# Computer Organization and Architecture

## Performance of Computer

Veena Thenkanidiyoor

National Institute of Technology Goa

## System Software

- Responsible for the coordination of all activities in a computing system
- Program that reads numbers stored in a file on disk, sorts the number in ascending order and displays the sorted list on the screen
  - Text editor
  - Command prompt – Operating system
  - Compiler
  - Linker and loader
  - **Operating system**
  - **.....**
- **Operating system**
  - Large program, collection of routines
  - Control the sharing and interaction of various computer units
  - Assigning resources to individual application programs

## Operating System as A Monitor

- Assign resources to an application program
  - Memory
  - Magnetic disk
  - Program and data file
  - Movement of data between memory and disk
  - I/O

- System having
  - One processor
  - One disk
  - One printer

Program : processData.c
- Read a data file from disk
- Process
- Print the result

– Compiled : ProcessData.out
- Disk

– $ ./ProcessData
- Loaded onto memory
- Linking
- Execution

OS → Program

ProcessData → "

Switching of control.

## Operating System as A Monitor

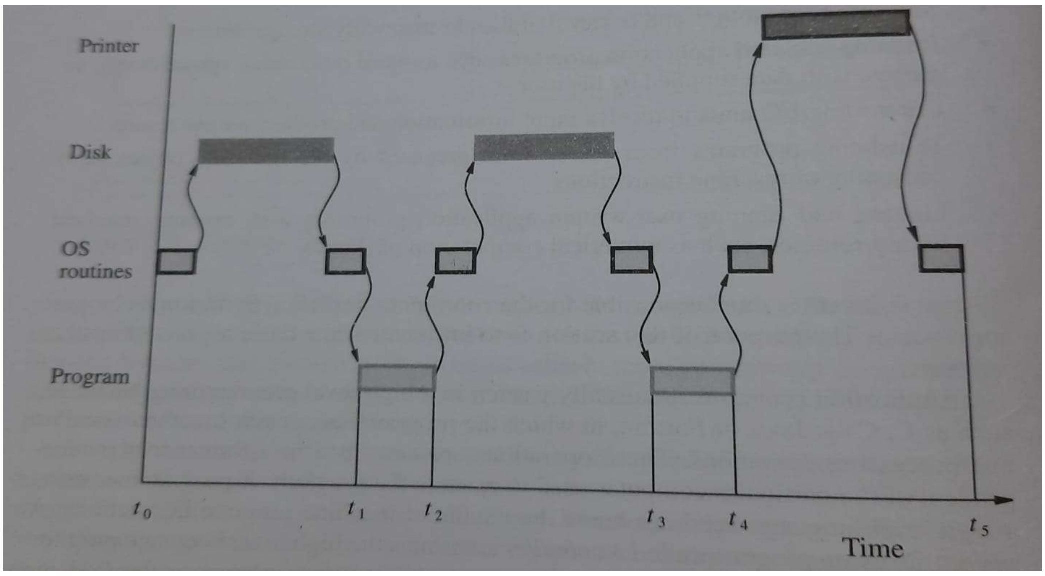


Figure curtesy: Computer Organization, Carl Hamacher

## Performance

- How quickly programs can be executed
- **Best performance**
  - Compiler, machine instruction set, hardware
- **Elapsed time**
  - Total time required to execute a program
  - Measure of entire computer system
  - Affected by the speed of the processor, disk and peripherals like printers
- **Processor time**
  - Performance of the processor
  - Periods during which the processor is active
  - Depends on the hardware involved in the execution of individual machine instruction
    - Processor and memory

## Basic Performance Equation

- *T*: The processor time required to execute a program that has been prepared in high level language
- *N*: **Actual** number of machine language instructions required to execute a program written in high level language
  - Loop-instructions get executed multiple times
  - Condition execution-some instructions may not get executed
- *S*: Average number of basic steps needed to execute one machine instruction
  - Each basic step is completed in one clock cycle
- *R*: Clock rate
- Program execution time

$$T = \frac{N \text{ x } S}{R}$$

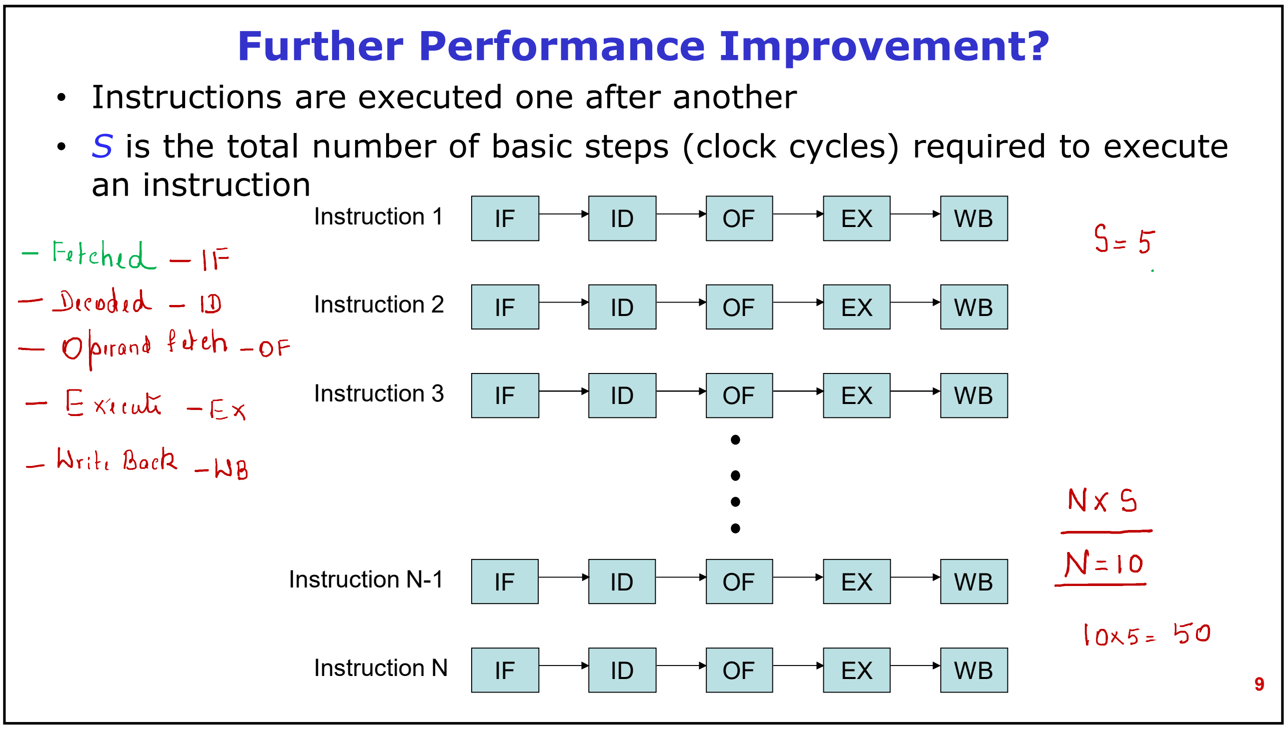
## High Performance

$$T = \frac{N \text{ x } S}{R}$$

- Reduce the value of *T*
  - Reduce *N* and *S*
  - Increase *R*
- Reduction in *N* – complexity of instruction increases
  - *S* increases
- Increasing *R*-using higher frequency clock
  - Time required to complete a basic execution step reduces
- *N*,*S*, and *R* are not independent parameters
  - Changing one may affect the other
- **Attempt to improve performance only by overall reduction of *T***

## Increasing Clock Rate *R*

- Reducing the amount of processing done in one basic step
  - Reduce the clock period *P*
  - Number of basic steps needed may increase *S*
- Improving the integrated-circuit (IC) technology makes logic circuits faster
  - Time needed to complete a basic step reduces
  - Clock period *P* reduces and clock rate *R* increases



## Further Performance Improvement?

- Instructions are executed one after another
- *S* is the total number of basic steps (clock cycles) required to execute an instruction

– Fetched – IF
– Decoded – ID
– Operand fetch – OF
– Execute – Ex
– Write Back – WB

Instruction 1: IF → ID → OF → EX → WB

Instruction 2: IF → ID → OF → EX → WB

Instruction 3: IF → ID → OF → EX → WB

•
•
•
•

Instruction N-1: IF → ID → OF → EX → WB

Instruction N: IF → ID → OF → EX → WB

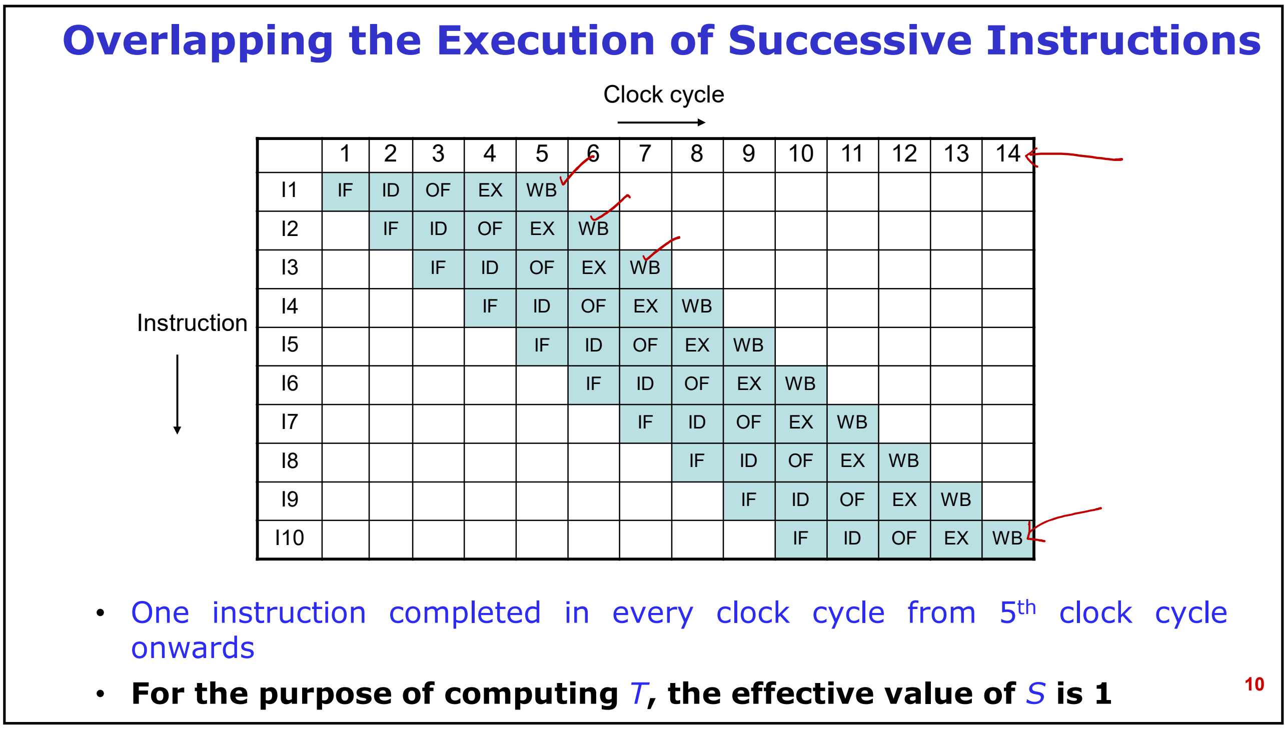
$S = 5$

$N \times S$

$N = 10$

$10 \times 5 = 50$

9



## Overlapping the Execution of Successive Instructions

Clock cycle →

Instruction ↓

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF | ID | OF | EX | WB | | | | | | | | | |
| I2 | | IF | ID | OF | EX | WB | | | | | | | | |
| I3 | | | IF | ID | OF | EX | WB | | | | | | | |
| I4 | | | | IF | ID | OF | EX | WB | | | | | | |
| I5 | | | | | IF | ID | OF | EX | WB | | | | | |
| I6 | | | | | | IF | ID | OF | EX | WB | | | | |
| I7 | | | | | | | IF | ID | OF | EX | WB | | | |
| I8 | | | | | | | | IF | ID | OF | EX | WB | | |
| I9 | | | | | | | | | IF | ID | OF | EX | WB | |
| I10 | | | | | | | | | | IF | ID | OF | EX | WB |

- One instruction completed in every clock cycle from 5th clock cycle onwards
- **For the purpose of computing *T*, the effective value of *S* is 1**

10

# Instruction Pipelining

Clock cycle →

Instruction ↓

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I1 | IF | ID | OF | EX | WB | | | | | | | | | |
| I2 | | IF | ID | OF | EX | WB | | | | | | | | |
| I3 | | | IF | ID | OF | EX | WB | | | | | | | |
| I4 | | | | IF | ID | OF | EX | WB | | | | | | |
| I5 | | | | | IF | ID | OF | EX | WB | | | | | |
| I6 | | | | | | IF | ID | OF | EX | WB | | | | |
| I7 | | | | | | | IF | ID | OF | EX | WB | | | |
| I8 | | | | | | | | IF | ID | OF | EX | WB | | |
| I9 | | | | | | | | | IF | ID | OF | EX | WB | |
| I10 | | | | | | | | | | IF | ID | OF | EX | WB |

Scalar pipeline Processor

- One instruction completed in every clock cycle from 5th clock cycle onwards
- **For the purpose of computing *T*, the effective value of *S* is 1**

# Superscalar Execution

- Multiple instruction pipelines
  - Multiple functional units
- Execution of several instructions per clock cycle
- Effective value of *S* can be reduced to **less than one**
- **Many of today's high performance processors are designed in this manner**

## Instruction Set: Simple Vs Complex Instructions

- **Simple instructions**
  - Require a small number of basic steps
  - A large number of instructions may be needed to perform a given programming task
  - Large value of *N* and small value of *S*

ADDM LocB, R0

ADD R1, R0

- **Complex instructions**
  - Require a large number of steps
  - Individual instructions perform complex operations
  - Fewer instructions will be needed
  - Lower value of *N* and a larger value of *S*
- **Design of an instruction set of a computer?**

## Instruction Set: Simple Vs Complex Instructions

- Support for pipelining
  - Easier in processors with simple instruction set
- **Complex Instruction Set Computer (CISC)**
- **Reduced Instruction Set Computer (RISC)**

## Compiler

- Translates a high-level language program into a sequence of machine instructions

- To reduce *N*, suitable machine instruction set a compiler that makes use of it

- An **optimizing compiler** takes advantage of various features of the target processor to reduce *N*x*S*

- A compiler is not a separate entity from the processor with which it is used
  - Should be closely linked to the processor architecture
  - Compiler and processor are designed together

## Performance of a Computer

- **Affected by**
  - Hardware
  - Machine language instructions-instruction set
  - Compiler

- For best performance, it is necessary to design the compiler, the machine instruction set, and he hardware in a coordinated way

- Performance of a computer is the execution time, *T*, for the program of interest

## Performance Measurement

- Measuring computer performance using **benchmark programs**
  - Benchmark: Standardised program
- **Performance measure**: Time it takes to execute a given benchmark
- Use a selection of real application programs to evaluate performance
  - **System Performance Evaluation Corporation (SPEC)** selects and publishes representative application programs for different application domains
    - Range from game playing, compiler and database applications to numerically intensive programs in astrophysics and quantum chemistry
- The same program is also run one computer selected as a reference

## Performance Measurement

- SPEC rating for a benchmark is given by

$$SPEC\ rating = \frac{Running\ time\ on\ the\ reference\ computer}{Running\ time\ under\ computer\ under\ test}$$

- The overall SPEC rating for the computer is given by

$$SPEC\ rating = \left(\prod_{i=1}^{n} SPEC_i\right)^{\frac{1}{n}}$$

## To Summarize

- Bus structure
- System software
- Performance of a computer
  - Basic performance equation
  - Factors affecting the performance
- Improving the performance
  - Pipelining
  - Superscalar processing
- Performance measurement

## Reference

- **Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill, 2002**

# Thank You