Computer Programming
CS F111

BITS Pilani

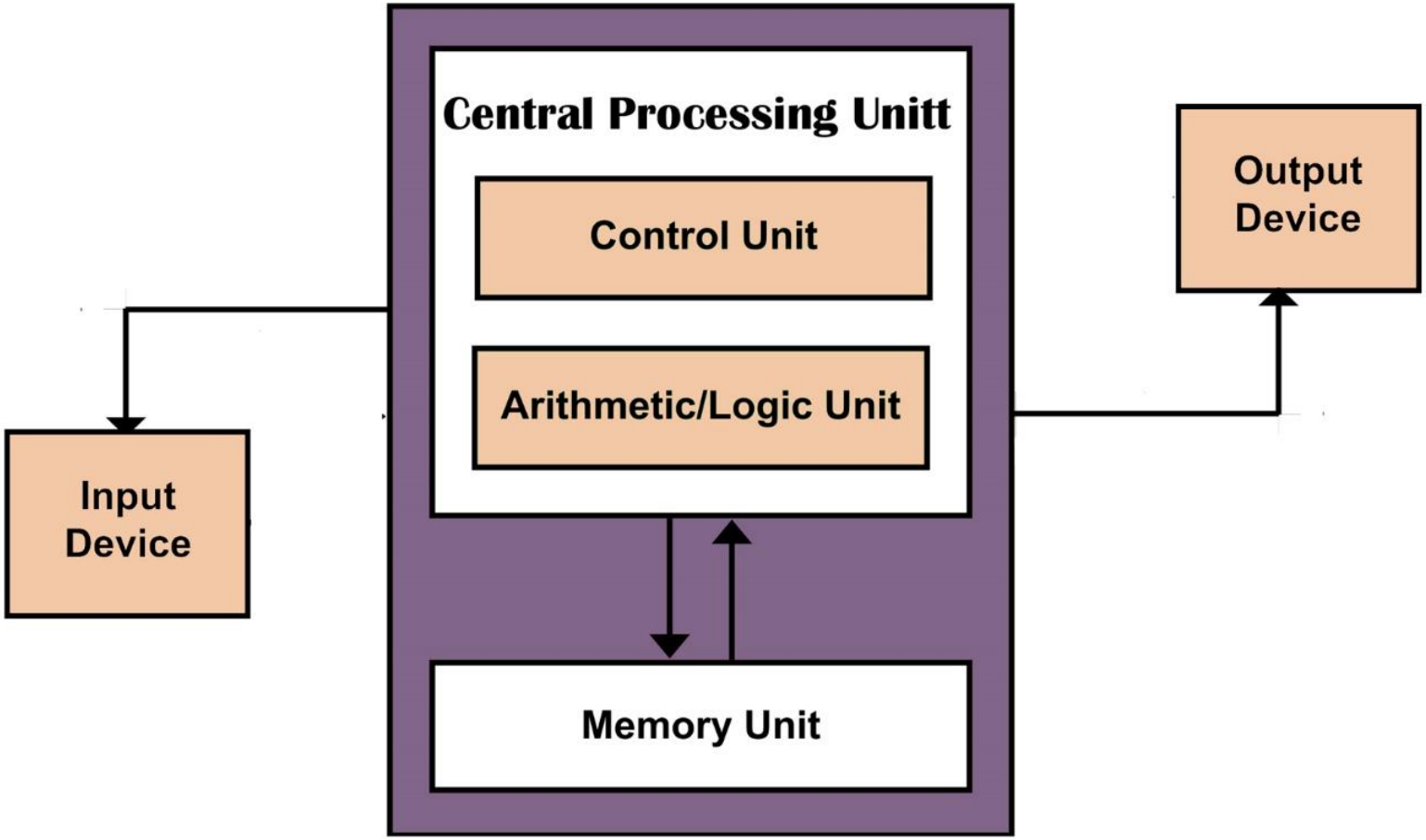
Pilani|Dubai|Goa|Hyderabad

Adapted from the slides of Dr. Sundaresan Raman, Assistant Professor, Dept. of Computer Science, BITS Pilani

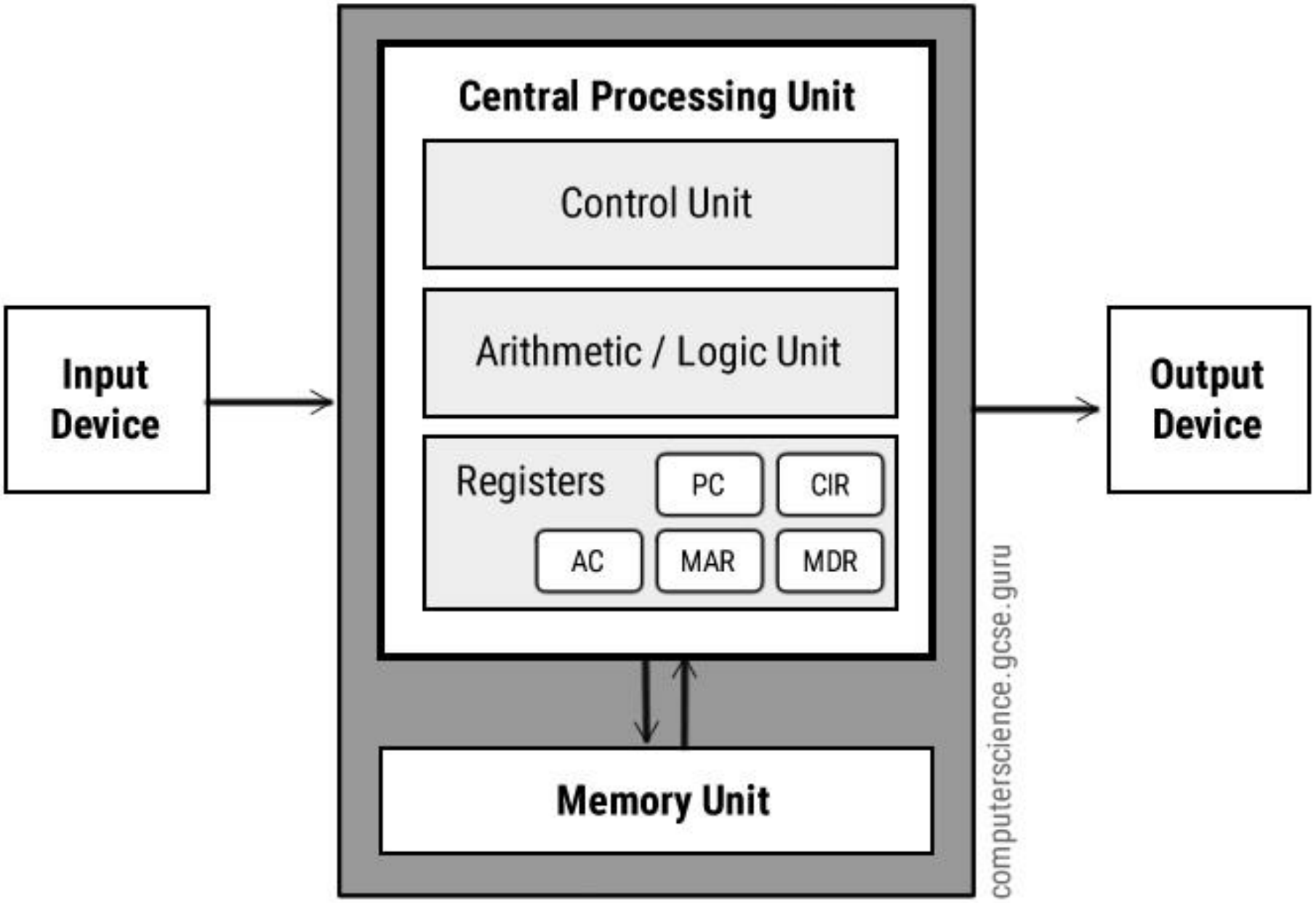
Course Overview

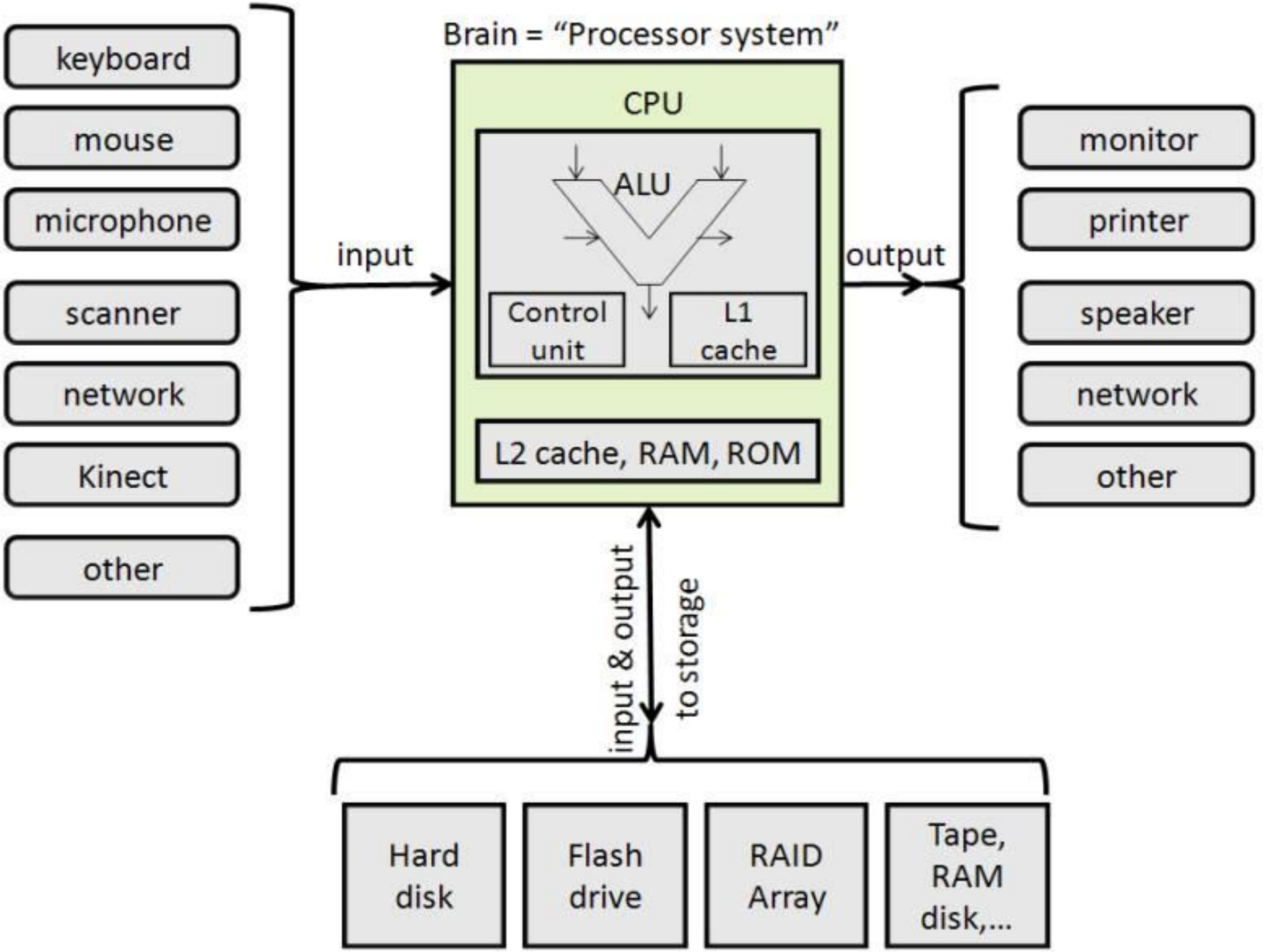
- Basic representation of data and processing it in a computer
- Problem solving using a programming language
- Learning programming methodology (using C)

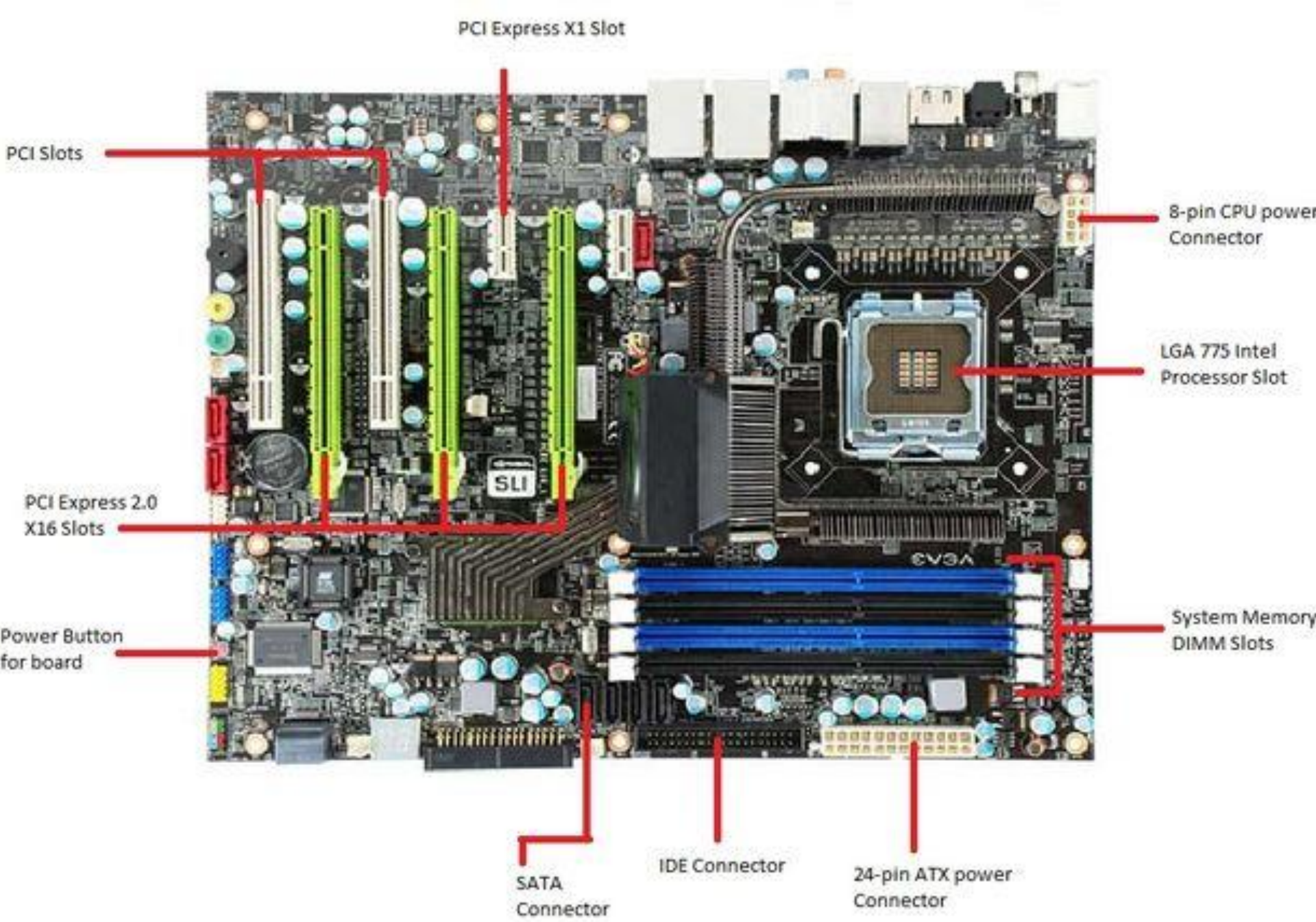
What is a Computer?



Von Neumann Architecture


Central Processing Unit
Control Unit
Arithmetic / Logic Unit
Registers
PC
CIR
AC
MAR
MDR
Input Device
Output Device
Memory Unit
computerscience.gcse.guru


keyboard
mouse
microphone
scanner
network
Kinect
other
input
Brain = "Processor system"
CPU
ALU
Control unit
L1 cache
L2 cache, RAM, ROM
output
monitor
printer
speaker
network
other
input & output
to storage
Hard disk
Flash drive
RAID Array
Tape, RAM disk,...


PCI Express X1 Slot
PCI Slots
8-pin CPU power Connector
LGA 775 Intel Processor Slot
PCI Express 2.0 X16 Slots
System Memory DIMM Slots
Power Button for board
SATA Connector
IDE Connector
24-pin ATX power Connector
SLI
EVGA

Levels of Transformations

Problems

Algorithms

Language

Machine (ISA) Architecture

Microarchitecture

Circuits

Devices

Programming Languages

- Computer language
  - Machine language with two symbols 0s and 1s
- Natural language
  - Ambiguous
  - Redundancy
  - Complex
  - Hence unsuitable for computers
- **Programming Language**
  - **Abstraction level between human and computer**

An Example

Machine Code in Hex	Assembly Code	High-Level Code
27BB0001	ldah gp, main	main()
23BD8050	lda gp, main	{
23DEFFF0	lda sp, -16(sp)	int a, b, c;
A61D8018	ldq r16, 8(gp)	a = 3;
A77D8010	ldq r27, printf	b = 4;
47E0F411	mov 7, r17	c = a + b;
B75E0000	stq r26, (sp)	printf("\n%d\n", c);
6B5B4000	jsr r26, printf	}
27BA0001	ldah gp, main	
A75E0000	ldq r26, (sp)	
23BD8050	lda gp, main	
47FF0400	clr r0	
23DE0010	lda sp, 16(sp)	
6BFA8001	ret r26	

High Level vs Low Level

- High level languages (HLLs)
  - Machine independent
  - Can be compiled to run on any machine
  - Much easier to write than Low level languages
  - Higher productivity and abstractions
    - Data structures like queues, stacks etc.
    - Control structures like loops, switch etc.

High Level vs Low Level

- Low level languages (LLLs)
  - Machine specific
    - Programmer exposed to the internal machine organization
  - They access very low level details
    - Number and type of registers
    - Specific instructions
    - Hardware features
  - Lower productivity but higher performing code
  - Generally called as assembly language

The C Language

- General-purpose programming language
- Closely associated with the UNIX system
- Language is not tied with any operating system or machine
- Knowing C programming helps in other disciplines too
- Product in itself – can create a career based on programming unlike other courses

Algorithms

- Step-by-step procedure that has
  - Finiteness (guarantees to terminate)
  - Definitiveness (each step precisely stated)
  - Effective computability (each step could be computed)

Programs

- Transform the algorithm into a computer program using a programming language
- Programming languages are mechanical and not natural; hence are unambiguous
- Many programming languages for variety of purposes

Levels of transformations

Problems

Algorithms

Language

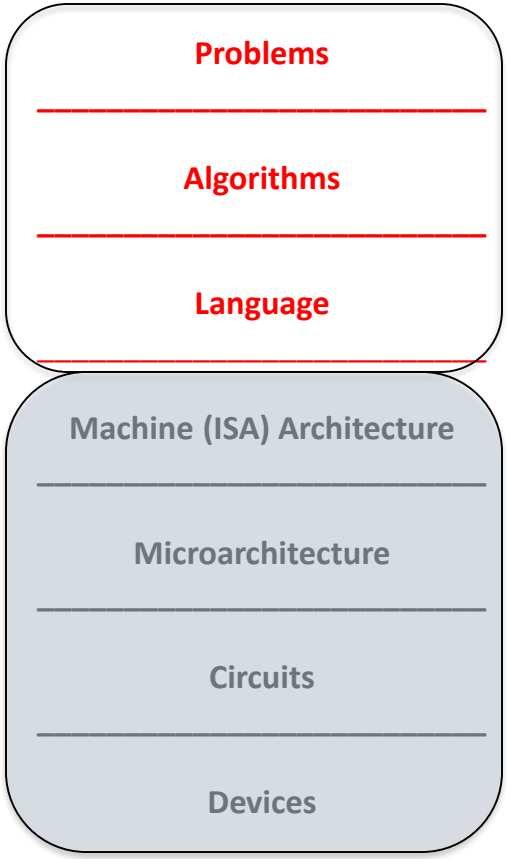
Machine (ISA) Architecture

Microarchitecture

Circuits

Devices

Levels of transformations



What is Operating System

What is Operating System

- An Operating System can be defined as the software that controls the hardware resources of the computer and provides an environment under which programs can run.
- It provides an interface between user and the hardware.
- Single-user single-tasking OS, single-user multi-tasking OS, multi-user, multi-tasking OS, real time OS

Hardware Resources

- CPU
- Memory
- I/O Devices
- Network

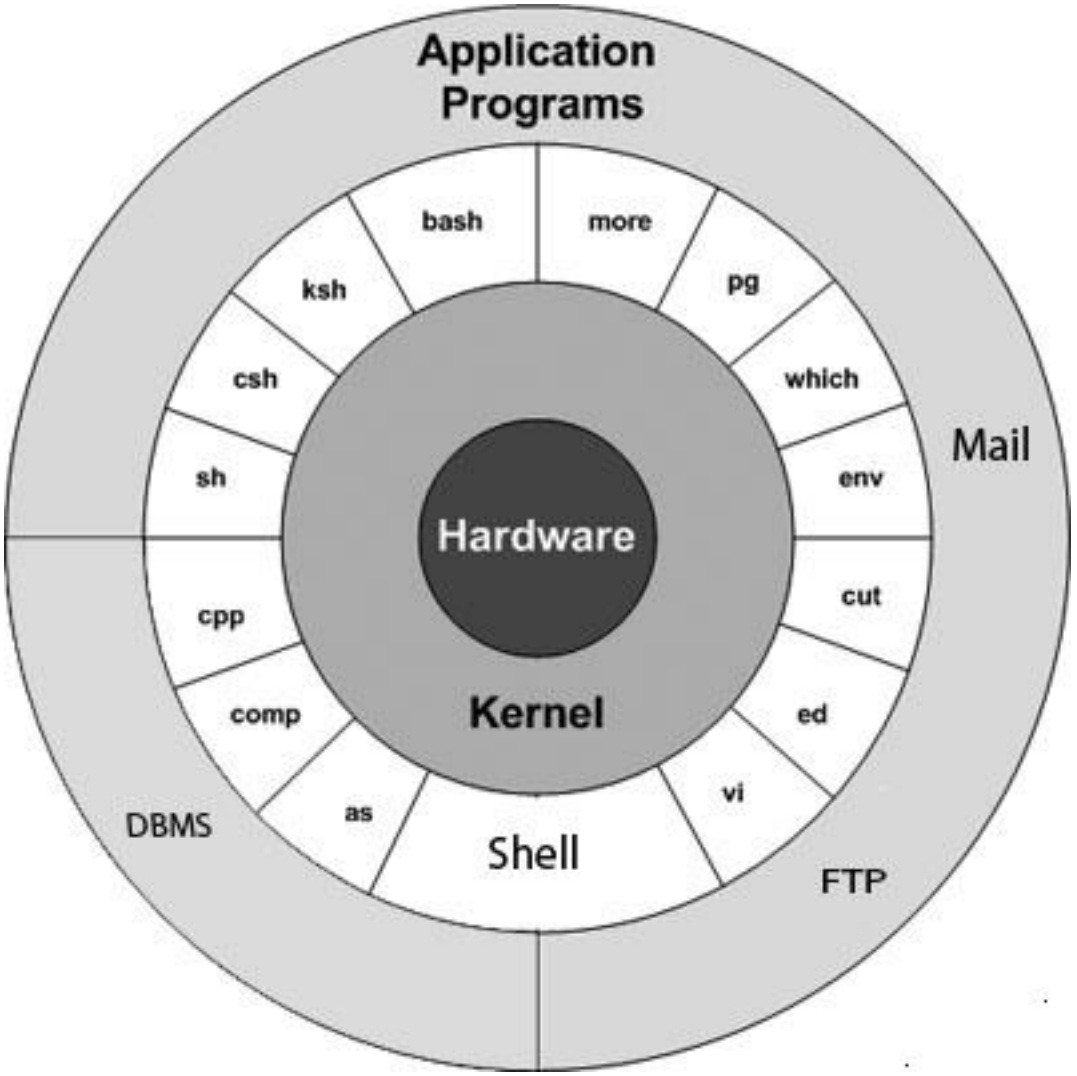
The UNIX Operating System

- Written in high-level language
- Simple user interface that can provide the services user wants
- Allows complex programs to be built from simpler programs
- Hierarchical file system allowing easy maintenance
- Multi-user multi-process system; each user can execute several processes
- Hides machine architecture from the user

The UNIX System

- Although written in C language, the UNIX systems support other languages including Fortran, Basic, Pascal, Ada, Cobol, Lisp and Prolog.

Layered architecture of the UNIX OS



Kernel

- Kernel is a program that constitutes the central core of the Operating System.
- Kernel provides basic services to all other parts of the Operating System including
  - Process Management
    - Creation, termination, suspension, communication
  - Memory Management
    - Allocation for executing process, swapping, paging system
  - File Management
    - Allocate secondary memory for efficient storage and retrieval, access rights, reclaims unused storage
  - I/O Management
    - Provide controlled access to I/O and network devices
  - Scheduling processes fairly

Kernel

- It can not directly interact with the user.
- But interacts with shell (or user program) through a set of interfaces and with hardware devices such as processor, memory, disk drives etc.

Shell

- Shell is a program that provides text-only interface to the user.
- It’s primary function is to read commands from the console and execute them.
- The term Shell comes from the fact that it is the outer most part of the OS.
- Shell commands