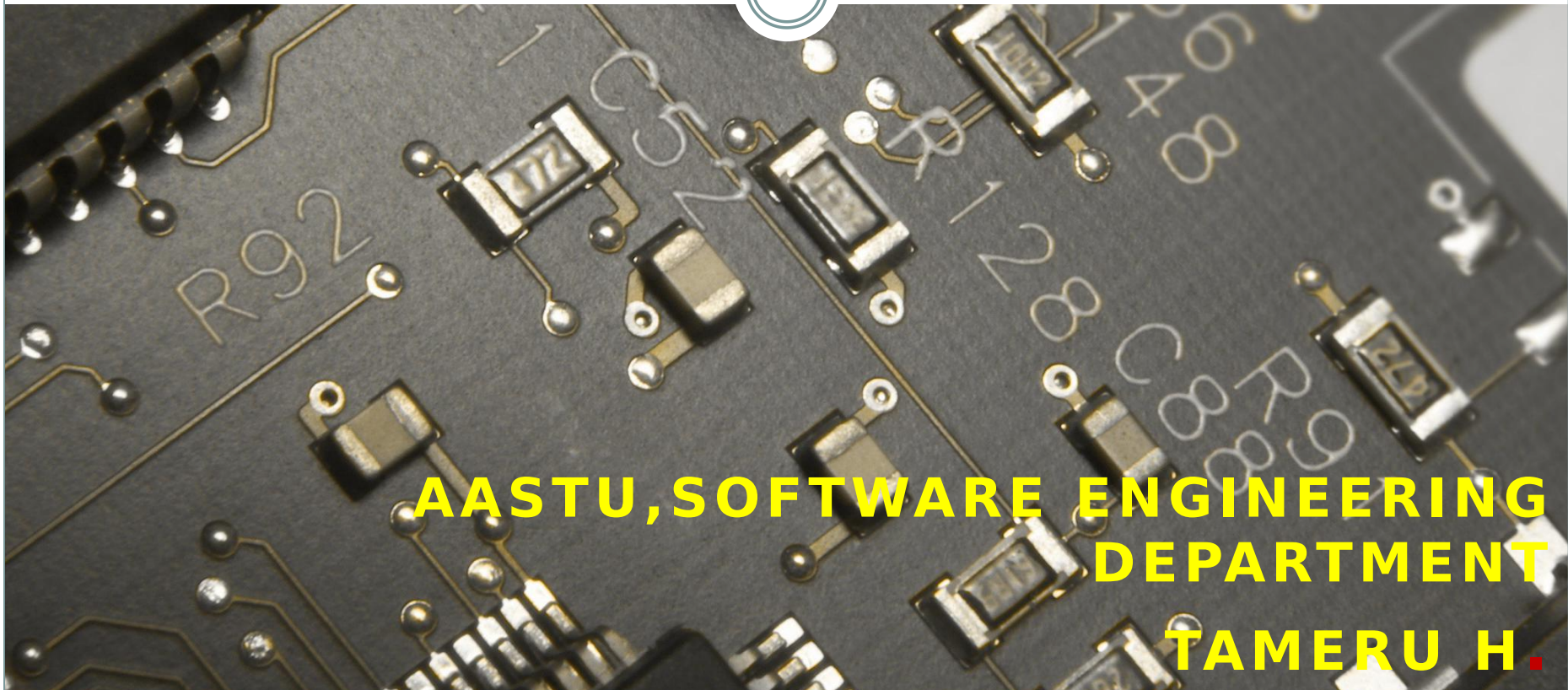


Computer Architecture and Assembly Language



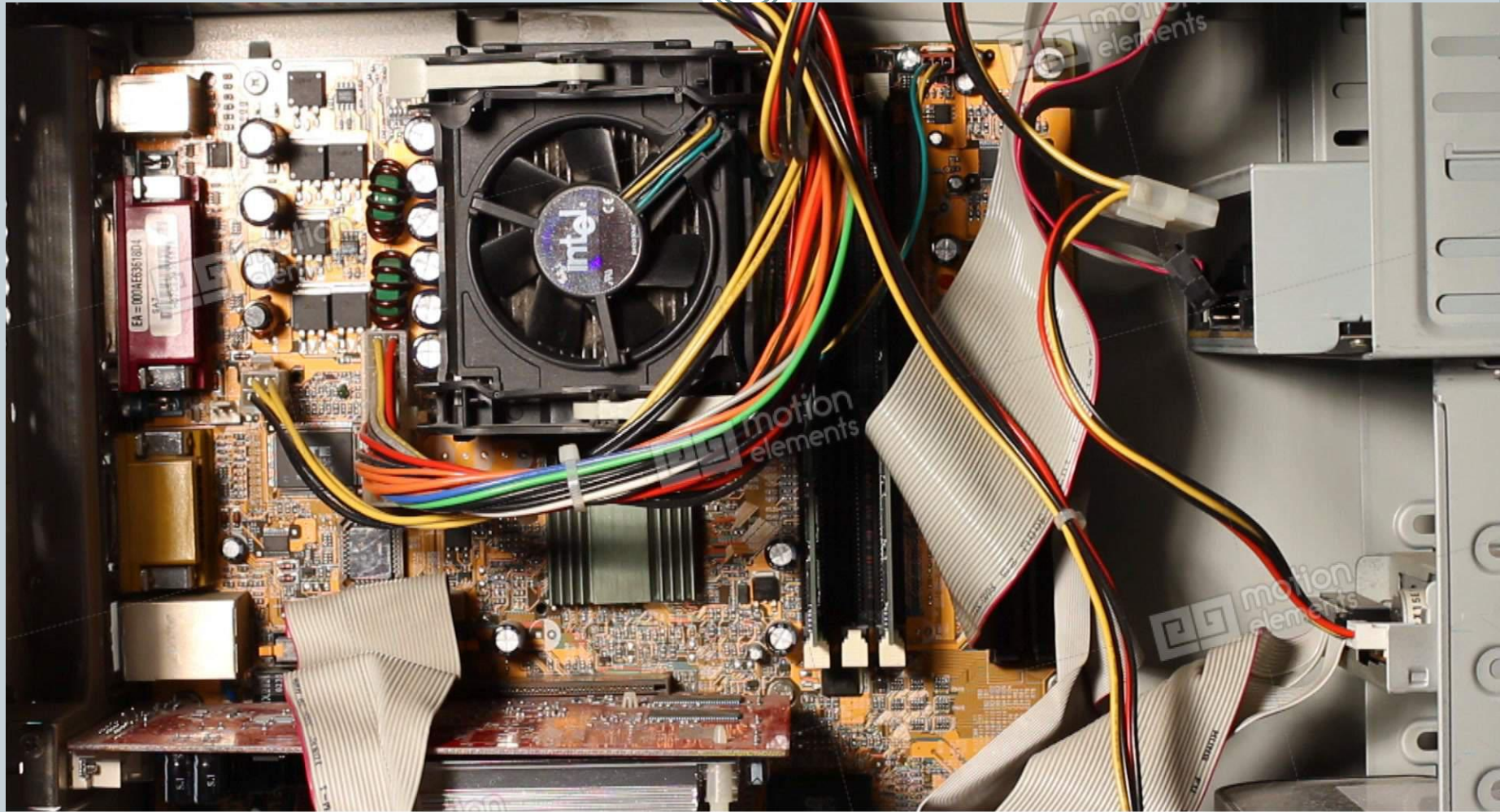
**AASTU, SOFTWARE ENGINEERING
DEPARTMENT
TAMERU H.**

What do you expect?

The background of the slide is a dark blue, abstract digital graphic. It features concentric circles, glowing blue lines, and patterns resembling binary code (0s and 1s) and circuitry. The overall effect is futuristic and technological.

COMPUTER ARCHITECTURE

Why study this course?



Cont'd...



- Architecture is a very **satisfying** profession
- A course in Architecture is a lot of fun.
- Architecture gives you the liberty to use the right side of your brain, the **creative part**.

What's Wrong With Assembly Language



Assembly language has a **pretty bad reputation**. Here are the reasons:

Assembly is hard to learn.

Assembly is hard to read and understand.

Assembly is hard to debug.

Assembly is hard to maintain.

Assembly is hard to write.

Why Assembly language?



An old joke goes something like this:

“There are three reasons for using assembly language: **speed**, **speed**, and **more speed**.”

Assembly language has several benefits other than speed:

Cont'd...



Speed. Assembly language programs are generally the fastest programs around.

Space. Assembly language programs are often the smallest.

Capability. You can do things in assembly which are difficult or impossible in HLLs.

Knowledge. Your knowledge of assembly language will help you write better programs, even when using HLLs.

What can you do Assembly?



- Anything you want in the software world
- boot loader, or device drivers that live in ROM
- Operating System

Course Outcome



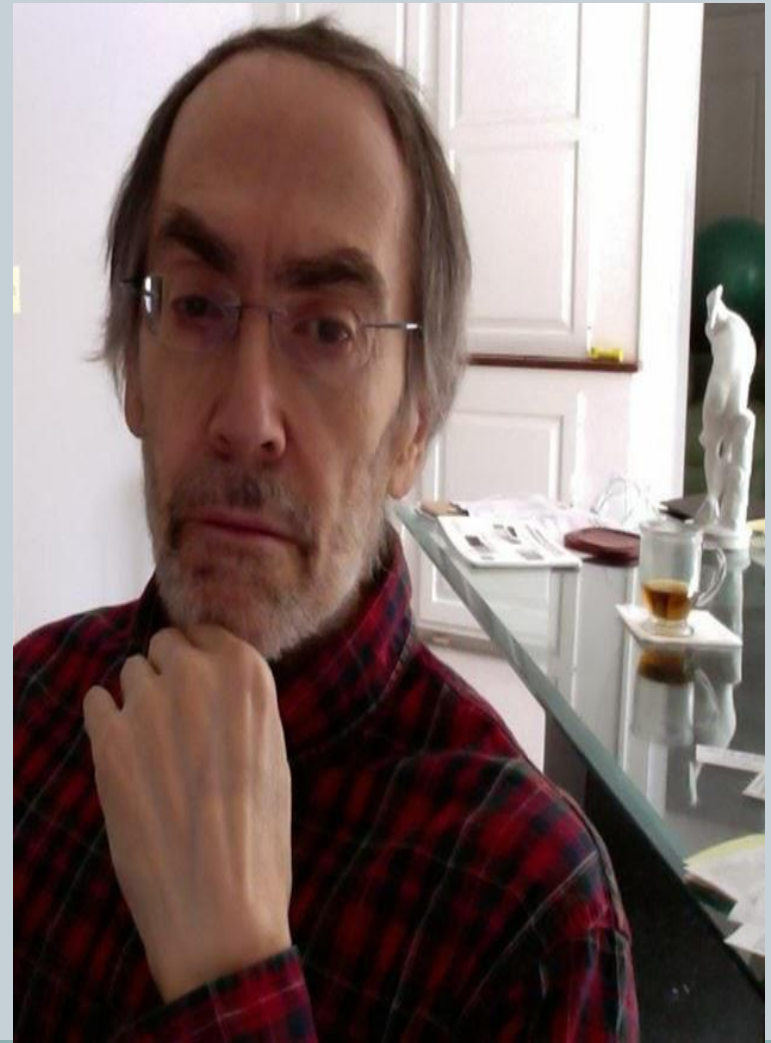
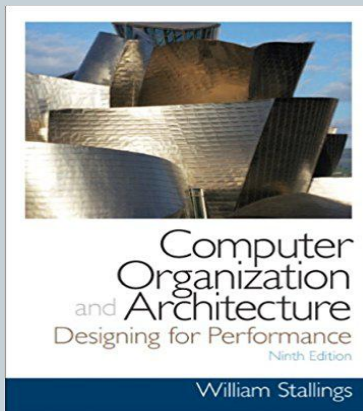
At the end of this course you'll be able to:

- Understand the basics of computer hardware and how software interacts with computer hardware
- Analyze and evaluate computer *performance*
- Understand how computers *represent and manipulate data*
- Understand basics of *Instruction Set Architecture* (ISA) – MIPS
- Understand Assembly Programming Language

Text Book



- American **Computer Science, Cryptography and Security** textbook author.
- Written textbooks on computer science topics such as **Operating System, computer networks, computer organization, and cryptography.**



Who is the Computer Architect ?



The computer architect need to have the following principles:

- **Look Backward** (to the past): understand trade-offs, analyse the past
- **Look forward** (to the future): be the dreamer, and create new design.
- **Look Up** (towards problems in the computing stack): understand problems, Develop architecture to solve them.
- **Look down** (towards device/circuit technology: platform for the future

Chapter 1: Introduction



Organization and Structure

- **Computer architecture** refers to those attributes of a system visible to a programmer.
- **Computer organization** refers to the **operational units** and their **interconnections** that realize the architectural specifications.

Cont'd..



Example:

Architectural design issue: whether a computer will have a multiply instruction.

An organizational issue: whether that instruction will be implemented **by a special multiply unit** or by a mechanism that makes repeated use of the add unit of the system.

Cont'd...



**How to make the organizational
decision on this issue?**



Cont'd...



The organizational decision may be based on:

- The anticipated *frequency* of use of the multiply instruction,
- The relative *speed* of the two approaches,
- And the *cost and physical size* of a special multiply unit.

STRUCTURE AND FUNCTION



- **Structure:** The way in which the components are interrelated.
- **Function:** The operation of each individual component as part of the structure.

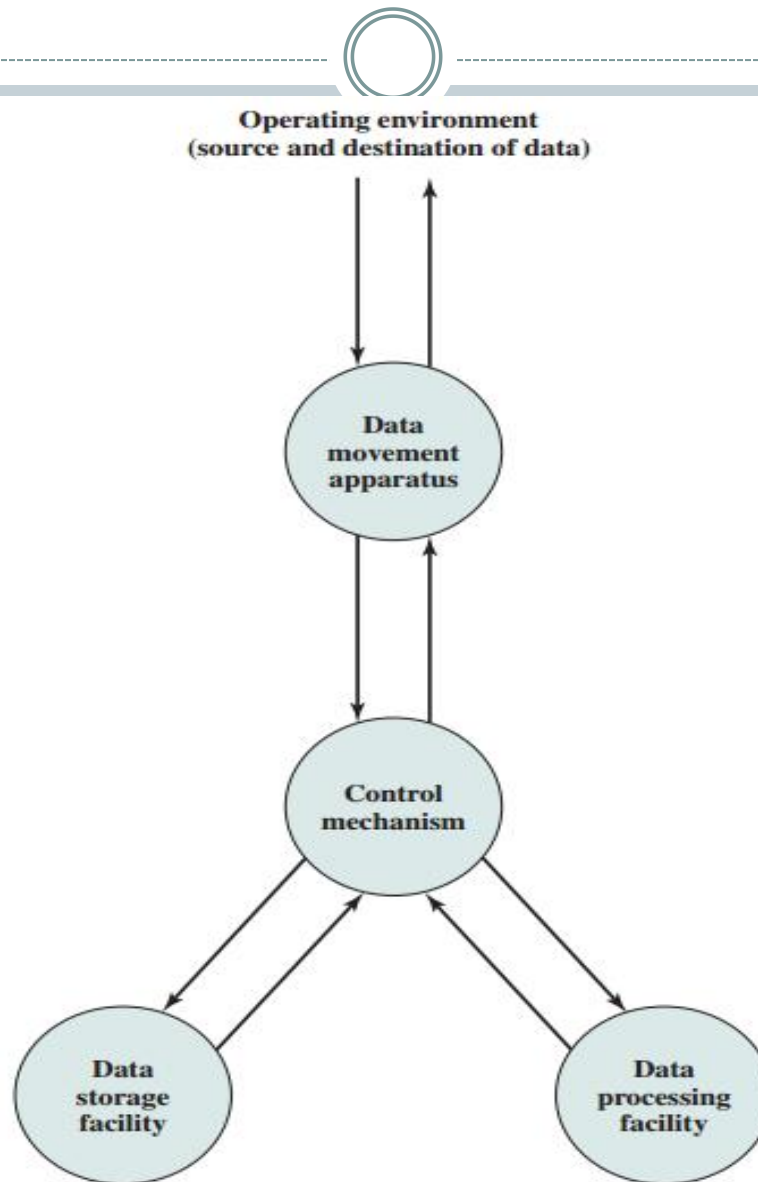
Function



In general terms, there are only four basic functions that a computer can perform:

- Data processing
- Data storage
- Data movement
- Control

Cont'd...



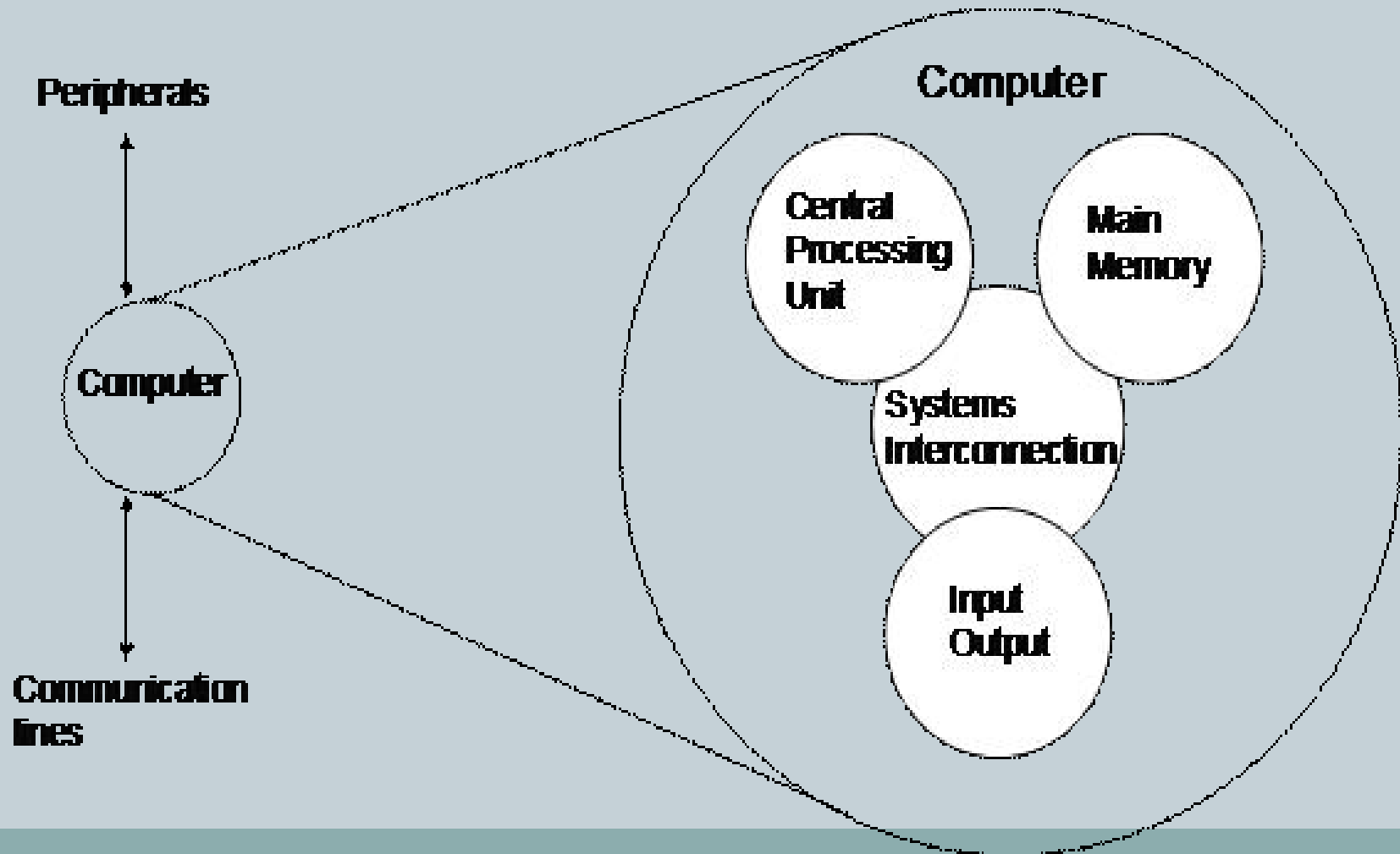
Structure



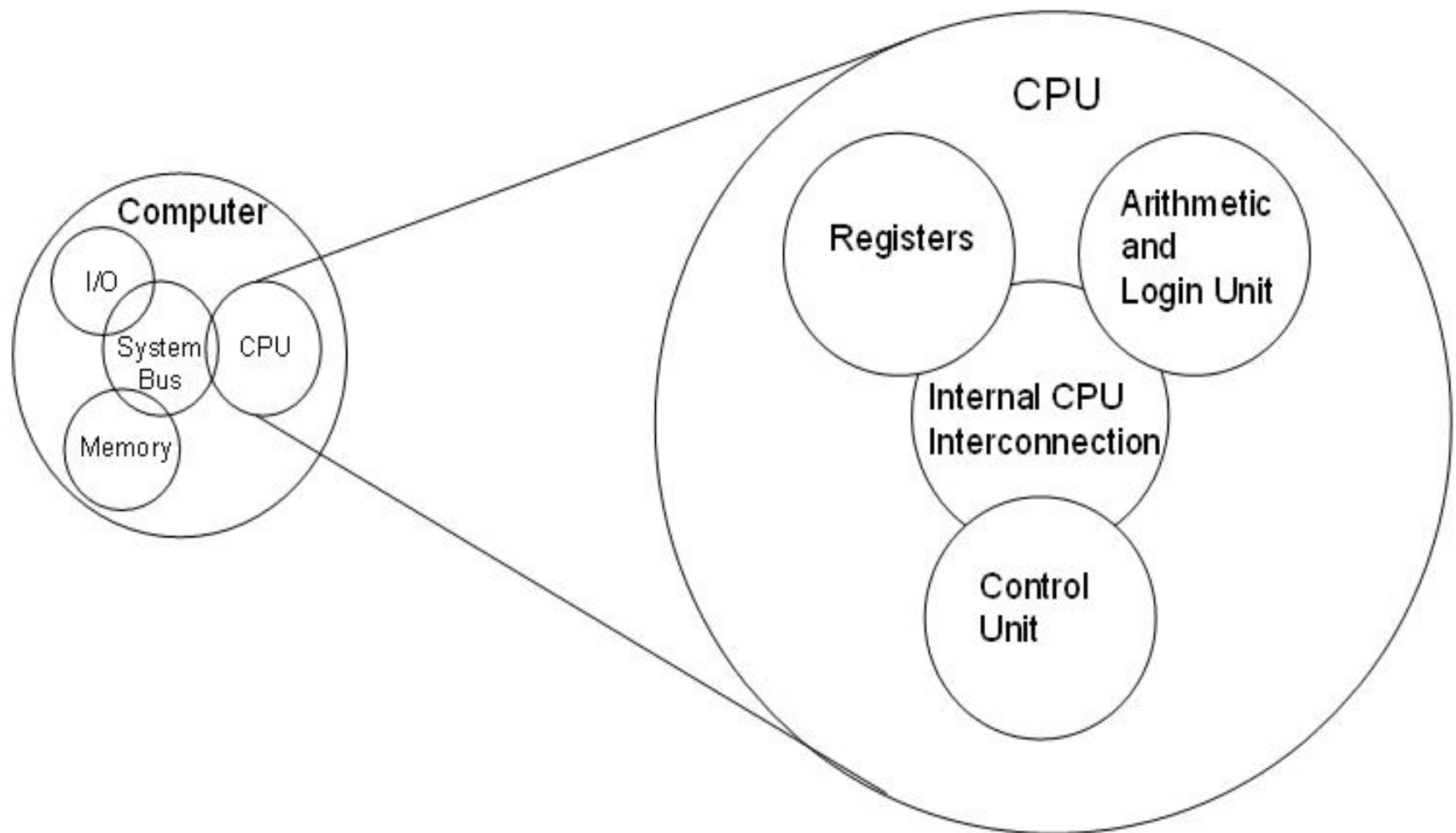
There are four main structural components:

- **Central processing unit (CPU):** Controls the operation of the computer and performs its data processing functions; often simply referred to as **processor**.
- **Main memory:** Stores data.
- **I/O:** Moves data between the computer and its external environment.
- **System interconnection:** Some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a **system bus**, consisting of a number of conducting wires to which all the other components attach.

The Computer: Top-Level Structure



Cont'd...



Question?



Next Lecture



Chapter Two

Computer Evolution and Performance