery life



Other Important Metrics

Bandwidth

- The amount of work (or data) during a period of time
  - Network or Disks: MB/sec, GB/sec, Gbps, Mbps
  - Game or Video: Frames per second
- Also called “throughput”, but with subtle differences
- “Work done” / “execution time”

Reliability

- Mean time to failure (MTTF)
  - Average time before a system stops working
  - Very complicated to calculate for complex systems
- Hardware can fail because of
  - Electromigration
  - Temperature
  - High-energy particle strikes

Summary

- Concepts of and basic factors that affect response time and bandwidth.
- Relative performance that is used to compare performance of different computers.
- Concepts and calculation of CPU time related factors.
- Concepts and calculation of CPI related factors.
- Knowledge of other performance metrics.

University of Nottingham

UK | CHINA | MALAYSIA

Stay Tuned.