Vector Processing

Introduction

- There is a class of **computational problems** that are beyond the capabilities of a conventional computer.
- These problems are characterized by the fact that they require a vast number of computations that will **take a conventional computer days or even weeks to complete.**
- In many science and engineering applications, the problems can be formulated in terms of vectors and matrices that lend themselves to **vector processing.**

Applications

- Computers with vector processing capabilities are in demand in specialized applications.
- The following are representative application areas where vector processing is of the utmost importance.

1. Long-range weather forecasting
2. Petroleum explorations
3. Seismic data analysis
4. Medical diagnosis
5. Aerodynamics and space flight simulations
6. Artificial intelligence and expert systems
7. Mapping the human genome
8. Image processing

Without sophisticated computers, many of the required computations cannot be completed within a reasonable amount of time.

To achieve the **required level of high performance** it is necessary to utilize the fastest and most reliable hardware and apply innovative procedures from vector and parallel processing techniques.

Vector Operations

- Many scientific problems require arithmetic operations on large arrays of numbers.
- These numbers are usually formulated as vectors and matrices of floating-point numbers.
- A vector is an ordered set of a one-dimensional array of data items.
- A vector V of length n is represented as a row vector by

$$V =[V1\ V2\ V3\ .\ .\ .\ Vn]$$

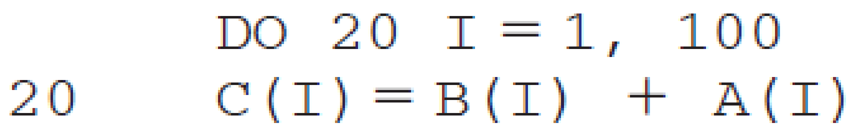
- It may be represented as a column vector if the data items are listed in a column.

Vector Operations

- A conventional sequential computer is capable of processing operands one at a time.
- Consequently, operations on vectors must be broken down into single computations with subscripted variables.
- 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.

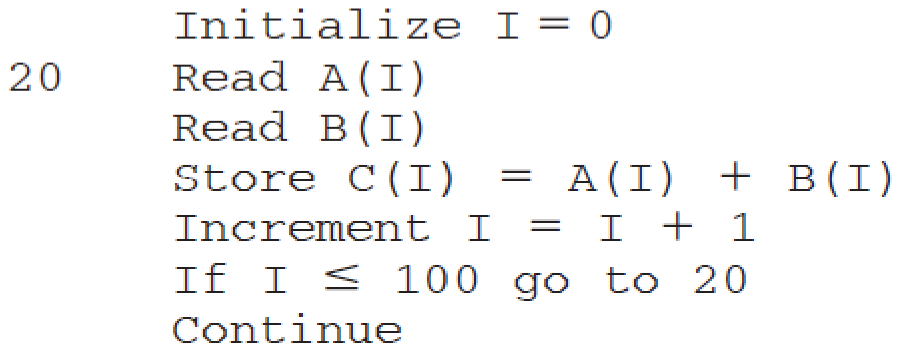
Vector Operations

- To examine the difference between a conventional scalar processor and a vector processor, consider the following Fortran DO loop:



```
      DO 20 I = 1, 100
20    C(I) = B(I)  +  A(I)
```

This is a program for adding two vectors *A* and *B* of length 100 to produce a vector *C*. This is implemented in machine language by the following sequence of operations.



```
        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
```

This constitutes a program loop that reads a pair of operands from arrays *A* and *B* and performs a floating-point addition. The loop control variable is then updated and the steps repeat 100 times.

Computer capable of Vector Processing

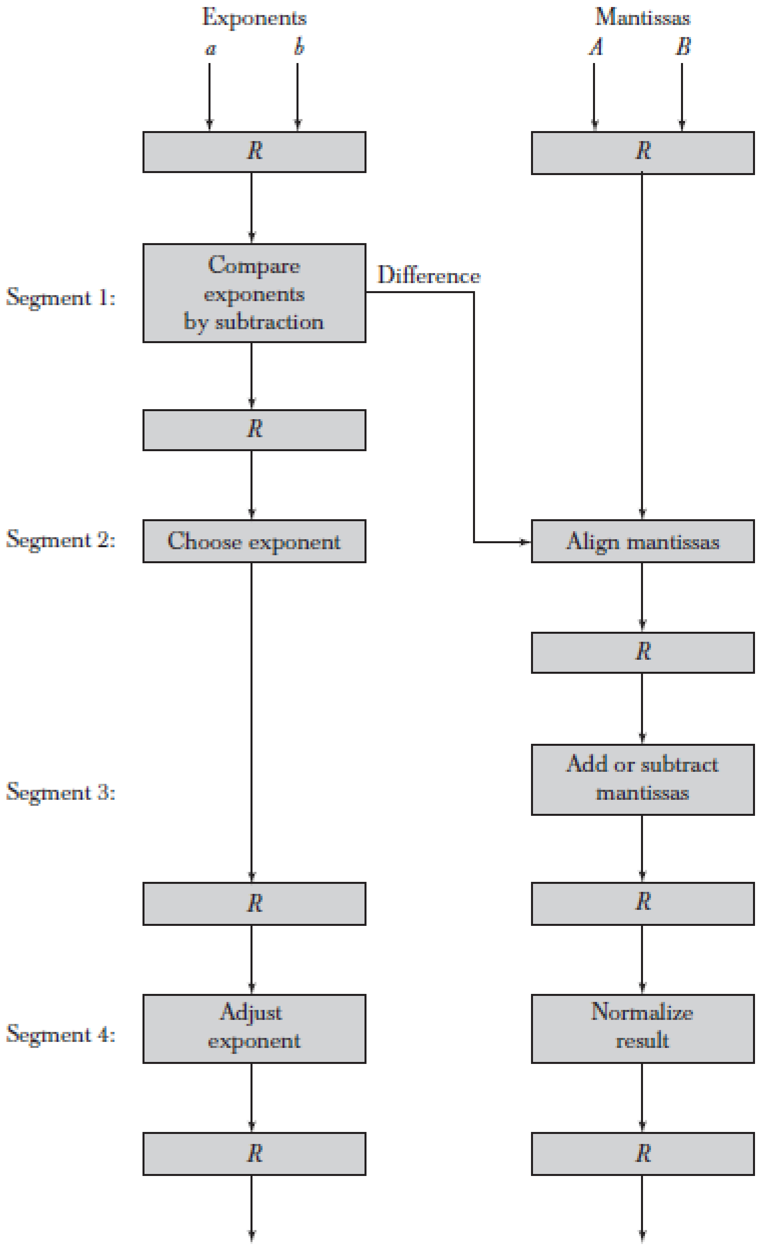
- A computer capable of vector processing eliminates the overhead associated with the time it takes to fetch and execute the instructions in the program loop.
- It allows operations to be specified with a single vector instruction of the form

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

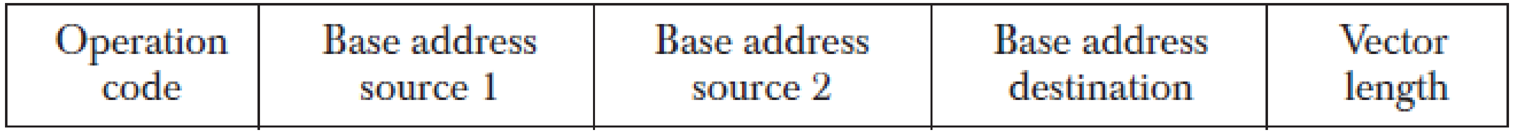
The vector instruction includes the initial address of the operands, the length of the vectors, and the operation to be performed, all in one composite instruction.

The addition is done with a **pipelined floating-point adder**

Pipeline for floating-point addition and subtraction.



Instruction format for vector processor.



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

This is essentially a three-address instruction with three fields specifying the base address of the operands and an additional field that gives the length of the data items in the vectors.

This assumes that the vector operands reside in memory.

It is also possible to design the processor with a large number of registers and store all operands in registers prior to the addition operation.

In that case the base address and length in the vector instruction specify a group of CPU registers.

Matrix Multiplication

- Matrix multiplication is one of the most computational intensive operations performed in computers with vector processors.

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

This requires three multiplications and (after initializing c_{11} to 0) three additions.

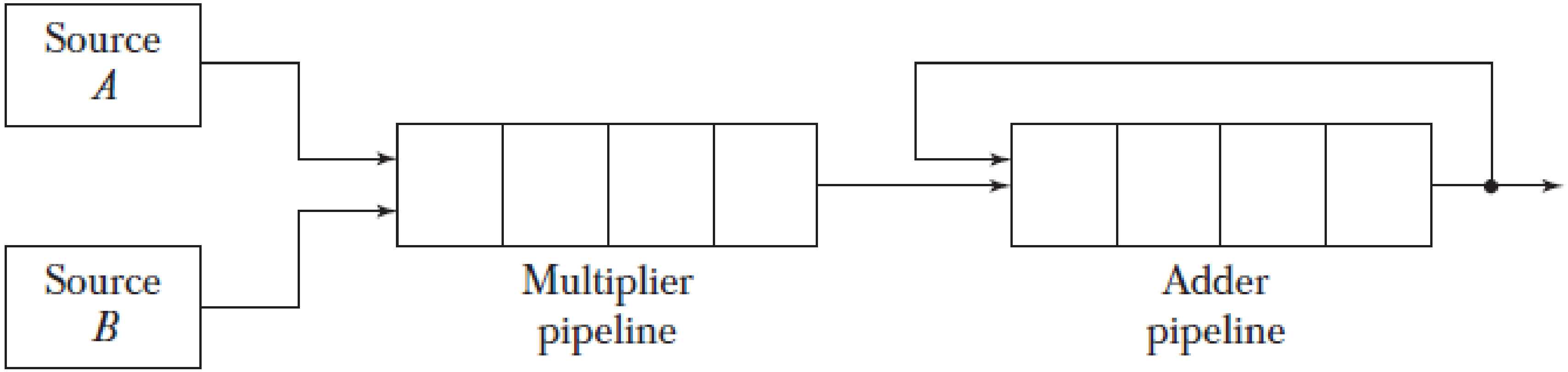
The total number of multiplications or additions required to compute the matrix product is 9 X 3 = 27.

In general, the inner product consists of the sum of *k* product terms of the form

$$C = A_1 B_1 + A_2 B_2 + A_3 B_3 + A_4 B_4 + \cdots + A_k B_k$$

In a typical application *k* may be equal to 100 or even 1000.

Pipeline for calculating an inner product.



$$
\begin{aligned}
C = {} & A_1B_1 + A_5B_5 + A_9B_9 + A_{13}B_{13} + \cdots \\
& + A_2B_2 + A_6B_6 + A_{10}B_{10} + A_{14}B_{14} + \cdots \\
& + A_3B_3 + A_7B_7 + A_{11}B_{11} + A_{15}B_{15} + \cdots \\
& + A_4B_4 + A_8B_8 + A_{12}B_{12} + A_{16}B_{16} + \cdots
\end{aligned}
$$

The values of A and B are either in memory or in processor registers.

The floating-point multiplier pipeline and the floating-point adder pipeline are assumed to have four segments each.

All segment registers in the multiplier and adder are initialized to 0.

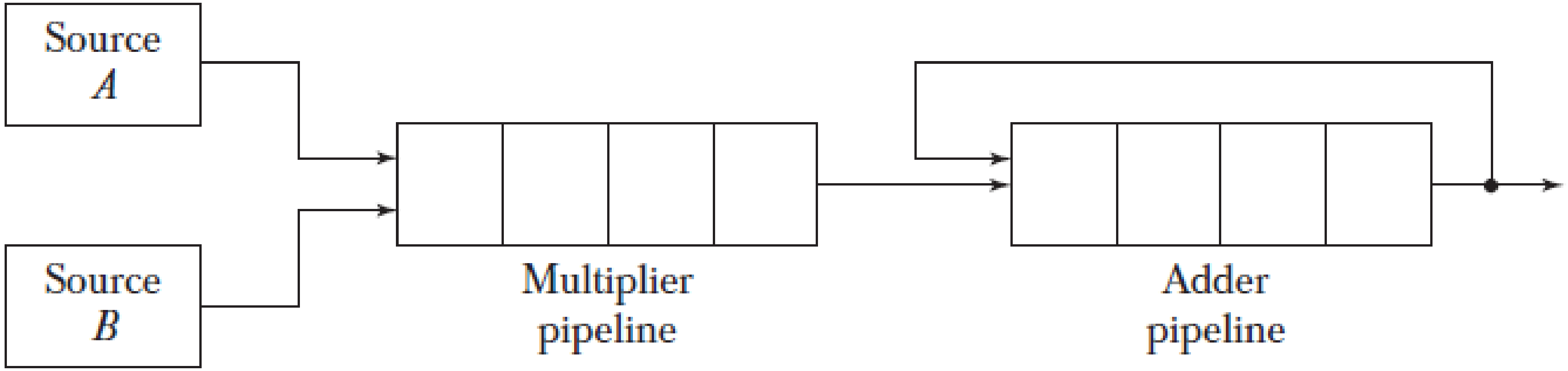
Therefore, the output of the adder is 0 for the first eight cycles until both pipes are full.

A_i and B_i pairs are brought in and multiplied at a rate of one pair per cycle.

After the first four cycles, the products begin to be added to the output of the adder.

During the next four cycles 0 is added to the products entering the adder pipeline.

Pipeline for calculating an inner product.



$$
\begin{aligned}
C = {} & A_1B_1 + A_5B_5 + A_9B_9 + A_{13}B_{13} + \cdots \\
& + A_2B_2 + A_6B_6 + A_{10}B_{10} + A_{14}B_{14} + \cdots \\
& + A_3B_3 + A_7B_7 + A_{11}B_{11} + A_{15}B_{15} + \cdots \\
& + A_4B_4 + A_8B_8 + A_{12}B_{12} + A_{16}B_{16} + \cdots
\end{aligned}
$$

At the end of the eighth cycle, the first four products *A*1 *B*1 through *A*4 *B*4 are in the four adder segments, and the next four products, *A*5 *B*5 through *A*8 *B*8, are in the multiplier segments.

At the beginning of the ninth cycle, the output of the adder is *A*1 *B*1 and the output of the multiplier is *A*5 *B*5.

Thus the ninth cycle starts the addition *A*1 *B*1 +*A*5 *B*5 in the adder pipeline.

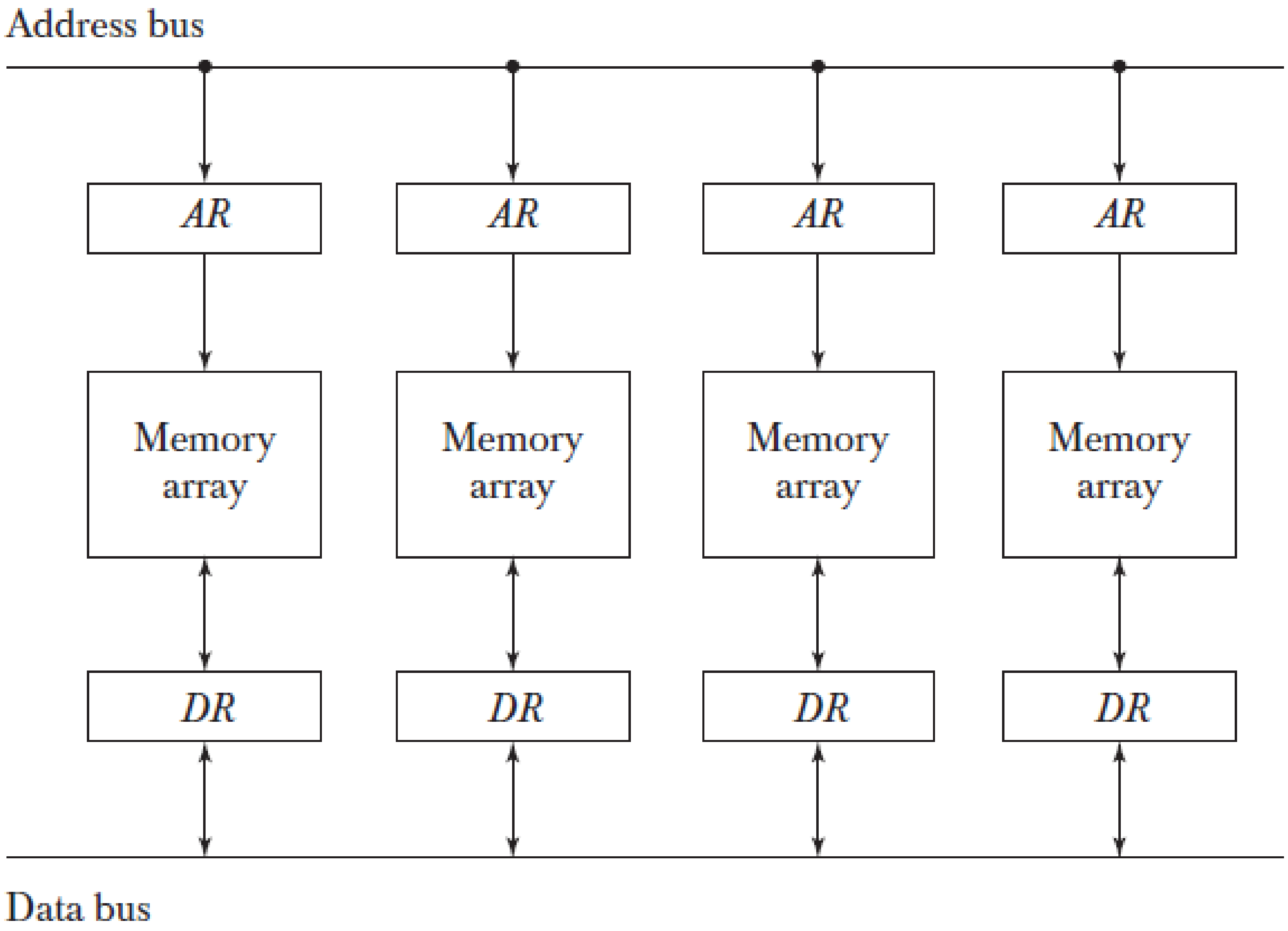
The tenth cycle starts the addition *A*2 *B*2 +*A*6 *B*6, and so on.

When there are no more product terms to be added, the system inserts four zeros into the multiplier pipeline.
The adder pipeline will then have one partial product in each of its four segments, corresponding to the four sums listed in the four rows in the above equation.
The four partial sums are then added to form the final sum.

Memory Interleaving

- Pipeline and vector processors often require simultaneous access to memory from two or more sources.
- An instruction pipeline may require the fetching of an instruction and an operand at the same time from two different segments.
- Similarly, an arithmetic pipeline usually requires two or more operands to enter the pipeline at the same time.
- Instead of using two memory buses for simultaneous access, the memory can be partitioned into a number of modules connected to a common memory address and data buses.
- A memory module is a memory array together with its own address and data registers.

Multiple module memory organization.



Explanation

- Each memory array has its own address register AR and data register DR.
- The address registers receive information from a common address bus and the data registers communicate with a bidirectional data bus.
- The two least significant bits of the address can be used to distinguish between the four modules.
- The modular system permits one module to initiate a memory access while other modules are in the process of reading or writing a word and each module can honor a memory request independent of the state of the other modules.

Explanation

- The advantage of a modular memory is that it allows the use of a technique called **interleaving.**
- In an interleaved memory, different sets of addresses are assigned to different memory modules.
- For example, in a two-module memory system, the even addresses may be in one module and the odd addresses in the other.
- When the number of modules is a power of 2, the least significant bits of the address select a memory module and the remaining bits designate the specific location to be accessed within the selected module.

Explanation

- A modular memory is useful in systems with pipeline and vector processing.
- A vector processor that uses an n-way interleaved memory can fetch n operands from n different modules.
- By staggering the memory access, the effective memory cycle time can be reduced by a factor close to the number of modules.
- A CPU with instruction pipeline can take advantage of multiple memory modules so that each segment in the pipeline can access memory independent of memory access from other segments.

Superscalar Processors

- A superscalar processor architecture has a form of parallelism on a single chip allowing the system as a whole to run much faster than it would otherwise be able to at a given clock speed.
- **A superscalar architecture fetches, executes, and returns results from more than one instruction during a single pipeline stage.**
- A scalar processor processes one data item at a time.
- In a vector processor, by contrast, a single instruction operates simultaneously on multiple data items.
- A superscalar processor is sort of a mixture of the two.
- Each instruction processes one data item, but there are multiple processing units so that multiple instructions can be processing separate data items at the same time.

Superscalar Processors

- A superscalar processor normally has an execution rate in excess of one instruction per machine cycle.
- But just processing multiple instructions at the same time does not make an architecture superscalar.
- Simple pipelining, where a processor may be loading an instruction while doing arithmetic for the previous one and storing the results from the one before that (thus executing three instructions at the same time) is not superscalar processing.
- In a superscalar processor, there are several functional units of the same type, along with additional circuitry to dispatch instructions to the units.
- For instance, most superscalar designs include more than one arithmetic and logic unit.
- The **dispatcher** reads instructions from memory and decides which ones can be run in parallel, dispatching them to the two units.

Superscalar Processors

- Seymour Cray’s CDC 6600 from 1965 is often mentioned as the first superscalar design.
- The Intel i960CA (1988) and the AMD 29000-series 29050 (1990) microprocessors were the first commercial single-chip superscalar microprocessors.
- The RS6000 from IBM was released in 1990 and was the world’s first superscalar RISC microprocessor.
- Intel followed in 1993 with the Pentium, which with its two ALUs brought the x86 world into the superscalar era.

Supercomputers

- A commercial computer with vector instructions and pipelined floating-point arithmetic operations is referred to as a supercomputer.
- Supercomputers are very powerful, high-performance machines used mostly for scientific computations.
- To speed up the operation, the components are packed tightly together to minimize the distance that the electronic signals have to travel.
- Supercomputers also use special techniques for removing the heat from circuits to prevent them from burning up because of their close proximity.

Supercomputers

- The instruction set of supercomputers contains the standard data transfer, data manipulation, and program control instructions of conventional computers.
- This is augmented by instructions that process vectors and combinations of scalars and vectors.
- A supercomputer is a computer system best known for its high computational speed, fast and large memory systems, and the extensive use of parallel processing.
- **It is equipped with multiple functional units and each unit has its own pipeline configuration.**
- Although the supercomputer is capable of general-purpose applications found in all other computers, it is specifically optimized for the type of numerical calculations involving vectors and matrices of floating-point numbers.

Supercomputers

- Supercomputers are not suitable for normal everyday processing of a typical computer installation.
- They are limited in their use to a number of scientific applications, such as numerical weather forecasting, seismic wave analysis, and space research.
- They have limited use and limited market because of their high price.

Supercomputers

- A measure used to evaluate computers in their ability to perform a given number of floating-point operations per second is referred to as flops.
- The term megaflops is used to denote million flops and gigaflops to denote billion flops.
- A typical supercomputer has a basic cycle time of 4 to 20 ns.
- If the processor can calculate a floating-point operation through a pipeline each cycle time, it will have the ability to perform 50 to 250 megaflops.
- This rate would be sustained from the time the first answer is produced and does not include the initial setup time of the pipeline.

History- “Super Computer”

- The first supercomputer developed in 1976 is the Cray-1 supercomputer.
- It uses vector processing with 12 distinct functional units in parallel.
- Each functional unit is segmented to process the incoming data through a pipeline.
- All the functional units can operate concurrently with operands stored in the large number of registers (over 150) in the CPU.
- A floating-point operation can be performed on two sets of 64-bit operands during one clock cycle of 12.5 ns.
- This gives a rate of 80 megaflops during the time that the data are processed through the pipeline.
- It has a memory capacity of 4 million 64-bit words.

History- “Super Computer”

- The memory is divided into 16 banks, with each bank having a 50-ns access time.
- This means that when all 16 banks are accessed simultaneously, the memory transfer rate is 320 million words per second.
- Cray research extended its supercomputer to a multiprocessor configuration called Cray X-MP and Cray Y-MP.
- The new Cray-2 supercomputer is 12 times more powerful than the Cray-1 in vector processing mode.

History- “Super Computer”

- Another early model supercomputer is the Fujitsu VP-200. It has a scalar processor and a vector processor that can operate concurrently.
- Like the Cray supercomputers, a large number of registers and multiple functional units are used to enable register-to-register vector operations.
- There are four execution pipelines in the vector processor, and when operating simultaneously, they can achieve up to 300 megaflops.
- The main memory has 32 million words connected to the vector registers through load and store pipelines.
- The VP-200 has 83 vector instructions and 195 scalar instructions.
- The newer VP-2600 uses a clock cycle of 3.2 ns and claims a peak performance of 5 gigaflops.

Array Processors

- An array processor is a processor that performs computations on large arrays of data.
- The term is used to refer to two different types of processors.
- ***An attached array processor*** is an auxiliary processor attached to a general-purpose computer.
  - It is intended to improve the performance of the host computer in specific numerical computation tasks.
- ***An SIMD array processor*** is a processor that has a single-instruction multiple-data organization.
  - It manipulates vector instructions by means of multiple functional units responding to a common instruction.
- Although both types of array processors manipulate vectors, their internal organization is different.

Attached Array Processor

- An attached array processor is designed as a peripheral for a conventional host computer, and its purpose is to enhance the performance of the computer by providing vector processing for complex scientific applications.
- It achieves high performance by means of parallel processing with multiple functional units.
- It includes an arithmetic unit containing one or more pipelined floating-point adders and multipliers.
- The array processor can be programmed by the user to accommodate a variety of complex arithmetic problems.

Attached array processor with host computer.

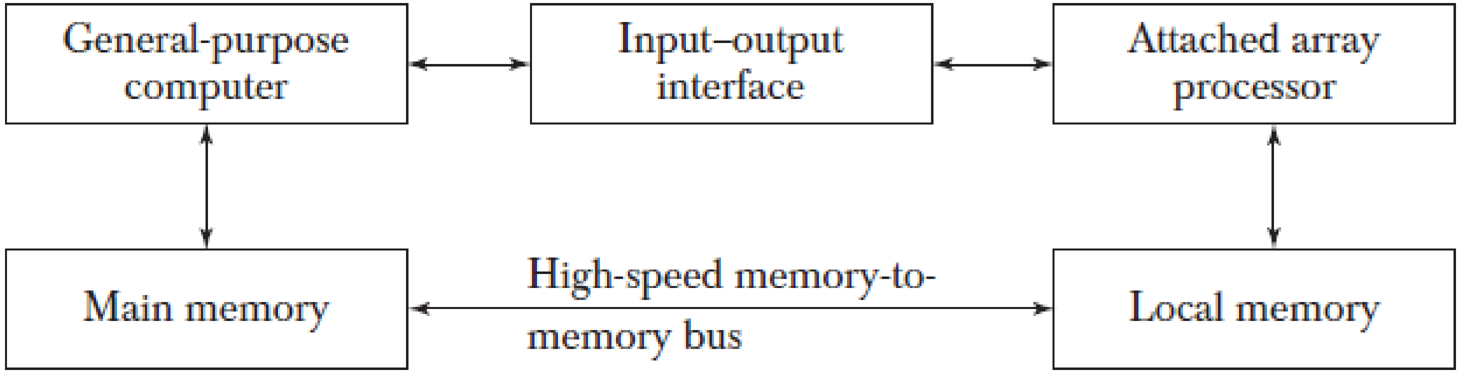


Figure shows the interconnection of an attached array processor to a host computer.

The host computer is a general-purpose commercial computer and the attached processor is a back-end machine driven by the host computer.

The array processor is connected through an input–output controller to the computer and the computer treats it like an external interface.
The data for the attached processor are transferred from main memory to a local memory through a high-speed bus.
The general-purpose computer without the attached processor serves the users that need conventional data processing.
The system with the attached processor satisfies the needs for complex arithmetic applications.

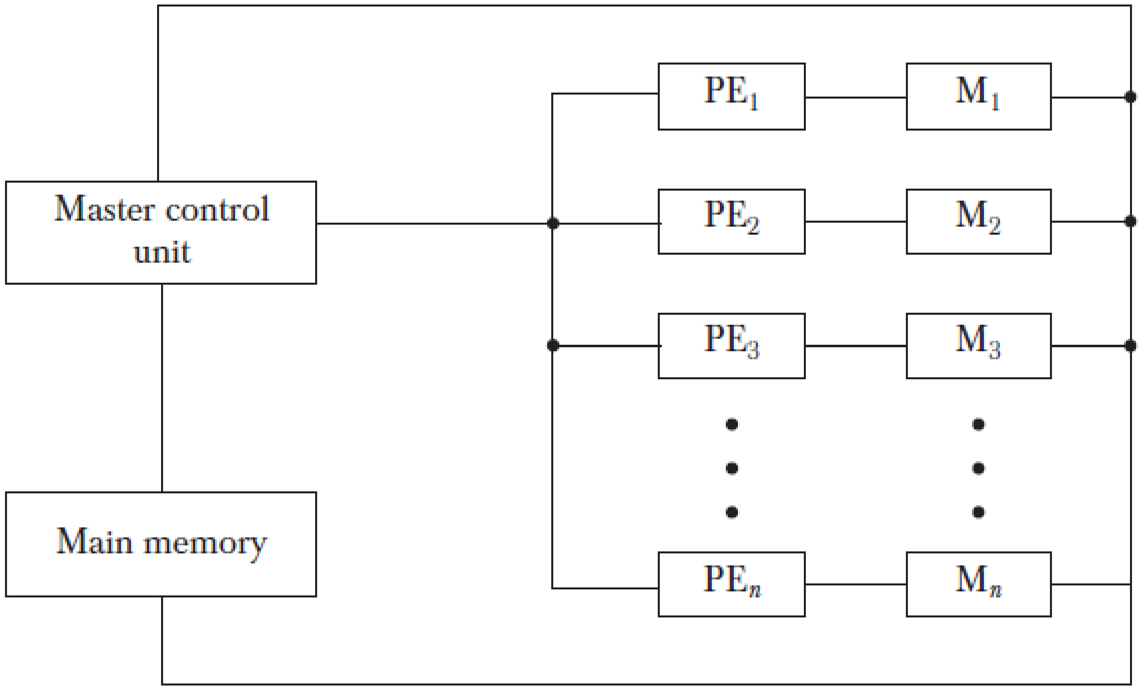
Attached Array Processor

- Some manufacturers of attached array processors offer a model that can be connected to a variety of different host computers.
- For example, when attached to a VAX 11 computer, the FSP-164/MAX from Floating-Point Systems increases the computing power of the VAX to 100 megaflops.
- The objective of the attached array processor is to provide vector manipulation capabilities to a conventional computer at a fraction of the cost of supercomputers.

SIMD Array Processor

- An SIMD array processor is a computer with multiple processing units operating in parallel.
- The processing units are synchronized to perform the same operation under the control of a common control unit, thus providing a single instruction stream, multiple data stream (SIMD) organization.

A general block diagram of an array processor



It contains a set of identical processing elements (PEs), each having a local memory *M.*

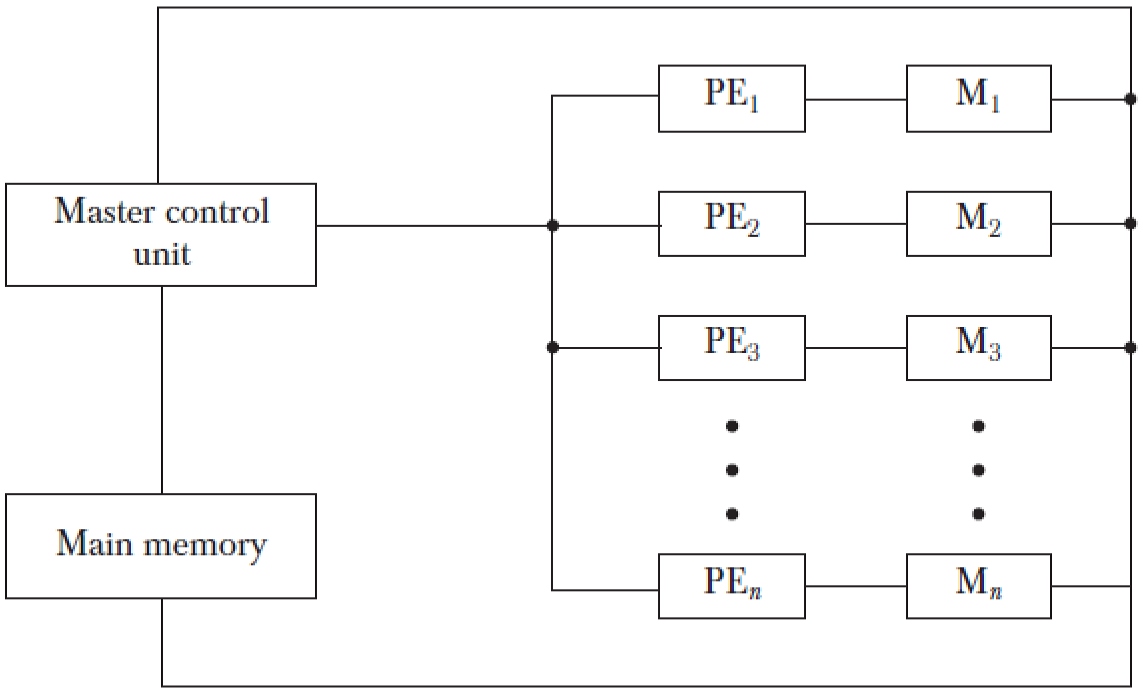
Each processor element includes an ALU, a floating-point arithmetic unit, and working registers.

The master control unit controls the operations in the processor elements.

The main memory is used for storage of the program.

The function of the master control unit is to decode the instructions and determine how the instruction is to be executed.

A general block diagram of an array processor



Scalar and program control instructions are directly executed within the master control unit.

Vector instructions are broadcast to all PEs simultaneously.

Each PE uses operands stored in its local memory.

Vector operands are distributed to the local memories prior to the parallel execution of the instruction.

Consider, for example, the vector addition $C = A + B$.

The master control unit first stores the ith components a_i and b_i of A and B in local memory M_i for $i = 1, 2, 3, \ldots, n$.

It then broadcasts the floating-point add instruction $c_i = a_i + b_i$ to all PEs, causing the addition to take place simultaneously.

The components of c_i are stored in fixed locations in each local memory.

This produces the desired vector sum in one add cycle.

Masking schemes are used to control the status of each PE during the execution of vector instructions.

Each PE has a flag that is set when the PE is active and reset when the PE is inactive.

This ensures that only those PEs that need to participate are active during the execution of the instruction.

For example, suppose that the array processor contains a set of 64 PEs. If a vector length of less than 64 data items is to be processed, the control unit selects the proper number of PEs to be active.

Vectors of greater length than 64 must be divided into 64-word portions by the control unit.

The best known SIMD array processor is the ILLIAC IV computer developed at the University of Illinois and manufactured by the Burroughs Corp.

This computer is no longer in operation.

SIMD processors are highly specialized computers.

They are suited primarily for numerical problems that can be expressed in vector or matrix form.

However, they are not very efficient in other types of computations or in dealing with conventional data processing programs.