## Vector processor

- Vector processing is used there where a vast no.of computation will take long time to complete that may be one day or one week or anything
- Vector processing can be used in different areas like-
    1. Long range weather forecasting
    2. Medical diagnosis
    3. Space flight simulation
    4. Genome mapping
    5. Image processing etc.
- The above areas take long time to complete or can't be completed in stipulated time. So to achieve high performance vector processing is used.
- Vector operation-

a) Large array of numbers are formulated as vectors. A vector is an ordered set of one-dimensional array of data items.

b) V= [V1 V2 V3.....Vn ], Where n= length of vector. It may be represented as a column vector if the data items are listed in a column.

c) Operation on vector must be broken down into single computation with subscripted variables.

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

e) Vector processing eliminates the overhead associated with the time it takes to fetch & execute the instruction in the program loop. It allows operation to be specified with a single vector instruction.

f) Vector instruction includes:-

The following 3 fields are included in one single composite vector instruction.

1. Initial address of operand
2. Length of vector
3. Operation to be performed

g) Instruction format for vector processing-

| OPERATION CODE | BASE ADDRESS SOURCE 1 | BASE ADDRESS SOURCE 2 | BASE ADDRESS DESTINATION | VECTOR LENGTH |
|---|---|---|---|---|

h) Example of vector instruction-

C (1:100) = A (1:100) + B (1:100).Here initial address, operation & vector length are in same instruction.
Where 1 = initial address
100 = length of vector
+ = operation to be performed.

i) Example of a (n × n) matrix multiplication using vector processing.

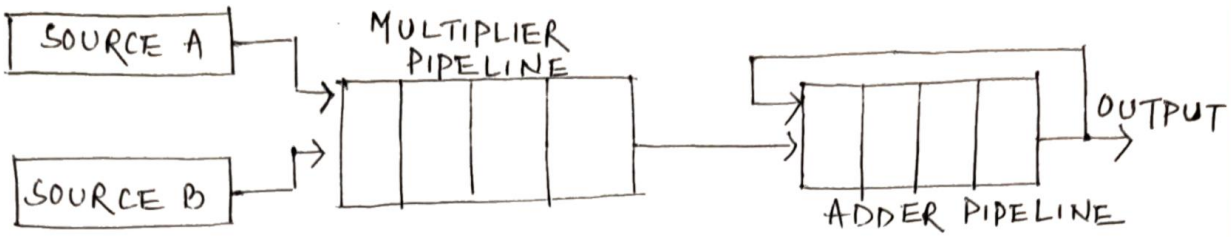
- When two (n × n) matrix get multiplied then the total no.of inner product in the resultant matrix = **n^2** & total no.of multiply-add operation in the resultant matrix is **n^3.**

$$C = A \times B$$

$$C_{ij} = \sum_{K=1}^{n} A_{iK} * B_{Kj}$$

$$\begin{bmatrix} C_{11} & C_{12} \rightarrow C_{1n} \\ C_{21} & C_{22} \rightarrow C_{2n} \\ \downarrow & \\ C_{n1} & C_{n2} \rightarrow C_{nn} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \rightarrow A_{1n} \\ A_{21} & A_{22} \rightarrow A_{2n} \\ \downarrow & \\ A_{n1} & A_{n2} \rightarrow A_{nn} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \rightarrow B_{1n} \\ B_{21} & B_{22} \rightarrow B_{2n} \\ \downarrow & \\ B_{n1} & B_{n2} \rightarrow B_{nn} \end{bmatrix}$$

- The arithmetic pipelines required for calculating inner products are - one multiplier pipeline & one adder pipeline.




- The above two pipeline consists of 4 segments because n=4 in (n × n) matrix. All the segments of both multiplier & adder pipeline is initialized to '0'. So output of adder pipeline is '0' for first eight clock cycle until both pipes are full.
- Ai & Bi are brought at a rate one pair per cycle. After first four cycles, the products begin to be added to the output of adder pipeline. So at the end of 8th clock cycle the adder pipeline contains A1B1 to A4B4. And the multiplier pipeline contains A5B5 to A8B8 at clock cycle 9.
- At clock cycle 9 the output of adder pipeline A1B1 & the output of multiplier pipeline A5B5 are added i:e A1B1+A5B5. At clock cycle 10 the output of adder pipeline is A2B2+A6B6 & so on.
- Continuing in the above manner, C (resultant matrix) = A1B1+A5B5+A9B9+.....
  A2B2+A6B6+A10B10+.....
  ↓
- When there are no more terms to be added then the system inserts four '0' in multiplier pipeline.
- The adder pipeline has one partial product in each of its four segments. The four partial sums are then added to form the final sum.

## Array processor

An array processor is a processor that performs computation on large arrays of data. Array processor is of two types. They are-

1. Attached array processor
2. SIMD array processor

Attached array processor

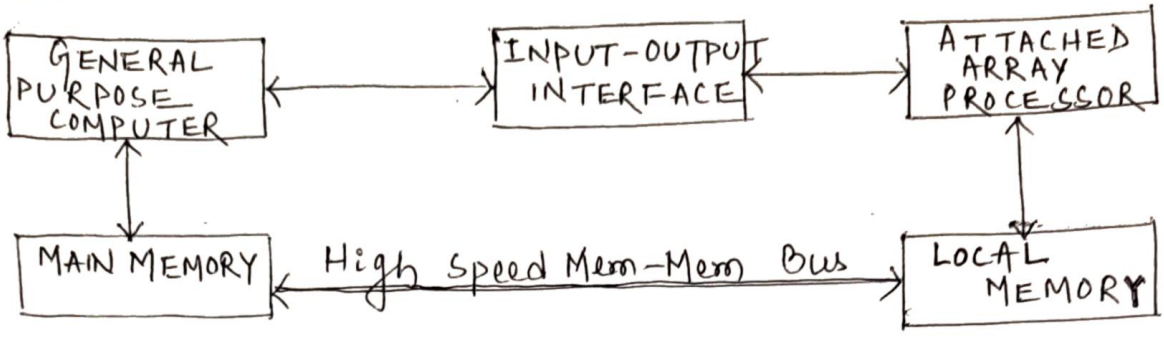
1. It is an auxillary processor attached to a general purpose computer.
2. It manipulates vectors.

SIMD array processor

1. It is a processor that has a single instruction, multiple data stream organization.
2. It also manipulates vectors.

Attached array processor

- It acts as a peripheral for a conventional computer. Its main purpose is to enhance the performance of computer by providing vector processing.
- It is used in complex scientific application.
- Attached array processor achieves high performance by means of parallel processing with multiple functional units. It includes arithmetic unit containing one or more floating-point adders & multipliers pipelines.
- The aim of attached array processor is to provide vector manipulation to a conventional computer at a low cost.

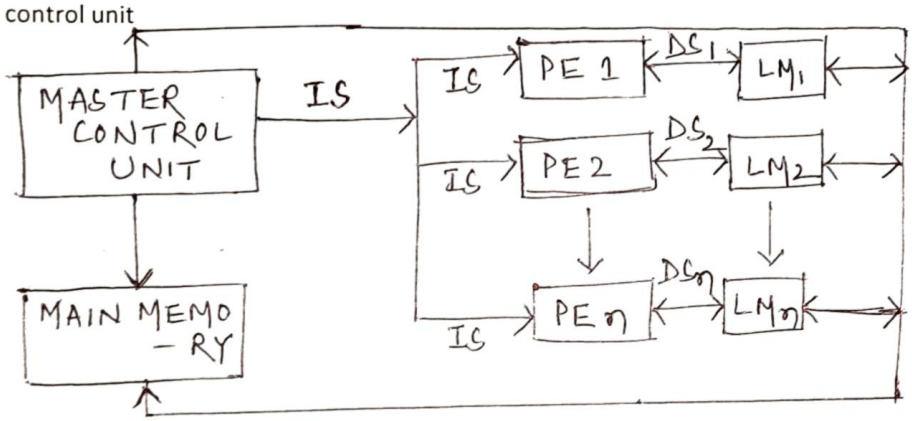



(The above figure shows interconnection of attached array processor with conventional computer)

- The array processor is connected through an IO interface to the computer.
- Data from attached array processor transferred from main memory to a local memory through a high speed bus.

SIMD array processor

- It is suitable for numerical operations which are expressed in vector.
- SIMD array processor is a computer with multiple functional units operating in parallel.
- The processing units are synchronized to perform the same operation under the control of a common control unit




- It contains a set of identical processing elements (PE), each having a local memory M. Each processing elements includes ALU, a floating-point arithmetic unit & registers.
- Main memory is used for storage of program (instructions & data)
- The master control unit controls the operations in processor elements. Master control unit decodes the instructions & determine how the instructions is to be executed.
- Vector instructions are broadcasted to all the PE, where each PE executes the same instruction on the operand stored in its local memory. Vector operands are distributed to the local memories before parallel execution of the instruction.

- Masking is used to control the status of each PE. Each PE has a flag that is SET when PE is active or in use & RESET if the PE is inactive or not in use. So just to reflect which PEs are currently in the state of active and how many are idle masking bits are used.
- E.g. if a SIMD array processor consist of 100 no.of PE & out of 100, 60 PEs are active. Now, consider a recent vector instruction requires 30 PEs. So the master control unit will find out 30 inactive PE from the rest 40.
- Example of processing a vector instruction through SIMD array processor, Ci=Ai+Bi

*Step1*- The master control unit stores ith component of Ai & Bi in local memory Mi. i.e.

A1 & B1 stored in M1

A2 & B2 stored in M2 & so on.

*Step2*-Master control unit then broadcast the floating-point addition instruction Ci=Ai+Bi to all the PEs to perform the addition operation in parallel.

*Step3*-The value of Ci are stored in a fixed location of each local memory. i.e.

C1 stored in M1

C2 stored in M2 & so on.