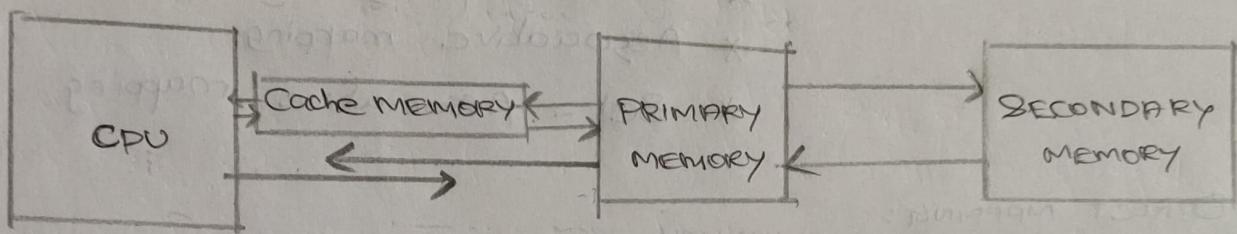


1. LEVEL OF CACHE MEMORY AND CACHE MAPPING:

CACHE MEMORY is a special very high speed memory. It is used to speed up and synchronizing with high-speed CPU. Cache memory is costlier than main memory or disk memory but economical than CPU Registers.

* Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instruction so that they are immediately available to the CPU when needed.



LEVEL OF MEMORY:

* LEVEL 1 OR Registers:

It is a type of memory in which data is stored and accessed that are immediately stored in CPU. Most commonly used register is accumulator, program counter, address register etc.

* LEVEL 2 OR CACHE MEMORY:

It is the fastest memory which has faster access time where data is temporarily stored for faster access.

* LEVEL 3 OR MAIN MEMORY:

It is memory on which computer works normally. It is small in size and once power is off data no longer stays in this memory.

* LEVEL 4 OR SECONDARY MEMORY:

* It is external memory which is not as fast as main memory but data stays permanently in this memory.

* CACHE MAPPING:

There are three different type of mapping used for the purpose of cache memory.

- * Direct Mapping
- * Associative mapping
- * Set - Associative mapping.

* DIRECT MAPPING:

The simplest, known as direct mapping, maps each block of main memory into one possible cache line or in direct mapping, assign each memory block to a specific line in the cache. If a line is previously taken up by memory block, when a new block needs to be loaded, the old block is erased. An address space is split into two parts index field and a tag field. Bit level mapping.

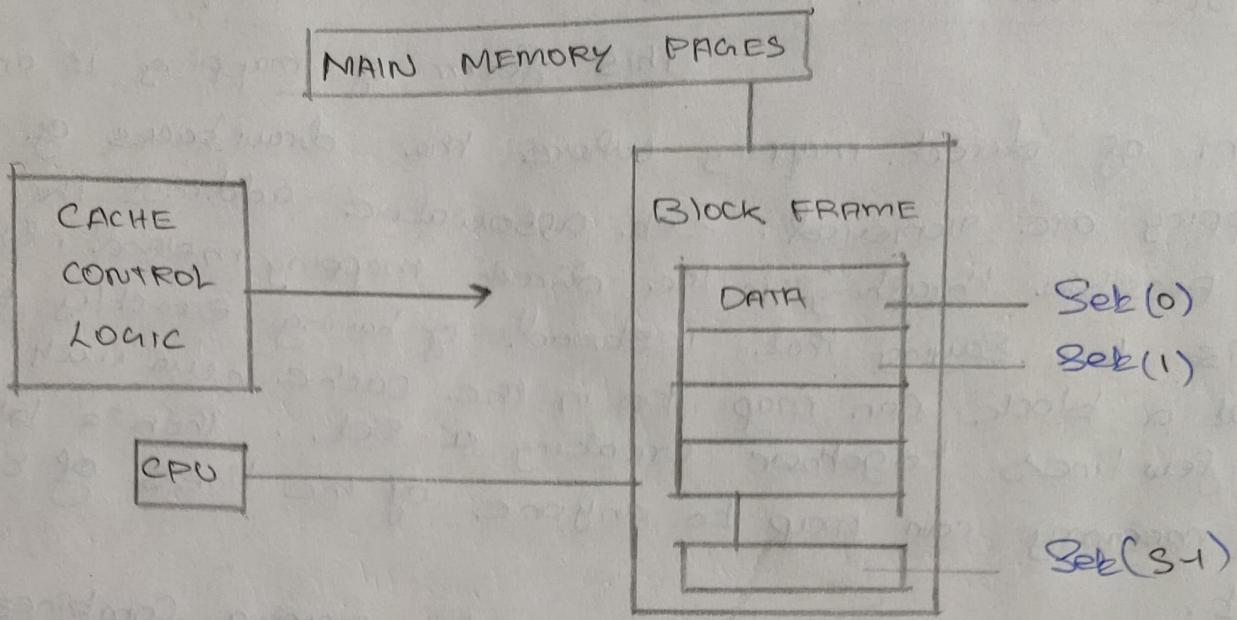
$$i = j \text{ modulo } m$$

where

i = cache line number

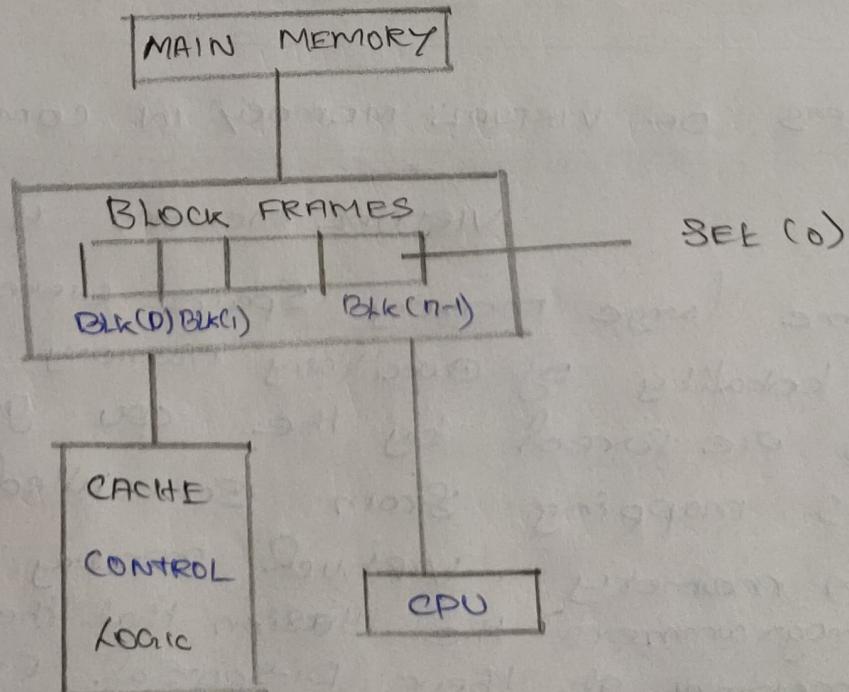
j = main memory block number

m = Number of lines in the cache.



* ASSOCIATIVE MAPPING :-

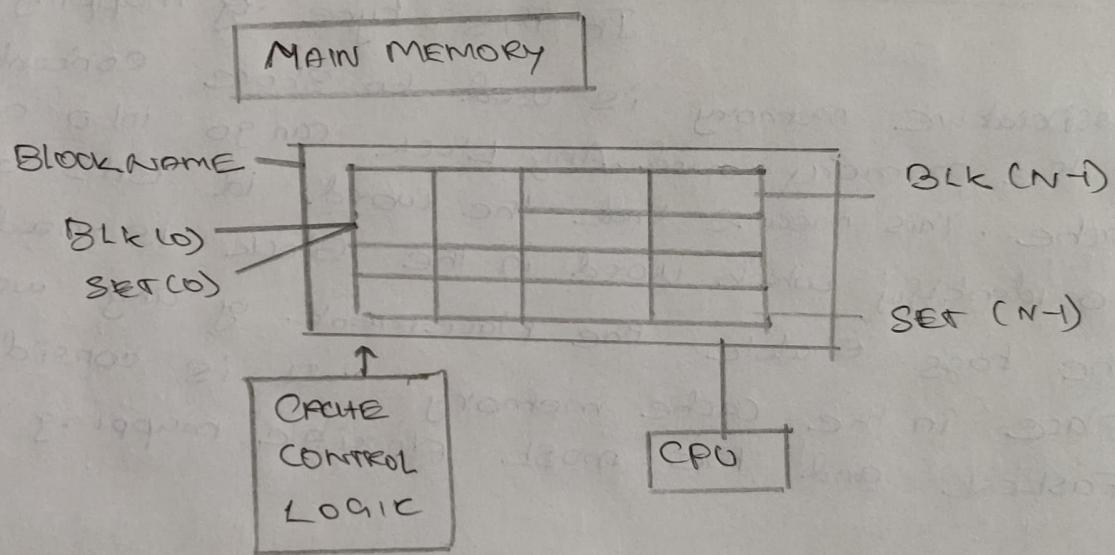
In this type of mapping, the associative memory is used to store content and address of the memory word. Any block can go into any line of the cache. This means that the word id bits are used to identify which word in the block is needed. But the tags enable the placement of any word at any place in the cache memory. It is considered to be the fastest and the most flexible mapping form.



* SET ASSOCIATIVE MAPPING:

This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are ignored. Set associative address the problem of possible thrashing in the direct mapping method. It does this by saying that instead of having exactly one line that a block can map to in the cache, we will group a few lines together creating a set. Then a block in memory can map to anyone of the lines of specific set.

Set associative Cache mapping combines the best of direct and associative cache mapping techniques.



2. CONCEPTS OF VIRTUAL MEMORY IN COMPUTER ORGANIZATION:

(a) VIRTUAL Memory is a concept used in some large memory space were available, equal to the locality of auxiliary memory. Each address that is generated by the CPU goes through an address mapping from so-called virtual address in main memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main

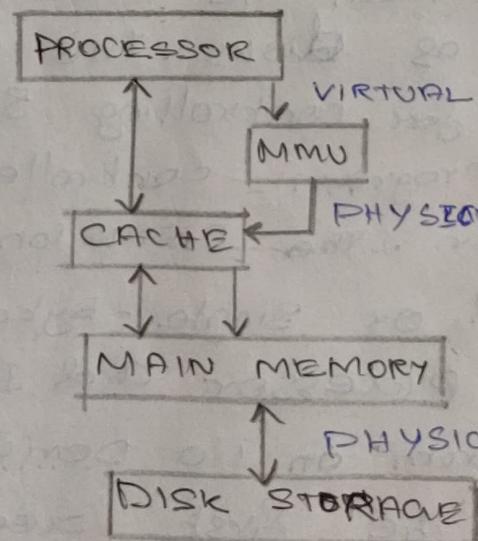
Memory.

* A virtual memory system provides a mechanism for translating program-generated addresses into correct main memory location. This is done dynamically while programs are being executed in the CPU. The translation of mapping is handled automatically by the hardware by means mapping table.

* Under a virtual memory system, program and hence the processor, reference instruction and data in an address space that is independent of the available physical memory space. The binary address that the processor issues for either instruction or data are called VIRTUAL OR LOGIC ADDRESS.

VIRTUAL MEMORY

A Special Hardware Unit called MEMORY MANAGEMENT UNIT [MMU], keeps tracks of which parts of the virtual address space are in the physical memory. When the desired data or instruction are in the main memory the MMU translates the virtual address into the corresponding physical address. Then the requested memory access proceeds in the usual manner.



* ADDRESS TRANSMISSION:

A simple method for translating virtual address into physical address is to assume that all program and data are composed of fixed length units called pages. Pages commonly range from 1k to 16k bytes in length, they constitute the basic unit of information that is transferred is required. Pages should not be small, because the access time of magnetic disk is much longer than the access time of main memory.

* PAGE FAULTS:

When a program generates an access request to a page that is not in the main memory, a page fault is said to have occurred. The entire page must be brought from the disk into memory before access can proceed.

3.

DIRECT MEMORY ACCESS [DMA].

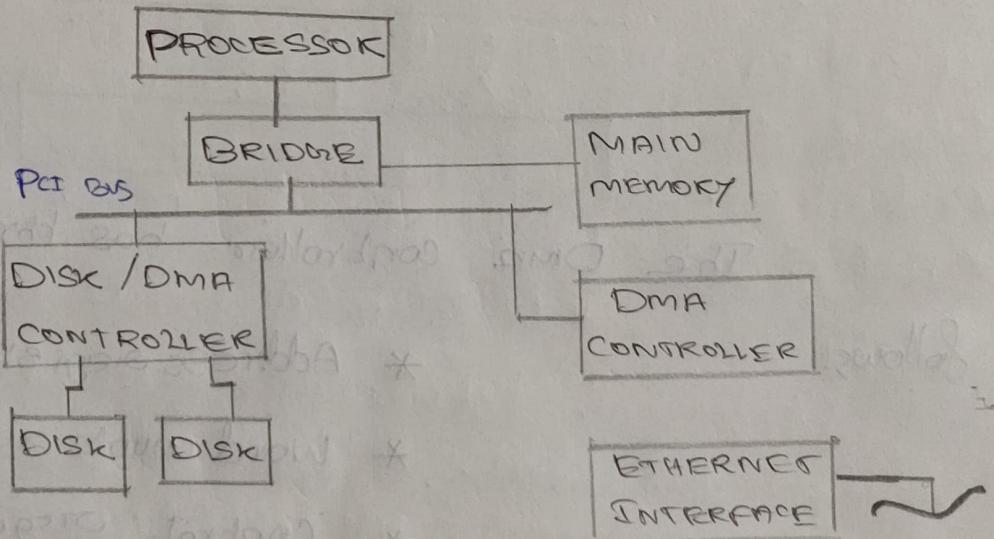
Blocks of data are often transferred between the main memory and I/O devices such as disks. This section discusses a technique for controlling such transfers without frequent program-controller intervention by the processor. The discussion concentrates on

single-word or single-byte data transfer between the processor and I/O devices. Data are transferred from an I/O device to the main memory by the first reading from the I/O devices using instructions.

LOAD R₂, DATA

which loads the data into a processor register than the data read are stored into a memory location. The reverse process take place for transferring data from the memory to an I/O Device.

An alternative approach is used to transfer blocks of data directly between the main memory and I/O devices such as disks. A special control unit is provided by the processor. This approach is called DIRECT MEMORY ACCESS or DMA. The unit that controls DMA transfer is referred as an DMA controller. It may be part of I/O device interface or it may be a separate unit shared by a number of I/O devices.



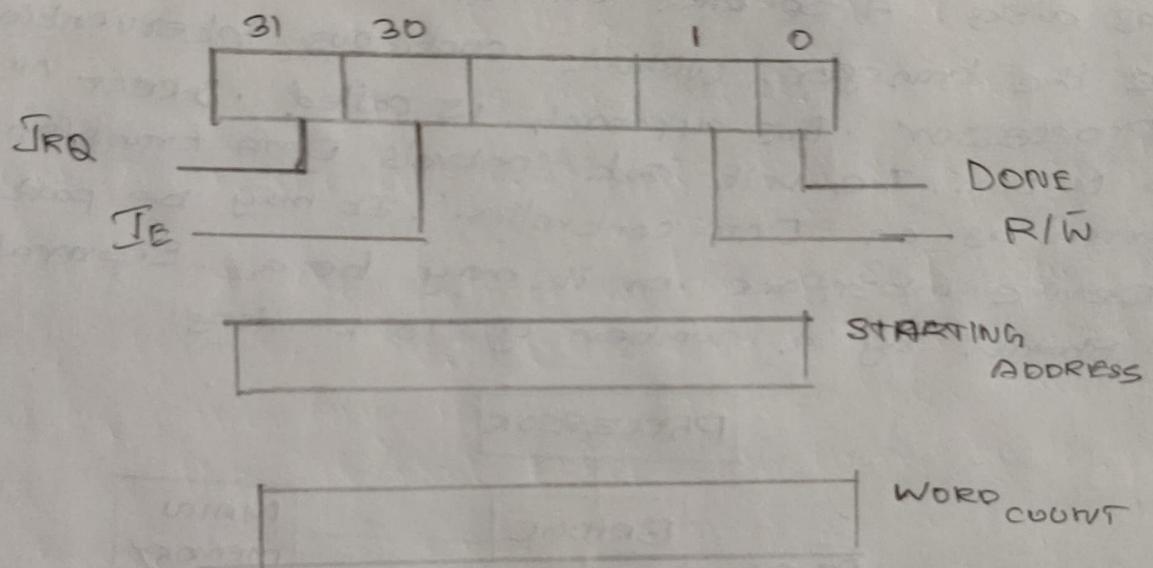
TRANSFER:

Even though the DMA controller transfer data without intervention by the processor. Its operating must be under the control of a program executed by the processor, usually an operating system routine.

To initiate the transfer of a blocks of words, the processor sends to the DMA controller the starting address the number of words in the block.

* Below Images an Example of the DMA Controller Register that are accessed by the processor to initiate data Transfer Operation. Two register are used for Storing address and the word count.

* The third register contains Status and Control flags the R/W determines the direction of the transfer. When this bit is set to 1 by a program instruction, the controller performs a Read operation, it's transfer data from the memory to the I/O devices, it's a perform a write operation.



The DMA controller has three registers as follows.

* Address register

* Word count register

* Control register.

* ADDRESS REGISTER :

It contains the address to specify the desired location in memory

* WORD COUNT REGISTER:

Ib contains the number of

words to be words transferred.

* CONTROL REGISTER:

IC specifies the transfer mode.

The IC, in turns typically asks for CPU Request by placing a status word in a prescribed location in memory to be examined later by the CPU program.

4. DESIGN A PROCEDURE FOR ALU DESIGN WITH NEAT Sketch.

* An Arithmetic logic unit [ALU] is a digital circuit that performs arithmetic and logical operation. The ALU is a fundamental building blocks of the control processing unit of the computer. And even the simplest microprocessor contains one for purpose such as masking timer.

* Mask of a processor operation are performed by one or more ALU. An ALU loads data from input registers executes and stores the results into an output register.

* A control unit tells the ALU what operations to perform on the DMT, mechanisms move data between these other registers and memory.

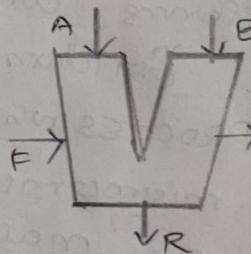
A Typical Schematic Symbol for an ALU.

- A and B are the inputs to the ALU
It is the data to be operated on called
Operands.

- R is the output, results of the
computation.

- F is the code from the control unit.

- D is an output Shaking, also
known as condition code, it
indicates cases such as carry-
in, carry out, overflow, Division
- By - zero, etc.



* SIMPLE OPERATION:

most ALU can perform the following

operation:

* Integer arithmetic operation
(addition, Subtraction, and sometimes multiplication
and division though there is more expensive).

* Bitwise logic operations

[AND, NOR, OR, XOR].

Bit Shifting Operations

[shifting or rotating a word by a specified
number of bits to the left or right, with or
without sign extension].

5. DIFFERENCE BETWEEN SERIAL AND PARALLEL TRANSMISSION.

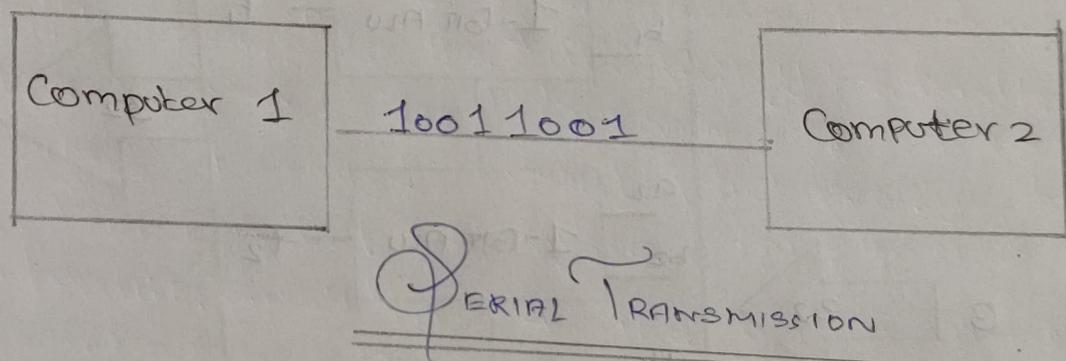
There are two methods used for transferring data between computers which are given below.

* SERIAL TRANSMISSION

* PARALLEL TRANSMISSION

SERIAL TRANSMISSION:

* In Serial Transmission, data-bit flows from one computer to another computer in bi-direction. In this transmission, one bit flows at one clock pulse. In Serial Transmission, 8 bits are transferred at a time having a start and stop bit, i.e. 0 and 1.

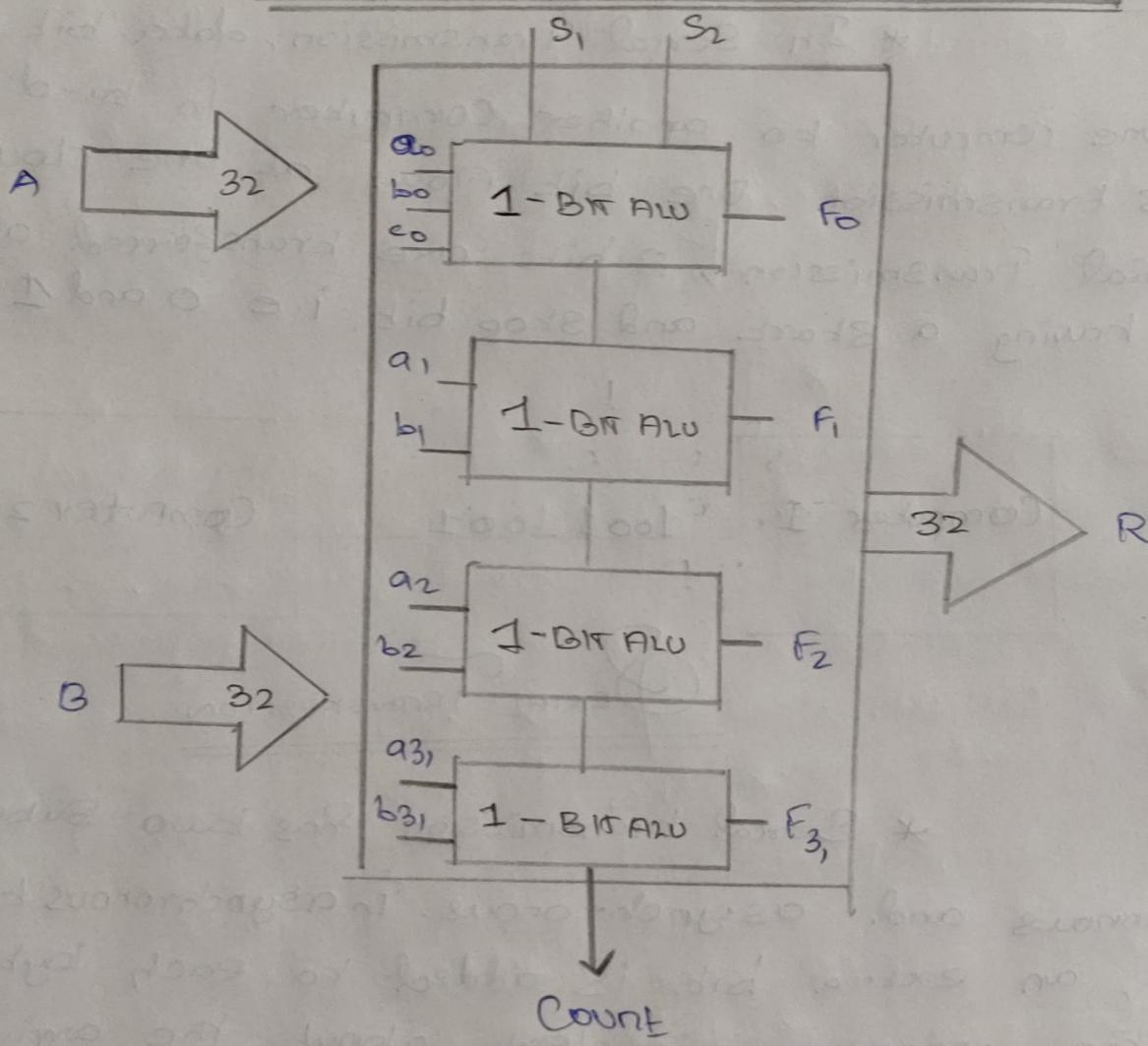


* Serial transmission has two subclasses Synchronous and asynchronous. In asynchronous transmission, an extra bit is added to each byte so that the receiver is aware about the arrival of new data. Usually, 0 is a start bit, and 1 is the stop bit. In synchronous transmission, no extra bit is added rather the data transferred in the form of frames with containing multiple bytes.

Shifts can be interpreted as multiplication by 2 and division by 2.

* A Basic ALU designs involves a collection of ALU "SLICES" which each can perform the specified operation on a single bit. There is one ALU slice for every bit in the operand.

DIAGRAM OF ALU ARCHITECTURE

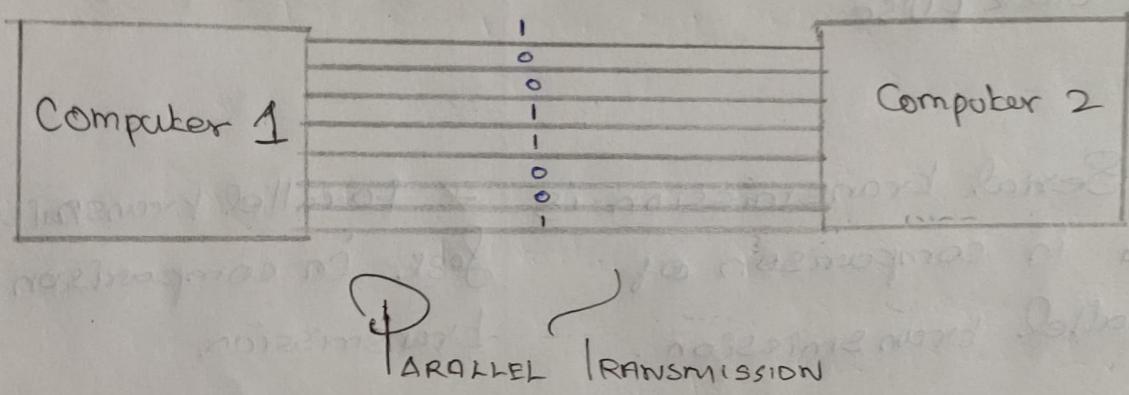


* The ALU is one of the most important components in a microprocessor and it's typically the part in the processor that is designed for.

* Once the ALU is designed, the rest of the microprocessor is implemented to feed operands and control codes to the ALU.

PARALLEL TRANSMISSION :

* In Parallel transmission, many bits are flow together simultaneously from one computer to another computer. Parallel transmission is faster than serial transmission to transmit the bits. Parallel transmission is used for short distance.



* Parallel transmission is advantageous because it conforms to the underlying hardware also, as the electronic devices like computer and communication hardware uses parallel circuitry internally. This is a reason the parallel interface complements the internal hardware well.

* The Installation and troubleshooting is easier than parallel transmission system due to its placement in a single physical cable.

* 4 lines that initiate handshaking

* 8 lines used to communicate and notify errors and

* 8 to transfer data.

SERIAL TRANSMISSION

- * In serial transmission data (bit) flows in bi-directional manner.
- * Serial transmission One bit transferred at one clock pulse.
- * Serial transmission is slow in comparison of parallel transmission.

* Generally, serial transmission is used for long-distance.

* The circuit used in serial transmission is simple.

* This cost is economical and only one number of communication channel required.

PARALLEL TRANSMISSION

- * In parallel transmission data flows in multiple lines.
- * Parallel transmission eight bit transferred at one clock pulse.

* Parallel transmission is fast in comparison of serial transmission.

* Generally, parallel transmission is used for short distance.

* The circuit used in parallel transmission is relatively complex.

* This cost is expensive and N number of communication channels are needed.