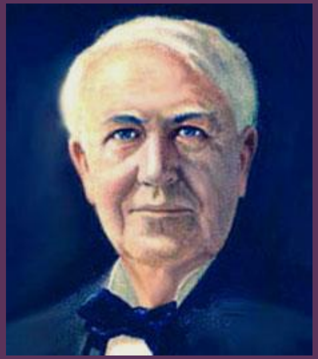
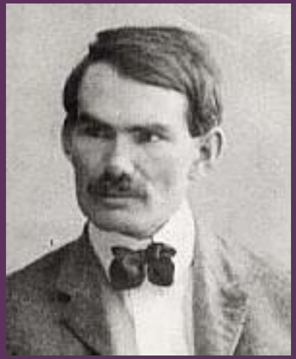
Microprocessors

Dr Mridul Gupta
Assistant Professor
Dept. of ECE

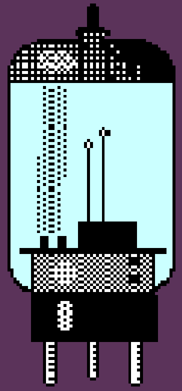
UNIT-1

Introduction



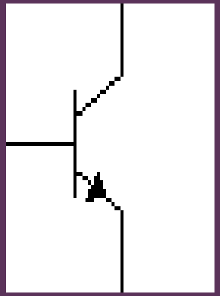



<u>Tubes</u>



- A computer of the **first generation** consisted of tubes.
- Tubes started with an effect Thomas Edison noticed while experimenting with light bulbs.
- John Ambrose Fleming discovered that one could exploit the effect to detect radio waves and convert them to electricity, but the signal was too small.
- Lee de Forest added to the device, making the triode; Edwin Armstrong pointed out it could be used to amplify signals.

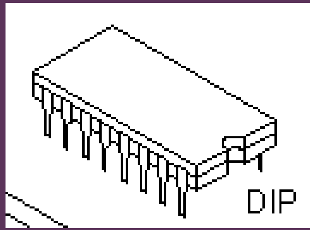
Introduction

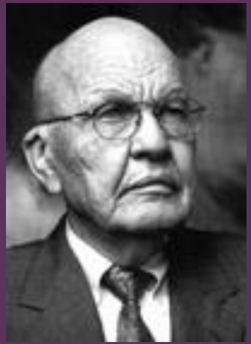


Transistors

- A computer of the **second generation** consisted of transistors.
- William Shockley, John Bardeen and Walter Brattain developed the transistor while working at Bell Labs in 1947. (Nobel Prize 1956)
- The transistor could play the same role as the vacuum tube but was significantly smaller – and thus faster and less power consuming.

Introduction





<u>Integrated Circuit</u>

- A computer of **third generation** consisted of integrated circuits.
- The problem with computers is that they required so many transistors connected to one another.
- This problem was solved by the "monolithic idea" – the idea the many circuit elements (mainly transistors) could be placed on the same piece of semiconductor, i.e. an integrated circuit (IC).
- In 1958 Jack Kilby of Texas Instruments invented the IC. In 1959 Robert Noyce of Fairchild Semiconductor independently developed a better-designed IC. (Nobel Prize 2000.)

Introduction

CPU

- The **Central Processing Unit** (CPU) is the "**brain**" of the computer.
- The CPU can be thought in terms of two basic pieces:
  - The **Arithmetic Logic Unit** (ALU) which modifies data by executing arithmetic and/or logical operations on it.
  - The **Control Unit** which takes the instructions from memory, decodes it, and then moves the data to the appropriate places and ensures that the ALU performs the desired operation.

Introduction

Generation

- The CPU is often just called the processor.
- For the ENIAC (Electronic Numerical Integrator and Computer- first programmable, electronic, general-purpose digital computer made in 1945) and other computers of the **first generation**, the processor was comprised of vacuum tubes.
- The processor's individual vacuum tubes were replaced by individual transistors in the **second generation** of computers.
- In **third generation** computers the processor consisted of several IC's.

Introduction

Processor → Microprocessor

- A computer of **fourth generation** has a microprocessor.
- Invented in 1971, the microprocessor evolved from the inventions of the transistor and the integrated circuit.
- Essentially a computer on a chip, it is the most advanced application of the transistor.

The influence of the microprocessor today is well known, but in 1971 the effect the microprocessor would have on every- day life was a vision beyond even those who created it.

-

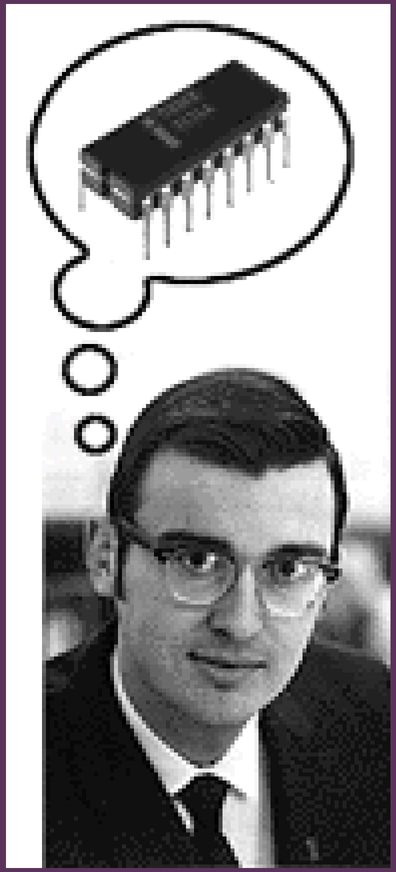
Introduction


Noyce


Moore

Intel

- Intel was founded by Bob Noyce and Gordon Moore who formerly worked for Fairchild Semiconductor.
- In a collaboration between Busicom (a Japanese firm) and Intel to make calculators, Ted Hoff devises a plan to put all the main circuitry on one chip instead of the original plan of twelve (in 1968).
- Intel made the 4004 (the first microprocessor in 1971) for Busicom. Knowing they had a good thing, they bought the rights to 4004 from Busicom for $60,000.
- The microprocessor was born.


Ted Hoff

Introduction

Microprocessor

- Processor means a device that processes whatever.
- To process means to manipulate.
- All of the components that made up the processor were placed on a single piece of silicon. The size became several thousand times smaller and the speed became several hundred times faster. The “Micro”Processor was born.

Definition of the Microprocessor

- The 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 other numbers as a result.

Evolution of Microprocessors

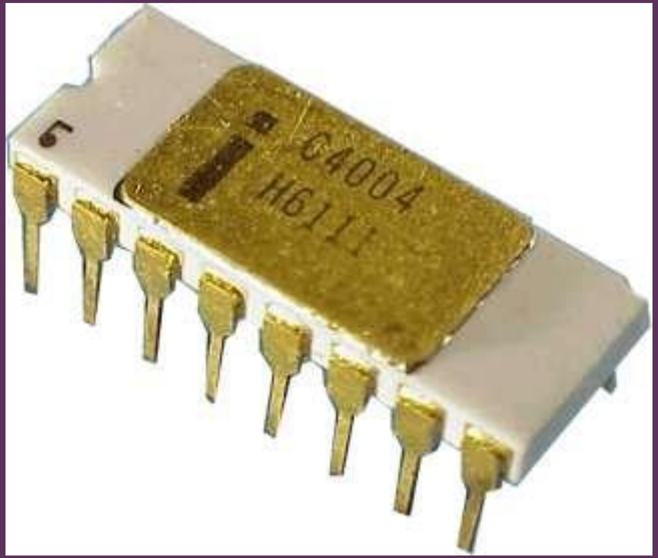
- Fairchild Semiconductors (founded in 1957) invented the first IC in 1959.
- In 1968, Robert Noyce, Gordan Moore, Andrew Grove resigned from Fairchild Semiconductors.
- They founded their own company Intel (Integrated Electronics).
- Intel grown from 3 man start-up in 1968

Evolution of Microprocessors

Classification

- 4-Bit Microprocessors
- 8-Bit Microprocessors
- 16-Bit Microprocessors
- 32-Bit Microprocessors
- 64-Bit Microprocessors

4-BIT MICROPROCESSORS



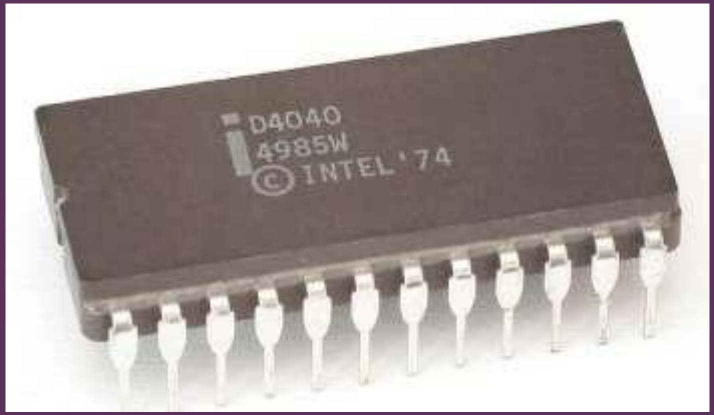
INTEL 4004

- Introduced in 1971.
- It was the first microprocessor by Intel intended for an electronic calculator.
- It was a 4-bit µP.
- Its clock speed was 740KHz.
- It had 2,300 transistors.
- It could execute around 60,000 instructions per second.

4-BIT MICROPROCESSORS

INTEL 4040

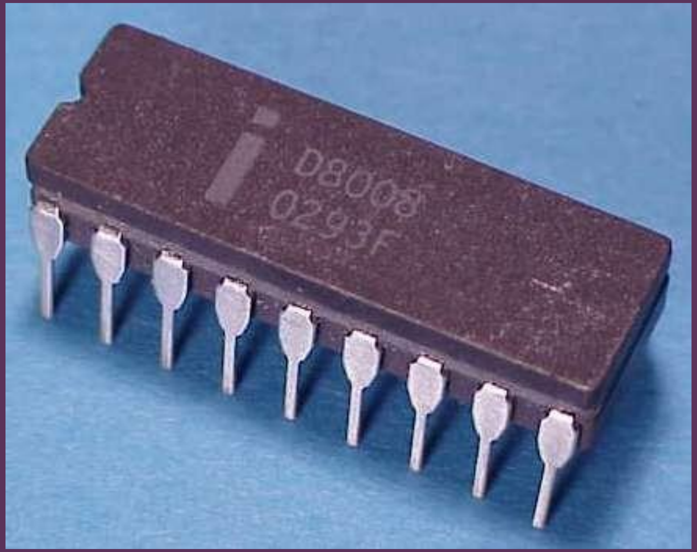
- Introduced in 1974.
- It was also 4-bit µP.



8-BIT MICROPROCESSORS

INTEL 8008

- Introduced in 1972.
- 8008 was designed for a computer terminal.
- It was first 8-bit µP.
- Its clock speed was 500 KHz.
- Could execute 50,000 instructions per second.



8-BIT MICROPROCESSORS

INTEL 8080

- Introduced in 1974.
- It was also 8-bit µP.
- Its clock speed was 2 MHz.
- It had 6,000 transistors.
- Was 10 times faster than 8008.
- Could execute 5,00,000 instructions per second.

8-BIT MICROPROCESSORS

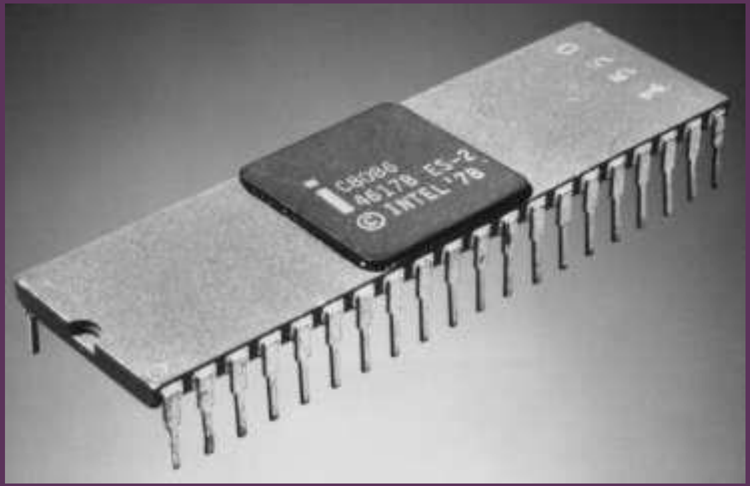
INTEL 8085

- Introduced in 1976.
- It was also 8-bit µP.
- Its clock speed was 3 MHz.
- Its data bus is 8-bit and address bus is 16-bit.
- It had 6,500 transistors.
- Could execute 7,69,230 instructions per second.
- It could access 64 KB of memory.
- It had 246 instructions.

16-BIT MICROPROCESSORS

INTEL 8086

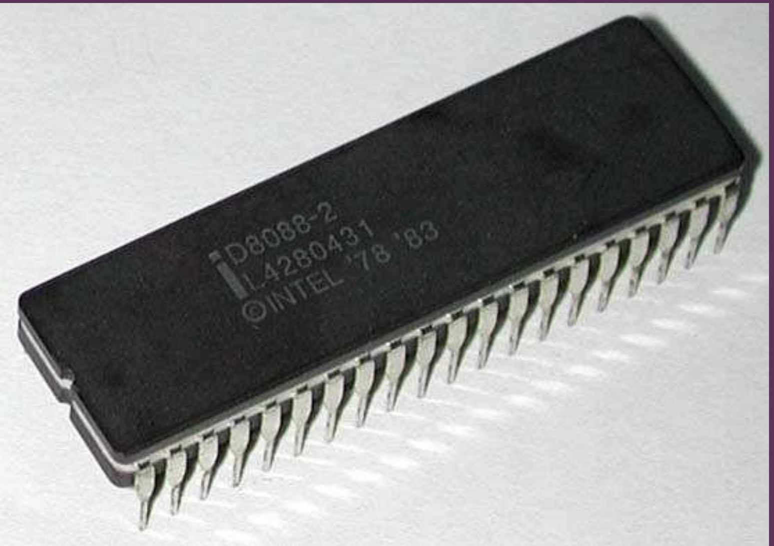
- Introduced in 1978. It was first 16-bit µP.
- Its clock speed is 4.77 MHz, 8 MHz and 10 MHz, depending on the version.
- Its data bus is 16-bit and address bus is 20-bit.
- It had 29,000 transistors.
- Could execute 2.5 million instructions per second.
- It could access 1 MB of memory.
- It had 22,000 instructions.
- It had Multiply and Divide instructions.



16-BIT MICROPROCESSORS

INTEL 8088

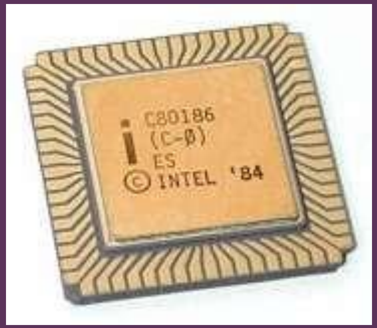
- Introduced in 1979.
- It was also 16-bit µP.
- It was created as a cheaper version of Intel's 8086.
- It was a 16-bit processor with an 8-bit external bus.

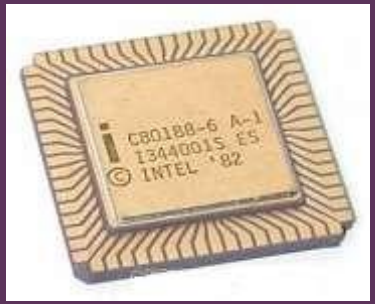


16-BIT MICROPROCESSORS

INTEL 80186 & 80188

- Introduced in 1982.
- They were 16-bit µPs.
- Clock speed was 6 MHz.





16-bit Microprocessors

INTEL 80286

- Introduced in 1982.
- It was 16-bit µP.
- Its clock speed was 8 MHz

32-bit Microprocessors

INTEL 80386

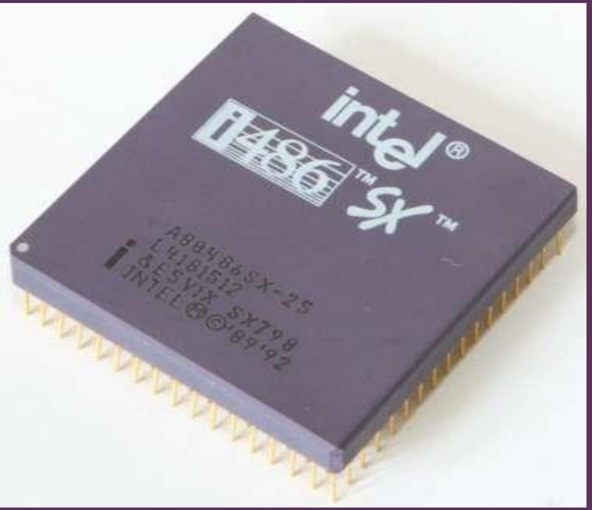
- Introduced in 1986.
- It was first 32-bit µP.
- Its data bus is 32-bit and address bus is 32-bit.
- It could address 4 GB of memory.

32-BIT MICROPROCESSORS

INTEL 80486

- Introduced in 1989.
- It was also 32-bit μP.
- It had 1.2 million transistors.
- Its clock speed varied from 16 MHz to 100 MHz

  depending upon the various versions



32-bit Microprocessors

INTEL PENTIUM

- Introduced in 1993.
- It was also 32-bit µP.
- It was originally named 80586.
- Its clock speed was 66 MHz.

32-bit Microprocessors

INTEL PENTIUM PRO

- Introduced in 1995.
- It was also 32-bit µP.

32-BIT MICROPROCESSORS

INTEL PENTIUM II

- Introduced in 1997.
- It was also 32-bit µP.

32-BIT MICROPROCESSORS

INTEL PENTIUM II XEON

- Introduced in 1998.
- It was also 32-bit µP.

32-BIT MICROPROCESSORS

INTEL PENTIUM III

- Introduced in 1999.
- It was also 32-bit µP.

32-BIT MICROPROCESSORS

INTEL PENTIUM IV

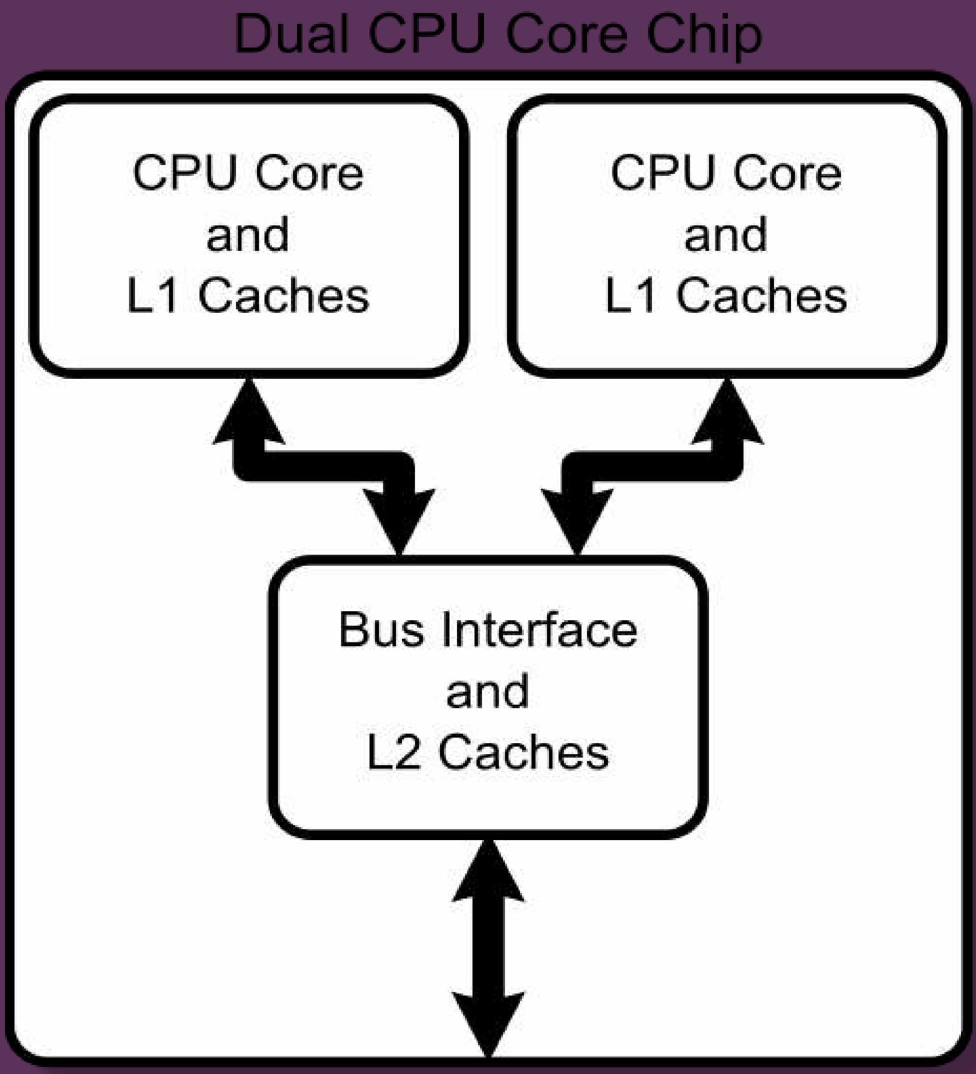
- Introduced in 2000.
- It was also 32-bit µP.

32-BIT MICROPROCESSORS

INTEL DUAL CORE

- Introduced in 2006.
- It is 32-bit or 64-bit µP.
- It has two cores.
- Both the cores have there own internal bus

  and L1 cache, but share the external bus and L2 cache



64-bit Microprocessors

Evolution of Microprocessors

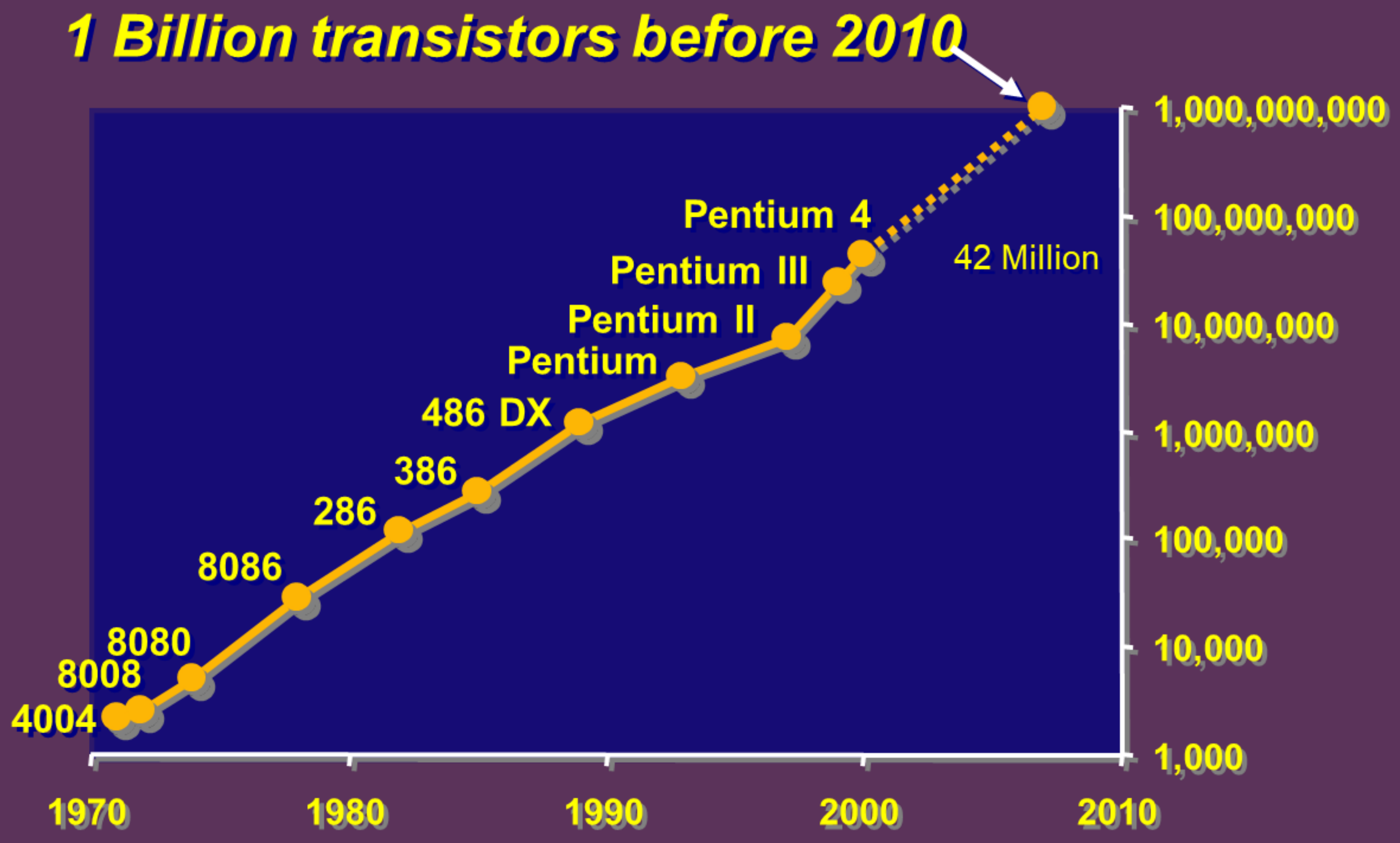
DIP: Dual Inline Package
PGA: Pin Grid Array
MCM: Multichip Module
SECC: Single Enge Contact Cartridge

Table 4.19 History of Intel microprocessors over three decades

Processor	Year	Feature Size (µm)	Transistors	Frequency (MHz)	Word size	Package
4004	1971	10	2.3k	0.75	4	16-pin DIP
8008	1972	10	3.5k	0.5–0.8	8	18-pin DIP
8080	1974	6	6k	2	8	40-pin DIP
8086	1978	3	29k	5–10	16	40-pin DIP
80286	1982	1.5	134k	6–12	16	68-pin PGA
Intel386	1985	1.5–1.0	275k	16–25	32	100-pin PGA
Intel486	1989	1–0.6	1.2M	25–100	32	168-pin PGA
Pentium	1993	0.8–0.35	3.2–4.5M	60–300	32	296-pin PGA
Pentium Pro	1995	0.6–0.35	5.5M	166–200	32	387-pin MCM PGA
Pentium II	1997	0.35–0.25	7.5M	233–450	32	242-pin SECC
Pentium III	1999	0.25–0.18	9.5–28M	450–1000	32	330-pin SECC2
Pentium 4	2001	0.18–0.13	42–55M	1400–3200	32	478-pin PGA

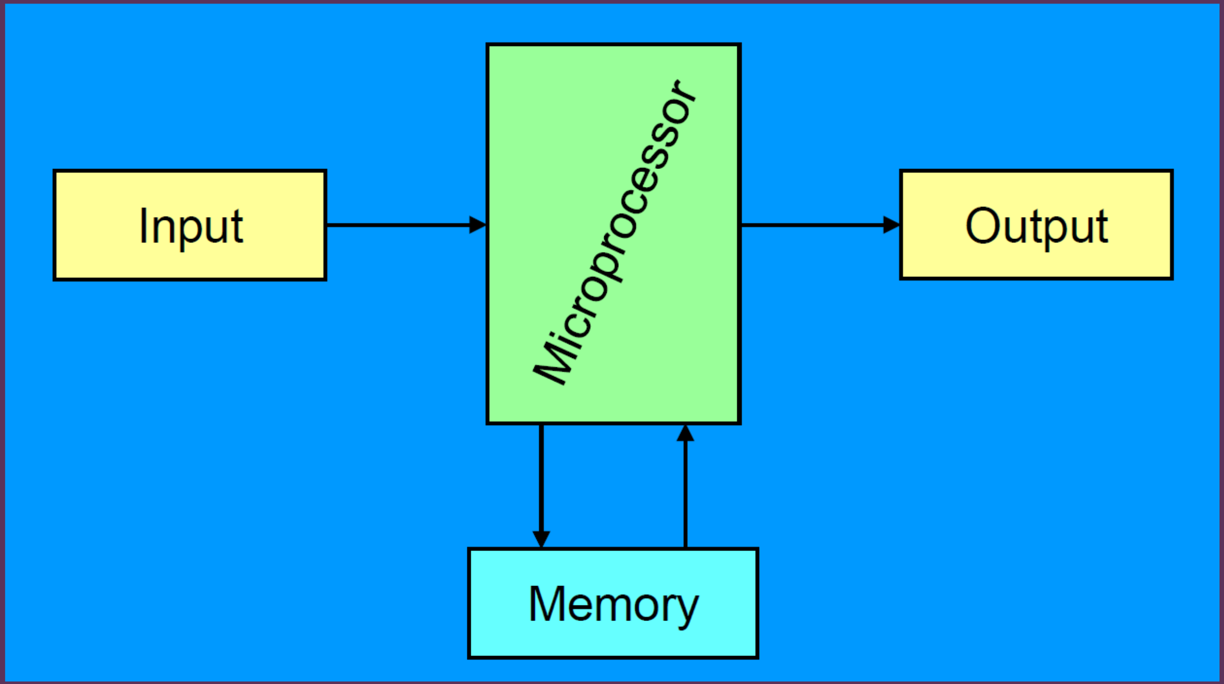
Moore's Law

The number of transistors on a chip will double every 18-24 months



A Microprocessor-based system

- The following block diagram is generally used to represent a microprocessor-based system:

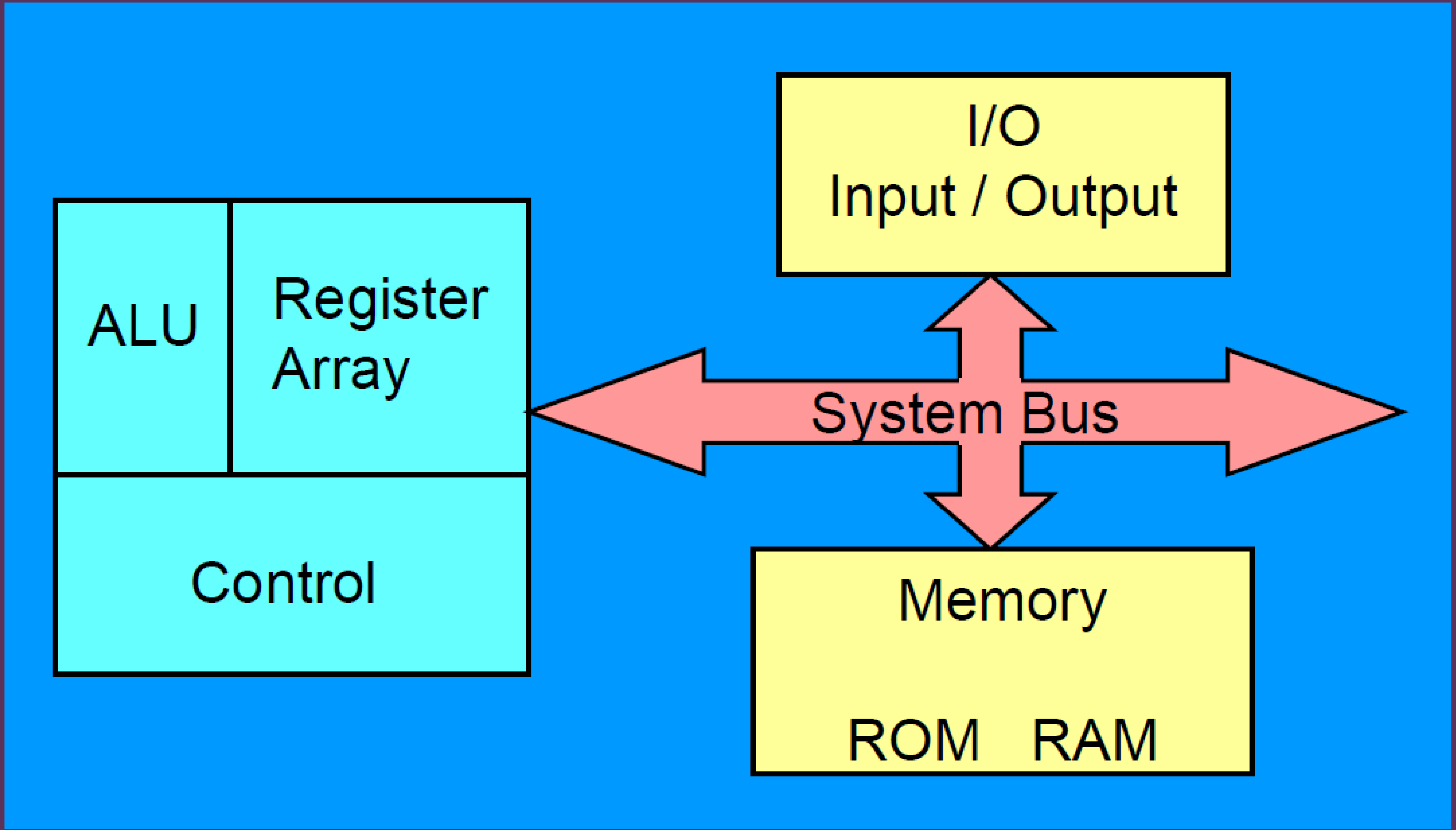


Inside The Microprocessor

- Internally, the microprocessor is made up of 3 main units.
    - The Arithmetic/Logic Unit (ALU)
    - The Control Unit.
    - An array of registers for holding data while it is being manipulated.

A microprocessor-based system

- The Microprocessor based system (single board microcomputer) consists of microprocessor as CPU, semiconductor memories like EPROM and RAM, input device, output device and interfacing devices.



A microprocessor-based system

- The work done by the processor can be classified into the following three groups.

1. Work done internal to the processor.
   - addition, subtraction, logical operations, data transfer operations, etc.

2. Work done external to the processor.
   - reading/writing the memory and reading/writing the J/O devices or the peripherals.

3. Operations initiated by the slaves or peripherals.
   - If the peripheral requires the attention of the master then it can interrupt the master and initiate an operation.

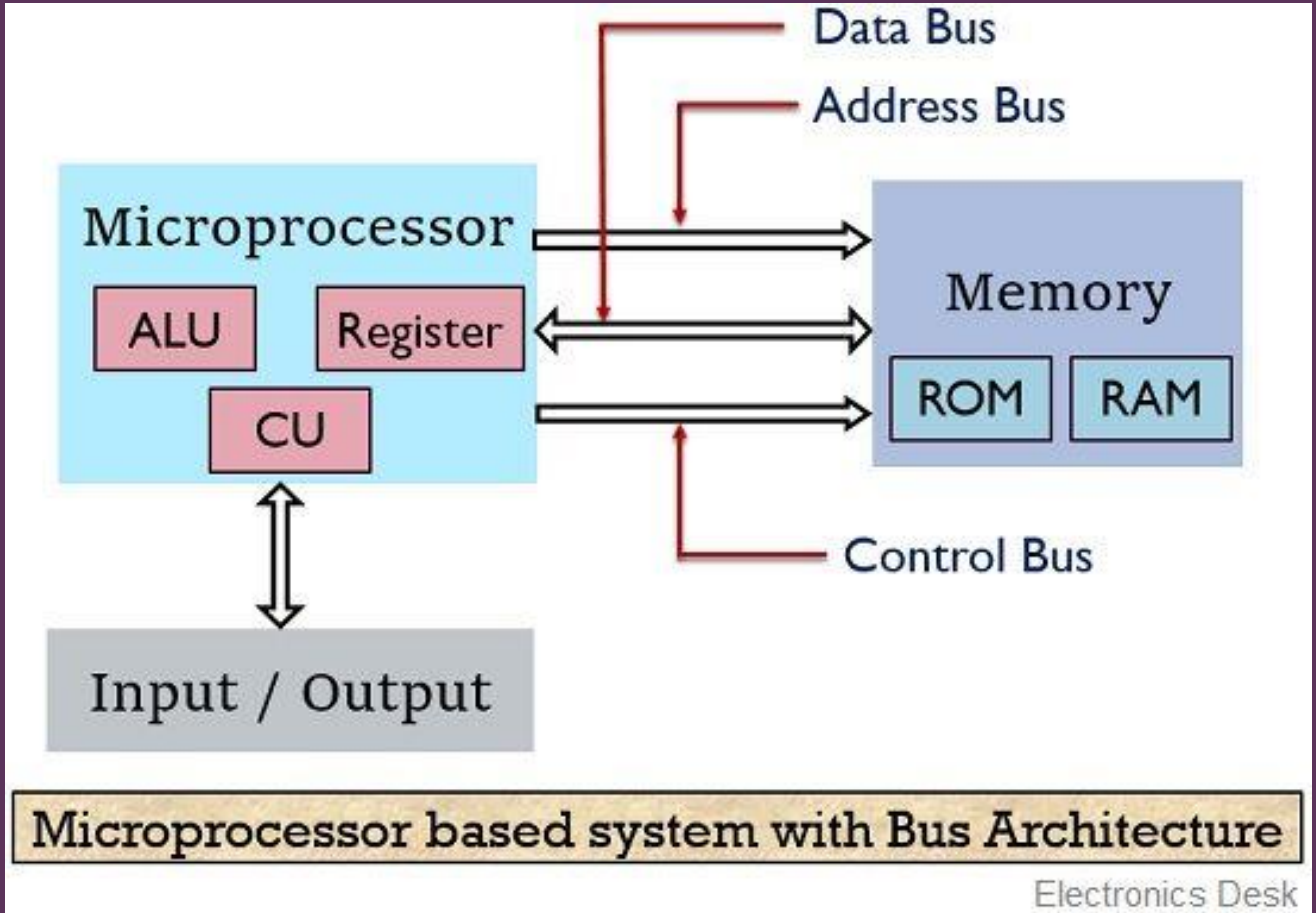
A microprocessor-based system

- Three types of Busses
  - ➢ **Address Bus**

    (To identify location in Memory)
  - ➢ **Data Bus**

    (To transfer/carry Data)
  - ➢ **Control Bus**

    (Read or Write ?)



Memory

- Memory **stores information such as instructions and data** in binary format (0 and 1).
- It provides this information to the microprocessor whenever it is needed.
- Usually, there is a memory "sub-system" in a microprocessor-based system. This sub-system includes:
  - The registers inside the microprocessor
  - **Read Only Memory (ROM):** used to store information that does not change.
  - **Random Access Memory (RAM)** also known as Read/Write Memory: used to store information supplied by the user. Such as programs and data.

Memory

- To execute a program:
  - The user enters its instructions in binary format into the memory.
  - The microprocessor then reads these instructions and whatever data is needed from memory, executes the instructions and places the results either in memory or produces it on an output device.

The three cycle instruction execution model

- To execute a program, the microprocessor "reads" each instruction from memory, "interprets" it, then "executes" it.

- The full instruction cycle use the right names for the cycles:
  - The microprocessor **fetches** each instruction from Memory,
  - **Decodes** it (as per the Opcode. Here digital Decoder is used),
  - Then **executes** it.

- This sequence is continued until all instructions are performed.

Machine Language

- The number of bits that form the “word” of a microprocessor is fixed for that particular processor.
  - These bits define a maximum number of combinations.
  - e.g. an 8-bit microprocessor can have at most $2^8 = 256$ different combinations.
- However, in most microprocessors, not all of these combinations are used.
  - Certain patterns are chosen and assigned specific meanings.
  - Each of these patterns forms an instruction for the microprocessor.
  - The complete set of patterns makes up the microprocessor’s machine language.

8085 Machine Language

- The 8085 (from Intel) is an 8-bit microprocessor uses a total of 246 bit patterns to form its instruction set.
  - These 246 patterns represent only 74 instructions.
  - The reason for the difference is that some (actually most) instructions have multiple different formats.
- Because it is very difficult to enter the bit patterns correctly, they are usually entered in hexadecimal instead of binary.
  - For example, the combination 0011 1100 which translates into "increment the number in the register called the accumulator", is usually entered as 3C.

Assembly Language

- Entering the instructions using hexadecimal is quite easier than entering the binary combinations.
- However, it still is difficult to understand what a program written in hexadecimal does.
  - So, each company defines a symbolic code for the instructions.
  - These codes are called "mnemonics".
  - The mnemonic for each instruction is usually a group of letters that suggest the operation performed.

Assembly Language

- Using the same example from before,
  - 00111100 translates to 3C in hexadecimal (OPCODE)
  - Its mnemonic is: "INR A".
  - INR stands for "increment register" and A is short for accumulator.
- Another example is: 1000 0000,
  - Which translates to 80 in hexadecimal.
  - Its mnemonic is "ADD B".
  - "Add register B to the accumulator and keep the result in the accumulator".

Assembly Language

- It is important to remember that a machine language and its associated assembly language are completely machine dependent.
- In other words, they are not transferable from one microprocessor to a different one.

- A program called an "**assembler**", translates assembly language into machine language.

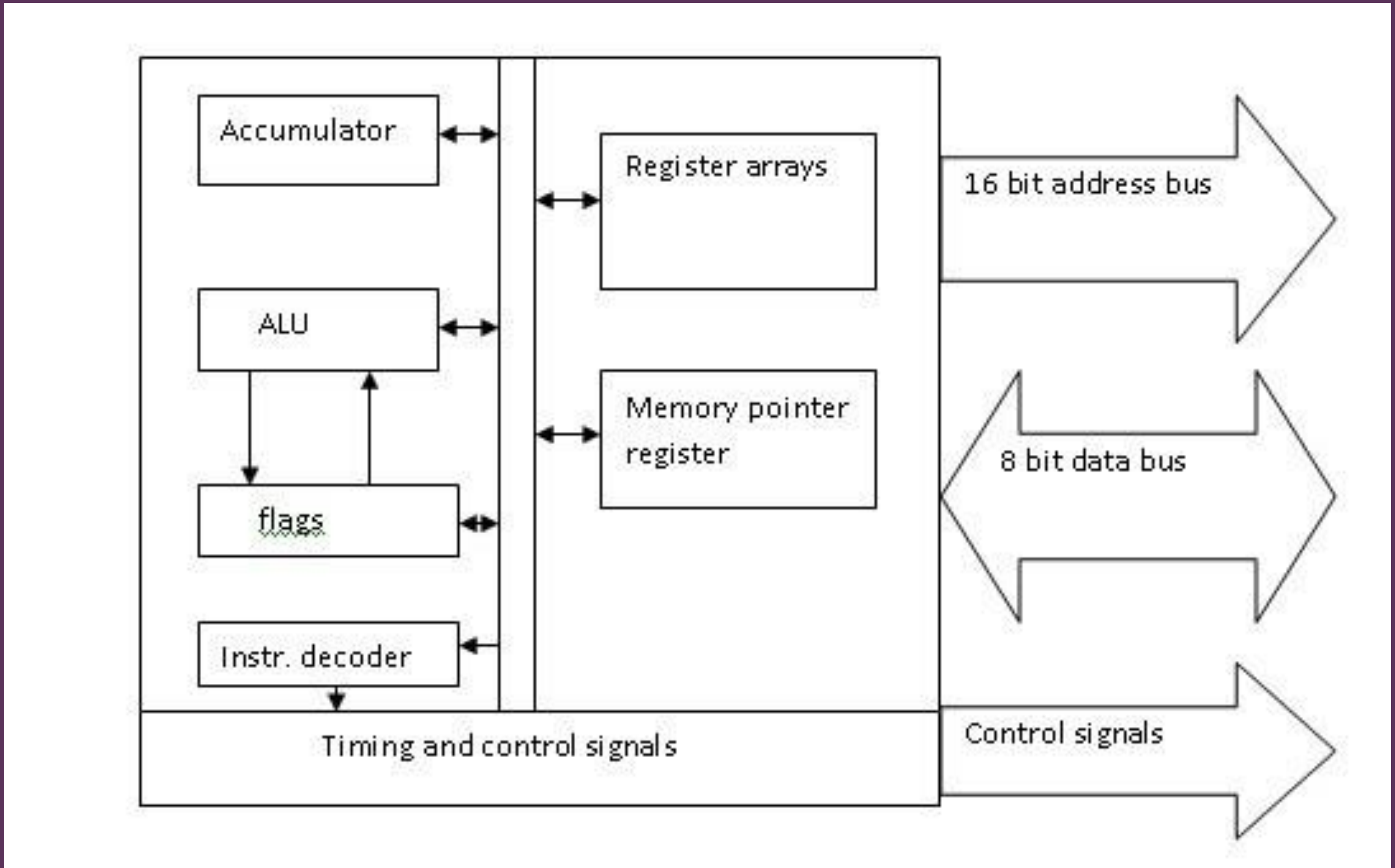
Example of 8085 based systems

- Nowadays, microprocessor can be seen in almost all types of electronics devices like mobile phones, printers, washing machines etc.
- Microprocessors are also used in advanced applications like radars, satellites and flights.
- Due to the rapid advancements in electronic industry and large scale integration of devices results in a significant cost reduction and increase application of microprocessors and their derivatives.

8085 Microprocessor Architecture

- 8-bit general purpose µp
- It has 16 address lines, hence it can access (2^{16}) 64 KBytes of memory
- Has 40 pins
- Requires +5 v power supply
- Can operate with 3 MHz clock
- 8085 upward compatible

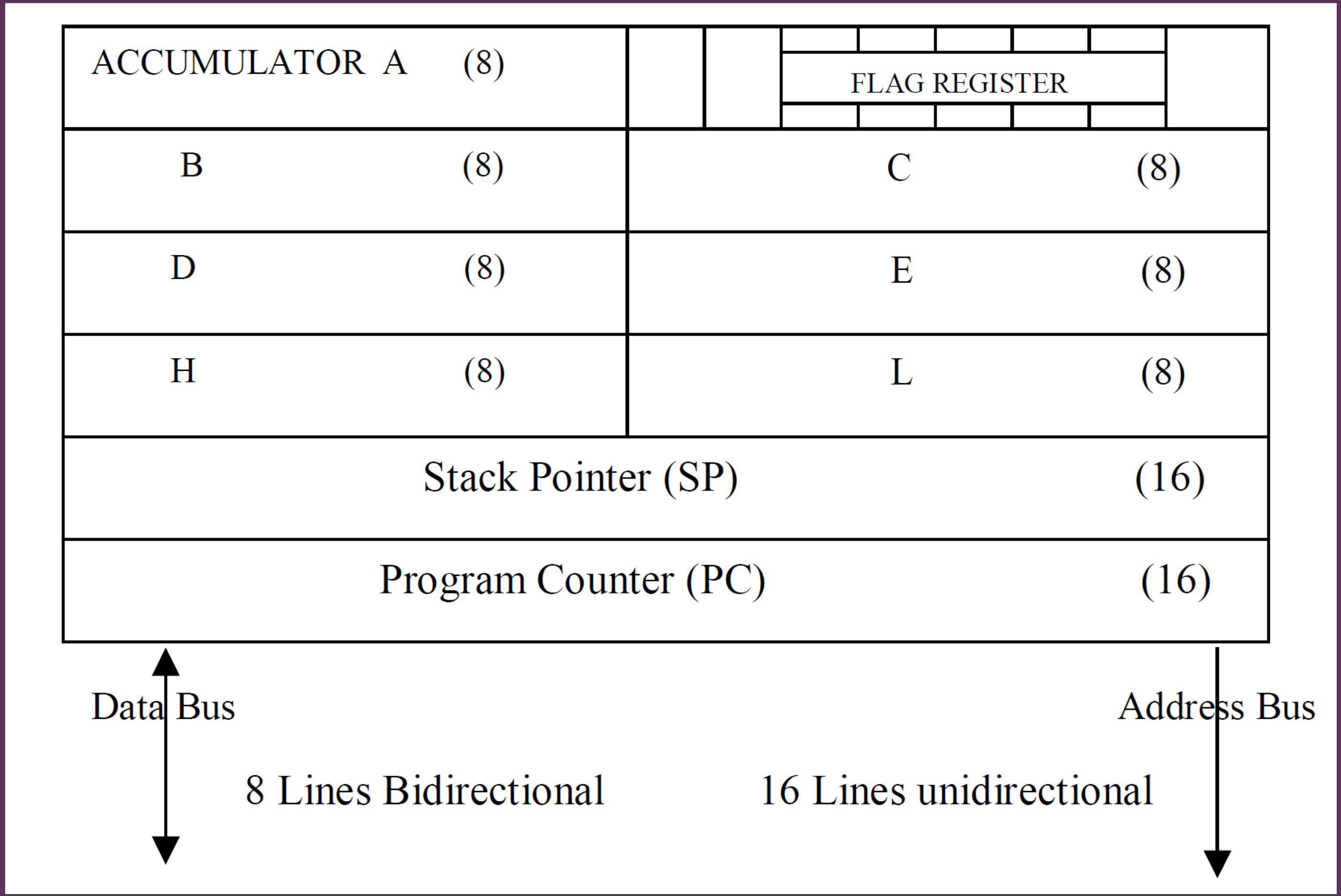
8085 Hardware Model



8085 Hardware Model

- The hardware model shows two major segments:
  - One segment includes arithmetic logic unit [ALU] and an 8 bit register called an accumulator, instruction decoder, and flags.
  - The second segment shows 8 bit and 16 bit registers.
- Both segments are connected with various internal connections called an internal bus.
- The arithmetic and logic operations are performed in the ALU.
- Results are stored in the accumulator, and flip-flops, called flags, are set or reset to reflect the results.
- There are 3 buses- a 16 bit unidirectional address bus, an 8 bit bidirectional data bus, and a control bus.

8085 Programming Model



8085 Programming Model

- The Programming model consists of some segments of the ALU and the registers.
- The 8085 programming model includes six registers, one accumulator, and one flag register.
- In addition, it has two 16-bit registers: the stack pointer and the program counter.

8085 Programming Model

Registers

- The 8085 has **six general - purpose registers** to store 8-bit data
  - B, C, D, E, H and L.
- They can be combined as register pairs - BC, DE, and HL - to perform some 16-bit operations.
- The programmer can use these registers to store or copy data into the registers by using data copy instructions.

8085 Programming Model

Accumulator

- The accumulator is an 8-bit register that is a part of arithmetic/logic unit (ALU).
- This register is used to store 8-bit data and to perform arithmetic and logical operations.
- The result of an operation is stored in the accumulator.
- The accumulator is also identified as register A.

8085 Programming Model

Flags

- **Flag Register** has five 1-bit flags.
- The ALU includes five flip-flops, which are set or reset after an operation according to data conditions of the result in accumulator and other registers.
- They are called Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags.
- The microprocessor uses these flags to test data conditions.

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

8085 Programming Model

Flags

- **Sign** - set if the most significant bit of the result is set.
  - Used for indicating the sign of the data in the accumulator
  - The sign flag is set if negative (1 –negative)
  - The sign flag is reset if positive (0 –positive)
- **Zero** - Is set if result obtained after an operation is 0.
- **Auxiliary carry** - This flag is set if there is an overflow out of bit-3 i.e., carry from lower nibble to higher nibble (D3 to D4 bit). This flag is used for BCD operations.

8085 Programming Model

Flags

- **Parity** - Parity is defined by the number of ones present in the accumulator.
    - Is set if parity is even
    - Is cleared if parity is odd
- **Carry** - set if there was a carry during addition, or borrow during subtraction/comparison/rotation.

8085 Programming Model

Program Counter (PC)

- This 16-bit register deals with sequencing the execution of instructions.
- This register is a memory pointer.
- The microprocessor uses this register to sequence the execution of the instructions.
- The function of the program counter is to point to the memory address from which the next byte is to be fetched.

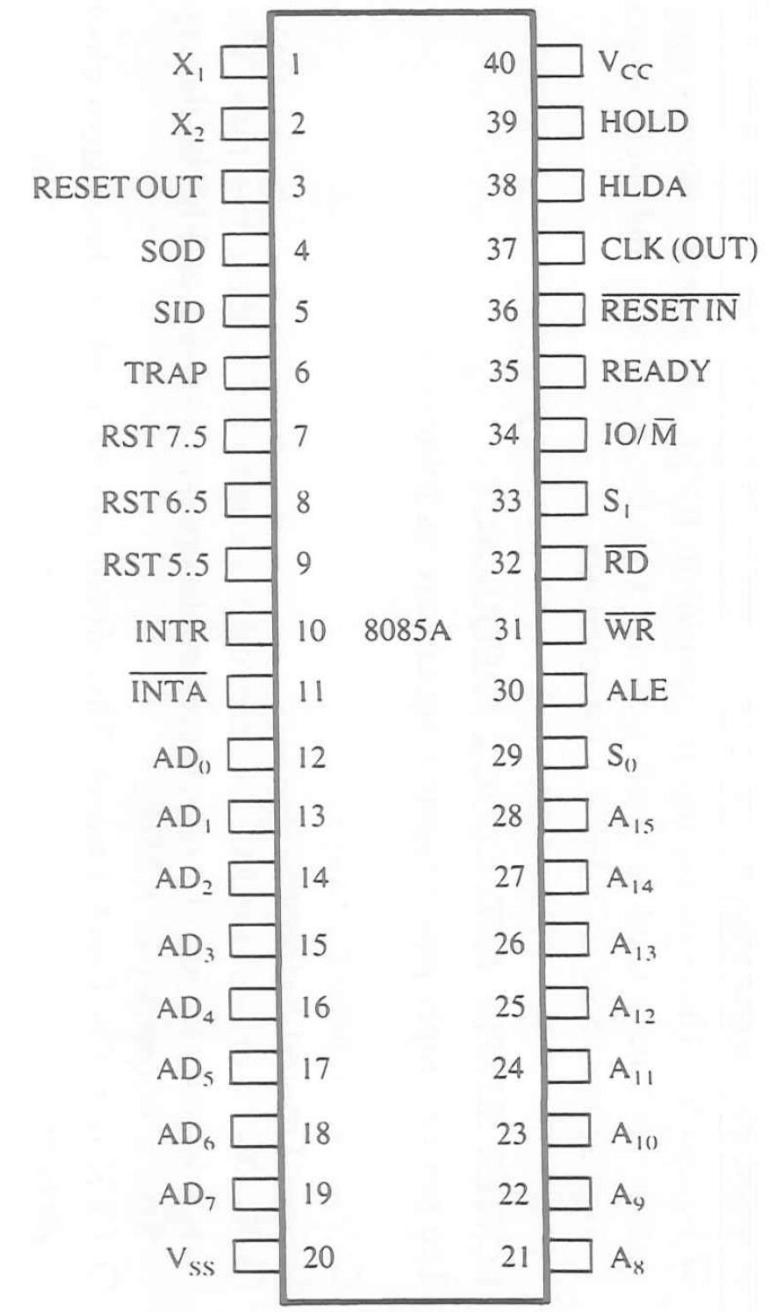
8085 Programming Model

The Stack pointer

- The stack pointer is also a 16-bit register that is used to point into memory.
- The memory this register points to is a special area called the stack.
- **The stack is an area of memory used to hold data that will be retreived soon.**
- The stack is usually accessed in a Last In First Out (LIFO) fashion.

8085

PIN Diagram



8085 Pinout

8085 PIN Diagram

The signals of 8085 can be classified into six groups according to their functions:

a) Address bus

b) Data bus

c) Control and status signals

d) Power supply and frequency signals

e) Externally initiated signals and

f) Serial I/O ports.

8085 PIN Diagram

ADDRESS BUS

- The 8085 has eight lines, **A_{15}-A_8**,which are unidirectional and used as the high order address bus.

MULTIPLEXED ADDRESS / DATA BUS

- The signal lines **AD_7-AD_0** are bidirectional: they serve a dual purpose.
- In executing an instruction, during the earlier part of the cycle, these lines are used as the low-order address bus.
- During the later part of the cycle, these lines are used as the data bus. (This is also known as multiplexing the bus.)
- The low-order address bus can be separated from these signals by using a latch.

8085 PIN Diagram

CONTROL AND STATUS SIGNALS

- This group of signals includes:
  - two control signals (RD and WR),
  - three status signals (IO/M, S_1, and S_0) to identify the nature of the operation,
  - one special signal (ALE) to indicate the beginning of the operation.

8085 PIN Diagram

CONTROL AND STATUS SIGNALS

Signal description:

- **RD’** — Read: This read control signal (active low). This signal indicates that the selected I/O or memory device is to be read and data are available on the data bus.
- **WR’** — Write: This is a write control signal (active low). This signal indicates that the data on the data bus are to be written on into a selected memory device or I/O location.

8085 PIN Diagram

CONTROL AND STATUS SIGNALS

Signal description:

- **IO/M'** : This is status signal used to differentiate between I/O and memory operations.
  - When it is high, it indicates an I/O operation;
  - when it is low, it indicates a memory operation.
  - This signal combines with RD and WR to generate I/O and memory control signals.
- **S_1 and S_0**: These are status signals. They distinguish the various types of operations such as halt, reading, instruction fetching or writing.

8085 PIN Diagram

CONTROL AND STATUS SIGNALS

Signal description:

- **S_1 and S_0**: These are status signals. They distinguish the various types of operations such as halt, reading, instruction fetching or writing.

IO/M′	S1	S0	Data Bus Status
0	1	1	Opcode fetch
0	1	0	Memory read
0	0	1	Memory write
1	1	0	I/O read
1	0	1	I/O write
1	1	1	Interrupt acknowledge
0	0	0	Halt

8085 PIN Diagram

CONTROL AND STATUS SIGNALS

Signal description:

- **ALE —** Address Latch Enable: This is positive going pulse generated every time the 8085 begins an operation (machine cycle);
- It indicates that the bits on AD_7-AD_0 are address bits.
- This is used primarily to latch the low-order address from the multiplexed bus and generate a separate set of eight address lines, A_7-A_0.

8085 PIN Diagram

POWER SUPPLY AND CLOCK FREQUENCY

- **V_{cc}**: +5V power supply.
- **V_{ss}**: Ground reference.
- **X_1, X_2**: A crystal is connected at these two pins. The frequency is internally divided by two; therefore, to operate a system at 3MHz and, the crystal should have frequency of 6MHz.
- **CLK (OUT)** — Clock out: This signal is used as the system clock for other devices.

8085 PIN Diagram

EXTERNALLY INITIATED SIGNALS

- The 8085 has **five interrupt** signals that can be used to interrupt a program execution. These are:

1. Interrupt Request (INTR)
    - INTR(input)- Interrupt Request: This is used as a general-purpose interrupt.
    - INTA(Output)-Interrupt Acknowledge: This is used to acknowledge an interrupt.
2. Restart Interrupts (RST5.5, RST 6.5, RST7.5)
    - These are vectored interrupts and transfer the program control to specific memory locations.
    - They have higher priorities than the INTR interrupt.
    - Among these three, the priority order is 7.5, 6.5 and 5.5.

8085 PIN Diagram

EXTERNALLY INITIATED SIGNALS

- The 8085 has **five interrupt** signals that can be used to interrupt a program execution. These are:

3. TRAP
    - This is a non maskable interrupt and has the highest priority.

- In addition to the interrupts, three pins — RESET, HOLD, and READY — accept the externally initiated signals as inputs.
- RESET IN: When the signal on this pin goes low, the program counter is set to zero, the buses are tri-stated, and the microprocessor is reset.
- RESET OUT: This signal indicates that the microprocessor is being reset. The signal can be used to reset other devices.

8085 PIN Diagram

EXTERNALLY INITIATED SIGNALS

- HOLD(input): This signal indicates that a peripheral such as a DMA controller is requesting the use of the address and data buses.

- HLDA-Hold Acknowledge: This signal acknowledges the HOLD request.

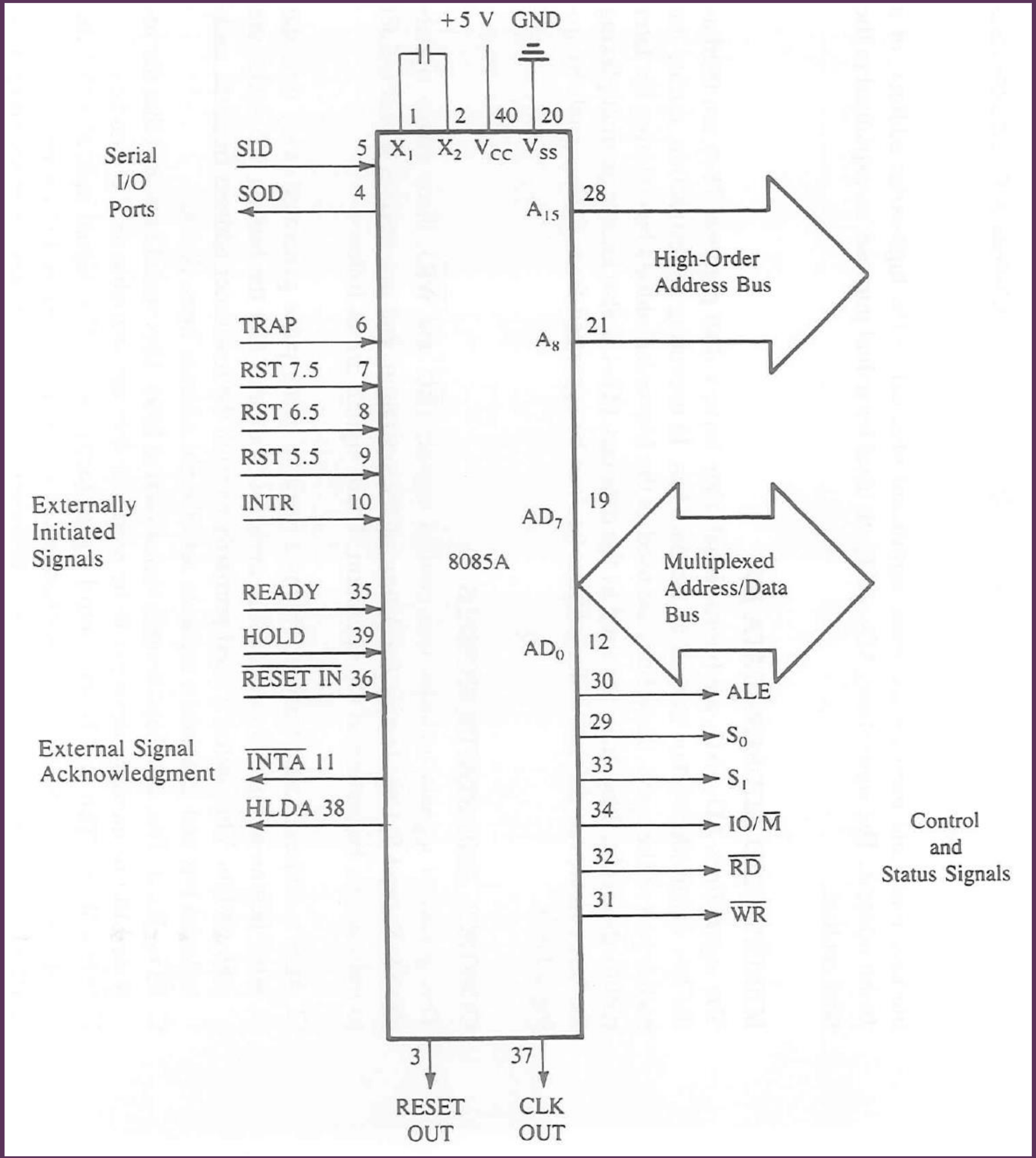
- READY: This signal is used to delay the microprocessor Read or Write cycles until a slow-responding peripheral is ready to send or accept data.
- When this signal goes low, the microprocessor waits for an integral number of clock cycles until it goes high.

8085 PIN Diagram

SERIAL I/O PORTS

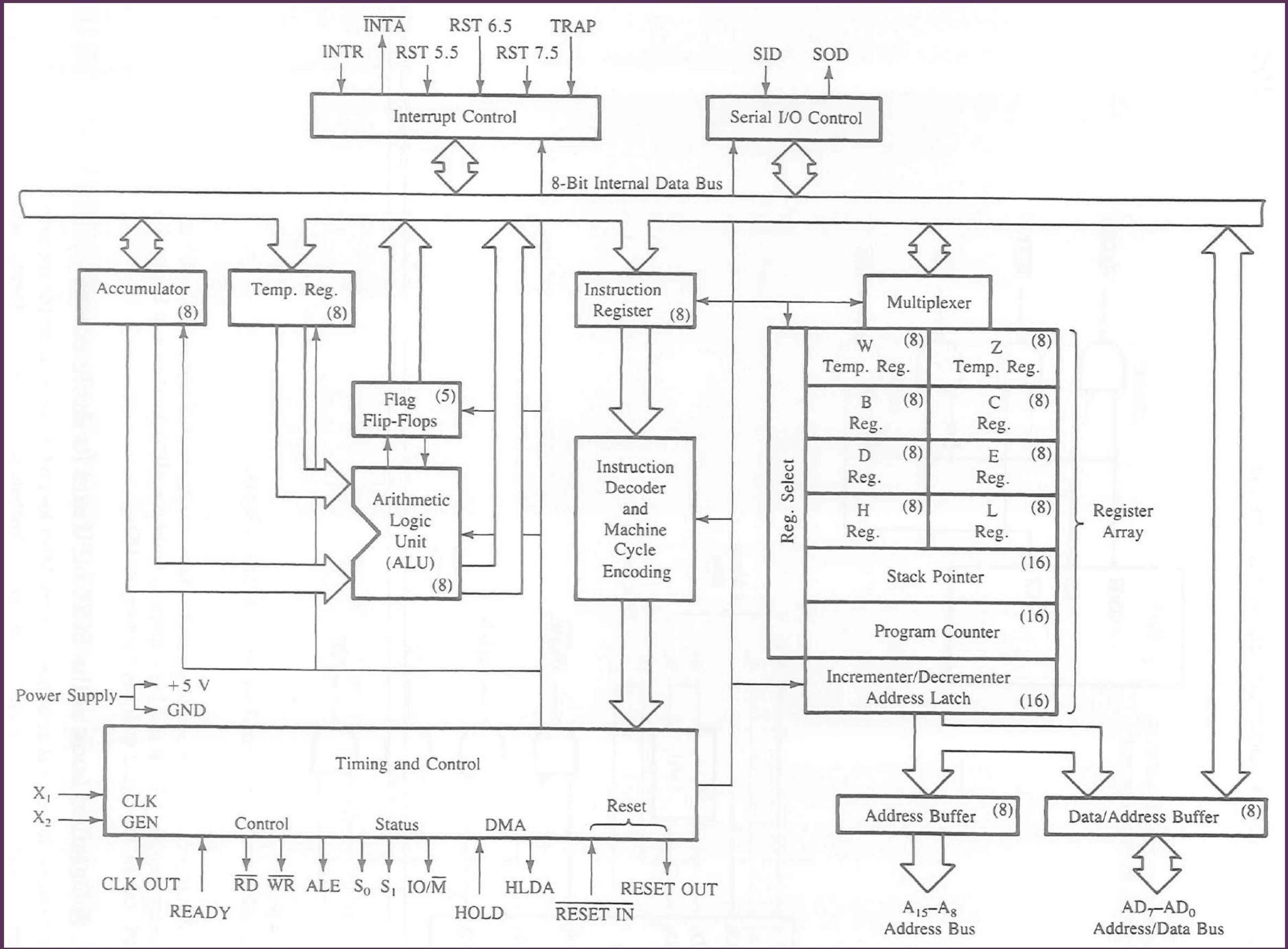
- The microprocessor has 2 pins specially designed for software –controlled serial I/O:
  - one is called SOD (Serial Output Data) and
  - the other is called SID (Serial Input Data).
- Data transfer is controlled through 2 instructions: SIM (Set Interrupt Mask) and RIM (Read Interrupt Mask).
- The instruction SIM is necessary to output data serially from the SOD line.
- Similarly instruction RIM is used to input serial data through the SID line.
- The SID and SOD lines in the 8085 eliminate the need for an input port and an output port in the software-controlled serial I/O.
- Essentially, the SID is a 1-bit port and SOD is a 1- bit output port.

8085 PIN Diagram with Grouping



8085

A R C H I T E C T U R E



8085 ARCHITECTURE

It consists of various functions blocks as listed below:

1) Registers

2) Arithmetic and logic unit

3) Instruction decoder and machine cycle encoder

4) Address Buffer

5) Address / Data Buffer

6) Incrementer / Decremented address batch

7) Serial I/O control

8) Timing and control circuitry

8085 ARCHITECTURE

Arithmetic and logic unit

- The 8085’s ALU performs arithmetic and logical functions on eight bit variables.
- The arithmetic unit performs bitwise fundamental, operation such as addition and subtraction.
- The logic unit performs the logical operations such as complement, AND, OR and EX-OR as well as rotate and clear.
- The ALU also looks after branching decisions.

8085 ARCHITECTURE

Instruction decoder and machine cycle encoder

- The processor first fetches the opcode of instruction from memory and stores this opcode in the instruction register.
- It is then sent to the instruction decoder which decodes it and accordingly gives the timing and control signals which control the register, data buffer, ALU, and external peripheral signals.
- The 8085 executes seven different types of machine cycles (Opcode Fetch Cycle, Memory Read, Memory Write, I/O Read, I/O Write, Interrupt Acknowledge, Bus Idle).
- It gives the information about which machine cycle is currently executing in the encoded.

8085 ARCHITECTURE

Address Buffer

- This is an 8-bit unidirectional buffer.
- It is used to drive external high order address (A_{15}-A_8).
- It is also used to tri-state the high order address bus under certain conditions such as reset, hold, halt and when address lines are not in use.

Address / Data buffer

- This is an 8-bit bidirectional buffer.
- It is used to drive multiplexed address/data bus i.e. low order address bus (A_7-A_0) and data bus (D_7-D_0).
- It is also used to tri-state the multiplexed address/data bus under certain conditions such as reset, hold, halt and when bus is not in use.

8085 ARCHITECTURE

Incrementer/Decrementer address latch

- This 16-bit register is used to increment or decrement the contents of program counter or stack pointer as a part of execution of instructions related to them.

8085 ARCHITECTURE

Interrupt Control

- The processor fetches, decodes and executes the instructions in a sequence.
- Sometimes it is necessary to have the processor automatically execute one of a collection of special routines whenever special condition exists within a program or the microcomputer system.
- After the execution of special routine, the program control must be transferred to the program which processor was executing before the occurrence of the special condition.
- The occurrence of this special condition is referred as interrupt.
- The interrupt control block has five interrupt inputs RST 5.5, RST 6.5, RST 7.5, TRAP and INTR and one acknowledge signal INTA.

8085 ARCHITECTURE

Serial I/O Control

- In situations like, data transmission over long distance and communication with cassette tapes or CRT terminal, it is necessary to transmit data bit by bit to reduce the cost of cabling.
- In serial communication one bit is transferred at a time over a single line.
- The 8085's serial I/O control provides two lines, SID and SOD for serial communication.
- The Serial Output Data (SOD) line is used to send data serially and Serial Input Data (SID) line is used to receive data serially.

8085 ARCHITECTURE

Timing and Control circuitry

- The control circuitry in the processor 8085 is responsible for all the operations.
- The control circuitry and hence the operations in 8085 are synchronized with the help of clock signal.
- Along with the control of fetching and decoding operations and generating appropriate signals for instruction execution, control circuitry, also generates signals required to interface external devices to the processor 8085.

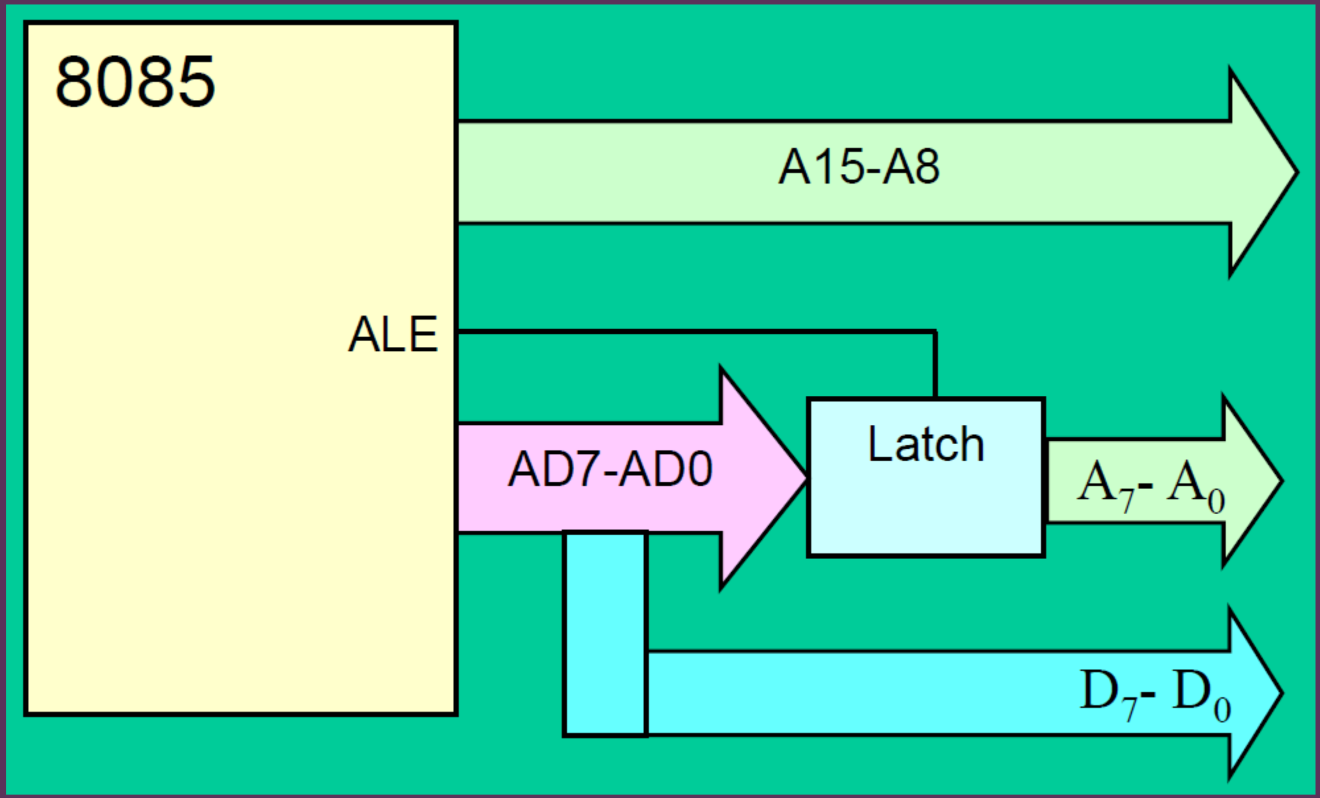
The Address and Data Busses

- The address bus has 8 signal lines A8 –A15 which are unidirectional.
- The other 8 address bits are multiplexed (time shared) with the 8 data bits.
  - So, the bits AD0 –AD7 are bi-directional and serve as A0 –A7 and D0 –D7 at the same time.
    - During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits.
  - In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes..

Demultiplexing AD7-AD0

- The high order bits of the address remain on the bus **for three clock periods**.
- However, the low order bits remain for **only one clock period** and they would be lost if they are not saved externally.
- Also, notice that the low order bits of the address disappear when they are needed most.
- To make sure we have the entire address for the full three clock cycles, we will **use an external latch** to save the value of AD7–AD0 (in some temporary register) when it is carrying the address bits.
- We use the ALE signal to enable this latch.

Demultiplexing AD7-AD0



Demultiplexing AD7-AD0

- A15-A8 continually hold higher order address for three clock cycles.
- In the first clock cycle of memory access, the bits are put on AD0 to AD7 and simultaneously the ALE signal is generated.
- Externally an 8-bit latch is used to store values of A7 to A0 into this latch.
- After the first clock cycle, the ALE signal will be taken off. As result, whatever the value is latched in first clock cycle will remain here for next two clock cycles.
- Given that ALE operates as a pulse during T1, we will be able to latch the address.
- Then when ALE goes low, the address is saved and the AD7–AD0 lines can be used for their purpose as the bi-directional data lines.