

Introduction

EE2222 Computer Interfacing and Microprocessors

Dr. Nimal Skandhakumar

University of Sri Jayewardenepura

Course Overview

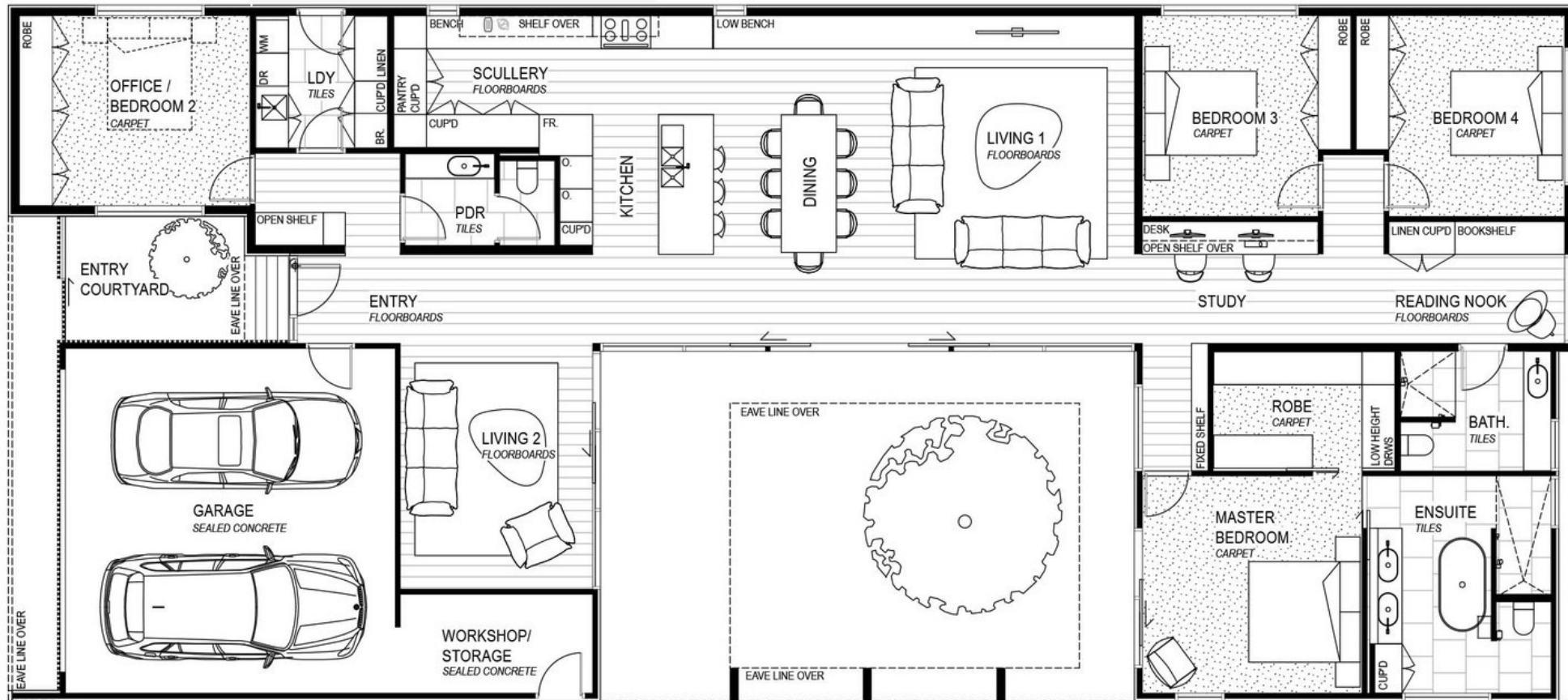
- Aims:
 - Basics of Microprocessor-based systems
 - Programmer's view of Computer Architecture
 - Interaction between hardware and software, i.e. 'Interfacing'.
- What not covered:
 - Designing computer hardware
 - High level language programming

Recall: Pre-Requisite

- Computers and Computing
 - C- Language Programming
 - The von Neumann model: memory/I-O/processing
 - The instruction set and execution cycle;
 - Registers and address spaces
 - An instruction set: operations and addressing modes
 - An expanded model of a computer: mass storage and I/O
 - The layered model of a computer: from gate- to user-level

Computer architecture

What is computer architecture?



Architecture and organization

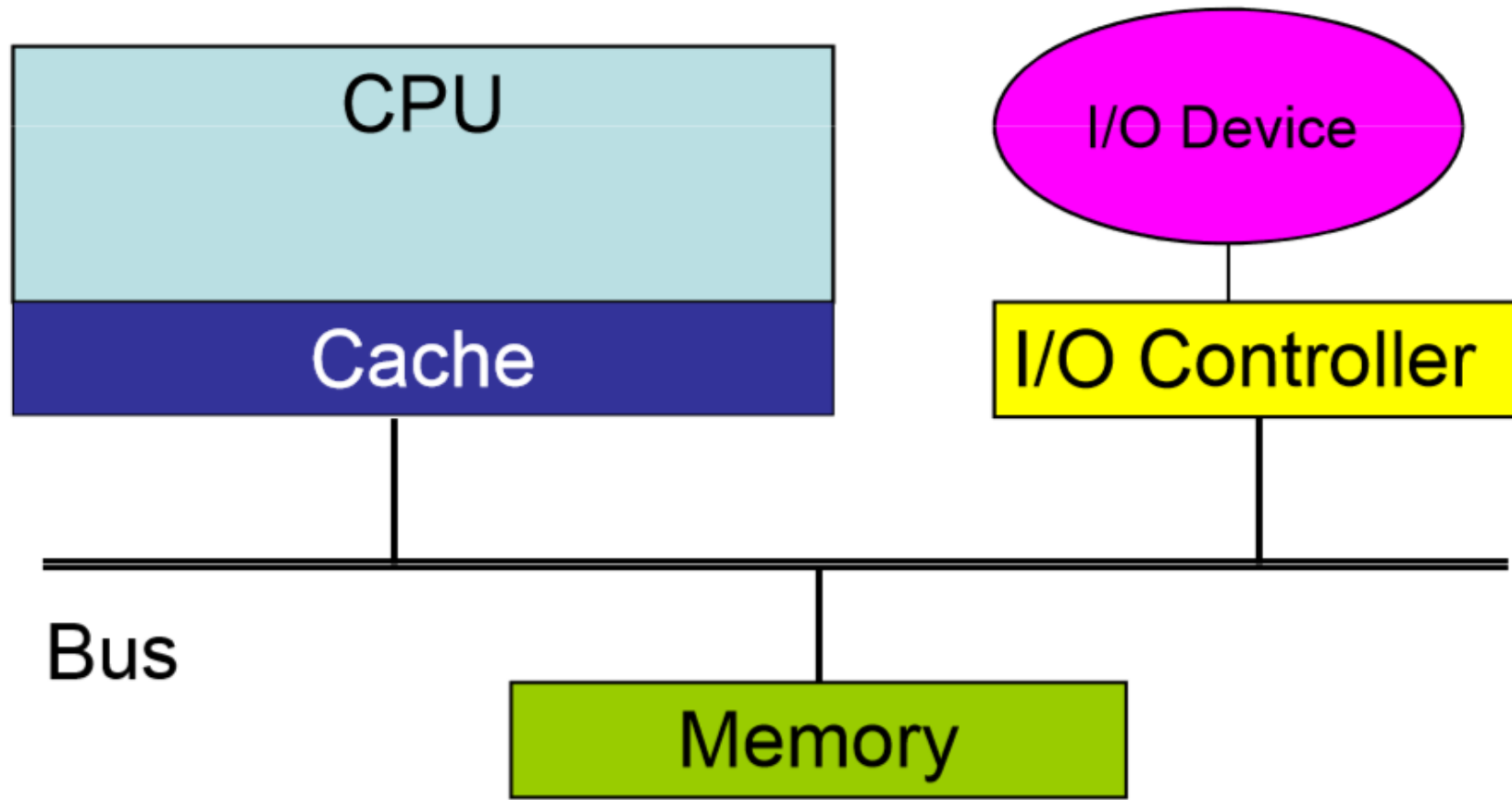
- Architecture is the design of the system visible to the assembly level programmer.
 - What instructions
 - How many registers
 - Memory addressing scheme
- Organization is how the architecture is implemented.
 - How much cache memory
 - Microcode or direct hardware
 - Implementation technology

Same architecture, different organization

- Almost every program that can run on a Core i3 can run on a Core i5.
- All computers in the Intel Core series have the same architecture.
- Each version of the Intel Core has a different organization or implementation, speed, and price.

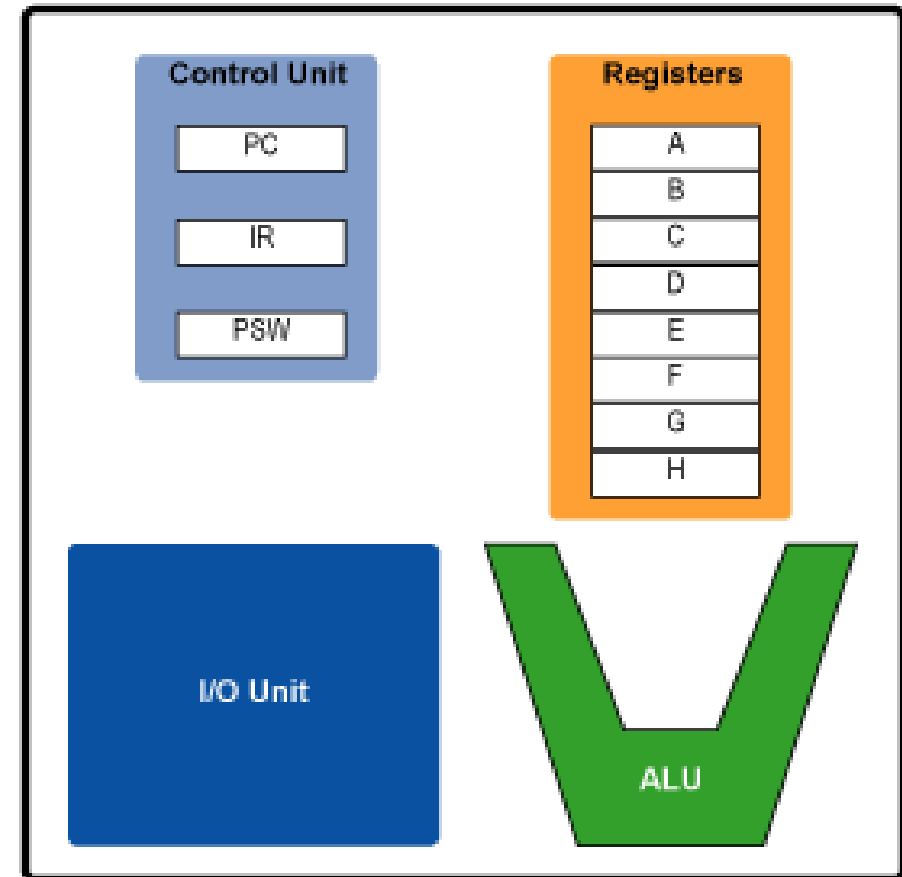


Basic computer components



Central Processing Unit

- Contains the control logic that initiates most activities in the computer.
- The Arithmetic Logic Units perform the math and logic calculations.
- Registers contain temporary data values.
- Program Counter contains the address of the next instruction to execute.

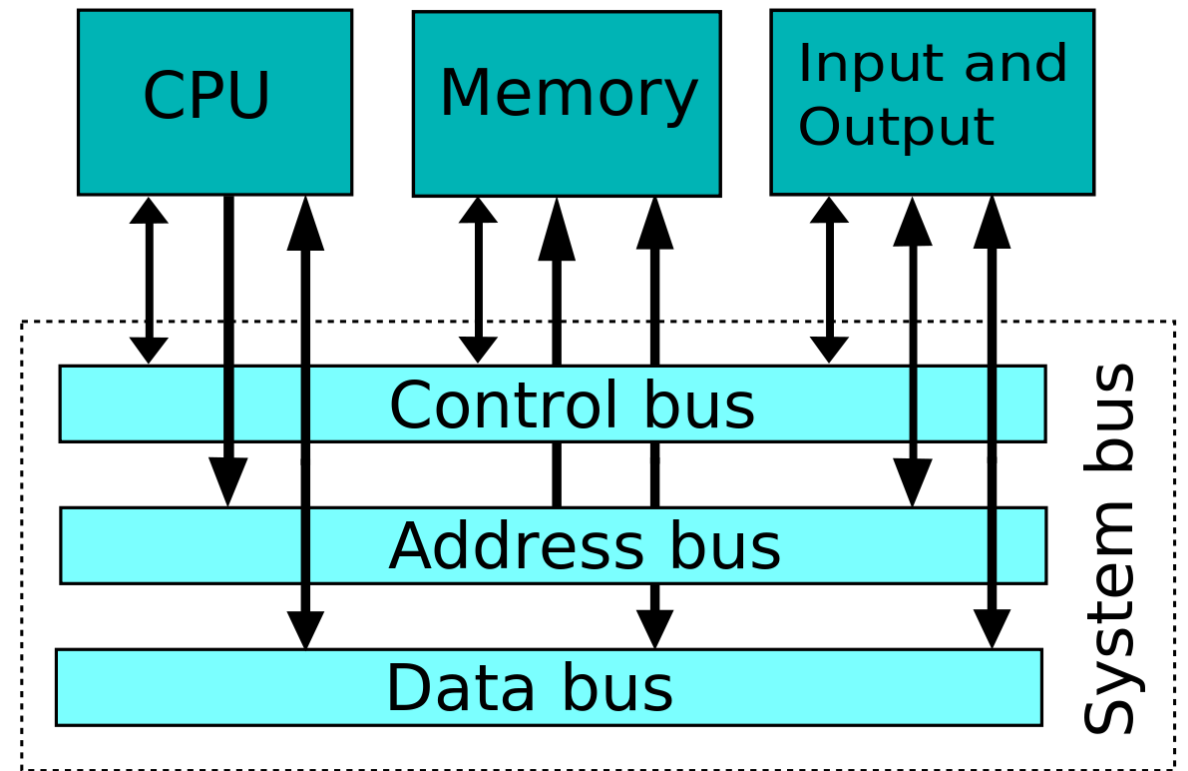


Registers

- The CPU has registers to temporarily hold data being acted upon.
- Different architectures have different number of registers.
- Some registers are available for the user programs to use directly.
- Some registers are used indirectly (such as the program counter).
- Some registers are used only by the operating system (i.e. program status reg)

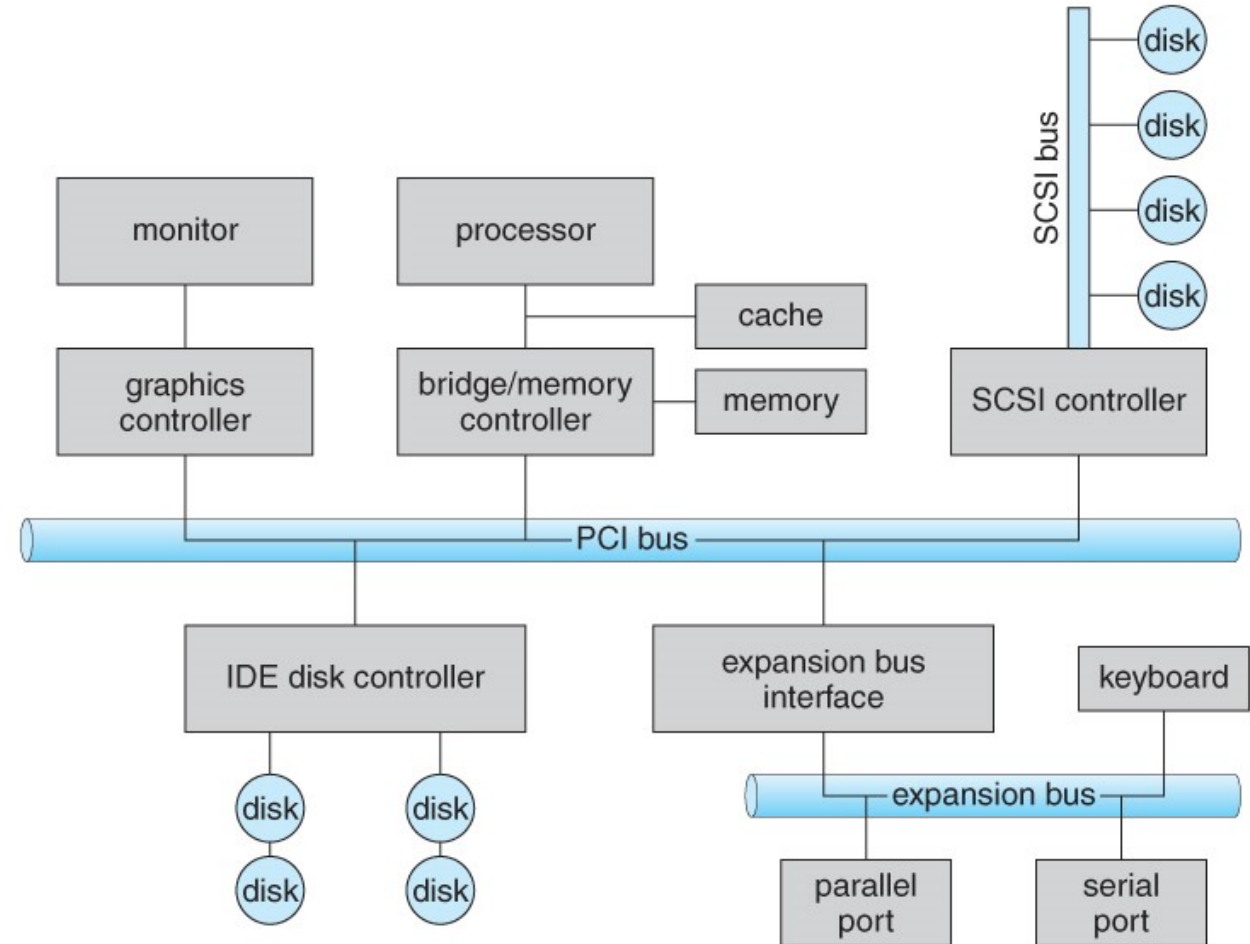
Bus

- The bus is a set of parallel wires that connect the CPU, memory and I/O controllers.
- It has logic (the chipset) to determine who can use the bus at any given instant.
- The width of the bus determines the maximum memory configuration



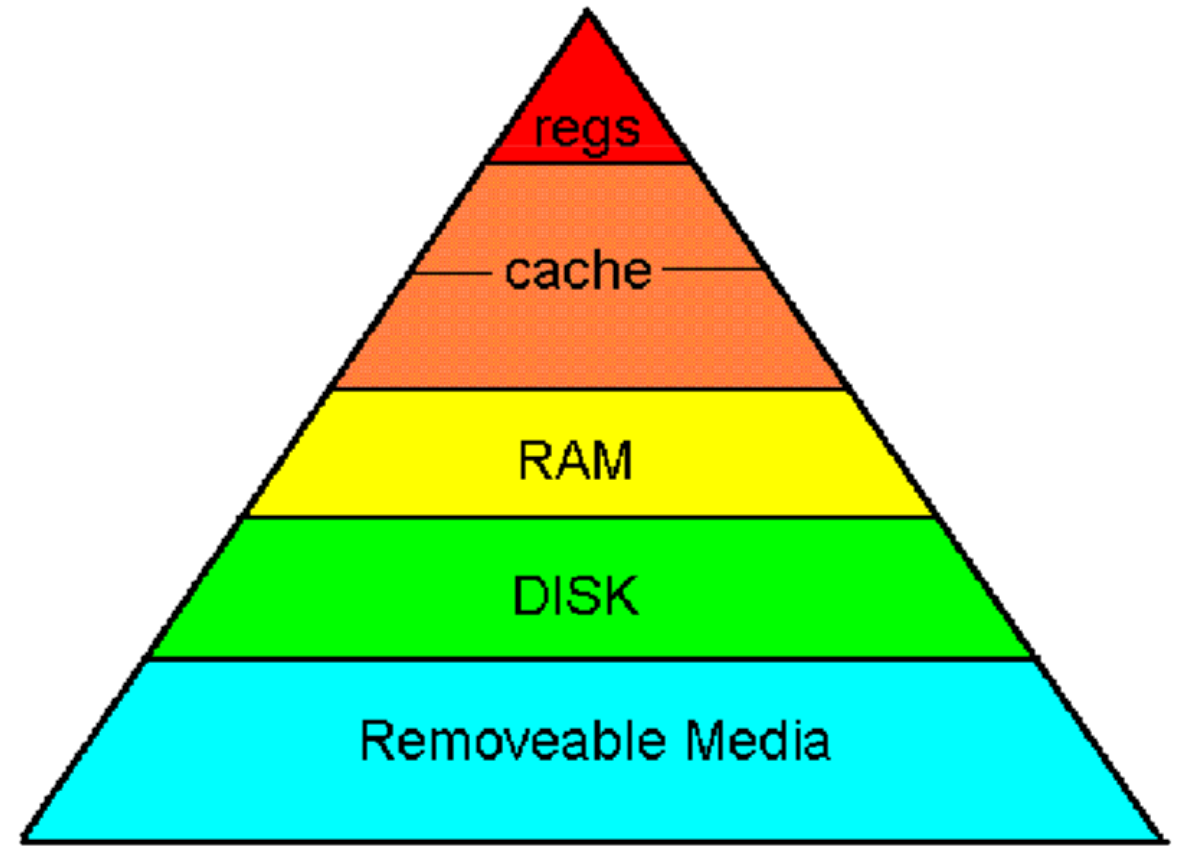
I/O controllers

- Direct the flow of data to and from I/O devices.
- CPU sends a request to the I/O controller to initiate I/O.
- I/O controllers run independently and in parallel with the CPU
- I/O controllers may interrupt the CPU upon completion of request or error.



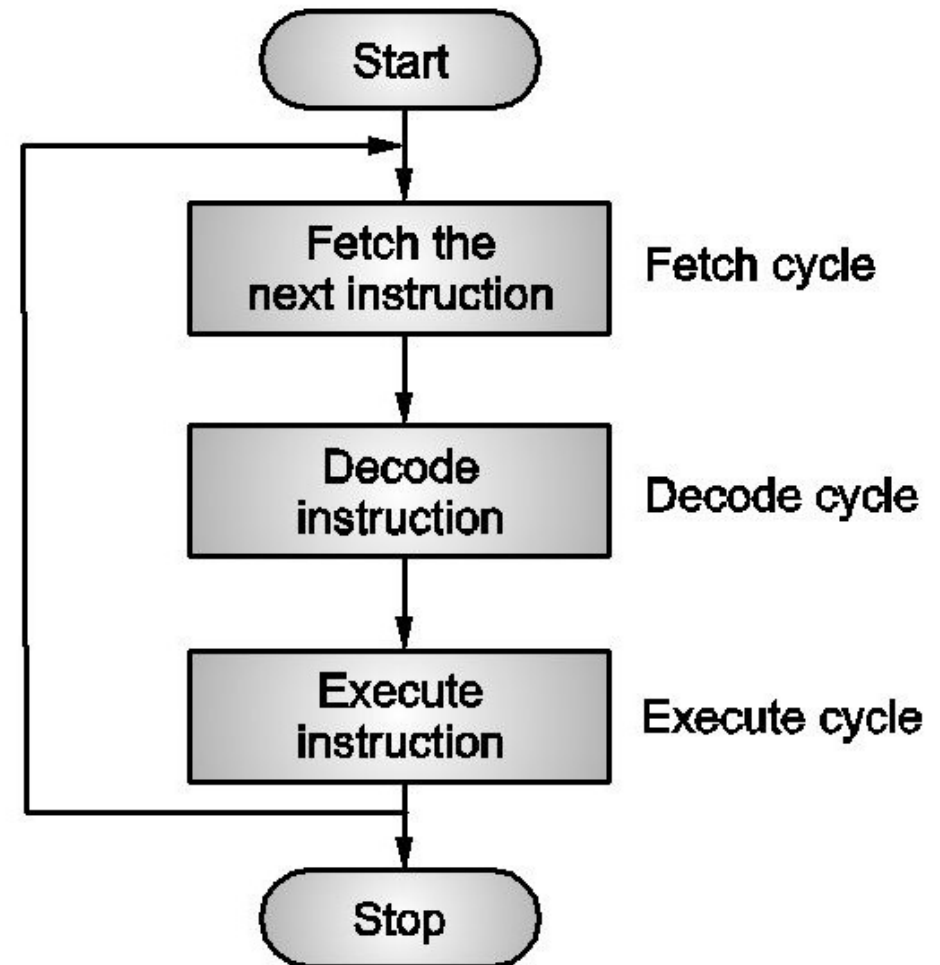
Memory hierarchy

- The internal memory is Random Access Memory (RAM).
- Both data and program instructions are kept in RAM.
- Instructions must be in RAM to be executed.



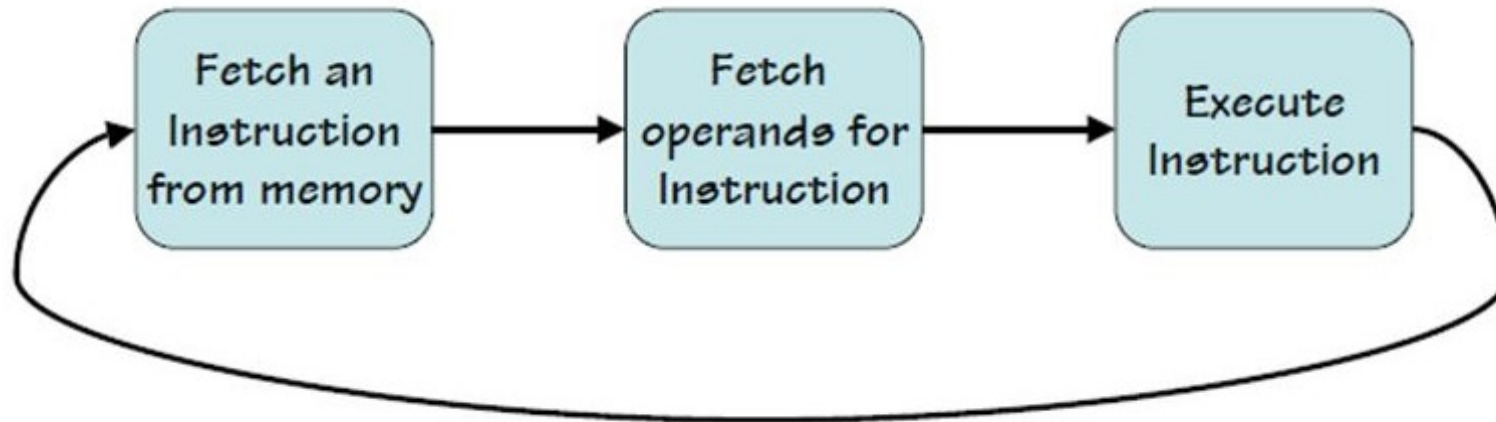
Instruction cycle

- Fetching the instruction from memory and executing the instruction
 1. Fetch the instruction from the memory address in the Program Counter register
 2. Increment the Program Counter
 3. Decode the type of instruction
 4. Fetch the operands
 5. Execute the instruction
 6. Store the results



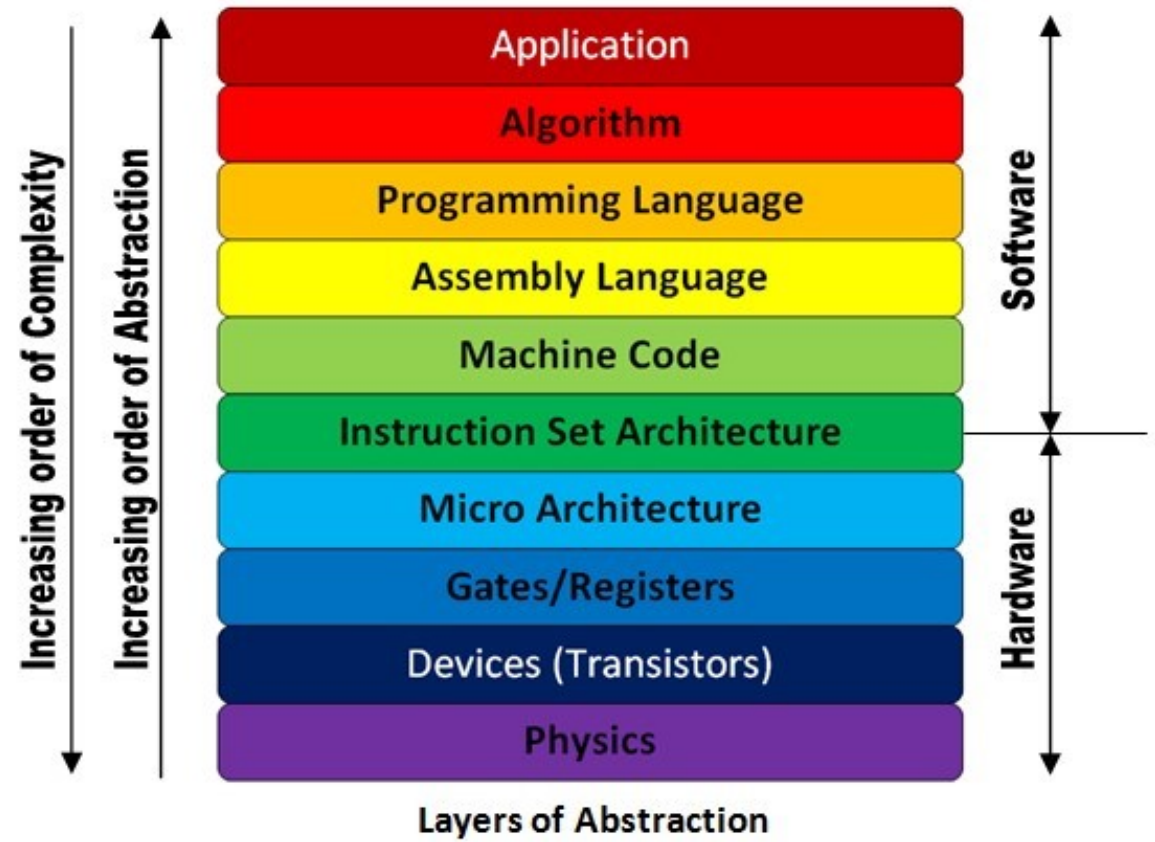
Simple model of execution

- Instruction sequence is determined by a simple conceptual control point.
- Each instruction is completed before the next instruction starts.
- One instruction is executed at a time.



Layers

- You can consider computer operation at many different levels.
 - Applications
 - Middleware
 - High level languages
 - Machine Language
 - Microcode
 - Logic circuits
 - Gates
 - Transistors
 - Silicon structures



Microprocessors & Microcontrollers

Processor

- A Processor is the heart of a computer system.

Microprocessor

- A microprocessor is a single chip semi conductor device also which is a computer on chip, but not a complete computer.
- Its CPU contains an ALU, a program counter, a stack pointer, some working register, a clock timing circuit and interrupt circuit on a single chip.
- To make complete micro computer, one must add memory usually ROM and RAM, memory decoder, an oscillator and a number of serial and parallel ports.

Microcontroller

- A microcontroller is a functional computer system-on-a-chip. It contains a processor, memory, and programmable input/output peripherals.
- Microcontrollers include an integrated CPU, memory (a small amount of RAM, program memory, or both) and peripherals capable of input and output.

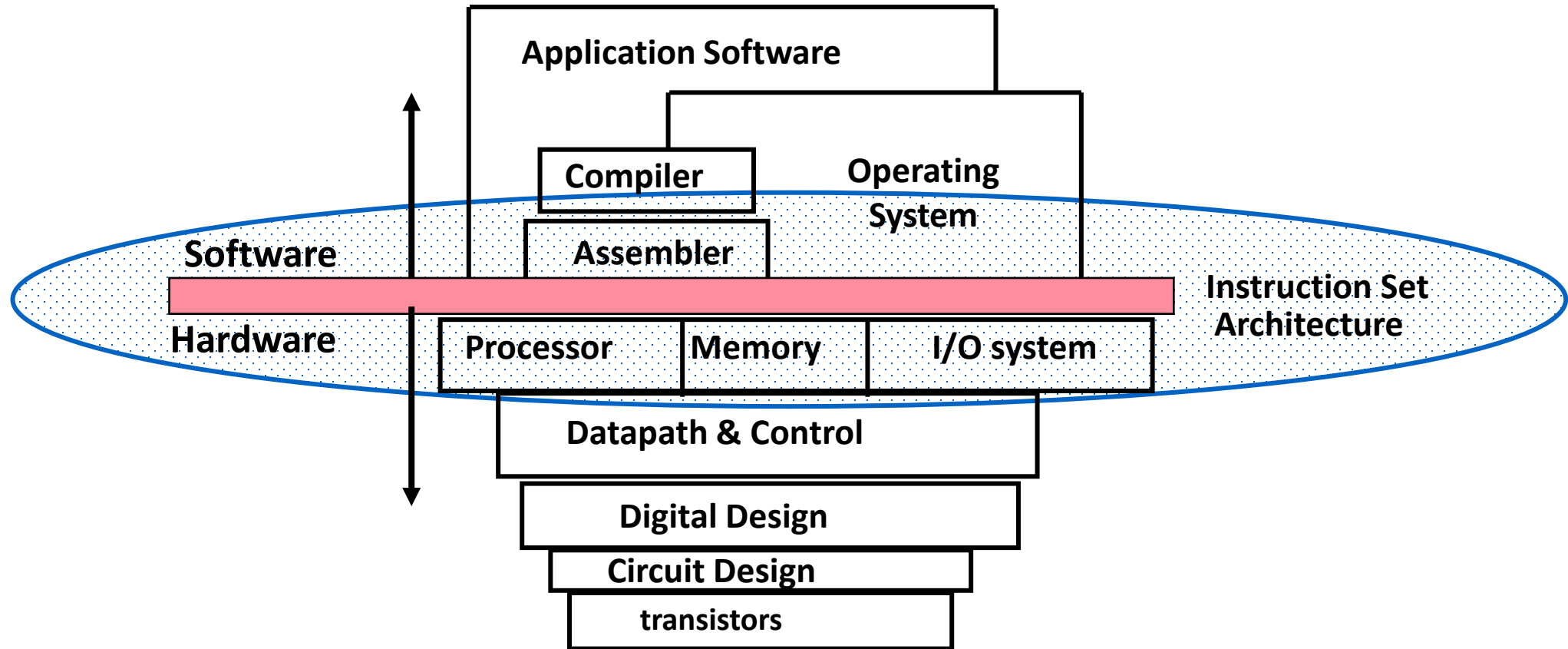
Various Microcontrollers

- INTEL
 - 8031,8032,8051,8052,8751,8752
- PIC
 - 8-bit PIC16, PIC18,
 - 16-bit DSPIC33 / PIC24,
 - PIC16C7x
- Motorola
 - MC68HC11

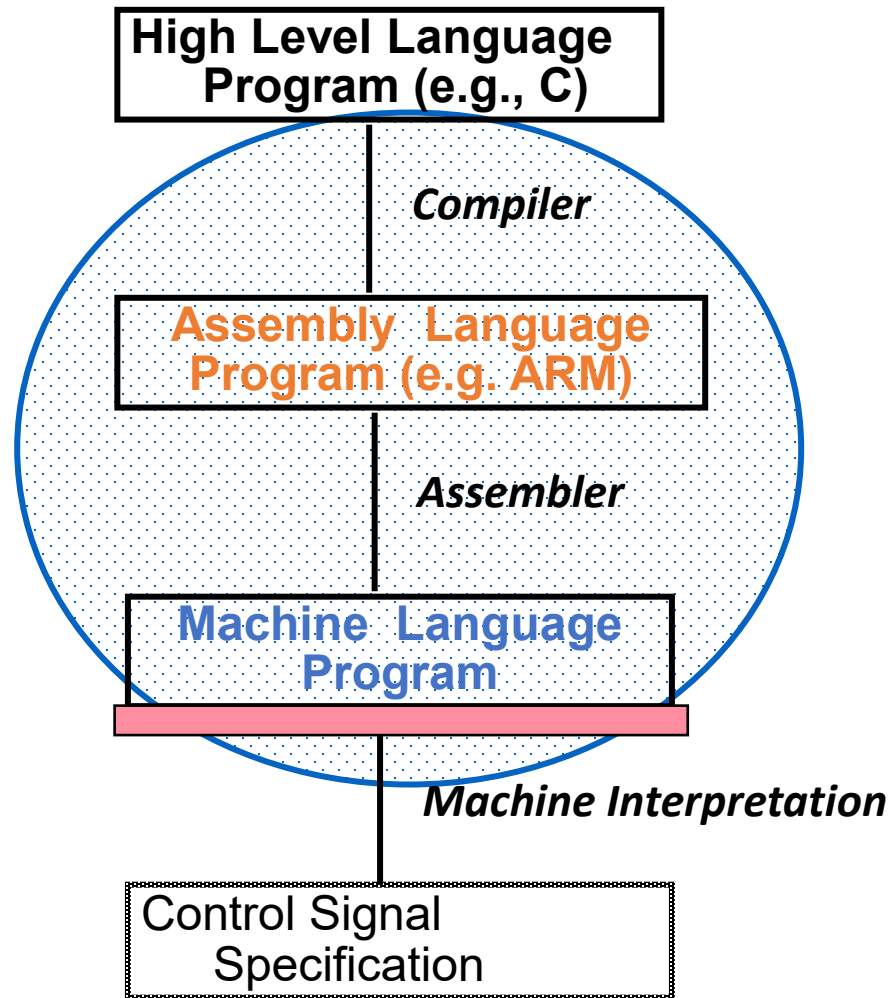
Microprocessor vs. Microcontroller

MICROPROCESSOR	MICROCONTROLLER
The functional blocks are ALU, registers, timing & control units	It includes functional blocks of microprocessors & in addition has timer, parallel i/o, RAM, EPROM, ADC & DAC
Bit handling instruction is less, One or two type only	Many type of bit handling instruction
Rapid movements of code and data between external memory & MP	Rapid movements of code and data within MC
It is used for designing general purpose digital computer systems	They are used for designing application specific dedicated systems

What is this course about?



Programming Levels of Representation



```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

```
ldr r0 , [r2, #0]  
ldr r1 , [r2, #4]  
str r1 , [r2, #0]  
str r0 , [r2, #4]
```

```
1110 0101 1001 0010 0000 0000 0000 0000  
1110 0101 1001 0010 0000 0000 0000 0100  
1110 0101 1000 0010 0001 0000 0000 0000  
1110 0101 1000 0010 0001 0000 0000 0100
```

```
ALUOP[0:3] <= InstReg[9:11] & MASK
```


Course Details

Topics Outline

1. Introduction
2. Digital number systems
3. Microprocessor systems and VLSI devices
4. Basic cells
5. Assembly language programming
6. Interfacing
7. Memory Design
8. A/D and D/A Conversion
9. Microcomputer System Software
10. Examples

Textbooks

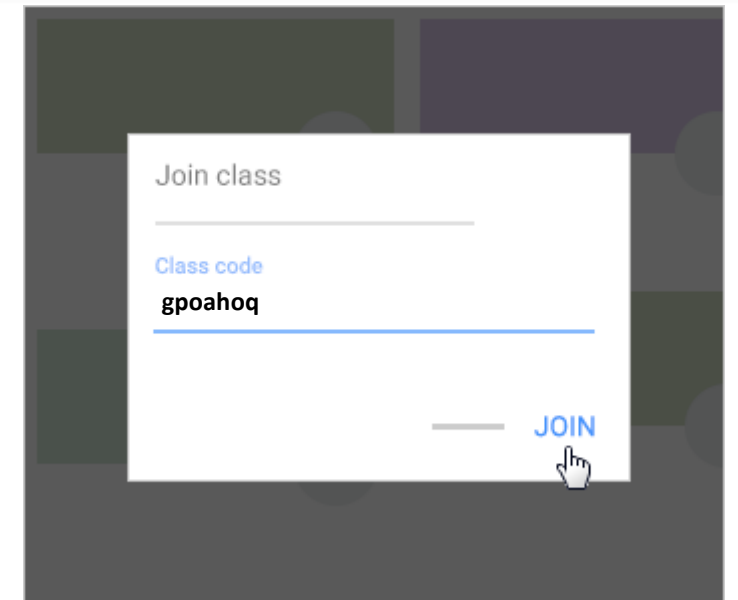
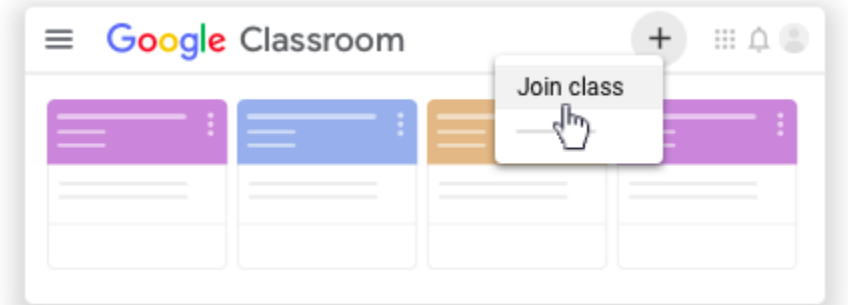
- Main references:
 - David Patterson and John Hennessy: Computer Organisation & Design: The HW/SW Interface
- Additional references:
 - Waldron, John: Introduction to RISC Assembly Language, Addison-Wesley Publishing

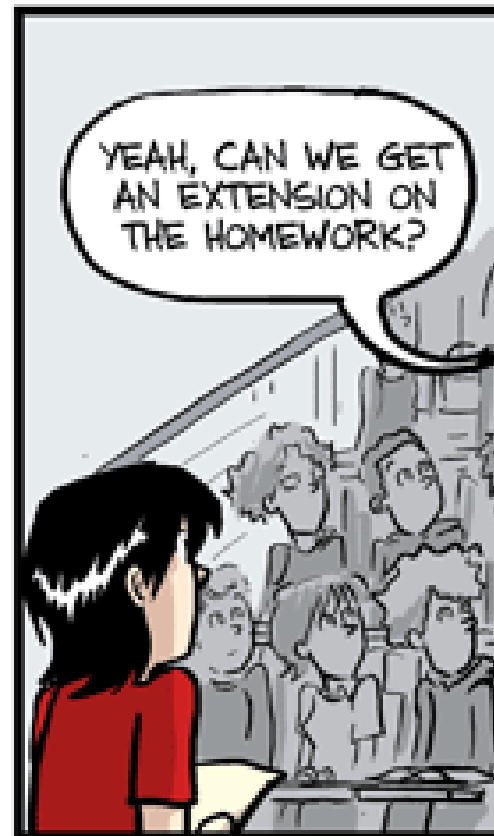
Quizzes

- Weekly/fortnightly online quizzes
- They are designed to help you stay up-to-date with the lectures
- You can do them from anywhere
- It is your responsibility to check for the quizzes and complete them.

Additional Information

- Course Homepage:
 - <https://academic.nimal.info/ee2222/>
- KDU LMS:
 - ???
- Google Classroom:
 - Go to <https://classroom.google.com/>
 - At the top, click Add + and then Join class.
 - Enter Class Code: **gpoahoq**
- Contact:
 - nimal@sjp.ac.lk





JORGE CHAM © 2008



WWW.PHDCOMICS.COM