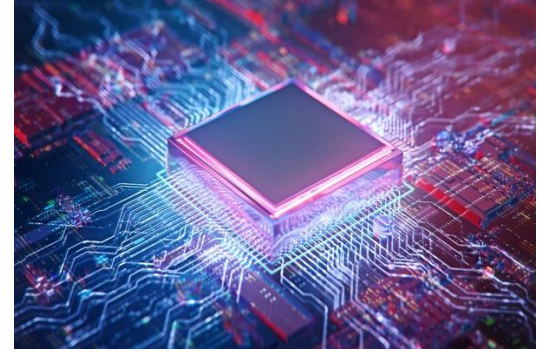




EL2003

Computer Organization and Assembly Language



Dr. Muhammad Usman Abbasi
Assistant Professor | Head of Electrical Engineering Department
National University of Computer and Emerging Sciences - FAST

1. Introduction to Assembly Language

- 1.1. Basic Computer Architecture
- 1.2. Registers
- 1.3. Instruction Groups
- 1.4. Intel iapx88 Architecture
- 1.5. History
- 1.6. Register Architecture
- 1.7. Our First Program
- 1.8. Segmented Memory Model

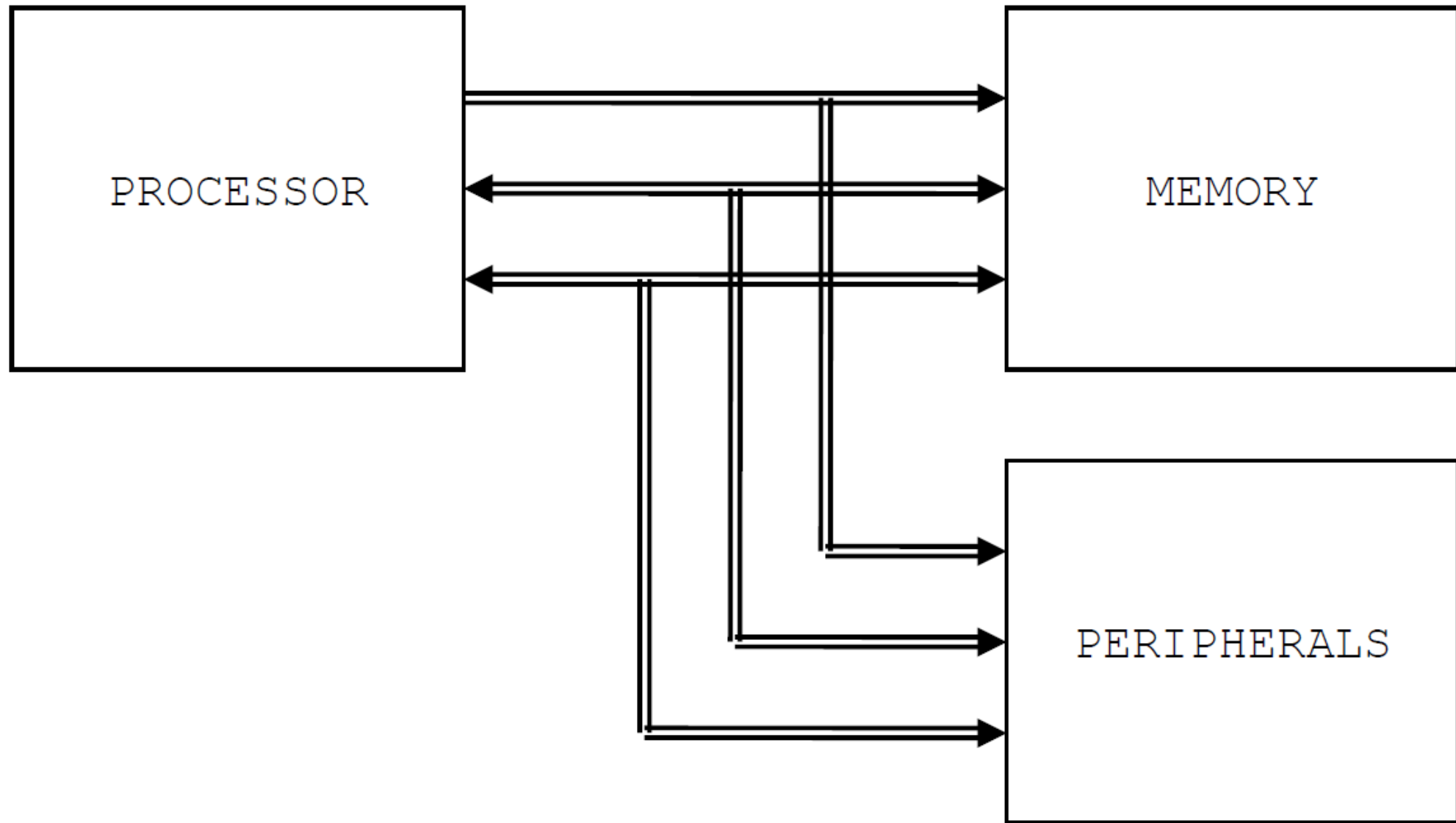
1.1. Basic Computer Architecture

- Address, Data, and Control Buses
 - Processor, memory, and I/O devices
 - I/O is used for interfacing with the external world
 - memory is the processor's internal world
- Memory is a dumb device
- Whole working of the computer is performing an operation by the processor on data, which resides in memory

Key points

- There must be a mechanism to inform memory that we want to do the read operation
- There must be a mechanism to inform memory that we want to read precisely which element
- There must be a mechanism to transfer that data element from memory to the processor

- Address bus:
 - The group of bits that the processor uses to inform the memory about which element to read or write.
- Data bus:
 - Move the data from the memory to the processor in a read operation and from the processor to the memory in a write operation.
- Control bus:
 - Miscellaneous independent lines used for control purposes.
 - For example one line of the bus is used to inform the memory about whether to do the read operation or the write operation



1.2. Registers

- The basic purpose of a computer is to perform operations.
- Operations need operands.
- Operands are the data on which we want to perform a certain operation.

- Consider the addition operation.
- It involves adding two numbers (8-bit, 16-bit, 32-bit, 64-bit etc.)
- We can have precisely one address on the address bus and consequently precisely one element on the data bus.
- At the very same instant the second operand cannot be brought inside the processor.
- As soon as the second is selected, the first operand is no longer there.
- For this reason there are temporary storage places inside the processor called *registers*.

More on registers...

- Registers are like a scratch pad ram inside the processor and their operation is very much like normal memory cells.
- They have precise locations and remember what is placed inside them.
- Names: r0, r1, r2, ... or A, B, C, D, ..., etc.
- X stands for an index register.

- Accumulator
- Pointer, Index, or Base Register
- Flags Register or Program Status Word
- Program Counter or Instruction Pointer

- Dosbox
- NASM
- AFD