Lecture-3

Bussed Architecture

The basic components of a microcomputer, as discussed earlier, are:

1) CPU

2) Program memory

3) Data memory

4) Output ports

5) Input ports

6) Clock generator.

The clock generator generates the appropriate clock pulses for the synchronized operation of different components of microcomputer. The clock generator is on-chip in Intel 8085A microprocessor. Now, the question comes, how the microprocessor is connected to other components - memory and I/O ports. One possibility is that all the memory chips and ports are connected separately to CPU as shown in fig.1.2.
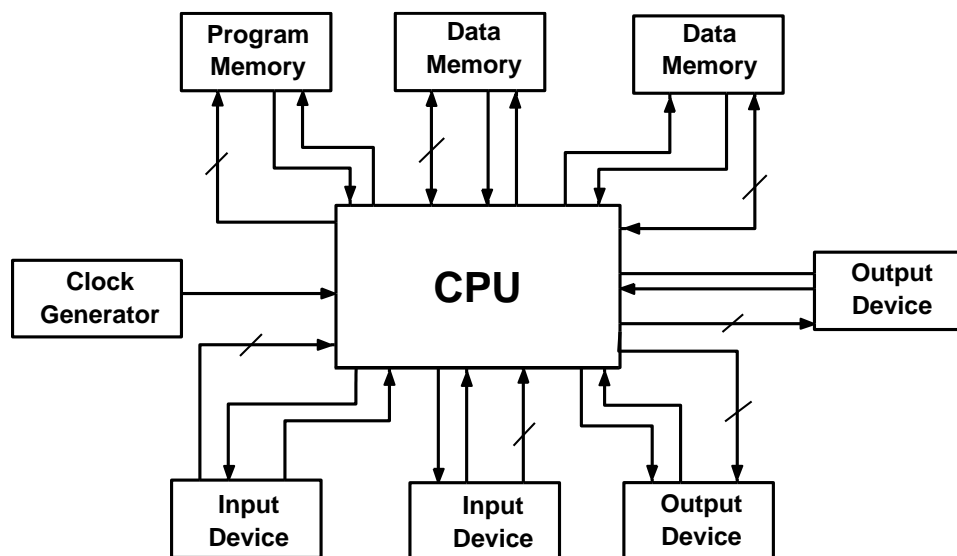

Fig.1.2 Separate Connection of Memory and I/O Devices with CPU

In this case, large no of address lines, signal lines, input and output lines are required from the CPU. The size of the CPU increases much if all the components are to be simultaneously controlled. The capability of expanding the system by adding more components will be limited. If a CPU is fabricated on a chip such that it provides interface to N devices, adding one more device will be impossible. The system becomes too complex, therefore, bussed architecture is used to connect component to the microprocessor.

The Bussed Architecture for Microprocessor:

The first question is what is a 'Bus'? Bus is a group of parallel lines that connect two or more devices. It carries information in bits. Whenever processor (CPU) needs to access any memory or I/O device of the microcomputer system, it does so by setting up signals on the address bus to identify the appropriate circuit. Data may be transferred by means of data bus, in required direction between the device and the processor. Signals on the control bus serve a number of purposes such as control the transfer of data direction.
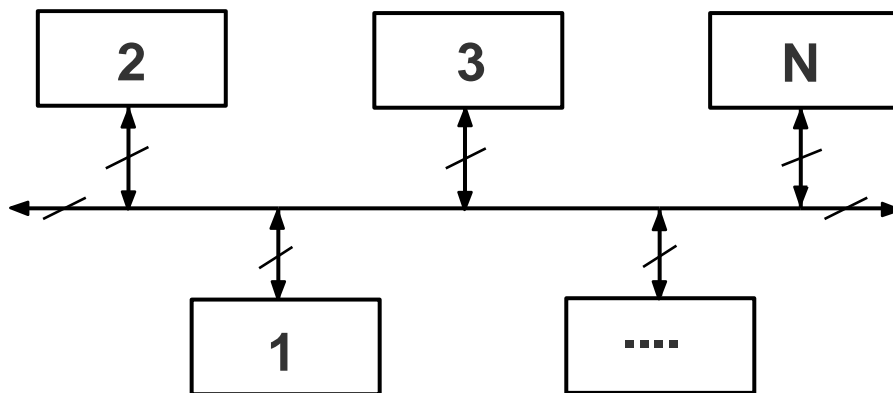
Fig.1.3 Data Bus Shared by 'N' Devices

Consider a situation where there are N devices connected to one single data bus as shown in fig.1.3. Some of them are input and some are output devices. Let us consider, device '1' wants to transfer data to device '2' using this line. This transfer of data can be performed provided:

a. Device '1' knows when to output data such that device '2' is in a position to receive the data. This can be easily ensured if device '2' has some means of signally device '1' to output data.

b. Other than device '1', no device outputs data on that signal line during this period. Device '1' should be the only device driving the signal line at this time.

c. Other than device '2', no device should accept the data from the data bus.

If the above conditions can be ensured, the same signal line can be shared by all N devices to transfer data between any two of the devices. The signal lines that are shared by a number of devices are referred to as the bus.

Normal gates are not suited for driving the bus in a bussed architecture. This is because such a gate will always be outputting either logic '1' or logic '0' it will not be possible to satisfy condition 'b'. Therefore, tri-state buffers are used for driving a bus.

The Microcomputer Bus:

The microcomputer contains three buses which carry all the address, data and control information involved in program execution. These buses connect the microprocessor to other elements - memory

and I/O devices so that transfer of information between the microprocessor & any of the elements can take place.

Address Bus:

In a microcomputer system, it is the CPU, which is the heart of the system, decides what action is to be taken in the system. Therefore, processor always selects the device for data transfer by putting the address of the device on the address bus. On address bus, information (address) flow takes place only in one direction, i.e., from the microprocessor to the memory or I/O devices. Therefore, this is called unidirectional address bus. The processor uses the address bus to identify an I/O device or memory. In the case of memory, this address also identifies the particular memory location inside the memory.

In 8085A processor, this bus is typically 16 bit long (A_0 to A_{15}). The CPU can generate 2^{16} or 65,536 different addresses on this bus. A memory location or an I/O device can be represented by each one of these addresses.

When the microprocessor wants to transfer information between itself and a certain memory location on I/O devices, it generates the 16-bit address (in 8085A processor) from an internal register on its 16 address pins, which then appear on the address bus. These address bits are decoded to determine the desired memory location on I/O devices. The decoding process normally requires logic circuit (decoders) in the microcomputer system. The logic circuit decodes the address to decide which I/O device or memory location is required to be involved in any data transfer

operation. If this decoding uniquely identifies only one port or memory location, then only data transfer takes place.

Data Bus:

A set of data lines (8 in 8085A processor)) referred to as the data bus is shared by number of devices to transfer data between microprocessor and peripherals. Care must be taken that at a time only one device should output data on the data bus, the other devices which can output data meet be in high-Z condition. The data can flow in both directions, i.e., to or from the microprocessor. Therefore, this is called bidirectional data bus (BDB). In some microprocessors, the data pins are also used to send other information such as address bits in addition to data. This means that the data pins are time shared or multiplexed. In Intel 8085A microprocessor lower 8-bits of the address (A_7-A_0) are time-multiplexed with the 8-bit data (D_7-D_0) and, therefore, this bus is called AD bus (AD_7-AD_0).

Control Bus:

The control bus is comprised of various single lines that carry control signals. These signals are used to synchronize the operation of the individual microcomputer elements. The microprocessor uses these signals for every operation it performs, like reading or writing a memory location or I/O device. These signals are also used to identify a memory location or an I/O device, e.g., $\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$. Some of the signals of the control bus are issued by the processor and some of the signals are received by the processor. Therefore, the control bus is called bidirectional control bus (BCB). The difference between BDB

and BCB is that in BDB all data lines are either in input mode or in output mode whereas in BCB the direction of signal flow on a line is fixed.

The bussed architecture of microprocessor is shown in fig.1.4
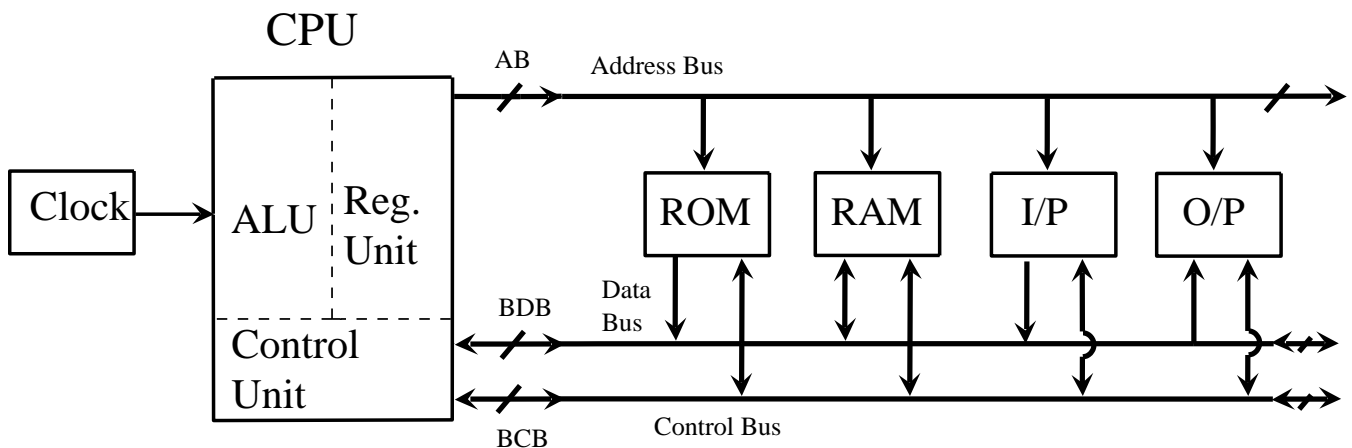
Fig.1.4 Bussed Architecture of Microprocessor

A microprocessor performs the function of the central processing unit. The microprocessor in combination with memory, I/O & a clock is essentially a microcomputer.

Computer Language:

Each machine has its own set of instructions based on the design of its microprocessor. To communicate with the computer one must give instruction in binary language or machine language the form in which it is stored in memory, i.e, as patterns of 1s & 0s. Since it is difficult for most users to write programs in machine language, computers manufactured have developed English like words to represent the binary instructions of a microprocessor. e.g. ADD, SUB or JMP etc. Users can write programs, called assembly language

programs (ALP), using these words called mnemonics. However, since the microprocessor can only executes the bit patterns of machine language instructions, the assembly language program must be converted to machine codes. This conversion can be carried out by hand, but this procedure is also time consumes and error prone. Special programs are available for each type of microprocessor that converts their assembly language programs to the equivalent machine codes. These programs are called assemblers and are run either on a microcomputer or minicomputer.

Because an assembly language is specific to a given machine, programs written in assembly language are not transferable from one machine to another. To circumvent this limitation, such general purpose languages as BASIC, FORTRAN, PASCAL, PL/M, C, have been devised, a program written in these languages are called high level languages (HLL). The programmes written in HLL are converted to machine language by another program called compiler or interpreter.

High–level languages do have same limitations in processor applications. The machine codes produced by compiler may be less efficient than that of the optimum equivalent ALP, increased memory requirement may not be important in view of cheap memory chips but increased executions time may be unacceptable in time critical applications. It is then desirable to write time critical parts of the program in assembly language. Further many peripheral device dependent operations may have to be programmed in ALP as such operation is often not supported by high level language.