Microprocessor and Assembly Language

Chapter-One

Introduction to microprocessor and Assembly Language

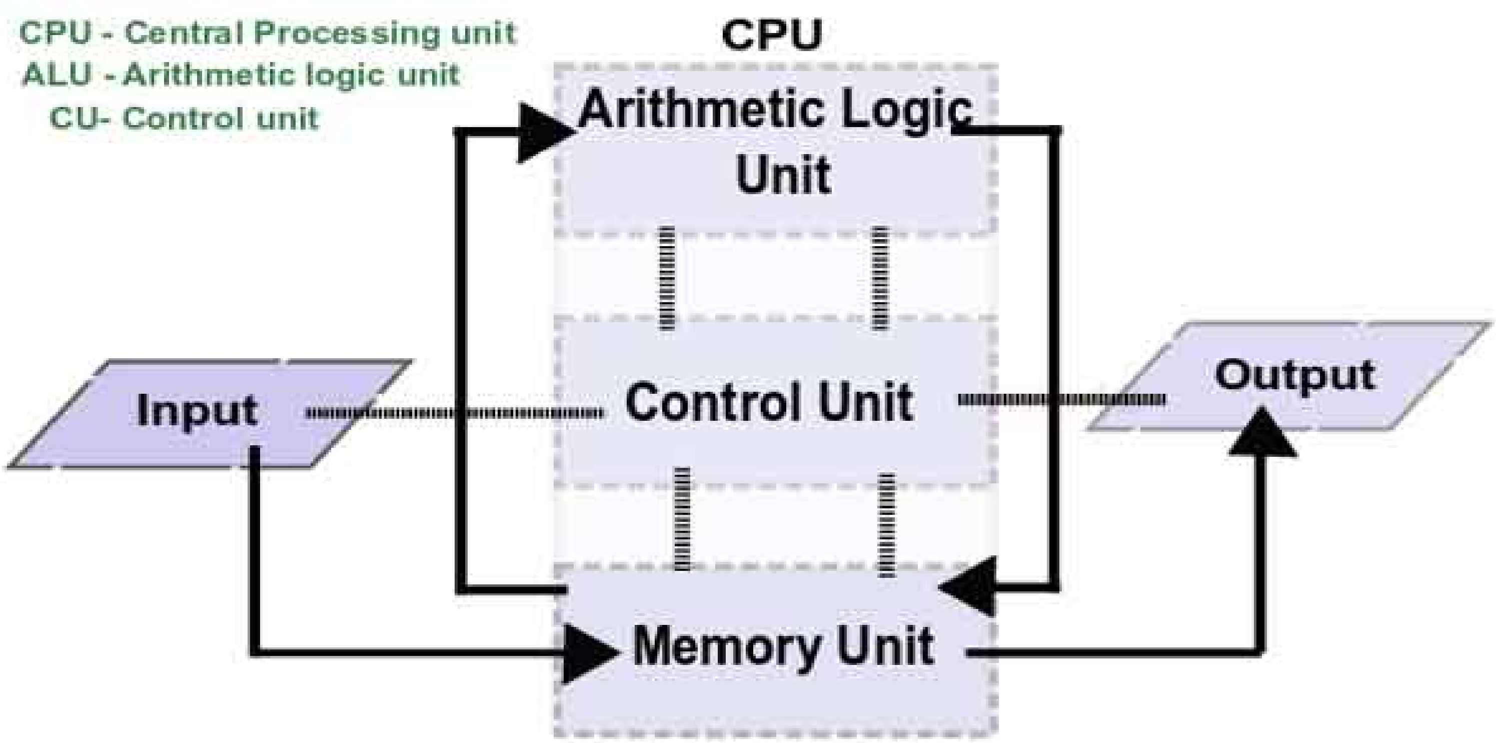
By Bekan . k

Outline

- Introduction
- Fundamentals of Microprocessors
- Language Translators
    - Compiler,
    - Interpreters and
    - Assemblers
- Instruction fields
- Evolution of microprocessor
- Intel family Microprocessors

What is Computer?

- **Computer:** is a programmable machine that receives input, stores and manipulates data/information, and provides output in a useful format.
- Basic computer system consist of **CPU**, **memory** and **I/O** unit.



What is a Microprocessor?

- The word microprocessor comes from the combination **micro** and **processor**.

✓ **Processor** means a device that processes whatever. In this context processor means a device that processes numbers, specifically **binary numbers**, 0’s and 1’s.

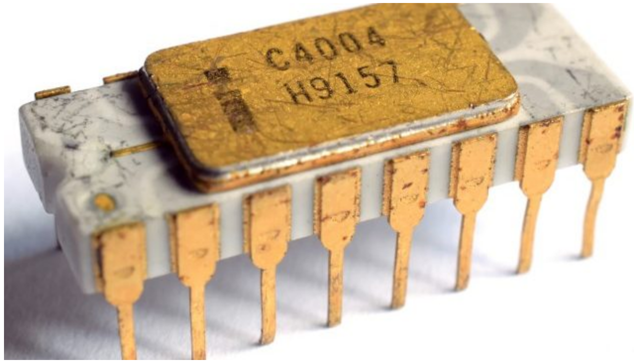
  - **To process** means **to manipulate** or to perform certain operations on the numbers that depend on the microprocessor’s design.

Cont….

- **Micro** is representation of size(**small size**).
- In this context it’s used to specify the size of processor. This processor can be of any size as long as it does the job. But as today's technology is going towards making **things as small as possible**, we also need processors that are small in size with all the necessary peripherals into it.
- You can think of sim cards that we use in mobile phones as an example. Earlier we called it sim, now we either say mini-sim or micro-sim.
- In the early 1970’s the **microchip** was invented. All of the components that made up the processor were now placed on a **single piece of silicon**.
  - The size now became several **thousand times smaller** and the **speed became several hundred times faster**.

Cont...

- Microprocessor is a **programmable** device that takes in numbers, performs on them arithmetic or logical operations according to the program stored in memory and then produces output as a result.
  - **Programmable -** perform different set of operation on the data depending on the sequence of instructions supplied by the programmer.



Oldest micro processor

Today's Microprocessor

Cont….

- It can also be defined as a **Central Processing Unit (CPU)** on a single chip that contains millions of transistors connected by wires.
- Its main components are:-
  - ALU
  - Registers
  - Control Unit
- **ALU:** Performs Arithmetic and Logic Operations. Every microprocessor has **arithmetic operations** such as add, subtract, multiply and divide as part of its instruction set. Of course the newer version have complex operations such as square root.
  - In addition, microprocessors have **logic operations** as well. Such as AND, OR, XOR, shift left, shift right, etc.

Cont….

- **Registers:** An array of storage for holding data while it is being manipulated.
- **Control Unit(CU):** to fetch and execute instructions from the memory of a computer.
  - ✓ It receives the input instruction/information from the user and converts it into control signals, which are then given to the CPU for further execution.

Then, what is microcomputers?

- Any microprocessor-based systems having limited number of resources are called **microcomputers**.
- Nowadays, microprocessor can be seen in almost all types of electronic devices like mobile phones, printers, washing machines etc.
- Microprocessors are also used in advanced applications like radars, satellites and flights.
- Due to rapid advancements in electronic industry and large scale integration of devices, there is an increase in the application of microprocessors.

Assembly Language and Machine Language

Assembly language

- Assembly language is a **low-level language** that needs compiler and interpreter, which converts high or low level language to machine language. And then it could be understood by a computer.
  - Readability of instructions is better than machine language.

Machine language

- Machine language is **series of bit patterns** (that is the binary form) that are directly executed by a computer.
  - Native to a processor: executed directly by hardware.
  - Instructions consist of binary code: 1s and 0s

Cont...

- Assembly language were developed to simplify the chore of entering binary code into a computer as an instruction. They uses mnemonic codes to represent operations.

  Example: the use of mnemonic codes, such as ADD for addition, in place of a binary numbers!

- Machine language is **not human readable**;
- Assembly language is a set of instructions which can be **read by human** and can be understood as well.
  - Here instead to remember the op-codes, “mnemonics”, are used.
  - It is however **less readable than high-level language**.

Compiler, Interpreters and Assemblers

- Computer programs are usually written on high level languages.
  - ✓ A **high level language** is one that can be understood by humans.
  - To make it clear, they contain words and phrases from the languages in common use, English/other languages.
- However, computers **cannot** understand high level languages as we humans do.

Cont…

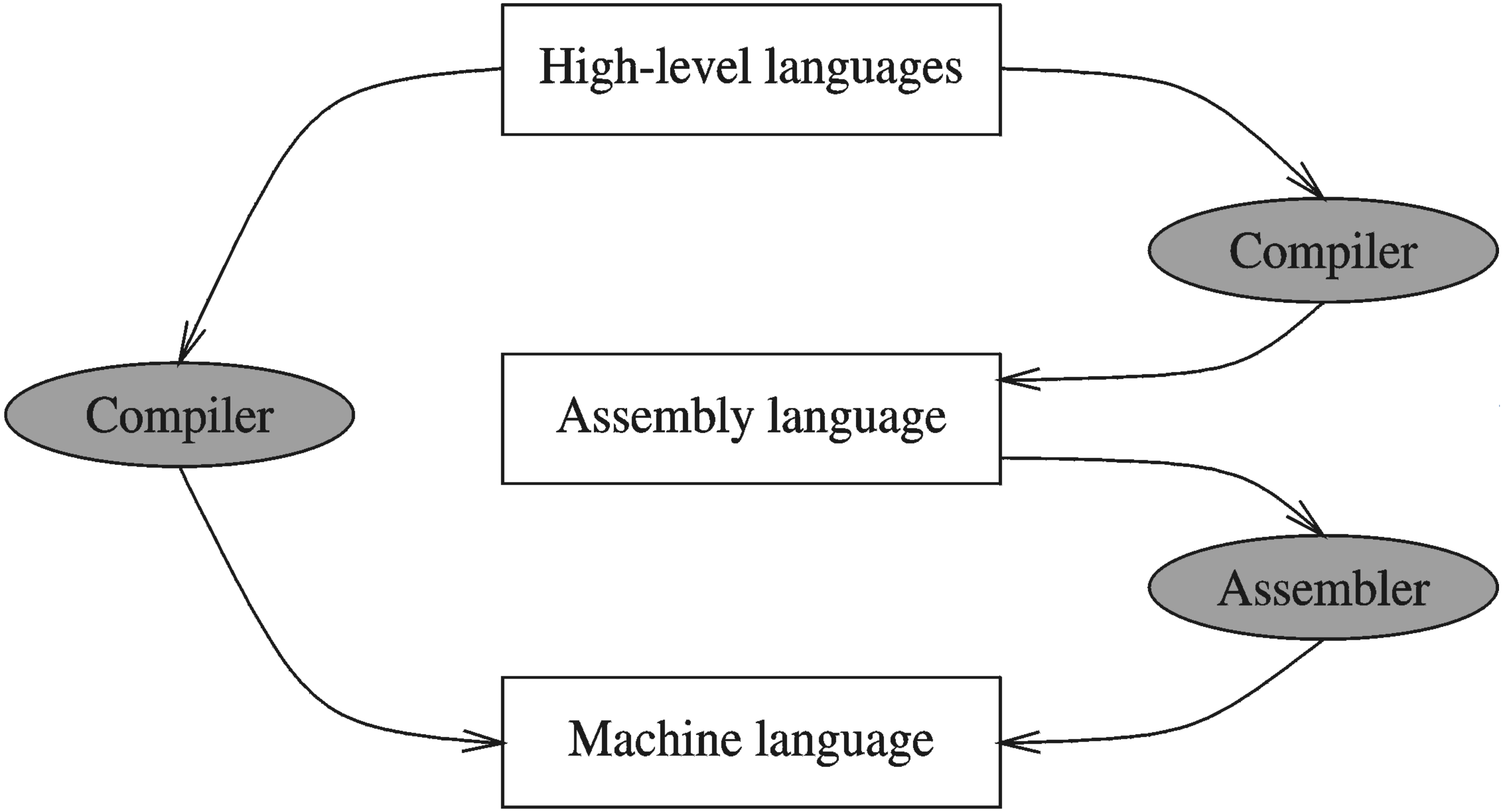
- They can only understand the programs that are developed in binary systems known as a machine code.
- Thus, this high level languages have to be converted into machine code so as to be understood by the computer.
- Those programs that are used to convert languages into machine codes are called **language translators**.
- These are compilers, interpreters and assemblers.
- Compilers and interpreters are programs that are used to convert high level language (Source Code) into machine codes.

Cont….

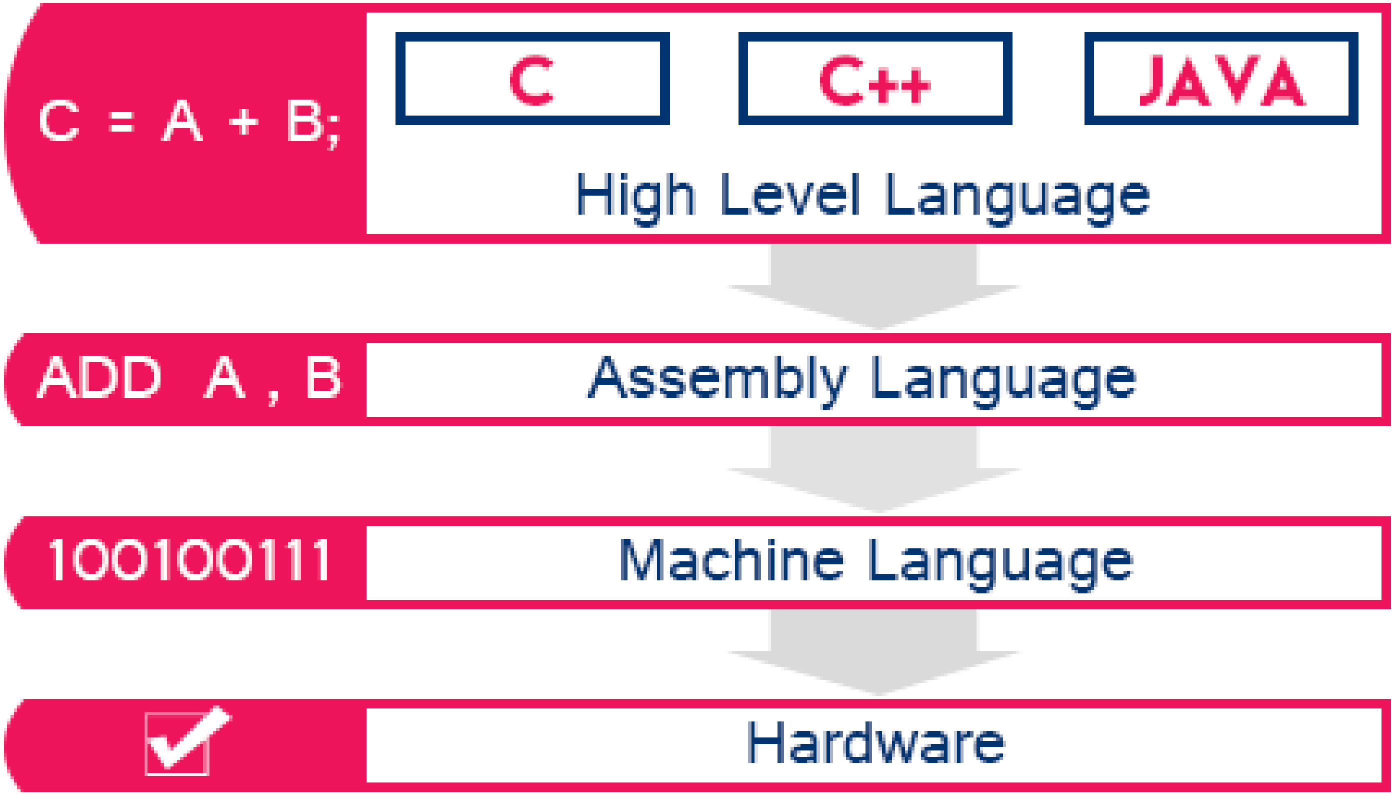
- **Compiler** analyze all the language statements that are written in high level language to check if they are correct. If it comes across something incorrect, it will give an error message. If there are no errors spotted, the compiler will convert the **whole source code into machine code at once**.
- **Interpreter** also convert the high level language into machine language **but line by line**.
- **Assembler** converts assembly language program to machine language.

Cont...

- Assemblers translate assembly to machine code.
- Compilers translate high-level programs to machine code.
  - Either directly, or
  - Indirectly via an assembler.



Machine, Assembly and high level languages



Instructions in Assembly Language

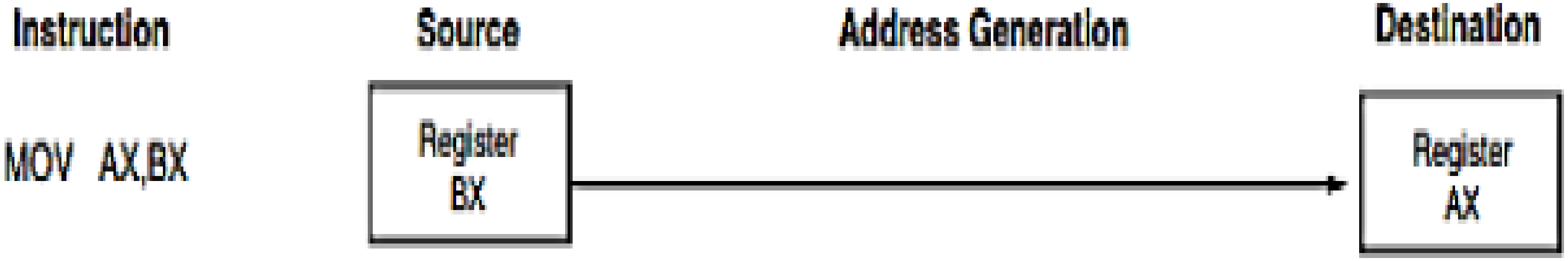
- Each command of a program is called an instruction.
- This instructions instructs or tells the computer what to do.
- Computers only deal with binary data, hence the instructions must be in binary format (0s and 1s) .
- The set of all instructions (in binary form) makes up the computer's machine language.
- This set of instruction also referred to as the *instruction set*.

Instructions Fields

- Assembly language instructions usually are made up of several fields. The major two fields are:
- **The Opcode**: it selects the operation (addition, subtraction, move, and so on) that is performed by the microprocessor.
  - Each operation has its unique opcode.
  - The opcode is either 1 or 2 bytes long for most machine language instructions.
- **MOD Field**: specifies the addressing mode for the selected instruction.
  - ✓ It specifies where to get the source and destination operands for the operation specified by the opcode.
  - ✓ Sometimes called **(Operand) modes.**

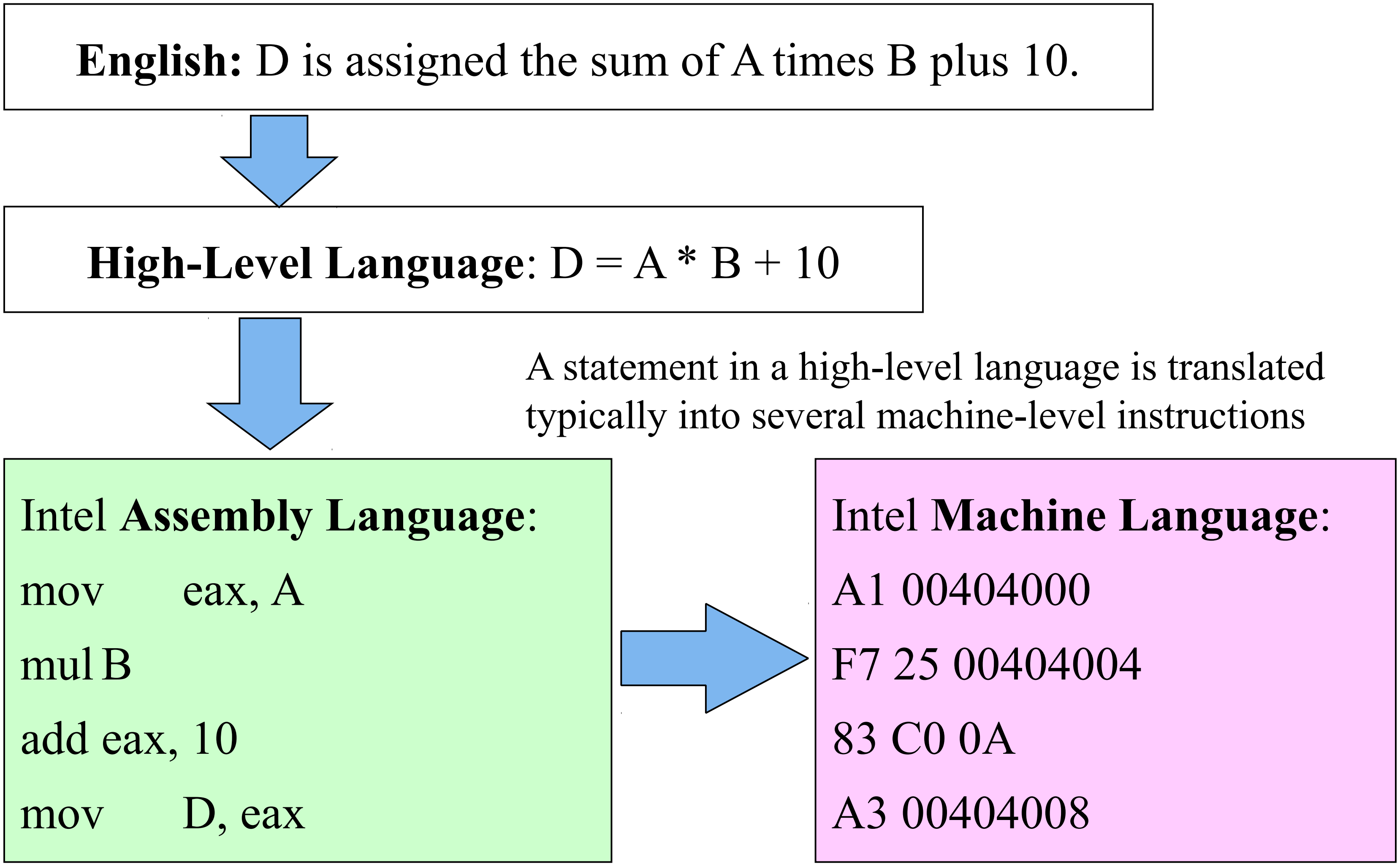
Instructions Fields(Cont...)

- The source/ destination of operands can be a constant, memory or one of the general-purpose registers.



  - This instruction copies the content of BX register into AX register(Register movement instruction).
  - Here the **Opcode is MOV**.

Translating Languages



English: D is assigned the sum of A times B plus 10.

High-Level Language: D = A * B + 10

A statement in a high-level language is translated typically into several machine-level instructions

Intel **Assembly Language**:

```
mov	eax, A
mul	B
add eax, 10
mov	D, eax
```

Intel **Machine Language**:

```
A1 00404000
F7 25 00404004
83 C0 0A
A3 00404008
```

The need to learn Assembly Language

- The main reason:
  - Accessibility to system hardware
- Some application of assembly languages are:
  - ✓ Real time system. E.g Traffic control system
  - ✓ Embedded system. Where there is no compiler.

    E.g Micro chips.
  - ✓ Operating system. Specially kernel part of operating system. Where direct access of hard ware is necessary.

Evolution of Microprocessor

- A microprocessor incorporates most or all of the functions of central processing unit (CPU) on a single integrated circuit (IC).
    - ✓ Sometimes referred to as the **CPU**, is the controlling element in a computer system.
    - ✓ The microprocessor controls memory and I/O through a series of connections called buses.
- A **bus** is a common group of wires that interconnect components in a computer system.
    - It transfer data between an I/O device or memory and the microprocessor, and control the I/O and memory system.

Cont...

- The world's **first microprocessor**, the **Intel 4004**, was a **4-bit** microprocessor,were used for electronic calculators.
    - It has a 4-bit-wide memory locations(often called nibble).
    - It contains only 45 instructions.
- Later, **Intel** Corporation released the **8008 -** an **extended 8-bit** version of the 4004 microprocessor.
    - ✓ The 8008 addressed an expanded memory size (16K bytes) and <u>contained additional instructions</u> (a total of 48).
- Later on, **Intel** introduced the **8080 microprocessor** in 1973 - the first **8-bit microprocessors**, that addresses more memory, executes additional instructions and 10 times faster than the 8008.
- The microprocessors after ward are named as the modern microprocessors.

Modern microprocessor

- In 1978, Intel released the **8086 microprocessor**; a year later, it released the **8088**.
  - Both devices are **16-bit microprocessors**.
  - Executes instructions in as little as 400 ns (2.5 MIPs, or 2.5 millions of instructions per second).
  - Addressed 1M byte of memory.
  - Has higher execution speed and larger memory size.

Cont...

- The **80286 microprocessor** (also a **16-bit architecture microprocessor**).
- Identical to the 8086 and 8088 except,
  - It addressed a **16M-byte memory system** instead of a 1M-byte system.
  - Contain few **additional instructions** that managed the extra 15M bytes of memory.
- Then **80386** represented a major overhaul of the 16-bit 8086–80286 architecture.
  - ✓ It was Intel's first practical 32-bit microprocessor that contained a 32-bit data bus and a 32-bit memory address.
  - ✓ It addressed up to 4G bytes of memory.

Modern microprocessor(Cont...)

- In 1989, Intel released **80486 microprocessor**(only internal structure modified from 80386).
- The **Pentium**, introduced in 1993, was similar to the 80386 and 80486 microprocessors. This microprocessor labeled the **P5** or **80586**.
  - Has a speed of 110 MIPs, millions of instructions/second.
- The most recent version of the Pentium is called the **Core2** by Intel.
  - ✓ **Core2** is available at speeds of up to 3 GHz.

Modern microprocessor(Cont...)

- Recently **Intel** has included new modifications to the Pentium 4 and Core2 that include a 64-bit core and multiple cores.
  - It allows the microprocessor to address more than 4G bytes of memory.
- Currently, 40 address pins in these newer versions allow up to 1T (terabytes) of memory to be accessed.
  - ✓ Intel manufactures dual and quad core versions, and many Cores are still on progress...

Intel family microprocessors

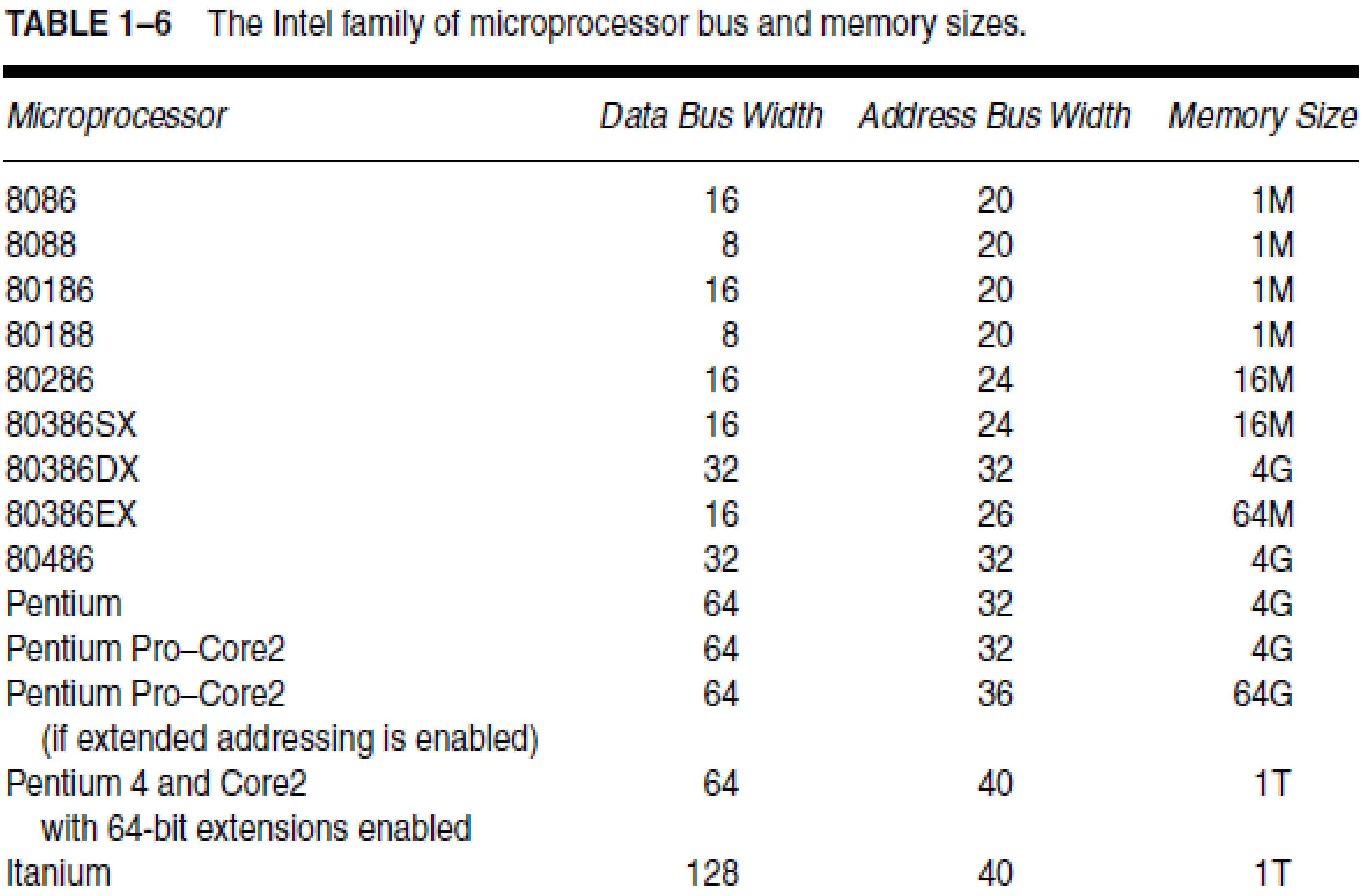


TABLE 1–6 The Intel family of microprocessor bus and memory sizes.

Microprocessor	*Data Bus Width*	*Address Bus Width*	*Memory Size*
8086	16	20	1M
8088	8	20	1M
80186	16	20	1M
80188	8	20	1M
80286	16	24	16M
80386SX	16	24	16M
80386DX	32	32	4G
80386EX	16	26	64M
80486	32	32	4G
Pentium	64	32	4G
Pentium Pro–Core2	64	32	4G
Pentium Pro–Core2 (if extended addressing is enabled)	64	36	64G
Pentium 4 and Core2 with 64-bit extensions enabled	64	40	1T
Itanium	128	40	1T