1. **What is parallel processing? Explain the benefits of parallel processing.**

- Parallel processing is a technique that are used to provide simultaneous data processing task for the purpose of increasing the computational speed of a computer system .Instead of processing each instruction sequentially in a conventional computer, a parallel processing system is able to perform concurrent data processing to achieve faster execution time .Purpose of parallel processing is to speed up the computer capability and increasing its throughput. The cost of hardware may increase due to its implementation. Some ways of achieving parallel processing are :
  - Using multiple functional unit
  - Pipelining
  - Multiple Processor
- We can achieve parallel processing in multiple ways with the intension of getting the result in more efficient time. It works on the principle of division of the work between multiple functional units or multiple processors or any other ways. It may implement proper instruction pipelining or reducing the instruction like vector processing. Some of the benefits of parallel processing are as follows :

i. Parallel computing saves time, allowing the execution of applications in a shorter time.
ii. Solve Larger Problems in a short interval time.
iii. Compared to serial computing, parallel computing is much better suited for modeling, simulating and understanding complex, real-world phenomena.
iv. Throwing more resources at a task will shorten it's time to completion, with potential cost savings. Parallel computers can be built from cheap, commodity components.
v. Many problems are so large and/or complex that it is impractical or impossible to solve them on a single computer, with limited computer memory.
vi. Many things can be done simultaneously by using multiple computing resources.
vii. We can use computer resources on the Wide Area Network(WAN) or even on the internet.
viii. It has large data storage and quick data computations.

2. **Explain the classifications of parallel processing by M. J. Flynn.**

- M.J flynn considers the organization of computer system by the number of instruction and data item manipulated simultaneously .
- Flynn's classification divides the computer in four major groups :

1. ***Single Instruction stream , Single data stream(SISD)***
2. ***Single Instruction stream , Multiple data stream(SIMD)***
3. ***Multiple Instruction stream ,Single data stream(MISD)***
4. ***Multiple Instruction stream , Multiple data stream(MIMD)***

- The sequence of instruction read from memory is Instruction stream
- The operation performed on the data in the processor is data stream

SISD

- It represents the organization of a single computer containing a control unit, processor unit and a memory unit. Instructions are executed sequentially and the system may or may not have internal parallel processing .Parallel processing can be achieved by pipelining or multiple functional units

SIMD

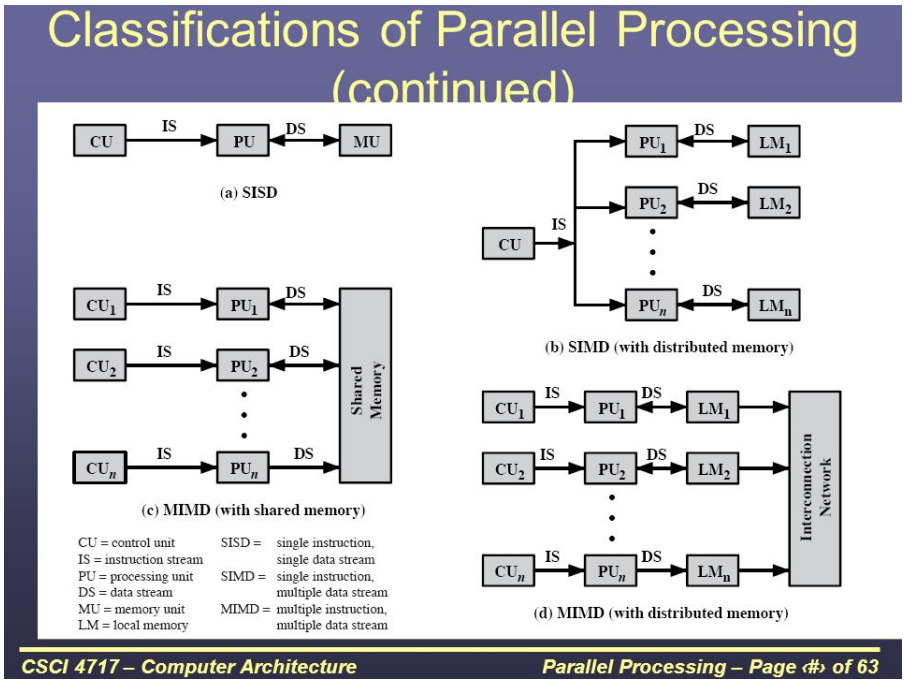
- It represents an organization that includes multiple processing units under the control of a common control unit. All processors receive the same instruction from control unit but operate on different parts of the data. The shared memory unit may contain multiple modules so that it can communicate with all processor simultaneously. Popular examples are vector and array computers.

MISD

- It consists of a single computer containing multiple processors connected with multiple control units and a common memory unit. It is capable of processing several instructions over single data stream simultaneously. MISD structure is only of theoretical interest since no practical system has been constructed using this organization.

MIMD

- It represents the organization which is capable of processing several programs at same time. It is the organization of a single computer containing multiple processors connected with multiple control units and a shared memory unit. The shared memory unit contains multiple modules to communicate with all processors simultaneously. Multiprocessors and multicomputer are the examples of MIMD.



3. What is pipelining? Explain the role of pipelining in computing.

- Pipelining is a process of decomposing a sequential process into sub operation, with each sub process being executed in a special dedicated segment that operates concurrently with other segment. 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.
- Without a pipeline, a computer processor gets the first instruction from memory, decodes the instruction and performs the operation and then again goes to get the next instruction from memory, and so on. The fetching and decoding circuits becomes idle during the execution time. Similarly,

while fetching the instruction, the arithmetic part of the processor is idle. It must wait until it gets the next instruction. Pipelining may be implemented with Arithmetic pipelining or Instruction pipeline. With implementing pipelining, the computer architecture allows the next instructions to be fetched while the decoding and execution of the previous instruction will be in process. Similarly the processor will be implementing the arithmetic operations with functional units and also other operations with other functional unit as per need .The staging of instruction fetching is continuous. The result is an increase in the number of instructions that can be performed during a given time period. So with pipelining we can use the different hardware of the computer for different purposes increasing the performance and time. Arithmetic pipelining can be used for floating point operation and instruction pipeline divides the fetch, decoding and execution for different instruction operation. The role of pipelining can be listed as:

- Pipelining is an implementation technique where multiple instructions are overlapped in execution.
- It has a high throughput
- In pipelining, many instructions are executed at the same time and execution is completed in fewer cycles.
- The pipeline is filled by the CPU scheduler from a pool of work which is waiting to occur. Each execution unit has a pipeline associated with it, so as to have work pre-planned.
- The efficiency of pipelining system depends upon the effectiveness of CPU scheduler.

4. Define instruction pipeline. Explain the four segment instruction pipeline.

- Instruction pipelining is a technique for implementing instruction-level parallelism within a single processor. Pipelining attempts to keep every part of the processor busy with some instruction by dividing incoming instructions into a series of sequential steps (the eponymous "pipeline") performed by different processor units with different parts of instructions processed in parallel. It allows faster CPU throughput than would otherwise be possible at a given clock rate, but may increase latency due to the added overhead of the pipelining process itself.
- In general, the computer needs to process each instruction with the following sequence of steps.

I. Fetch instruction from memory.
II. Decode the instruction.
III. Calculate the effective address.
IV. Fetch the operands from memory.
V. Execute the instruction.
VI. Store the result in the proper place.

A **four-segment instruction** pipeline combines two or more different segments and makes it as a single one. The 4 segments of 4 segment pipeline are :

Segment 1:

- The instruction fetch segment can be implemented using first in, first out (FIFO) buffer.

Segment 2:

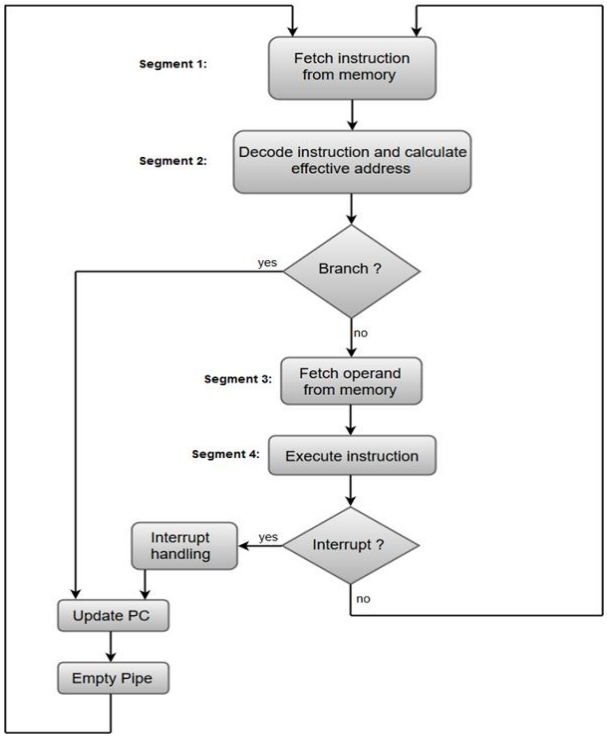
- The instruction fetched from memory is decoded in the second segment, and eventually the effective address is calculated in a separate arithmetic circuit.

Segment 3:

- An operand from memory is fetched in the third segment.

Segment 4:

- The instructions are finally executed in the last segment of the pipeline organization.



5. **Explain the different pipeline hazards (conflicts)?**
   - A pipeline hazard is a situation that prevents an instruction from executing during its designated clock cycle .There are three major types :

i. Structural

- Structural hazard are caused by conflicts among the instruction executing in a pipeline in accessing certain resources. For example suppose a pipelined processor uses a single memory in which the instruction and data are stored. When an instruction is fetched from memory, some preceding instruction may be reading or writing in memory in same pipeline cycle. As two different operations cannot be carried out in memory in the same cycle there is resource conflict .Can be resolved using separate memories for data and instruction.

ii. Data hazard

- A data conflict arises when an instruction depends on the result produced from previous instruction and the result has not yet been produced by the time it is needed. It can be resolved using one of the method as a)hardware interlock b)operand Forwarding c)Delayed Load

iii. Control hazard

- Usually each time the next instruction is fetched by incrementing the PC by the size of instruction. But when a branch or a jump occurs, fetching should start from the address of the branch. This is called as branch hazard or control hazard. It can be resolved as a) Pre fetch the target instruction b)Branch Prediction c) Delayed Branch

6. How to handle the branch instruction in pipeline? Explain.

- One of the major problems designing an instruction pipe line instruction pipelining is the occurrence of branch instruction .The branch instruction breaks the normal sequence of the instruction stream causing difficulties in the operating of the instruction pipeline. The primary problem is the conditional branches instruction until the instruction is actually executed, it is impossible to determine whether the branch will be taken or not.

  A variety of approaches have been taken for dealing with conditional branches:

  Prefetch Branch target:
  In this method ,the branch instruction which are to be executed are prefetched to detect if any error are present in the branch before execution.
  Loop Buffer :
  The loop buffer is very high speed memory device When every loop is to be executed in the computer. The complete loop will be transferred in to the loop buffer memory and will be executed in the cache memory.
  Branch prediction:
  Before the branch is to be executed the instruction along with the error checking condition are checked .Therefore we are not going in unnecessary branch.
  Delayed Branch:
  The delayed branch concept is same as the delayed load process in which we are delaying the execution of branch process before all the data is fetched by the system for beginning the CPU.

7. Define vector processing. Explain the application areas of vector processing.

- Due to the advancement in technology some problem requires vast number of computation and it takes a conventional computer days or even weeks to complete. In many science and engineering applications, the problem can be formulated in terms of vector and matrices that lend themselves to vector processing.

  Some of the application areas of vector processing are:

  i. Long range weather forecasting:
  ii. Petroleum exploration
  iii. Seismic data analysis
  iv. Medical diagnosis
  v. Aerodynamics and space simulation

vi. Artificial Intelligence and expert system
vii. Mapping the human genome
viii. Image Processing

8. Explain the characteristics of multiprocessor system.

- A multi processor is an interconnection of two or more CPUs with memory and input-output equipment. The term processor in multiprocessor system can mean either a central processing unit(CPU) or an input-output processor(IOP).Multiprocessors are classified as multiple instruction stream multiple data stream(MIMD) systems. The main difference between a multicomputer and a multiprocessor system is, in multi computer system computers are interconnected with each other to form a network and they may or may not communicate with each other. A multiprocessor system is controlled by one operating system that provides interaction between processors and all the components of the system cooperate in the solution of a problem. Multiprocessing improves the reliability of the system so that a failure in one part has little effect in the system. If a fault causes one processor to fail, another processor can be used to perform the functions of disabled processor. The benefit derived from a multiprocessor organization is an improved system performance. Computations can proceed in parallel in one of the two ways:
- Multiple independent jobs can be made to operate in parallel.
- A single job can be partitioned into multiple parallel tasks.
- Multiprocessors are classified by the way their memory is organized.

Shared Memory

- Multiprocessor system with common shared memory is classified as a shared memory or tightly coupled multiprocessor. Information can be passed by placing that in common global-memory.

Distributed Memory

- Another type is the distributed memory or loosely-coupled system. Each processor element in a loosely coupled system has its own private local memory.

9. Explain the interconnection structure of multiprocessor system.

Multiprocessor system is an interconnection of two or more CPUs with memory and input-output equipment The components that forms multiprocessor are CPUs IOPs connected to input –output devices, and memory unit that may be partitioned into a number of separate modules. Multiprocessor are classified as multiple instruction stream, multiple data stream (MIMD) system.
There are several physical forms available for establishing an interconnection network.

I. Time-shared common bus
II. Multiport memory

III. Crossbar switch
IV. Multistage switching network
V. Hypercube system

Time-shared common bus

A common-bus multiprocessor system consists of a number of processors connected through a common path to a memory unit. Disadv.:• o Only one processor can communicate with the memory or another processor at any given time. o As a consequence, the total overall transfer rate within the system is limited by the speed of the single path A more economical implementation of a dual bus structure is depicted in Fig. below.• Part of the local memory may be designed as a cache memory attached to the CPU.•

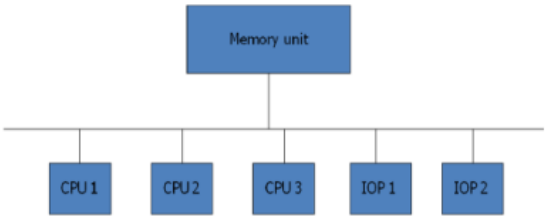


Fig: Time shared common bus organization

Multiport memory

A multiport memory system employs separate buses between each memory module and each CPU. The module must have internal control logic to determine which port will have access to• memory at any given time. Memory access conflicts are resolved by assigning fixed priorities to each memory port.

Adv : The high transfer rate can be achieved because of the multiple paths.

Disadv.: It requires expensive memory control logic and a large number of cables and connections.

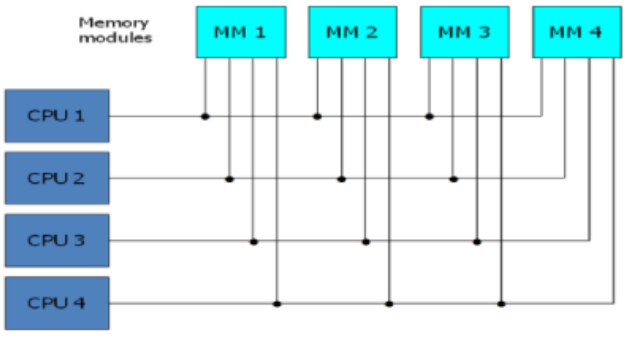


Fig: Multiport memory organization

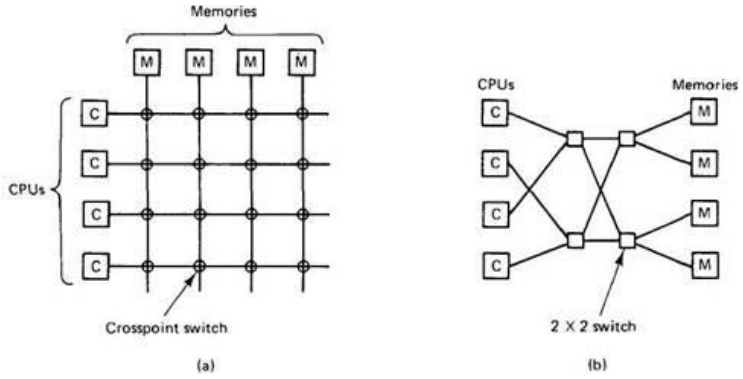
Crossbar switch

Consists of a number of crosspoints that are placed at intersections between processor buses and memory module paths. The small square in each crosspoint is a switch that determines the path from a processor• to a memory module.

Advantage:

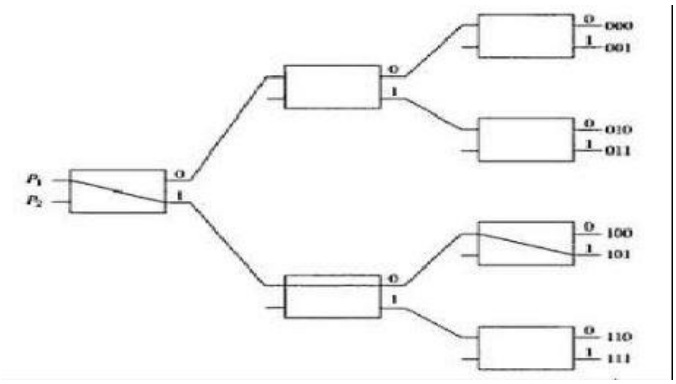
Supports simultaneous transfers from all memory modules

Disadv.:
The hardware required to implement the switch can become quite large and complex. Below fig. shows the functional design of a crossbar switch connected to one memory module.



Multistage switching network

The basic component of a multistage network is a two- input , two- output interchange switch.



10. Write short notes on: Arithmetic pipeline, vector operations, matrix multiplications.

Arithmetic pipeline:

- Arithmetic Pipelines are mostly used in high-speed computers. They are used to implement floating-point operations, multiplication of fixed-point numbers, and similar computations used in scientific problems. Data enters a stage of pipeline, which performs some arithmetic operation on data. The results are then passed to the next stage, which performs its operation and so on until the final computation has been performed.

- To understand the concepts of arithmetic pipeline in a more convenient way, let us consider an example of a pipeline unit for floating-point addition and subtraction. The inputs to the floating-point adder pipeline are two normalized floating-point binary numbers defined as:The input to the Floating Point Adder pipeline is: $X = A*2^a$ *and* $Y = B*2^b$ Here A and B are mantissas (significant digit of floating point numbers), while **a** and **b**are exponents.
- The floating point addition and subtraction is done in 4 parts:

i. Compare the exponents.
ii. Align the mantissas.

iii. Add or subtract mantissas
iv. Produce the result.

- Lets take a example X=0.9504 * 10^3 and Y=0.8200 * 10^2

1. Compare the exponents.

- 3-2=1 so, they are not equal . In 3 and 2 the larger 3 is chosen

2. Align the mantissas.

- X=0.9504 * 10^3 and Y= Y=0.0820 * 10^3

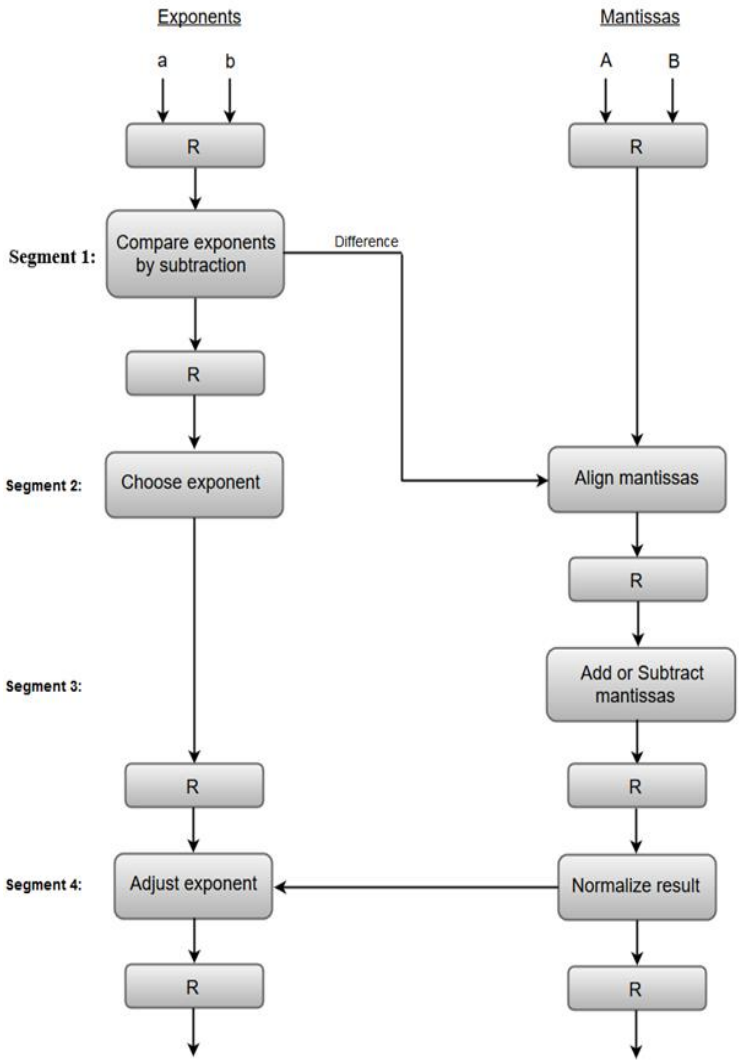
3. Add or subtract mantissas

- 1.0324* 10^3

4. Produce the result.

- 0.10324* 10^4

Pipeline organization for floating point addition and subtraction:



Vector Operations

- A vector processor is a central processing unit that can work on an entire vector in one instruction. The instruction to the processor is in the form of one complete vector instead of its element. Vector processors are used because they reduce the time for fetching and executing multiple instruction replaced with the single instruction.
- Many scientific problems require arithmetic operations on large arrays of numbers .These number are usually formulated as vector and metrics of floating point number.A vector is an ordered set of on-dimensional array of data items .A vector V of length n is represented as row vector by:

  V = [V1 V2 V3 …….. Vn]

- The element Vi of vector V is written as V(I) and the index I refers to a memory address or register where the number is stored.

- Let us consider the program in assembly language that two vectors A and B of length 100 and put the result in vector C.

```
      Initialize I = 0
20    Read A(I)
      Read B(I)
      Store C(I) = A(I) + B(I)
      Increment I = I + 1
      If I ≤ 100 go to 20
      Continue
```

Computer capable of vector processing eliminates the overhead associated with the time it takes to fetch and execute the instructions in the program loop. It allows operations to be specified with a single vector instruction of the form:

$$C(1:100) = A(1:100) + B(1:100)$$

- The vector instruction includes the initial address of the operands, the length of the vectors, and the operation to be performed , all in one composite instruction. The addition is done with a pipelined floating-point adder.

Operation code	Base address Source 1	Base address Source 2	Base address destination	Vector length

Matrix Multiplications:

- A matrix is a set of numerical and non-numerical data arranged in a fixed number of rows and column. Matrix multiplication is an important multiplication design in parallel computation. Applications of matrix multiplication in computational problems are found in many fields like scientific computing and pattern recognition and also counting the paths through a graph. The multiplication of N * N matrices consists of n^2 inner product and N^3 multiply add operation. Let us consider the multiplication of two 3*3 matrix A and B.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

The product matrix C is a 3×3 matrix whose elements are related to the elements of A and B by the inner product:

$$c_{ij} = \sum_{k=1}^{3} a_{ik} \times b_{kj}$$

3×3 Matrix Multiplication Formula:

The product of two matrices $A = (a_{ij})_{3\times3}$ and $B = (a_{ij})_{3\times3}$ is determined by the following formula

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{pmatrix}$$

- Here C11 requires three multiplication and three additions . The total number of multiplications or additions required to compute the matrix product is 9*3= 27 .