FUNDAMENTAL CONCEPTS

Microcomputer: A programmable machine that processes binary data.

There are three major parts of a microcomputer system

- **The CPU (Central Processing Unit)** which acts as the brain coordinating all Activities within the computer
  - i.e. The group of circuits that process data and provide control signals and timing.

- **The memory unit** where the program instructions and data are temporarily stored
- **The I/O (Input/Output) devices** which allow the computer to input information for processing and then output the result

Today the CPU circuitry has been reduced to ICs called the microprocessor,

**Microprocessor:**

Microprocessor is a multipurpose, programmable, clock-driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions, and provides as output.

**CPU (Central Processing Unit)**
- includes ALU, registers, control unit. etc.

  **ALU:** Arithmetic Logic unit- the group of circuits that perform arithmetic and logic-operations. The ALU is a part of the CPU.

  **Control unit:** The group of circuits that provides timing and signals to all operations in the computer and controls data flow. The basic timing of the computer is controlled by a square wave oscillator or a clock generator circuit.
Register unit: This area of CPU consists of various registers. The registers are used primarily to store data temporarily during the execution of a program. Registers can be viewed as a memory location where the memory resides within the CPU.

Scale of Integration

SSI- Small Scale Integration.
The process of designing a few circuits on a single chip. The term refers to the technology used to fabricate discrete logic gates on a chip.

MSI- Medium Scale Integration.
The process of designing more than hundreds of gates on a single chip

LSI- Large Scale Integration.
The process of designing more than a thousand gates on a single chip.

Similarly the term VLSI (Very Large Scale Integration) and SLSI (Super Scale Integration) are used to indicate the scale of integration.

Reprogrammable systems:
In Reprogrammable systems such as microcomputers, the microprocessor unit can be programmed according to the user’s need. Reprogrammable systems are capable of handling large data.

Embedded Systems:
In Embedded Systems, the microprocessor unit is not available for reprogramming to the end user. The user can select one of the many predefined operations supported by the Embedded System. Washing machines, Microwave ovens, Copy-Machines are examples of the Embedded Systems.

Programming a microprocessor:
A microprocessor is a programmable circuit. Microprocessor is programmable because it can be instructed to perform given tasks within its capability. Microprocessor is designed to understand and execute many binary instructions.
Programming a microprocessor means teaching all the steps necessary to complete a job. A microprocessor can be instructed by a user to furnish the following tasks:

1. To take data from the user
2. To modify the above data in a way the user wants
3. To return the modified data back to the user

**Instruction Set**

The list of all recognizable instructions by the instruction decoder is called the instruction set.

Based on instruction set, two types of computer systems:

CISC (Complex Instruction Set Computers), e.g., 80x86 family has more than 3000 instructions

RISC (Reduced Instruction Set Computers) - A small number of very fast executing instructions

**Three Bus System Architecture**

- A collection of electronic signals all dedicated to a particular task is called a *bus*

  - *address bus*
  - *data bus*
  - *control bus*

**Data Bus**

- The width of the data bus determines how much data the processor can read or write in one memory or I/O cycle.
- 8-bit microprocessor has an 8-bit data bus

**Address Bus**

- The address bus is used to identify the memory location or I/O device (also called port) the processor intends to communicate with.

- 20 bits for the 8086 and 8088
- 32 bits for the 80386/80486 and the Pentium processors

- The total number of memory locations addressable by a given CPU is always equal to $2^N$ where N is the number of address bits, regardless of the data bus.
Control Bus
Control Bus is used to transfer control signals. Typical control signals are
– Memory Read
– Memory Write
– I/O Read
– I/O Write

• Control and address lines are output lines only but the data bus is bidirectional

Some Important Terminology
• Bit is a binary digit that can have the value 0 or 1
• A byte is defined as 8 bits
• A nibble is half a byte
• A word is two bytes
• A double word / long word is four bytes.
Memory :

**Standard Memory:**  
Memory devices which are not connected to user devices or I/O devices are treated as Standard Memory. For Example, ROM, R/W Memory etc.

**Port Memory:**  
Port Memory is a memory that is connected to user or I/O devices.

**Interface controller:**

I/O devices are not directly connected to the microprocessor BUS because, the I/O devices are slower in compared to the speed of the microprocessor. Connecting the I/O devices (Peripherals) directly to the microprocessor BUS will cause the MP to wait for the peripheral until the peripheral manages to send data to the Microprocessor (for example, Keyboard) or the peripheral manages to receive data from the Microprocessor (for example, display device). The microprocessor can’t do any other operation as long as the Read/Write operation with the peripherals in progress.

Interface controllers are devices that are placed in between the peripheral and the microprocessor providing speed synchronization between the microprocessor and the I/O devices. In Fig, M1, M3 are interface controllers.
Absolute address of a memory chip:

The address assigned to the memory locations of a memory chip when it is not connected to any system is called the absolute address. The starting address always starts with ‘0’.

System address of a memory chip:

The address assigned to the memory locations of a memory chip while it is connected to a system is called the System address. In this scheme, two memory location cannot have the same system address.
Interfacing memories using Memory Decoder:

Fig: Microcomputer System using Memory Decoder
<table>
<thead>
<tr>
<th>Upper A3 A2</th>
<th>Lower A1 A0</th>
<th>Memory Selected</th>
<th>Memory locations on a particular memory chip</th>
<th>Chip select line enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>M0</td>
<td>0-3H</td>
<td>CS0/</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>M0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>1 0</td>
<td>M0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>1 1</td>
<td>M0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>M1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
<td>M1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>1 0</td>
<td>M1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>1 1</td>
<td>M1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>M2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>0 1</td>
<td>M2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>1 0</td>
<td>M2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>1 1</td>
<td>M2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>M3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>0 1</td>
<td>M3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>1 0</td>
<td>M3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>1 1</td>
<td>M3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Higher order address bits of the address bus are used for chip selection and lower order bits are used for selecting one of many memory locations in the selected chip.

**Example:**

To select the 0th Memory location of the Memory chip M2, The address bus will contain the following address

\[
\begin{align*}
A3 & A2 & A1 & A0 \\
0 & 0 & 0 & 0
\end{align*}
\]

Selects the M2 Chip

Select 0th address of the M2 Chip
Operation Types in a Microprocessor

All of the operations/Functions of a microprocessor can be classified into one of three types:

- Microprocessor Initiated Operations
- Internal Operations
- Peripheral Initiated Operations

Microprocessor Initiated Operations

- These are operations that the microprocessor itself starts.
- These are usually one of 4 operations:
  - Memory Read
  - Memory Write
  - I/O Read (Get data from an input device)
  - I/O write (Send data to an output device)

- It is important to note that the microprocessor treats memory and I/O devices the same way.
  - Input and output devices simply look like memory locations to the microprocessor.
    - For example, the keyboard may look like memory address A3F2H. To get what key is being pressed, the microprocessor simply reads the data at location A3F2H.

- The communication process between the microprocessor and peripheral devices consist of the following three steps:
  - Identify the address.
  - Transfer the binary information.
  - Provide the right timing signals.

To read the contents of a memory location, the following steps take place:

1. The microprocessor places the 16-bit address of the memory location on the address bus.
2. The microprocessor activates a control signal called “memory read” which enables the memory chip.
3. The memory decodes the address and identifies the right location.
4. The memory places the contents on the data bus.
5. The microprocessor reads the value of the data bus after a certain amount of time.
Internal Data Operations

- The 8085 can perform a number of internal operations. Such as:
  a. Storing data
  b. Perform Arithmetic & Logic operations,
  c. Testing for condition.
  d. Sequence the execution of instructions
  e. Store data temporarily during the execution in the defined R/W memory locations called **Stack**

To perform these operations, the microprocessor needs an internal architecture similar to the following:
The Program Counter (PC)

- This is a register that is used to control the sequencing of the execution of instructions.
- This register always holds the address of the next instruction.
- Since it holds an address, it must be 16 bits wide.

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 retrieved soon.
- The stack is usually accessed in a Last In First Out (LIFO) fashion

Example:

```
2000: 06  MVIB 76H
2001: 78
2002: 3E  MVIA F2H
2003: F2
2004: 80  ADD B
2005: 76  HLT
```

** See Page 62 of your book for the detailed steps (Fifth Edition)

Externally Initiated Operations

- External devices can initiate (start) one of the 4 following operations:
  - **Reset**
    - All operations are stopped and the program counter is reset to 0000.
  - **Interrupt**
    - The microprocessor’s operations are interrupted and the microprocessor executes what is called a “service routine”.
    - This routine “handles” the interrupt, (perform the necessary operations). Then the microprocessor returns to its previous operations and continues.
  - **Ready**
    - The 8085 has a pin called RDY. This pin is used by external devices to stop the 8085 until they catch up.
    - As long as the RDY pin is low, the 8085 will be in a wait state.
  - **Hold**
    - The 8085 has a pin called HOLD. This pin is used by external devices to gain control of the busses.
• When the HOLD signal is activated by an external device, the 8085 stops executing instructions and stops using the busses.
• This would allow external devices to control the information on the busses

**Accessing Information in Memory**

• For the microprocessor to access (Read or Write) information in memory (RAM or ROM), it needs to do the following:
  – Select the right memory chip (using part of the address bus).
  – Identify the memory location (using the rest of the address bus).
  – Access the data (using the data bus).

**Tri-State Buffers**

• Tri-State Buffer is an important circuit element that is used extensively in memory.
• This buffer is a logic circuit that has three states:
  – Logic 0, logic 1, and high impedance.
  – When this circuit is in high impedance mode it looks as if it is disconnected from the output completely.
• This circuit has two inputs and one output.
  – The first input behaves like the normal input for the circuit.
  – The second input is an “enable”.
    • If it is set high, the output follows the proper circuit behavior.
    • If it is set low, the output looks like a wire connected to nothing.

**Example:**

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Hi-Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Hi-Z</td>
</tr>
</tbody>
</table>
The Basic Memory Element

- The basic memory element is similar to a D latch.
- This latch has an input where the data comes in. It has an enable input and an output on which data comes out.

- However, this is not safe.
  - Data is always present on the input and the output is always set to the contents of the latch.
  - To avoid this, tri-state buffers are added at the input and output of the latch.

- The WR signal controls the input buffer.
  - The bar over WR means that this is an active low signal.
  - So, if WR is 0 the input data reaches the latch input.
  - If WR is 1 the input of the latch looks like a wire connected to nothing.
- The RD signal controls the output in a similar manner.
A Memory “Register”

- If we take four of these latches and connect them together, we would have a 4-bit memory register