

Sardar Patel College Of Engineering, Bakrol, Anand

Name: Shah Yash R.

Enrollment no.: 181240116051

Subject: Computer Organization and Architecture

Subject code: 3140707

Branch: Information Technology

Prepared by: Prof. Nirali Shah

(Assistant Prof. in Information Technology Department)

Teacher’s Signature:

H.O.D. Signature:

**Assignment 3**

**Q-1. Explain subroutine call and return with micro-operation.**

Answer

* A set of Instruction which are used repeatedly in a program can be referred to as Subroutine.
* A call subroutine Instruction calls the subroutine.
* When a Subroutine is required it can be called many times during the Execution of a particular program
* The CALL instruction interrupts the flow of a program by passing control to an internal or external subroutine. An internal subroutine is part of the calling program.
* An external subroutine is another program. The RETURN instruction returns control from a subroutine back to the calling program and optionally returns a value.
* When calling an internal subroutine, CALL passes control to a label specified after the CALL keyword. When the subroutine ends with the RETURN instruction, the instructions following CALL are processed.

**Q-2.** **Explain Flynn’s classification of computer**.

**Answer**

* M.J. Flynn proposed a classification for the organization of a computer system by the number of instructions and data items that are manipulated simultaneously.

### Flynn's classification divides computers into four major groups that are:

1. [Single instruction stream, single data stream (SISD)](https://www.javatpoint.com/sisd)
2. [Single instruction stream, multiple data stream (SIMD)](https://www.javatpoint.com/simd)
3. [Multiple instruction stream, single data stream (MISD)](https://www.javatpoint.com/misd)
4. [Multiple instruction stream, multiple data stream (MIMD)](https://www.javatpoint.com/mimd)

1. SISD

* **SISD** stands for **'Single Instruction and Single Data Stream'**. It represents the organization of a single computer containing a control unit, a processor unit, and a memory unit.
* Instructions are executed sequentially, and the system may or may not have internal parallel processing capabilities.
* Most conventional computers have SISD architecture like the traditional Von-Neumann computers.

### 2. SIMD

* **SIMD** stands for **'Single Instruction and Multiple Data Stream'**. It represents an organization that includes many processing units under the supervision of a common control unit.
* All processors receive the same instruction from the control unit but operate on different items of data.
* The shared memory unit must contain multiple modules so that it can communicate with all the processors simultaneously.

3. MISD

* **MISD** stands for **'Multiple Instruction and Single Data stream'***.*
* MISD structure is only of theoretical interest since no practical system has been constructed using this organization.
* In MISD, multiple processing units operate on one single-data stream. Each processing unit operates on the data independently via separate instruction stream.

4. MIMD

* **MIMD** stands for **'Multiple Instruction and Multiple Data Stream'***.*
* In this organization, all processors in a parallel computer can execute different instructions and operate on various data at the same time.
* In MIMD, each processor has a separate program and an instruction stream is generated from each program.

**Q-3.**  Explain Inter-process communication

Answer

* A process can be of two type:
* Independent process.
* Co-operating process.
* An independent process is not affected by the execution of other processes while a co-operating process can be affected by other executing processes.
* Though one can think that those processes, which are running independently, will execute very efficiently but in practical, there are many situations when co-operative nature can be utilised for increasing computational speed, convenience and modularity. Inter process communication (IPC) is a mechanism which allows processes to communicate each other and synchronize their actions.
* The communication between these processes can be seen as a method of co-operation between them. Processes can communicate with each other using these two ways:

1. Shared Memory

2. Message passing

* An operating system can implement both method of communication. First, we will discuss the shared memory method of communication and then message passing.
* Communication between processes using shared memory requires processes to share some variable and it completely depends on how programmer will implement it.
* One way of communication using shared memory can be imagined like this: Suppose process1 and process2 are executing simultaneously and they share some resources or use some information from other process, process1 generate information about certain computations or resources being used and keeps it as a record in shared memory.
* When process2 need to use the shared information, it will check in the record stored in shared memory and take note of the information generated by process1 and act accordingly.
* Processes can use shared memory for extracting information as a record from other process as well as for delivering any specific information to other process.

**Q-4.**  Explain parallel processing.

**Ans**wer

* Parallel processing is a method in computing of running two or more [processors](https://whatis.techtarget.com/definition/processor) (CPUs) to handle separate parts of an overall task.
* Breaking up different parts of a task among multiple processors will help reduce the amount of time to run a program. Any system that has more than one CPU can perform parallel processing, as well as [multi-core](https://searchdatacenter.techtarget.com/definition/multi-core-processor) processors which are commonly found on computers today.
* Parallel processing is commonly used to perform complex tasks and computations. Data scientists will commonly make use of parallel processing for compute and data-intensive tasks

### How parallel processing works.

* Typically a computer scientist will divide a complex task into multiple parts with a software tool and assign each part to a processor, then each processor will solve its part, and the data is reassembled by a software tool to read the solution or execute the task.
* Typically each processor will operate normally and will perform operations in parallel as instructed, pulling data from the computer’s memory. Processors will also rely on software to communicate with each other so they can stay in sync concerning changes in data values. Assuming all the processors remain in sync with one another, at the end of a task, software will fit all the data pieces together.
* Computers without multiple processors can still be used in parallel processing if they are networked together to form a [cluster](https://whatis.techtarget.com/definition/cluster).

### Types of parallel processing

* There are multiple types of parallel processing, two of the most commonly used types include [SIMD](https://whatis.techtarget.com/definition/Single-Instruction-Multiple-Data-SIMD) and MIMD. SIMD, or single instruction multiple data, is a form of parallel processing in which a computer will have two or more processors follow the same instruction set while each processor handles different data. SIMD is typically used to analyze large data sets that are based on the same specified benchmarks.
* MIMD, or multiple instruction multiple data, is another common form of parallel processing which each computer has two or more of its own processors and will get data from separate data streams.
* Another, less used, type of parallel processing includes MISD, or multiple instruction single data, where each processor will use a different algorithm with the same input data.

**Q-5.**  Explain different types of Interrupts.

**Answer**

* A program *interrupt* refers to the transfer of program control from a currently running program to another service program as a result of an external or internal generated request.
* There are mainly three types of *interrupts*:

1. External interrupts: It arises due to external call from I/O devices. For e.g. I/O devices requesting transfer of data, power failure, etc.
2. Internal interrupts: It arises due to illegal and erroneous use of an instruction or data. For e.g. stack overflow, division by zero, invalid opcode, etc. These are also called *traps*.
3. Software interrupts: It is initiated by executing an instruction. It can be used by the programmer to initiate an interrupt at the desired point in the program.

* External and internal interrupts are initiated from signals that occur in the hardware of the CPU whereas Software interrupts occur from the instructions.

**Q-6.**  Explain paging and address translation with example.

**Answer:**

* **Paging** is a [memory management](https://en.wikipedia.org/wiki/Memory_management) scheme by which a computer stores and retrieves data from [secondary storage](https://en.wikipedia.org/wiki/Computer_data_storage#Secondary_storage)[[a]](https://en.wikipedia.org/wiki/Paging#cite_note-1) for use in [main memory](https://en.wikipedia.org/wiki/Computer_data_storage#Primary_storage).
* In this scheme, the operating system retrieves data from secondary storage in same-size [blocks](https://en.wikipedia.org/wiki/Block_(data_storage)) called [*pages*](https://en.wikipedia.org/wiki/Page_(computer_memory)).
* Paging is an important part of [virtual memory](https://en.wikipedia.org/wiki/Virtual_memory) implementations in modern operating systems, using secondary storage to let programs exceed the size of available physical memory.

### Example-If Logical Address = 31 bit, then Logical Address Space = 231 words = 2 G words (1 G = 230)

**Address Translation:**

* The addresses generated by the machine while executing in user mode are logical addresses. The paging hardware translates these addresses to physical addresses as described below.
* Address translation is done using the Page Table. Logical address/512 gives the logical page number which is the index of the page table entry for the logical page in the page table. Since each page table entry has 2 words, multiplying the logical page number by 2 and adding the base address of the page table given in the PTBR to it gives the location of the page table entry corresponding to the given logical address.

#### Q-7. Explain push and pop operation on stack.

**Answer:**

* The computers which use Stack-based CPU Organization are based on a data structure called**stack**. The stack is a list of data words.
* It uses **Last In First Out (LIFO)** access method which is the most popular access method in most of the CPU.
* A register is used to store the address of the topmost element of the stack which is known as **Stack pointer (SP)**. In this organisation, ALU operations are performed on stack data.
* It means both the operands are always required on the stack. After manipulation, the result is placed in the stack.
* The main two operations that are performed on the operators of the stack are **Push** and **Pop**. These two operations are performed from one end only.

**Push:** This operation results in inserting one operand at the top of the stack and it decrease the stack pointer register It inserts the data word at specified address to the top of the stack

**Pop:** This operation results in deleting one operand from the top of the stack and it increase the stack pointer register. It deletes the data word at the top of the stack to the specified address

**Q-8.** **C**ompare SRAM and DRAM.

Answer

* **SRAM (Static Random Access Memory)** is made up of **CMOS technology** and uses six transistors.
* Its construction is comprised of two cross-coupled inverters to store data (binary) similar to flip-flops and extra two transistors for access control. It is relatively faster than other RAM types such as DRAM. It consumes less power. SRAM can hold the data as long as power is supplied to it.
* It is more faster
* It is smaller
* It is more expensive
* It is used in Cache memory
* Power consumption is low

**DRAM (Dynamic Random Access Memory)** is also a type of RAM which is constructed using capacitors and few transistors. The capacitor is used for storing the data where bit value 1 signifies that the capacitor is charged and a bit value 0 means that capacitor is discharged. Capacitor tends to discharge, which result in leaking of charges.

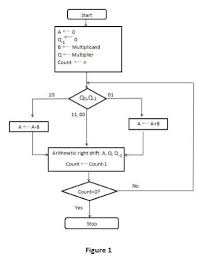
* It is slower
* It is larger
* It is less expensive
* It is used in main memory
* Power consumption is high

**Q-9.**  Explain Booth’s algorithm with flowchart.

**Answer:**

* This is a kind of algorithm which uses a more straightforward approach.
* This algorithm also has the benefit of the speeding up the multiplication process and it is very efficient too.
* Binary multiplication which has signed number uses this type of algorithms named as **Booth's algorithm**.
* As in all multiplication schemes, booth algorithm requires examination **of the multiplier bits** and shifting of the partial product.
* Prior to the shifting, the multiplicand may be added to the partial product, subtracted from the partial product, or left unchanged according to following rules:

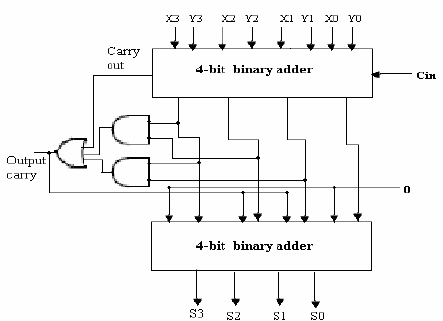
1. The multiplicand is subtracted from the partial product upon encountering the first least significant 1 in a string of 1’s in the multiplier
2. The multiplicand is added to the partial product upon encountering the first 0 (provided that there was a previous ‘1’) in a string of 0’s in the multiplier.
3. The partial product does not change when the multiplier bit is identical to the previous multiplier bit.

****

**Q-10.**  Explain BCD Adder with its block diagram.

**Answer:**

* BCD stand for binary coded decimal. Suppose, we have two 4-bit numbers A and B. The value of A and B can varies from 0(0000 in binary) to 9(1001 in binary) because we are considering decimal numbers.
* The digital systems handle the decimal number in the form of binary coded decimal numbers (**BCD**). A **BCD adder** is a circuit that adds two **BCD** digits and produces a sum digit also in **BCD**. The logic circuit to detect sum greater than 9 can be determined by simplifying the Boolean expression of given truth Table.
* The output will varies from 0 to 18, if we are not considering the carry from the previous sum. But if we are considering the carry, then the maximum value of output will be 19 (i.e. 9+9+1 = 19).
* When we are simply adding A and B, then we get the binary sum. Here, to get the output in BCD form, we will use BCD Adder.

****

**Assignment 4**

**Q-1.**  Give the feature of multiprocessor system.

**Answer:**

Characteristics of Multiprocessors

* A multiprocessor system is an interconnection of two or more CPU, with memory and input-output equipment. As defined earlier, multiprocessors can be put under MIMD category.
* The term multiprocessor is some times confused with the term multicomputers. Though both support concurrent operations, there is an important difference between a system with multiple computers and a system with multiple processors.
* In a multicomputers system, there are multiple computers, with their own operating systems, which communicate with each other, if needed, through communication links.
* A multiprocessor system, on the other hand, is controlled by a single operating system, which coordinate the activities of the various processors, either through shared memory or interprocessor messages.  
    
  **The advantages of multiprocessor systems are:**
* Increased reliability because of redundancy in processors
* Increased throughput because of execution of
* multiple jobs in parallel  
  \
* portions of the same job in parallel
* A single job can be divided into independent tasks, either manually by the programmer, or by the compiler, which finds the portions of the program that are data independent, and can be executed in parallel.
* The multiprocessors are further classified into two groups depending on the way their memory is organized. The processors with shared memory are called tightly coupled or shared memory processors.
* The information in these processors is shared through the common memory. Each of the processors can also have their local memory too. The other class of multiprocessors is loosely coupled or distributed memory multi-processors.
* In this, each processor have their own private memory, and they share information with each other through interconnection switching scheme or message passing.
* The principal characteristic of a multiprocessor is its ability to share a set of main memory and some I/O devices. This sharing is possible through some physical connections between them called the interconnection structures.

**Q-2.**  Differentiate tightly coupled and loosely coupled systems

**Answer:**

**Loosely Coupled Multiprocessor System:**

* It is a type of multiprocessing system in which, There is distributed memory instead of shared memory. In loosely coupled multiprocessor system, data rate is low rather than tightly coupled multiprocessor system. In loosely coupled multiprocessor system, modules are connected through MTS (Message transfer system) network.
* There is distributed memory in loosely coupled multiprocessor system.
* Loosely Coupled Multiprocessor System has low data rate.
* The cost of loosely coupled multiprocessor system is less.
* The cost of loosely coupled multiprocessor system is less.

**Tightly Coupled Multiprocessor System:**

* It is a type of multiprocessing system in which, There is shared memory. In tightly coupled multiprocessor system, data rate is high rather than loosely coupled multiprocessor system. In tightly coupled multiprocessor system, modules are connected through PMIN, IOPIN and ISIN networks.
* There is shared memory, in tightly coupled multiprocessor system.
* Tightly coupled multiprocessor system has high data rate.
* Tightly coupled multiprocessor system is more costly.
* While there is PMIN, IOPIN and ISIN networks.

**Q-3.**  Explain the role of associative memory.

**Answer**

* **Associative memory**searches stored data only by the data value itself rather by an address. This type of search helps in reducing the search time by a large extent.

**How associative memory works**

Given below is a series of steps that depicts working of associative memory:

* Data is stored at the very first empty location found in memory.
* In associative memory when data is stored at a particular location then no address is stored along with it.
* When the stored data need to be searched then only the key (i.e. data or part of data) is provided.
* A sequential search is performed in the memory using the specified key to find out the matching key from the memory.
* If the data content is found then it is set for the next reading by the memory

**Advantages of associative memory**

* Associative memory searching process is fast.
* Associative memory is suitable for parallel searches.

**Disadvantages of associative memory**

* Associative memory is expensive than RAM

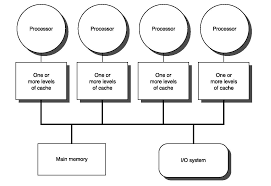
**Q-4.** Draw and explain shared memory architecture of multiprocessor system.

**Answer**

* A system with multiple CPUs “sharing” the same main memory is called multiprocessor.
* In a multiprocessor system all processes on the various CPUs share a unique logical address space, which is mapped on a physical memory that can be distributed among the processors.
* Each process can read and write a data item simply using load and store operations, and process communication is through shared memory.
* It is the hardware that makes all CPUs access and use the same main memory
* Since all CPUs share the address space, only a single instance of the operating system is required.
* When a process terminates or goes into a wait state for whichever reason, the O.S. can look in the process table (more precisely, in the ready processes queue) for another process to be dispatched to the idle CPU.
* On the contrary, in systems with no shared memory, each CPU must have its own copy of the operating system, and processes can only communicate through message passing.
* The basic issue in shared memory multiprocessor systems is memory itself, since the larger the number of processors involved, the more difficult to work on memory efficiently.

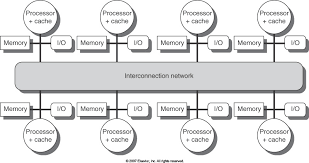
1. Uniform Memory Access (UMA): the name of this type of architecture hints to the fact that all processors share a unique centralized primary memory, so each CPU has the same memory access time.

* Owing to this architecture, these systems are also called Symmetric Shared-memory Multiprocessors (SMP)

****

1. Non Uniform Memory Access (NUMA): these systems have a shared logical address space, but physical memory is distributed among CPUs, so that access time to data depends on data position, in local or in a remote memory (thus the NUMA denomination)

* These systems are also called Distributed Shared Memory (DSM) architectures

****

**Q-5.** How virtual memory helps in increasing the storage capacity of a system?

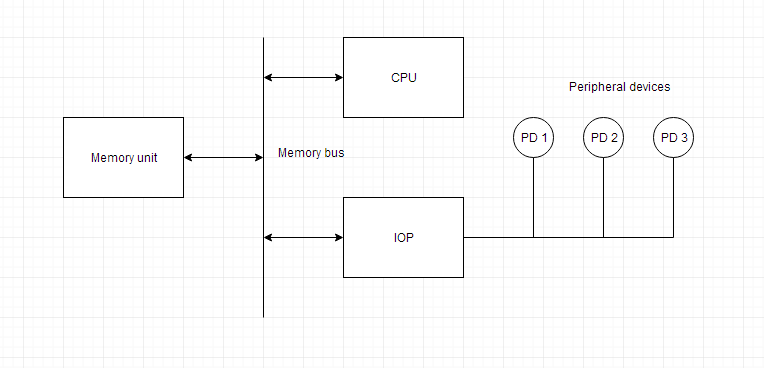
**Answer**

* In [computing](https://en.wikipedia.org/wiki/Computing), **virtual memory** (also **virtual storage**) is a [memory management](https://en.wikipedia.org/wiki/Memory_management_(operating_systems)) technique that provides an "idealized abstraction of the storage resources that are actually available on a given machine"[[1]](https://en.wikipedia.org/wiki/Virtual_memory#cite_note-2) which "creates the illusion to users of a very large (main) memory".[[2]](https://en.wikipedia.org/wiki/Virtual_memory#cite_note-3)
* The computer's [operating system](https://en.wikipedia.org/wiki/Operating_system), using a combination of hardware and software, maps [memory addresses](https://en.wikipedia.org/wiki/Memory_address) used by a program, called [*virtual addresses*](https://en.wikipedia.org/wiki/Virtual_address_space), into *physical addresses* in [computer memory](https://en.wikipedia.org/wiki/Computer_memory). [Main storage](https://en.wikipedia.org/wiki/Main_storage#Primary_storage), as seen by a process or task, appears as a contiguous [address space](https://en.wikipedia.org/wiki/Address_space) or collection of contiguous [segments](https://en.wikipedia.org/wiki/Memory_segmentation).
* The operating system manages [virtual address spaces](https://en.wikipedia.org/wiki/Virtual_address_space) and the assignment of real memory to virtual memory. Address translation hardware in the CPU, often referred to as a [memory management unit](https://en.wikipedia.org/wiki/Memory_management_unit) (MMU), automatically translates virtual addresses to physical addresses.
* Software within the operating system may extend these capabilities to provide a virtual address space that can exceed the capacity of real memory and thus reference more memory than is physically present in the computer.
* The primary benefits of virtual memory include freeing applications from having to manage a shared memory space, increased security due to memory isolation, and being able to conceptually use more memory than might be physically available, using the technique of [paging](https://en.wikipedia.org/wiki/Paging).

**Q-6.**  What is the use of IOP? Explain its communication with CPU.

**Answer**

* An input-output processor (IOP) is a processor with direct memory access capability. In this, the computer system is divided into a memory unit and number of processors.
* Each IOP controls and manage the input-output tasks. The IOP is similar to CPU except that it handles only the details of I/O processing. The IOP can fetch and execute its own instructions. ->These IOP instructions are designed to manage I/O transfers only.
* Below is a block diagram of a computer along with various I/O Processors. The memory unit occupies the central position and can communicate with each processor.
* The CPU processes the data required for solving the computational tasks. The IOP provides a path for transfer of data between peripherals and memory. The CPU assigns the task of initiating the I/O program.
* The IOP operates independent from CPU and transfer data between peripherals and memory.

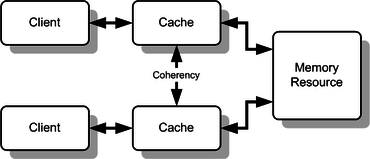
****

* The communication between the IOP and the devices is similar to the program control method of transfer. And the communication with the memory is similar to the direct memory access method.
* In large scale computers, each processor is independent of other processors and any processor can initiate the operation.
* The CPU can act as master and the IOP act as slave processor. The CPU assigns the task of initiating operations but it is the IOP, who executes the instructions, and not the CPU. CPU instructions provide operations to start an ->I/O transfer. The IOP asks for CPU through interrupt.
* Instructions that are read from memory by an IOP are also called *commands* to distinguish them from instructions that are read by CPU. Commands are prepared by programmers and are stored in memory. Command words make the program for IOP. CPU informs the IOP where to find the commands in memory.
* The communication between CPU and input/output devices is implemented using an interface unit.
* In a computer system, data is transferred from an input device to the processor and from the processor to an output device. Each input and output device is provided with a device controller, which is used to manage the working of various peripheral devices.
* Actually, the CPU communicates with the device controllers for performing the I/O operations.
* In the computer system, the interface unit works as an intermediary between the processor and the device controllers of various peripheral devices.
* The interface unit accepts the control commands from the processor and interprets the commands so that they can be easily understood by the device controllers for performing the required operations.
* Hence, the interface unit is responsible for controlling the input and output operations. The processor to I/O devices communication involves two important operations-I/O read and I/O write.

**Q-7.** Signify the cache coherence in memory.

**Answer**

* In [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture), **cache coherence** is the uniformity of shared resource data that ends up stored in multiple [local caches](https://en.wikipedia.org/wiki/Cache_(computing)).
* When clients in a system maintain [caches](https://en.wikipedia.org/wiki/CPU_cache) of a common memory resource, problems may arise with incoherent data, which is particularly the case with [CPUs](https://en.wikipedia.org/wiki/Central_processing_unit) in a [multiprocessing](https://en.wikipedia.org/wiki/Multiprocessing) system.
* In the illustration on the right, consider both the clients have a cached copy of a particular memory block from a previous read. Suppose the client on the bottom updates/changes that memory block, the client on the top could be left with an invalid cache of memory without any notification of the change. Cache coherence is intended to manage such conflicts by maintaining a coherent view of the data values in multiple caches.



**Q-8.** Give the use and role of C

**Answer**

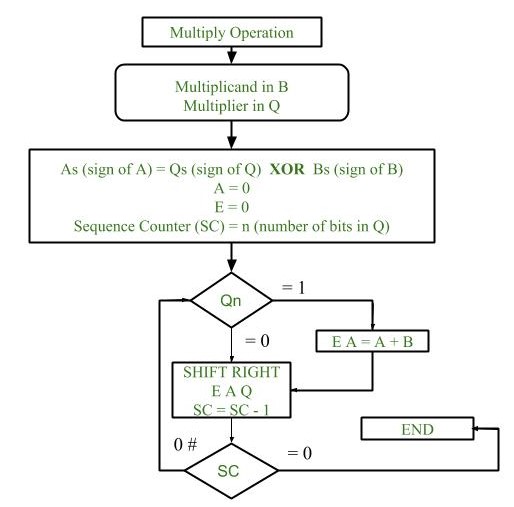
* **C** is a general-purpose programming language that is extremely popular, simple and flexible. It is machine-independent, structured programming language which is used extensively in various applications.
* C was the basic language to write everything from operating systems (Windows and many others) to complex programs like the Oracle database, Git, Python interpreter and more.
* It is said that 'C' is a god's programming language. One can say, C is a base for the programming. If you know 'C,' you can easily grasp the knowledge of the other programming languages that uses the concept of 'C'
* It is essential to have a background in computer memory mechanisms because it is an important aspect when dealing with the C programming language.

**Assignment 5**

**Q-1.** Draw flowchart hardware multiplication algorithm and explain it.

**Answer**

* **Flowchart of Multiplication:**

****

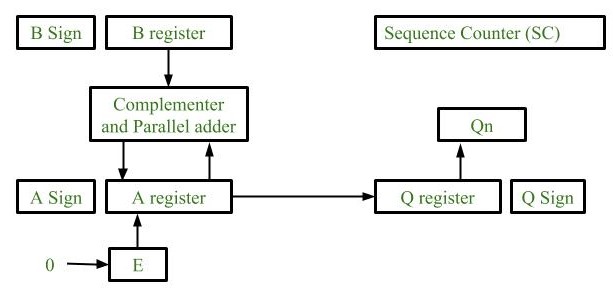
1. Initially multiplicand is stored in B register and multiplier is stored in Q register.
2. Sign of registers B (Bs) and Q (Qs) are compared using **XOR** functionality (i.e., if both the signs are alike, output of XOR operation is 0 unless 1) and output stored in As (sign of A register).

**Note:** Initially 0 is assigned to register A and E flip flop. Sequence counter is initialized with value n, n is the number of bits in the Multiplier.

1. Now least significant bit of multiplier is checked. If it is 1 add the content of register A with Multiplicand (register B) and result is assigned in A register with carry bit in flip flop E. Content of E A Q is shifted to right by one position, i.e., content of E is shifted to most significant bit (MSB) of A and least significant bit of A is shifted to most significant bit of Q.
2. If Qn = 0, only shift right operation on content of E A Q is performed in a similar fashion.
3. Content of Sequence counter is decremented by 1.

* Check the content of Sequence counter. (SC), if it is 0, end the process and the final product is present in register A and Q, else repeat the process

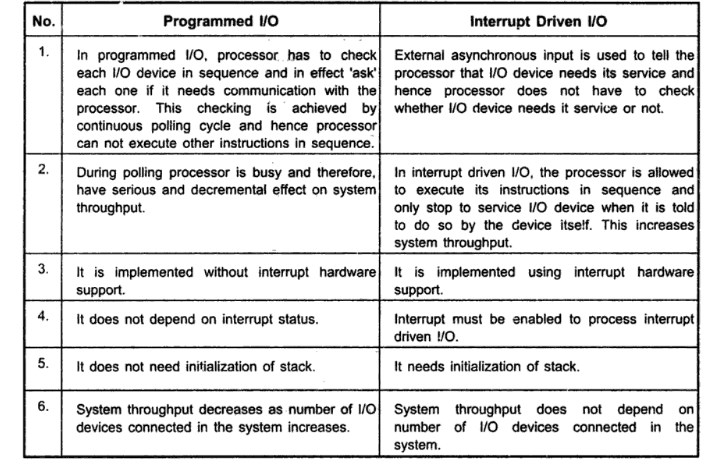
**Hardware Implementation:**

****

1. **Registers:**  
   Two Registers B and Q are used to store multiplicand and multiplier respectively.  
   Register A is used to store partial product during multiplication.  
   Sequence Counter register (SC) is used to store number of bits in the multiplier.
2. **Flip Flop:**  
   To store sign bit of registers we require three flip flops (A sign, B sign and Q sign).  
   Flip flop E is used to store carry bit generated during partial product addition.
3. **Complement and Parallel adder:**  
   This hardware unit is used in calculating partial product i.e, perform addition required.

**Q-2.** Differentiate Programmed I/O and Interrupt initiated I/O.

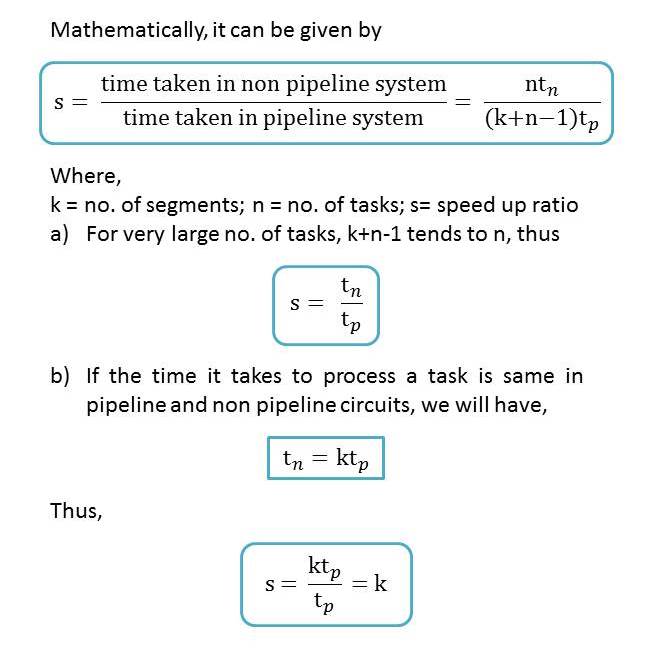
**Answer**

****

**Q-3.** What is speedup? Derive the equation of speedup for k-segment pipeline processing for task.

**Answer**

* In [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture), **speedup** is a number that measures the relative performance of two systems processing the same problem. More technically,
* it is the improvement in speed of execution of a task executed on two similar architectures with different resources.
* The notion of speedup was established by [Amdahl's law](https://en.wikipedia.org/wiki/Amdahl%27s_law), which was particularly focused on [parallel processing](https://en.wikipedia.org/wiki/Parallel_computing). However, speedup can be used more generally to show the effect on performance after any resource enhancement.

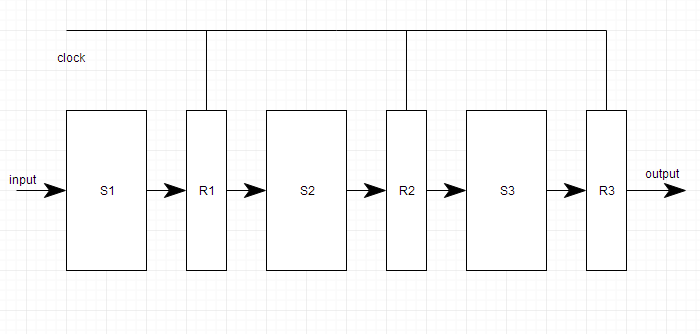
****

**Q-4.** Explain pipeline processing conflict.

**Answer**

* Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as **pipeline processing**.
* ->Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.
* Pipelining increases the overall instruction throughput.
* ->In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it. The output of combinational circuit is applied to the input register of the next segment.

Pipeline system is like the modern day assembly line setup in factories. For example in a car manufacturing industry, huge assembly lines are setup and at each point, there are robotic arms to perform a certain task, and then the car moves on ahead to the next arm.



There are some factors that cause the pipeline to deviate its normal performance. Some of these factors are given below:

### 1. Timing Variations

All stages cannot take same amount of time. This problem generally occurs in instruction processing where different instructions have different operand requirements and thus different processing time.

### 2. Data Hazards

When several instructions are in partial execution, and if they reference same data then the problem arises. We must ensure that next instruction does not attempt to access data before the current instruction, because this will lead to incorrect results**3. Branching**

In order to fetch and execute the next instruction, we must know what that instruction is. If the present instruction is a conditional branch, and its result will lead us to the next instruction, then the next instruction may not be known until the current one is processed**4. Interrupts**

Interrupts set unwanted instruction into the instruction stream. Interrupts effect the execution of instruction.

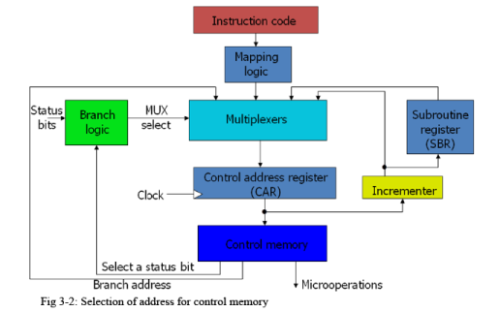
**5. Data Dependency**

It arises when an instruction depends upon the result of a previous instruction but this result is not yet available.

**Q-5.** List out address sequencing capabilities required in control memory.

**Answer**

* The **address sequencing capabilities required** in a **control memory** are: Incrementing **of** the **control address** register. Unconditional branch or conditional branch, depending on status bit conditions. A mapping process from the bits **of** the instruction to an **address** for **control memory.**

****

**Q-6.** Explain Daisy chain priority interrupt.

**Answer**

* The daisy-chaining method involves connecting all the devices that can request an interrupt in a serial manner. This configuration is governed by the priority of the devices. The device with the highest priority is placed first followed by the second highest priority device and so on. The given figure depicts this arrangement.

**WORKING:**  
There is an interrupt request line which is common to all the devices and goes into the CPU.

* When no interrupts are pending, the line is in HIGH state. But if any of the devices raises an interrupt, it places the interrupt request line in the LOW state.
* The CPU acknowledges this interrupt request from the line and then enables the interrupt acknowledge line in response to the request.
* This signal is received at the PI(Priority in) input of device 1.
* If the device has not requested the interrupt, it passes this signal to the next device
* through its PO(priority out) output. (PI = 1 & PO = 1)
* However, if the device had requested the interrupt, (PI =1 & PO = 0)

The device consumes the acknowledge signal and block its further use by placing 0 at its PO(priority out) output.

****

* + The device then proceeds to place its interrupt vector address(VAD) into the data bus of CPU.
  + The device puts its interrupt request signal in HIGH state to indicate its interrupt has been taken care of.

**NOTE:** VAD is the address of the service routine which services that device.

* If a device gets 0 at its PI input, it generates 0 at the PO output to tell other devices that acknowledge signal has been blocked. (PI = 0 & PO = 0)
* Hence, the device having PI = 1 and PO = 0 is the highest priority device that is requesting an interrupt. Therefore, by daisy chain arrangement we have ensured that the highest priority interrupt gets serviced first and have established a hierarchy. The farther a device is from the first device, the lower its priority.

**Q-7.** . Discuss associative mapping and direct mapping in organization of cache memory.

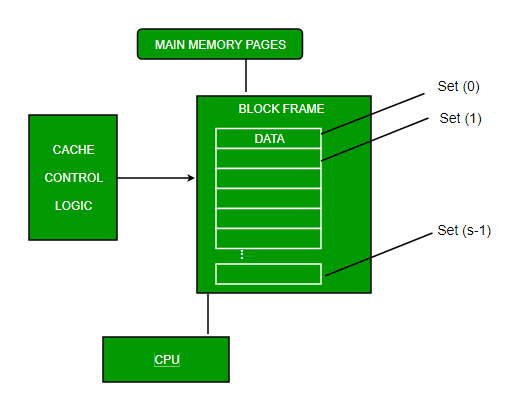
**Answer**

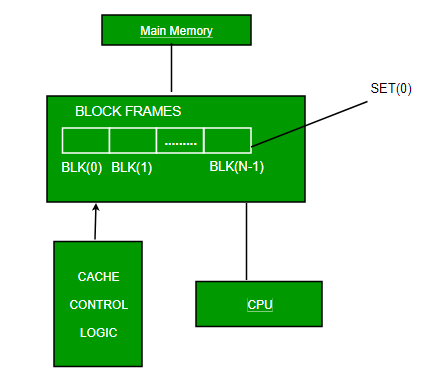
**Direct Mapping –**

* The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. or  
  In Direct mapping, assigne each memory block to a specific line in the cache.
* If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field. The cache is used to store the tag field whereas the rest is stored in the main memory.
* Direct mapping`s performance is directly proportional to the Hit ratio.

**Associative Mapping –**

* In this type of mapping, the associative memory is used to store content and addresses 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 tag becomes all of the remaining bits. This enables 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.

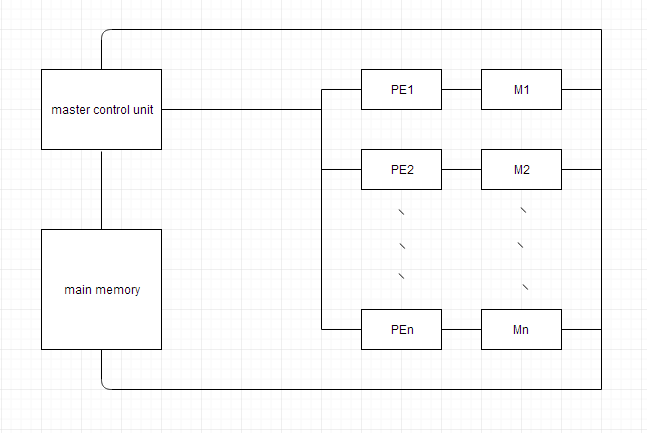
****

****

**Q-8.** Describe SIMD array processor.

**Answer**

* SIMD is the organization of a single computer containing multiple processors operating in parallel. The processing units are made to operate under the control of a common control unit, thus providing a single instruction stream and multiple data streams.
* A general block diagram of an array processor is shown below. It contains a set of identical processing elements (PE's), each of which is having a local memory M. Each processor element includes an **ALU** and **registers**. The master control unit controls all the operations of the processor elements. It also decodes the instructions and determines how the instruction is to be executed.
* -.The main memory is used for storing the program. The control unit is responsible for fetching the instructions. Vector instructions are send to all PE's simultaneously and results are returned to the memory.
* The best known SIMD array processor is the **ILLIAC IV** computer developed by the **Burroughs corps**. SIMD processors are highly specialized computers. They are only suitable for numerical problems that can be expressed in vector or matrix form and they are not suitable for other types of computations.

****

**Q-9.** Write an assembly program to add 10 numbers from memory.

**Answer**

* LXI H, 9000H
* LXI D, 9100H
* MVI C, 09H
* MVI B, 00H
* MOV A, M
* UP:
* INX H
* ADD M
* JNC DOWN
* INR B
* DOWN:
* DCR C
* JNZ UP
* STAX D
* MOV A, B
* STA 9101H
* HLT

**Q-10.** How main memory is useful in computer system? Explain the memory address map of RAM and ROM.

**Answer**

* The main memory acts as the central storage unit in a computer system. It is a relatively large and fast memory which is used to store programs and data during the run time operations.
* ->The primary technology used for the main memory is based on semiconductor integrated circuits. The integrated circuits for the main memory are classified into two major units.

1. RAM (Random Access Memory) integrated circuit chips
2. ROM (Read Only Memory) integrated circuit chips

**RAM integrated circuit chips –**

* ->The primary compositions of a static RAM are flip-flops that store the binary information.
* The nature of the stored information is volatile, i.e. it remains valid as long as power is applied to the system.
* The static RAM is easy to use and takes less time performing read and write operations as compared to dynamic RAM.
* The dynamic RAM exhibits the binary information in the form of electric charges that are applied to capacitors.
* The capacitors are integrated inside the chip by MOS transistors.
* The dynamic RAM consumes less power and provides large storage capacity in a single memory chip.

ROM integrated circuit –

* The primary component of the main memory is RAM integrated circuit chips, but a portion of memory may be constructed with ROM chips.
* A ROM memory is used for keeping programs and data that are permanently resident in the computer.
* Apart from the permanent storage of data, the ROM portion of main memory is needed for storing an initial program called a **bootstrap loader**. The primary function of the **bootstrap loader** program is to start the computer software operating when power is turned on.