

## Multiplexer:

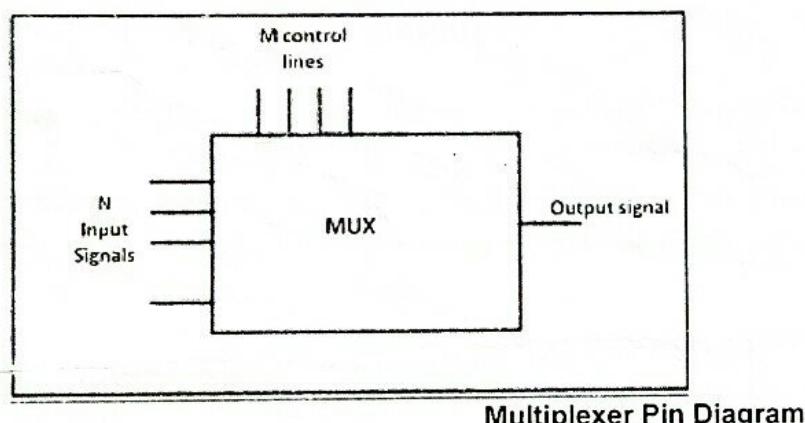
Multiplexer means many into one. A multiplexer is a circuit used to select and route any one of the several input signals to a signal output. An simple example of an non electronic circuit of a multiplexer is a single pole multiposition switch.

Multiposition switches are widely used in many electronics circuits. However circuits that operate at high speed require the multiplexer to be automatically selected. A mechanical switch cannot perform this task satisfactorily. Therefore, multiplexer used to perform high speed switching are constructed of electronic components.

Multiplexer handle two type of data that is analog and digital. For analog application, multiplexer are built of relays and transistor switches. For digital application, they are built from standard logic gates.

The multiplexer used for digital applications, also called digital multiplexer, is a circuit with many input but only one output. By applying control signals, we can steer any input to the output. Few types of multiplexer are 2-to-1, 4-to-1, 8-to-1, 16-to-1 multiplexer.

Following figure shows the general idea of a multiplexer with n input signal, m control signals and one output signal.

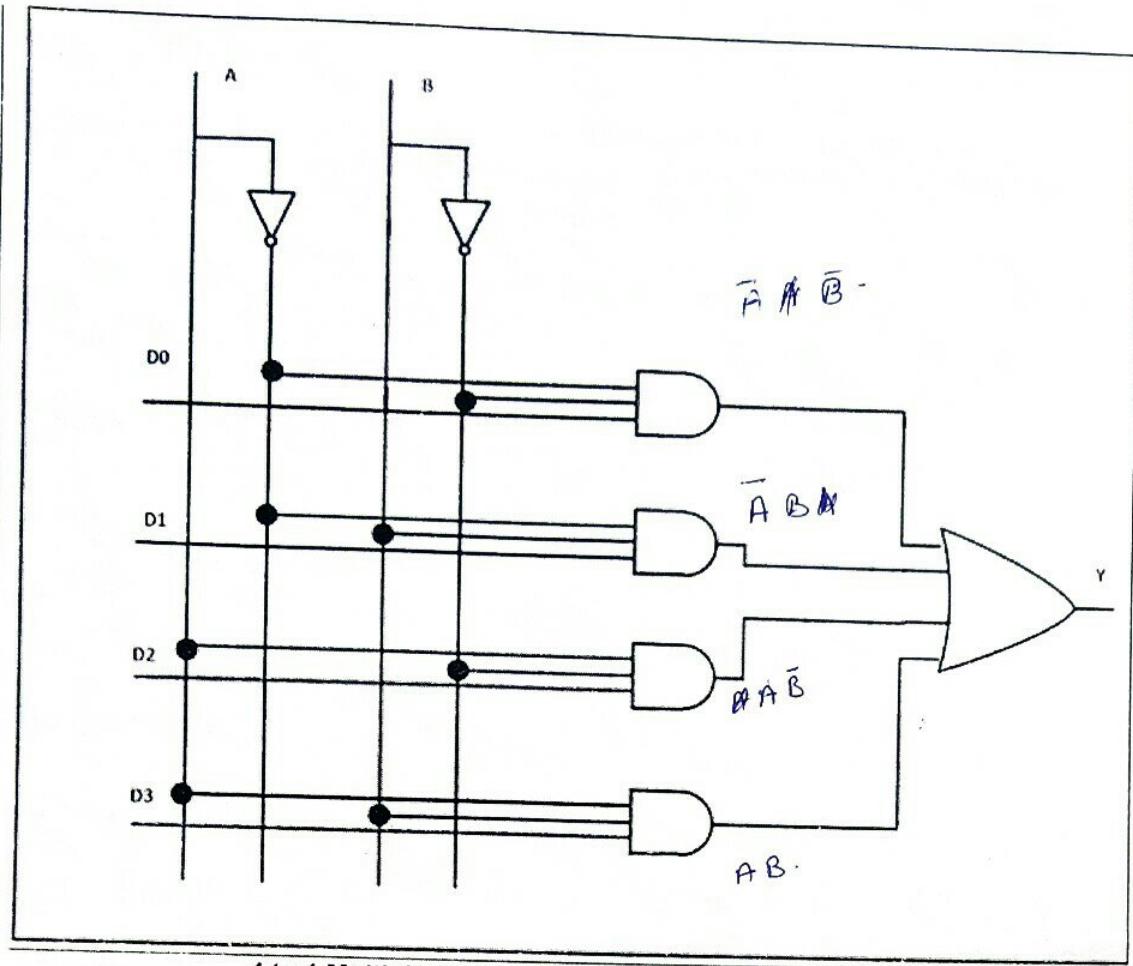


Multiplexer Pin Diagram

## Understanding 4-to-1 Multiplexer:

The 4-to-1 multiplexer has 4 input bit, 2 control bits, and 1 output bit. The four input bits are D<sub>0</sub>, D<sub>1</sub>, D<sub>2</sub> and D<sub>3</sub>. only one of this is transmitted to the output y. The output depends on the value of AB which is the control input. The control input determines which of the input data bit is transmitted to the output.

For instance, as shown in fig. when AB = 00, the upper AND gate is enabled while all other AND gates are disabled. Therefore, data bit D<sub>0</sub> is transmitted to the output, giving Y = D<sub>0</sub>.



**4 to 1 Multiplexer Circuit Diagram – ElectronicsHub.Org**

If the control input is changed to  $AB = 11$ , all gates are disabled except the bottom AND gate. In this case,  $D_3$  is transmitted to the output and  $Y = D_3$ .

- An example of 4-to-1 multiplexer is IC 74153 in which the output is same as the input.
- Another example of 4-to-1 multiplexer is 45352 in which the output is the compliment of the input.
- Example of 16-to-1 line multiplexer is IC74150.

### **Applications of Multiplexer:**

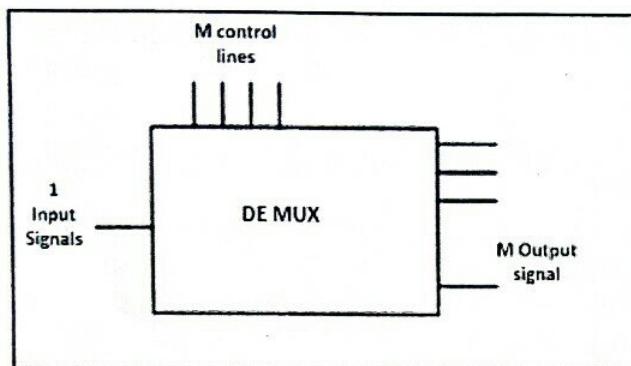
Multiplexers are used in various fields where multiple data need to be transmitted using a single line. Following are some of the applications of multiplexers –

1. **Communication system** – Communication system is a set of system that enable communication like transmission system, relay and tributary station, and communication network. The efficiency of communication system can be increased considerably using multiplexer. Multiplexer allow the process of transmitting different type of data such as audio, video at the same time using a single transmission line.
2. **Telephone network** – In telephone network, multiple audio signals are integrated on a single line for transmission with the help of multiplexers. In this way, multiple audio signals can be isolated and eventually, the desire audio signals reach the intended recipients.
3. **Computer memory** – Multiplexers are used to implement huge amount of memory into the computer, at the same time reduces the number of copper lines required to connect the memory to other parts of the computer circuit.
4. **Transmission from the computer system of a satellite** – Multiplexer can be used for the transmission of data signals from the computer system of a satellite or spacecraft to the ground system using the GPS (Global Positioning System) satellites.

## Demultiplexer:

Demultiplexer means one to many. A demultiplexer is a circuit with one input and many output. By applying control signal, we can steer any input to the output. Few types of demultiplexer are 1-to 2, 1-to-4, 1-to-8 and 1-to 16 demultiplexer.

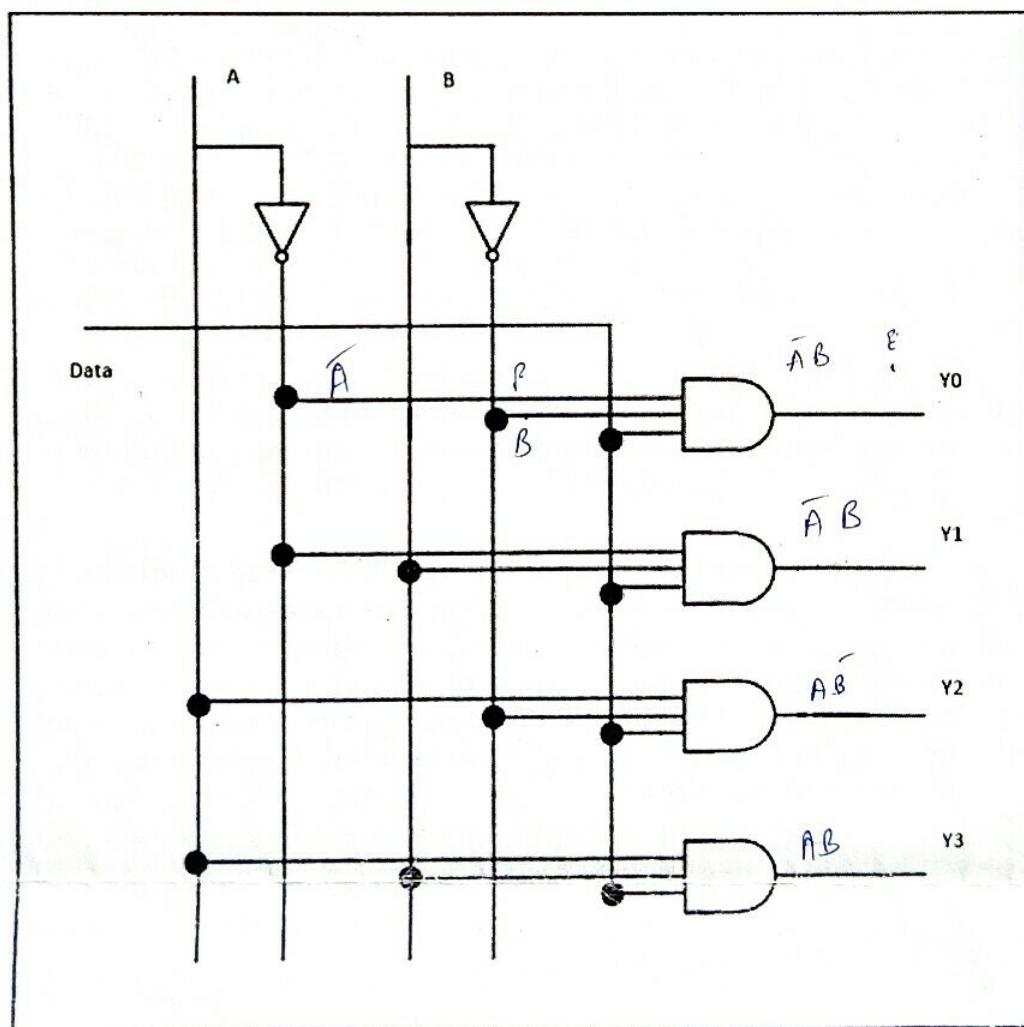
Following figure illustrate the general idea of a demultiplexer with 1 input signal, m control signals, and n output signals.



Demultiplexer Pin Diagram

## Understanding 1- to-4 Demultiplexer:

The 1-to-4 demultiplexer has 1 input bit, 2 control bit, and 4 output bits. An example of 1-to-4 demultiplexer is IC 74155. The 1-to-4 demultiplexer is shown in figure below-



1 to 4 Demultiplexer Circuit Diagram – ElectronicsHub.Org

The input bit is labelled as Data D. This data bit is transmitted to the data bit of the output lines. This depends on the value of AB, the control input.

When  $AB = 01$ , the upper second AND gate is enabled while other AND gates are disabled. Therefore, only data bit D is transmitted to the output, giving  $Y_1 = \text{Data}$ .

If D is low,  $Y_1$  is low. If D is high,  $Y_1$  is high. The value of  $Y_1$  depends upon the value of D. All other outputs are in low state.

If the control input is changed to  $AB = 10$ , all the gates are disabled except the third AND gate from the top. Then, D is transmitted only to the  $Y_2$  output, and  $Y_2 = \text{Data}$ .

Example of 1-to-16 demultiplexer is IC 74154 it has 1 input bit, 4 control bits and 16 output bit.

### **Applications of Demultiplexer:**

1. Demultiplexer is used to connect a single source to multiple destinations. The main application area of demultiplexer is communication system where multiplexer are used. Most of the communication system are bidirectional i.e. they function in both ways (transmitting and receiving signals). Hence, for most of the applications, the multiplexer and demultiplexer work in sync. Demultiplexer are also used for reconstruction of parallel data and ALU circuits.
2. **Communication System** – Communication system use multiplexer to carry multiple data like audio, video and other form of data using a single line for transmission. This process make the transmission easier. The demultiplexer receive the output signals of the multiplexer and converts them back to the original form of the data at the receiving end. The multiplexer and demultiplexer work together to carry out the process of transmission and reception of data in communication system.
3. **ALU (Arithmetic Logic Unit)** – In an ALU circuit, the output of ALU can be stored in multiple registers or storage units with the help of demultiplexer. The output of ALU is fed as the data input to the demultiplexer. Each output of demultiplexer is connected to multiple register which can be stored in the registers.
4. **Serial to parallel converter** – A serial to parallel converter is used for reconstructing parallel data from incoming serial data stream. In this technique, serial data from the incoming serial data stream is given as data input to the demultiplexer at the regular intervals. A counter is attach to the control input of the demultiplexer. This counter directs the data signal to the output of the demultiplexer where these data signals are stored. When all data signals have been stored, the output of the demultiplexer can be retrieved and read out in parallel.

## **Parallel Adder**

Adders are the combinatorial circuits which are used to add two binary numbers. The nature of the adders chosen depends on the characteristics of the binary numbers which need to be added. Say for example, if one needs to add two single bit binary digits, then one can use half adder while if there is an additional carry which needs to be added along with them, then one may resort to the use of full adder. However what if we want to add a binary number which has multiple bits in it. In such a case, the need arises to use a parallel adder.

### **Structure of Parallel Adder**

Parallel adder is nothing but a cascade of several full adders. The number of full adders used will depend on the number of bits in the binary digits which require to be added. Such a n-bit adder formed by cascading n full adders ( $FA_1$  to  $FA_n$ ) is as shown by Figure 1 and is used to add two n-bit binary numbers  $A_n A_{n-1} \dots A_2 A_1$  and  $B_n B_{n-1} \dots B_2 B_1$ .

- Frequency counters
- Digital clock
- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator.

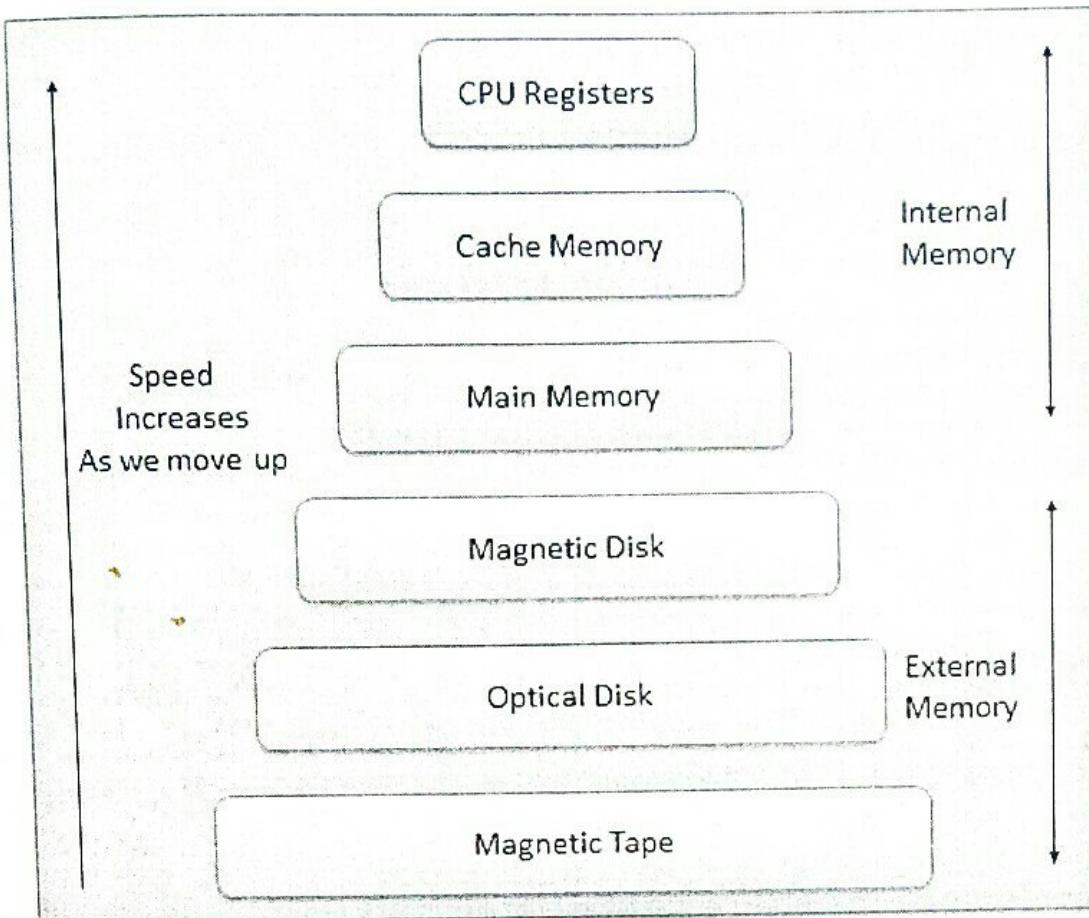
A **memory** is just like a human brain. It is used to store data and instruction. Computer memory is the storage space in computer where data is to be processed and instructions required for processing are stored.

The memory is divided into large number of small parts. Each part is called a cell. Each location or cell has a unique address which varies from zero to memory size minus one.

For example if computer has 64k words, then this memory unit has  $64 \times 1024 = 65536$  memory location. The address of these locations varies from 0 to 65535.

Memory is primarily of two types

- **Internal Memory** – cache memory and primary/main memory
- **External Memory** – magnetic disk / optical disk etc.



Characteristics of Memory Hierarchy are following when we go from top to bottom.

- Capacity in terms of storage increases.
- Cost per bit of storage decreases.
- Frequency of access of the memory by the CPU decreases.
- Access time by the CPU increases.

## RAM

A RAM constitutes the internal memory of the CPU for storing data, program and program result. It is read/write memory. It is called random access memory (RAM).

Since access time in RAM is independent of the address to the word that is, each storage location inside the memory is as easy to reach as other location & takes the same amount of time. We can reach into the memory at random & extremely fast but can also be quite expensive.

RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. Hence, a backup uninterruptible power system (UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold.

RAM is of two types

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

### Static RAM (SRAM)

The word **static** indicates that the memory retains its contents as long as power remains applied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not have to be refreshed on a regular basis.

Because of the extra space in the matrix, SRAM uses more chips than DRAM for the same amount of storage space, thus making the manufacturing costs higher.

Static RAM is used as cache memory needs to be very fast and small.

### Dynamic RAM (DRAM)

DRAM, unlike SRAM, must be continually **refreshed** in order for it to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory because it is cheap and small. All DRAMs are made up of memory cells. These cells are composed of one capacitor and one transistor.

## ROM

ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture.

A ROM, stores such instruction as are required to start computer when electricity is first turned on, this operation is referred to as bootstrap. ROM

chip are not only used in the computer but also in other electronic items like washing machine and microwave oven.

Following are the various types of ROM –

### **MROM (Masked ROM)**

The very first ROMs were hard-wired devices that contained a pre-programmed set of data or instructions. These kind of ROMs are known as masked ROMs. It is inexpensive ROM.

### **PROM (Programmable Read Only Memory)**

PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM programmer. Inside the PROM chip there are small fuses which are burnt open during programming. It can be programmed only once and is not erasable.

### **EPROM (Erasable and Programmable Read Only Memory)**

The EPROM can be erased by exposing it to ultra-violet light for a duration of upto 40 minutes. Usually, an EPROM eraser achieves this function. During programming an electrical charge is trapped in an insulated gate region. The charge is retained for more than ten years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid). This exposure to ultra-violet light dissipates the charge. During normal use the quartz lid is sealed with a sticker.

### **EEPROM (Electrically Erasable and Programmable Read Only Memory)**

The EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of re-programming is flexible but slow.

## **Serial Access Memory**

Sequential access means the system must search the storage device from the beginning of the memory address until it finds the required piece of data. Memory device which supports such access is called a Sequential Access Memory or Serial Access Memory. Magnetic tape is an example of serial access memory.

## Direct Access Memory

Direct access memory or Random Access Memory, refers to conditions in which a system can go directly to the information that the user wants. Memory device which supports such access is called a Direct Access Memory. Magnetic disks, optical disks are examples of direct access memory.

## Cache Memory

Cache memory is a very high speed semiconductor memory which can speed up CPU. It acts as a buffer between the CPU and main memory. It is used to hold those parts of data and program which are most frequently used by CPU. The parts of data and programs, are transferred from disk to cache memory by operating system, from where CPU can access them.

### Advantages

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

### Disadvantages

- Cache memory has limited capacity.
- It is very expensive.

Virtual memory is a technique that allows the execution of processes which are not completely available in memory. The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory is the separation of user logical memory from physical memory.

This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available. Following are the situations, when entire program is not required to be loaded fully in main memory.

- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.
- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.
- A program would no longer be constrained by the amount of physical memory that is available.
- Each user program could take less physical memory, more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.

## Auxiliary Memory

Auxiliary memory is much larger in size than main memory but is slower. It normally stores system programs, instruction and data files. It is also known as secondary memory. It can also be used as an overflow/virtual memory in case the main memory capacity has been exceeded. Secondary memories cannot be accessed directly by a processor. First the data/information of auxiliary memory is transferred to the main memory and then that information can be accessed by the CPU. Characteristics of Auxiliary Memory are following –

- **Non-volatile memory** – Data is not lost when power is cut off.

- **Reusable** – The data stays in the secondary storage on permanent basis until it is not overwritten or deleted by the user.
- **Reliable** – Data in secondary storage is safe because of high physical stability of secondary storage device.
- **Convenience** – With the help of a computer software, authorised people can locate and access the data quickly.
- **Capacity** – Secondary storage can store large volumes of data in sets of multiple disks.
- **Cost** – It is much lesser expensive to store data on a tape or disk than primary memory.

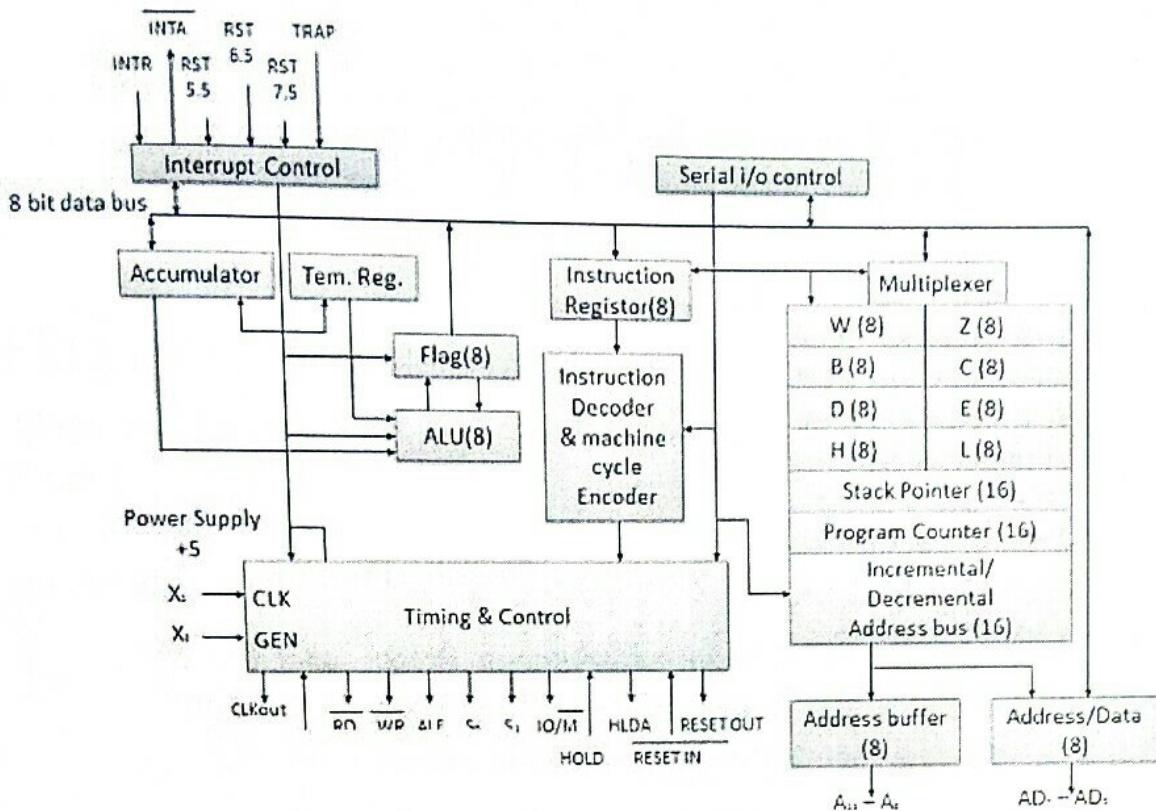
Microprocessing unit is synonymous to central processing unit, CPU used in traditional computer. Microprocessor (MPU) acts as a device or a group of devices which do the following tasks.

- communicate with peripherals devices
- provide timing signal
- direct data flow
- perform computer tasks as specified by the instructions in memory

## 8085 Microprocessor

The 8085 microprocessor is an 8-bit general purpose microprocessor which is capable to address 64k of memory. This processor has forty pins, requires +5 V single power supply and a 3-MHz single-phase clock.

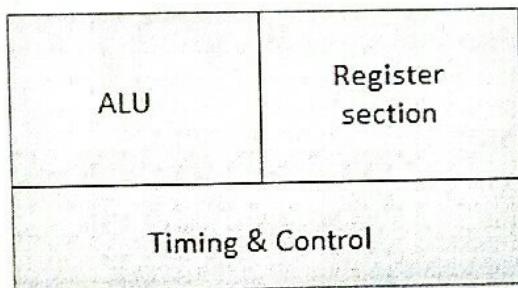
### Block Diagram



## ALU

The ALU perform the computing function of microprocessor. It includes the accumulator, temporary register, arithmetic & logic circuit & and five flags. Result is stored in accumulator & flags.

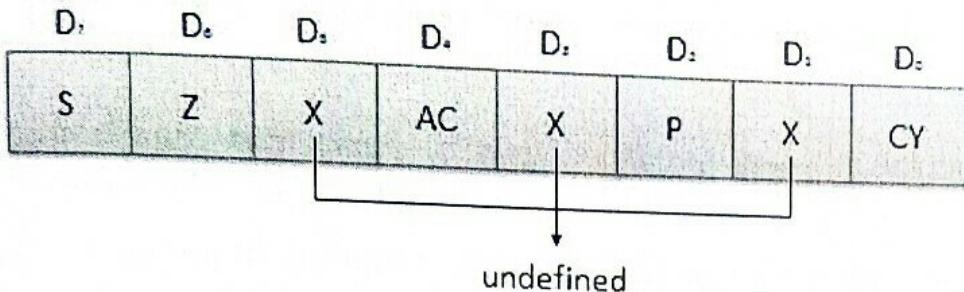
### Block Diagram



## Accumulator

It is an 8-bit register that is part of ALU. This register is used to store 8-bit data & in performing arithmetic & logic operation. The result of operation is stored in accumulator.

### Diagram



## Flags

Flags are programmable. They can be used to store and transfer the data from the registers by using instruction. The ALU includes five flip-flops that are set and reset according to data condition in accumulator and other registers.

- **S (Sign) flag** – After the execution of an arithmetic operation, if bit D<sub>7</sub> of the result is 1, the sign flag is set. It is used to signed number. In a given byte, if D<sub>7</sub> is 1 means negative number. If it is zero means it is a positive number.
- **Z (Zero) flag** – The zero flag is set if ALU operation result is 0.
- **AC (Auxiliary Carry) flag** – In arithmetic operation, when carry is generated by digit D<sub>3</sub> and passed on to digit D<sub>4</sub>, the AC flag is set. This flag is used only internally BCD operation.
- **P (Parity) flag** – After arithmetic or logic operation, if result has even number of 1s, the flag is set. If it has odd number of 1s, flag is reset.
- **C (Carry) flag** – If arithmetic operation result is in a carry, the carry flag is set, otherwise it is reset.

## Register section

It is basically a storage device and transfers data from registers by using instructions.

- **Stack Pointer (SP)** – The stack pointer is also a 16-bit register which is used as a memory pointer. It points to a memory location in Read/Write memory known as stack. In between execution of program, sometime data to be stored in stack. The beginning of the stack is defined by loading a 16-bit address in the stack pointer.
- **Program Counter (PC)** – This 16-bit register deals with fourth operation to sequence the execution of instruction. This register is also a memory pointer. Memory location have 16-bit address. It is used to store the execution address.

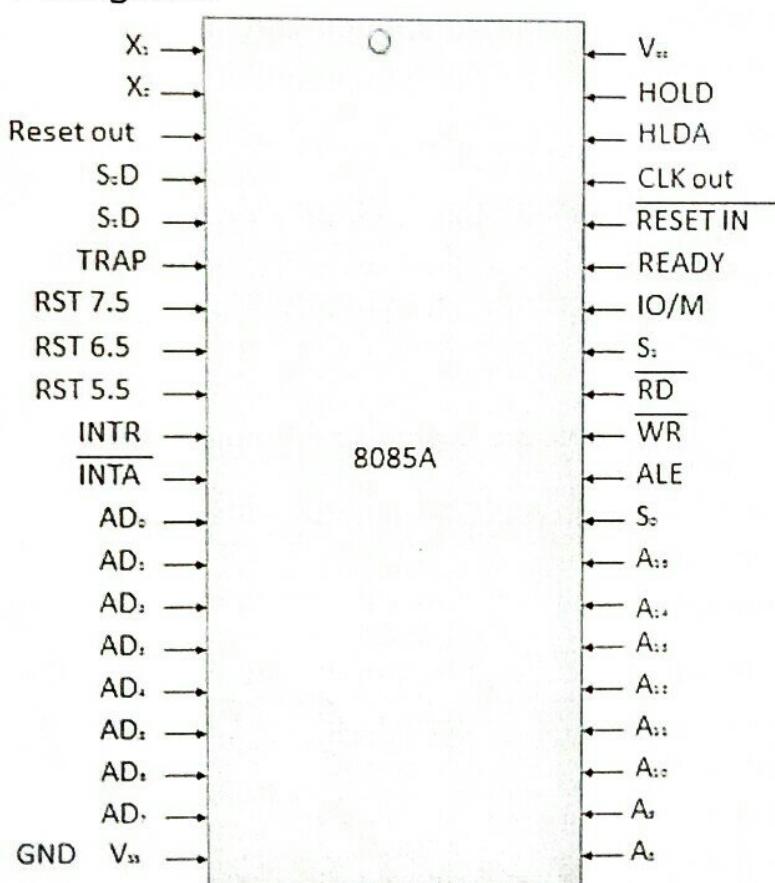
The function of the program counter is to point to memory address from which next byte is to be fetched.

- **Storage registers** – These registers store 8-bit data during a program execution. These registers are identified as B, C, D, E, H, L. They can be combined as register pair BC, DE and HL to perform some 16 bit operations.

## Time and Control Section

This unit is responsible to synchronize Microprocessor operation as per the clock pulse and to generate the control signals which are necessary for smooth communication between Microprocessor and peripherals devices. The RD bar and WR bar signals are synchronous pulses which indicates whether data is available on the data bus or not. The control unit is responsible to control the flow of data between microprocessor, memory and peripheral devices.

### PIN diagram



All the signal can be classified into six groups

S.N.	Group	Description
1	<b>Address bus</b>	The 8085 microprocessor has 8 signal line, A <sub>15</sub> - A <sub>8</sub> which are unidirectional and used as a high order address bus.
2	<b>Data bus</b>	The signal line AD7 - AD0 are bidirectional for dual purpose. They are used as low order address bus as well as data bus.
3	<b>Control signal and Status signal</b>	<p>Control Signal</p> <p><b>RD bar</b> – It is a read control signal (active low). If it is active then memory read the data.</p> <p><b>WR bar</b> – It is write control signal (active low). It is active when written into selected memory.</p> <p>Status signal</p> <p><b>ALU (Address Latch Enable)</b> – When ALU is high. 8085 microprocessor use address bus. When ALU is low. 8085 microprocessor is use data bus.</p> <p><b>IO/M bar</b> – This is a status signal used to differentiate between i/o and memory operations. When it is high, it indicate an i/o operation and when it is low, it indicate memory operation.</p>

#### 4 Power supply and frequency signal

**S<sub>1</sub> and S<sub>0</sub>** – These status signals, similar to i/o and memory bar, can identify various operations, but they are rarely used in small system.

**V<sub>cc</sub>** – +5v power supply.

**V<sub>ss</sub>** – ground reference.

**X, X** – A crystal is connected at these two pins. The frequency is internally divided by two operate system at 3-MHz, the crystal should have a frequency of 6-MHz.

**CLK out** – This signal can be used as the system clock for other devices.

**INTR (i/p)** – Interrupt request.

**INTA bar (o/p)** – It is used as acknowledge interrupt.

**TRAP (i/p)** – This is non maskable interrupt and has highest priority.

**HOLD (i/p)** – It is used to hold the executing program.

**HLDA (o/p)** – Hold acknowledge.

**READY (i/p)** – This signal is used to delay the microprocessor read or write cycle until a slow responding peripheral is ready to

#### 5 Externally initiated signal

accept or send data.

**RESET IN bar** – When the signal on this pin goes low, the program counter is set to zero, the bus are tri-stated, & MPU is reset.

**RESET OUT** – This signal indicate that MPU is being reset. The signal can be used to reset other devices.

**RST 7.5, RST 6.5, RST 5.5 (Request interrupt)** – It is used to transfer the program control to specific memory location. They have higher priority than INTR interrupt.

## 6      **Serial I/O ports**

The 8085 microprocessor has two signals to implement the serial transmission serial input data and serial output data.

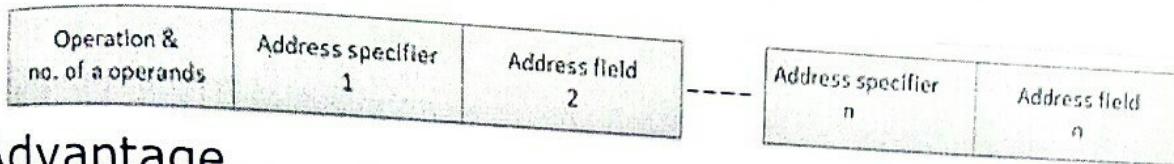
# Instruction Format

Each instruction is represented by a sequence of bits within the computer. The instruction is divided into group of bits called field. The way instruction is expressed is known as instruction format. It is usually represented in the form of rectangular box. The instruction format may be of the following types.

## Variable Instruction Formats

These are the instruction formats in which the instruction length varies on the basis of opcode & address specifiers. For Example, VAX instruction vary between 1 and 53 bytes while X86 instruction vary between 1 and 17 bytes.

### Format



## Advantage

These formats have good code density.

## Drawback

These instruction formats are very difficult to decode and pipeline.

## Fixed Instruction Formats

In this type of instruction format, all instructions are of same size. For Example, MIPS, Power PC, Alpha, ARM.

## Format

Operation	Address field 1	Address field 2	Address field 3
-----------	-----------------	-----------------	-----------------

## Advantage

They are easy to decode & pipeline.

## Drawback

They don't have good code density.

## Hybrid Instruction Formats

In this type of instruction formats, we have multiple format length specified by opcode. For example, IBM 360/70, MIPS 16, Thumb.

## Format

Operation	Address specifier	Address field	
Operation	Address specifier 1	Address specifier 2	Address field
Operation	Address specifier	Address field 1	Address field 2

## Advantage

These compromise between code density & instruction of these type are very easy to decode.

## Addressing Modes

Addressing mode provides different ways for accessing an address to given data to a processor. Operated data is stored in the memory location, each instruction required certain data on which it has to operate. There are various techniques to specify address of data. These techniques are called Addressing Modes.

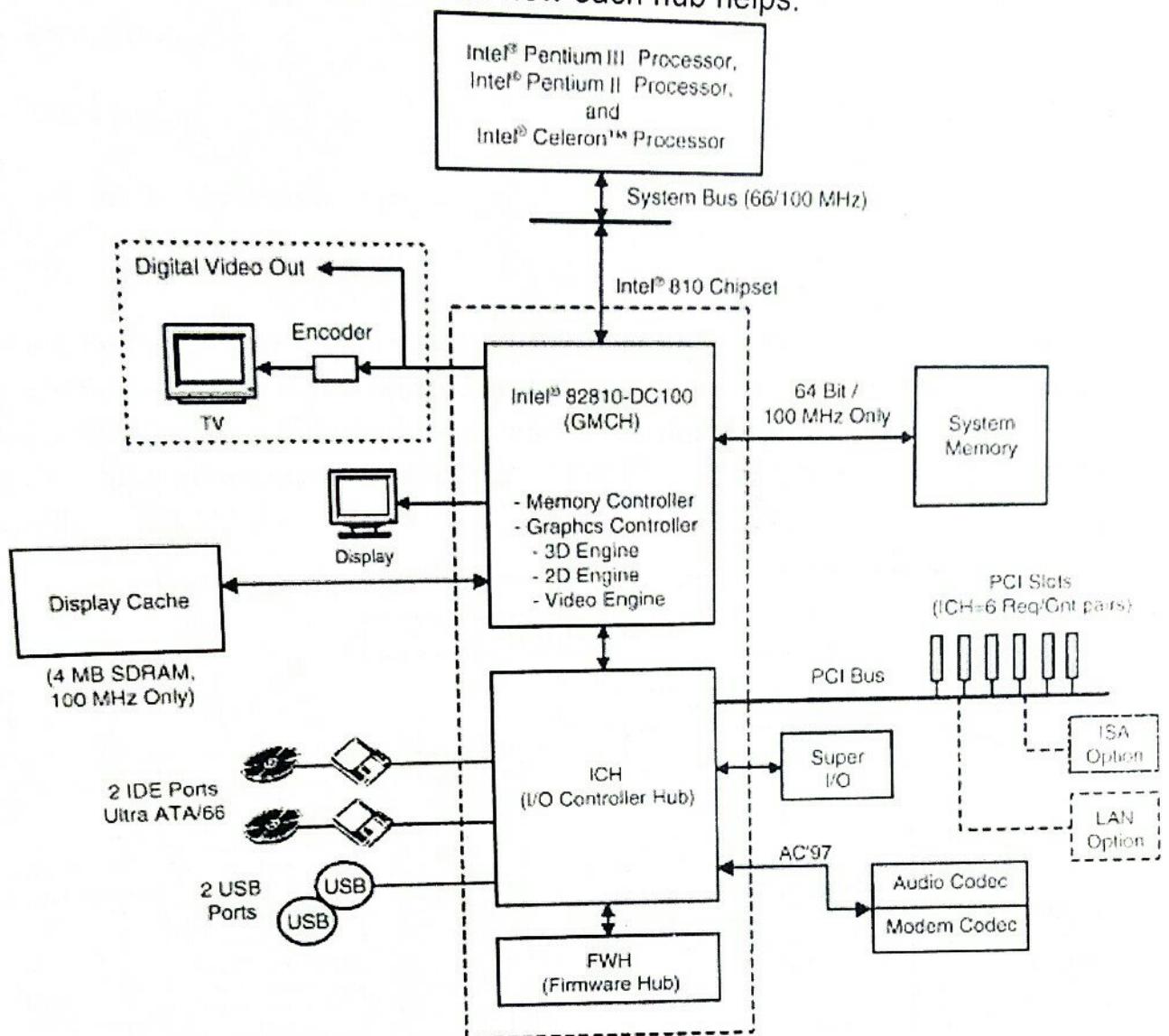
- **Direct addressing mode** – In the direct addressing mode, address of the operand is given in the instruction and data is available in the memory location which is provided in instruction. We will move this data in desired location.
- **Indirect addressing mode** – In the indirect addressing mode, the instruction specifies a register which contain the address of the operand. Both internal RAM and external RAM can be accessed via indirect addressing mode.
- **Immediate addressing mode** – In the immediate addressing mode, direct data is given in the operand which move the data in accumulator. It is very fast.
- **Relative addressing mode** – In the relative address mode, the effective address is determined by the index mode by using the program counter in stead of general purpose processor register. This mode is called relative address mode.
- **Index addressing mode** – In the index address mode, the effective address of the operand is generated by adding a content value to the contents of the register. This mode is called index address mode.

Intel launched the i810 chipset and the Celeron-466 on the 26th of April and is marketing the i810 chipset specifically with Celeron-based systems in mind. The best part is that it uses a new architecture that is far more efficient than the previous generations and it might just set the tone for the future.

## The i810 Architecture

Traditionally everything on the system communicates with the CPU and memory via the slow and perhaps saturated central PCI bus, but with the introduction of the i810 chipset, a new architecture is born and is called the **Accelerated Bus Architecture** (AHA). AHA allows direct communication of the CPU, memory, graphics (built-in to the

chipset) and other peripherals. It replaces the PCI as the central bus and the typical North and Southbridge chipsets with a **three-hub architecture**. The three hubs (which are distinct chips) are the **Graphics and Memory Controller Hub (GMCH)**, the I/O Controller Hub (ICH) and the Firmware Hub (FWH). The below diagram will help to explain how the new AHA works and how each hub helps:



Between the ICH and GMCH, there is an exclusive bus with a bandwidth of 266MB/s (double that of the PCI bus) to make sure that there is no slowdown for any of the connected peripherals like HDDs, AC97 codec, PCI cards and others, to communicate with the processor or system memory. Another point to note is that the i810 chipset can only support a single CPU unlike the popular Intel 440 BX chipset that can support dual CPUs. It's not much of a matter because the i810 chipset is for the value market or system integrators.

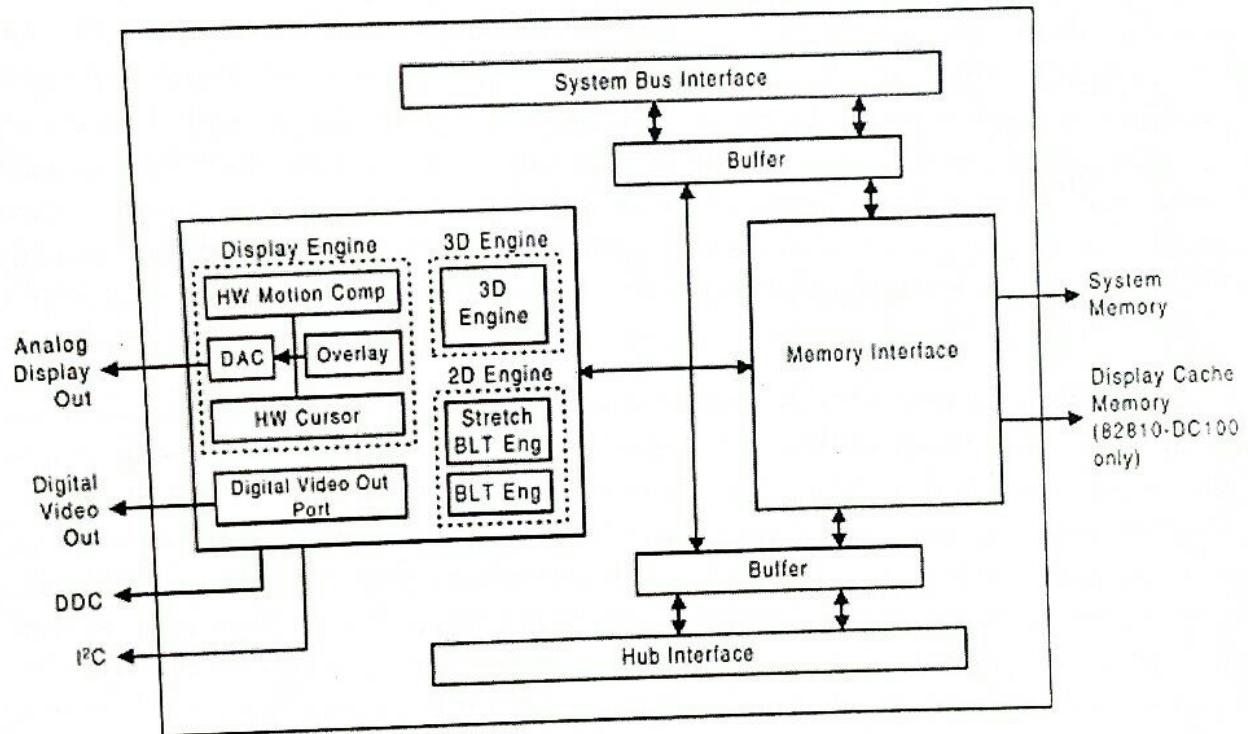
### The GMCH

The GMCH as the name suggests contains the graphics and memory controllers and the following its break down of components:

- System Bus Interface

- System Memory Interface
- Display Interface
- Display Cache Interface (82810-DC100 only)
- Digital TV-Out
- Clock Signals
- Miscellaneous Interface Signals
- Hub Interface

There are 2 versions of the GMCH, the 82810 GMCH and the 82810-DC100 GMCH. The main difference between them is that the DC100 version has the Display Cache Interface integrated which supports 4MB of external display cache. Both versions of the GMCH are pin-compatible, hence one manufacturer can have a low cost model and the premium model without the need to redesign the base model. Here's a schematic of the GMCH:-



While the Intel BX chipset is able to support up to 1GB of memory, the new i810 can only support 256MB of memory. This is the reason why you'll see up to 4 DIMM slots in the BX chipset based motherboard, which has 8 addressing lines to cater to 4 DIMM slots that can take in single or double-sided DIMMs mixed. The i810 has only 4 memory addressing lines, therefore motherboards based on the i810 chipset come with 2 DIMM slots that can take in either a pair of single or double-sided DIMMs mixed. Take note that double-sided DIMMs need 2 addressing lines each (and single-sided DIMMs need 1 addressing line) and these are usually available in either 128MB or 256MB DIMM modules. Since the

i810 is for integrated cost-effective systems that won't run in servers that use ECC memory for safety, ECC support has been removed in the i810 chipset.

## The Graphics Core

Graphics is handled by an integrated i752 core, which is basically a spruced up i740. Its key features include dual-rendering pipelines, motion-compensation for DVD, 1600 x 1200 pixels desktop resolution, 230MHz RAMDAC, 16-bit colour support in 3D and a Direct-AGP connection that is essentially a direct access to the system memory to store all graphics data as the GMCH itself has no local memory. What's more, the memory bus is locked at 100MHz, while the FSB can asynchronously run at 66/100MHz (officially, but there are the usual list of unofficial ones implemented by the motherboard vendors). This means even if you intend to overclock the system by changing the FSB, the memory bus can still operate at 100MHz. This also means any standard PC-100 memory will do the job without you needing to hunt for those speedier and more expensive RAM. However you can't use your older PC-66 memory unless they happen to be of good quality and are happy to operate at 100MHz.

Now back to the Direct-AGP connection, since the graphics controller in the GMCH has a direct connection to the system memory, the bandwidth is 800MB/s (64-bit memory bus / 8-bits x 100MHz) and that is more than AGP 2X transfer rate of 533MB/s (32-bit / 8-bits x 133MHz)! Don't get excited yet because all the graphics adapters have their own local memory that has far higher bandwidth between the graphics chipset and the onboard memory, but once the graphics data (usually texture data) gets too large for the local memory, they'll have to spillover to the system memory if the graphics adapter uses an AGP connection. In that situation, the GMCH's integrated graphics becomes more favorable because of its higher bandwidth to the system's memory. Even so, this won't be much beneficial as the graphics engine itself isn't as powerful as Intel's marketing claims (would you believe NVIDIA Riva TNT-like performance from an i752 graphics core?)

Fortunately, the costlier version of the GMCH labeled as the i82810-DC100 can support an optional 4MB of local cache for the integrated graphics (the cheaper i82810 GMCH doesn't have support for the extra cache). The DC100 descriptor just means that it uses 100MHz cache. There is a possibility of a DC133 part because the BIOS of the test motherboard supports a cache speed setting of either 100 or 133MHz. The odd thing is that there is a separate bus from the GMCH to the optional cache but it's only 32-bits wide, meaning it can only sustain a bandwidth of 400MB/s. Seems odd considering that the Direct-AGP to system memory connection sports a wider 800MB/s bandwidth but only half of that for the dedicated local cache. Fortunately the cache and its memory bandwidth is dedicated to the purpose of serving the integrated graphics engine and it helps reduce the system memory's burden. Overall the idea of dedicated cache and the Direct-AGP connection is far better than the old UMA-architecture in older motherboard designs. Last but not least, take note that 1MB of your system memory is reserved for display frame buffer purposes from the second you power up your PC. Once in the Windows OS, it will use the Intel Dynamic Video Memory (D.V.M.) technology to freely allocate itself an appropriate amount of system memory as and when it needs to. Here's a snippet from Intel:-

The internal graphics device on both the 82810 and 82810-DC100 support Intel Dynamic Video Memory Technology (D.V.M.). With D.V.M., the allocated size of system memory used for display graphics can be dynamically altered. For example, if 2 MB of system memory is needed, the driver allocates this amount.

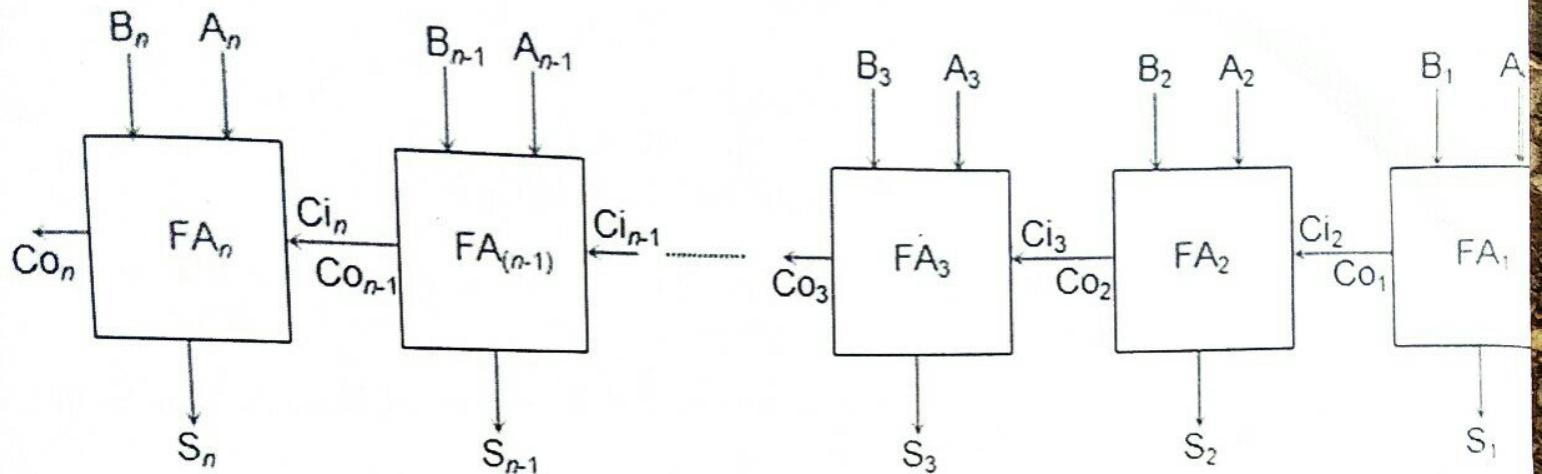


Figure 1  $n$ -bit Parallel Adder Circuit

Here, every single bit of the numbers to be added is provided at the input pins of every single full adder. That is, the first bits  $A_1$  and  $B_1$  are provided as the inputs to full adder ( $FA_1$ ), the second bits  $A_2$  and  $B_2$  to the inputs of full adder 2 ( $FA_2$ )... and the last bits  $A_n$  and  $B_n$  to the  $n^{\text{th}}$  full adder  $FA_n$ . Next, the carry out pin of each full adder in the circuit is connected to the carry in pin of its succeeding full adder (except in the case of last full adder). For example, the carry out pin of  $FA_1$  ( $Co_1$ ) is connected to carry in pin of  $FA_2$  ( $Ci_2$ ), the carry out pin of  $FA_2$  ( $Co_2$ ) is connected to carry in pin of  $FA_3$  ( $Ci_3$ ) and so on and so forth.

### Working of Parallel Adder

In the circuit shown by Figure 1, first,  $FA_1$  adds  $A_1$  with  $B_1$  to generate  $S_1$  (the first bit of sum output) and  $Co_1$ . Next,  $FA_2$  uses this  $Co_1$  as its carry in bit and adds it with its input bits  $A_2$  and  $B_2$  to generate the second bit of the sum output  $S_2$  and  $Co_2$ . Next, this  $Co_2$  is considered as an input by  $FA_3$  which adds it with the bits  $A_3$  and  $B_3$ . This process continues till the  $n^{\text{th}}$  full adder in the sequence which adds the carry out bit of  $(n-1)^{\text{th}}$  full adder ( $Co_{n-1}$ ) with its inputs  $A_n$  and  $B_n$ . When this happens, we would get the output bits  $S_n$  and  $Co_n$  which are the last bits of our sum output and the expected carry bit, respectively.

### Drawback of Parallel Adders

From the discussion presented we can say that in the case of  $n$ -bit **parallel adder**, each adder has to wait for the carry term to be generated from its preceding adder in order to finish its task of adding. This can be visualized as if the carry term propagates along the chain in the fashion of a ripple. Thus these kind of adders are even referred to as ripple carry adders. Further, the delay associated with the travelling of carry bit is called carry propagation delay and is found to worsen with an increase in the length of the binary numbers which require to be added. For example, if each full adder is considered to have a delay of 10 ns, then the total delay required to produce the output of a 4-bit **parallel adder** would be  $4 \times 10 = 40$  ns.

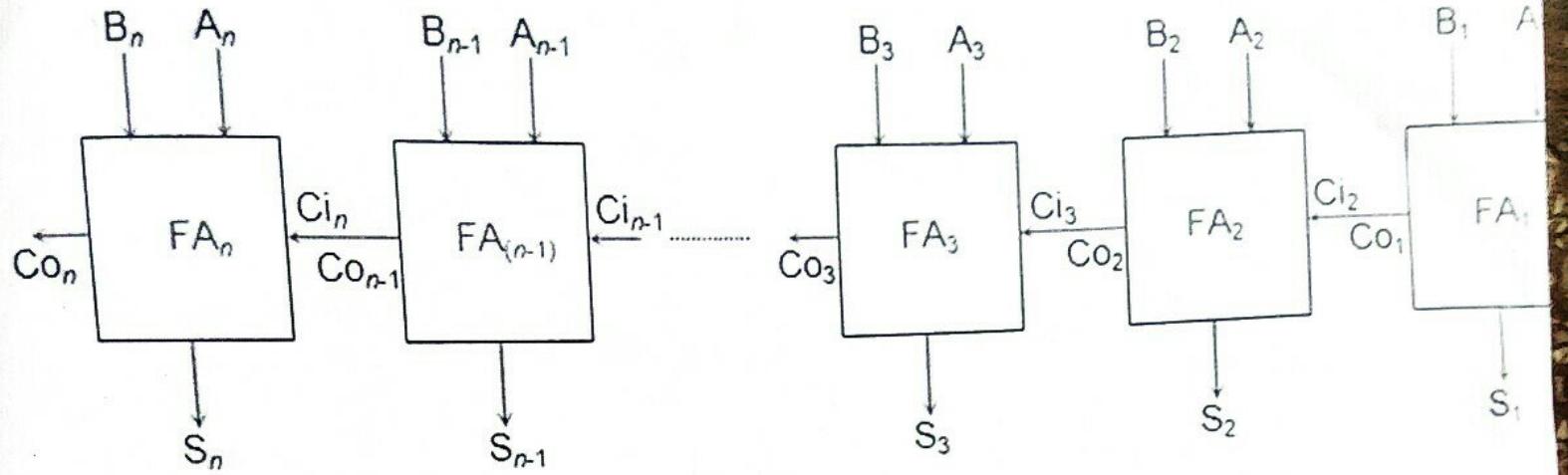


Figure 1 *n*-bit Parallel Adder Circuit

Here, every single bit of the numbers to be added is provided at the input pins of every single full adder. That is, the first bits  $A_1$  and  $B_1$  are provided as the inputs to full adder ( $FA_1$ ), the second bits  $A_2$  and  $B_2$  to the inputs of full adder 2 ( $FA_2$ )... and the last bits  $A_n$  and  $B_n$  to the  $n^{\text{th}}$  full adder  $FA_n$ . Next, the carry out pin of each full adder in the circuit is connected to the carry in pin of its succeeding full adder (except in the case of last full adder). For example, the carry out pin of  $FA_1(Co_1)$  is connected to carry in pin of  $FA_2(Ci_2)$ , the carry out pin of  $FA_2(Co_2)$  is connected to carry in pin of  $FA_3(Ci_3)$  and so on and so forth.

### Working of Parallel Adder

In the circuit shown by Figure 1, first,  $FA_1$  adds  $A_1$  with  $B_1$  to generate  $S_1$  (the first bit of sum output) and  $Co_1$ . Next,  $FA_2$  uses this  $Co_1$  as its carry in bit and adds it with its input bits  $A_2$  and  $B_2$  to generate the second bit of the sum output  $S_2$  and  $Co_2$ . Next, this  $Co_2$  is considered as an input by  $FA_3$  which adds it with the bits  $A_3$  and  $B_3$ . This process continues till the  $n^{\text{th}}$  full adder in the sequence which adds the carry out bit of  $(n-1)^{\text{th}}$  full adder ( $Co_{n-1}$ ) with its inputs  $A_n$  and  $B_n$ . When this happens, we would get the output bits  $S_n$  and  $Co_n$  which are the last bits of our sum output and the expected carry bit, respectively.

### Drawback of Parallel Adders

From the discussion presented we can say that in the case of *n*-bit parallel adder, each adder has to wait for the carry term to be generated from its preceding adder in order to finish its task of adding. This can be visualized as if the carry term propagates along the chain in the fashion of a ripple. Thus these kind of adders are even referred to as ripple carry adders. Further, the delay associated with the travelling of carry bit is called carry propagation delay and is found to worsen with an increase in the length of the binary numbers which require to be added. For example, if each full adder is considered to have a delay of 10 ns, then the total delay required to produce the output of a 4-bit parallel adder would be  $4 \times 10 = 40$  ns.