UNIT – 1

Introduction to Parallel Processing

Basic concepts of parallel processing on high-performance computers are introduced in this unit. Parallel computer structures will be characterized as Pipelined computers, array processors and multiprocessor systems.

Evolution of Computer Systems

Over the past four decades the computer industry has experienced four generations of development. The first generation used Vacuum Tubes (1940 – 1950s) to discrete diodes to transistors (1950 – 1960s), to small and medium scale integrated circuits (1960 – 1970s) and to very large scale integrated devices (1970s and beyond). Increases in device speed and reliability and reduction in hardware cost and physical size have greatly enhanced computer performance. The relationships between data, information, knowledge and intelligence are demonstrated.

Parallel processing demands concurrent execution of many programs in a computer. The highest level of parallel processing is conducted among multiple jobs through multiprogramming, time sharing and multiprocessing.

Over the past four decades the computer industry has experienced four generations of development.

a) Generations of Computer Systems

First Generation (1938-1953) - Vacuum Tube

- 1938 - John W. Mauchly and J. Presper Eckert built the first digital electronic computer, ENIAC (Electronic Numerical Integrator & Computer)
- Electro mechanical relays were used as switching devices in the 1940s, and vacuum tubes were used in 1950s.
- CPU structure to be bit srial: arithmetic is done on a bit by bit fixed-point basis
- In 1950, the first stored program computer, EDVAC (Electronic Discrete Variable Automatic Computer), was developed.
- In 1952, IBM had announced its 701 electronic calculator

Second Generation Computers (1952 -1963) – Transistor

- Transistors were invented in 1948.
- TRADIC (Transistorized digital Computer), was built by Bell Laboratories in 1954.
    - Discrete transistors and diodes are used as building blocks
    - Printed circuits appeared
    - Assembly Languages, Fortan used in 1956 and Algol used in 1960
- The first IBM scientific, transistorized computer, IBM 1620, became available in 1960.

Third Generation Computers (1962 -1975) – IC

- SSI & MSI circuits are the basic building blocks
- Multilayered Printed circuits were used
- 1968 - DEC introduced the first "mini-computer", the PDP-8, named after the mini-skirt.
- 1969 - Development began on ARPAnet
- 1971 - Intel produced large scale integrated (LSI) circuits that were used in the digital delay line, the first digital audio device.

Fourth Generation (1971-1991) – microprocessor

- 1971 - Gilbert Hyatt at Micro Computer Co. patented the microprocessor
- 1972 - Intel made the 8-bit 8008 and 8080 microprocessors
- 1974 - Xerox developed the Alto workstation at PARC, with a monitor, a graphical user interface, a mouse, and an Ethernet card for networking
- 1984 - Apple Computer introduced the Macintosh personal computer January 24.

Fifth Generation (1991 and Beyond)

- 1991 - World-Wide Web (WWW) was developed by Tim Berners-Lee and released by CERN.
- 1993 - The first Web browser called Mosaic was created by student Marc Andreesen and programmer Eric Bina at NCSA in the first 3 months of 1993.
- 1994 - Netscape Navigator 1.0 was released Dec. 1994
- 1996 - Microsoft failed to recognize the importance of the Web, but finally released the much improved browser Explorer 3.0 in the summer.

b) Trends towards Parallel Processing

From an application point of view, the mainstream of usage of computer is experiencing a trend of four ascending levels of sophistication:

- Data processing
- Information processing
- Knowledge processing
- Intelligence processing

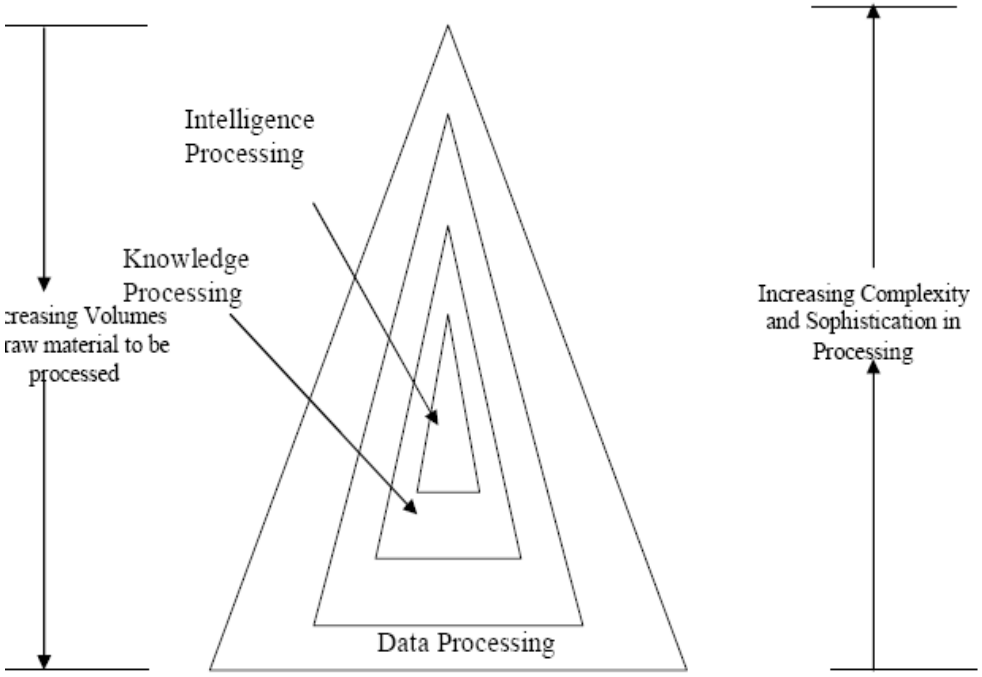

Figure 1.1 The spaces of data, information, knowledge and intelligence from the viewpoint of computer processing

Computer usage started with data processing, while is still a major task of today's computers. With more and more data structures developed, many users are shifting to computer roles from pure data processing to information processing. A high degree of parallelism has been found at these levels. As the accumulated knowledge bases expanded rapidly in recent years, there grew a strong demand to use computers for knowledge processing. Intelligence is very difficult to create; it's processing even more so.

From an operating point of view, computer systems have improved chronologically in four phases:

- batch processing
- multiprogramming
- time sharing
- multiprocessing

In these four operating modes, the degree of parallelism increase sharply from phase to phase.

We define parallel processing as “

- Parallel processing is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process”.
- Concurrency implies parallelism, simultaneity, and pipelining.
- Parallel processing demands concurrent execution of many programs in the computer. The highest level of parallel processing is conducted among multiple jobs or programs through multiprogramming, time sharing, and multiprocessing.

Parallel processing can be challenged in four programmatic levels:

- Job or program level
- Task or procedure level
- Inter instruction level
- Intra instruction level

The highest job level is often conducted algorithmically. The lowest intra-instruction level is often implemented directly by hardware means. Hardware roles increase from high to low levels. On the other hand, software implementations increase from low to high levels.

PARALLELISM IN UNIPROCESSOR SYSTEMS

a) Basic Uniprocessor Architecture

The typical Uniprocessor system consists of three major components:

1. Main memory,
2. Central processing unit (CPU)
3. Input-output (I/O) sub-system.

The architectures of two available Uniprocessor computers are described below:

1. Fig below shows the architectural components of the super minicomputer VAX-11/780.

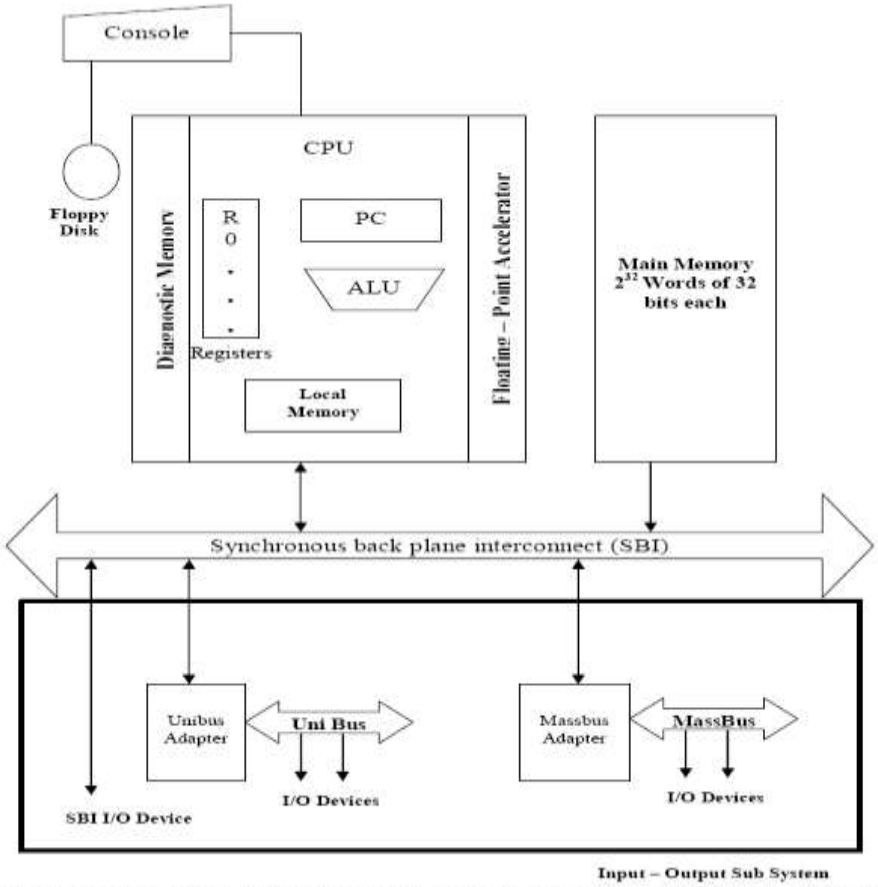

Figure 1.2 The system architecture of the super mini VAX – 11/780 microprocessor system

The **CPU** contains the **Master Controller** of the VAX system. There are sixteen 32-bit general purpose registers, one which serve as **Program Counter (PC)**.There is also a special CPU status register containing information about the current state of the processor and of the program being

executed. The CPU contains an **arithmetic and logic unit (ALU)** with an optional floating-point accelerator, and some local cache memory with an optional diagnostic memory.

The CPU, the main memory and the I/O subsystems are all connected to a common bus, the S**ynchronous Backplane Interconnect** (SBI) through this bus, all I/O device scan communicate with each other, with the CPU, or with the memory. Peripheral storage or I/O devices can be connected directly to the SBI through the **unibus** and its controller or through a **mass bus** and its Controller.

2. Fig below shows the architectural components of the main frame computer IBM System 370/model 168 Uniprocessor.

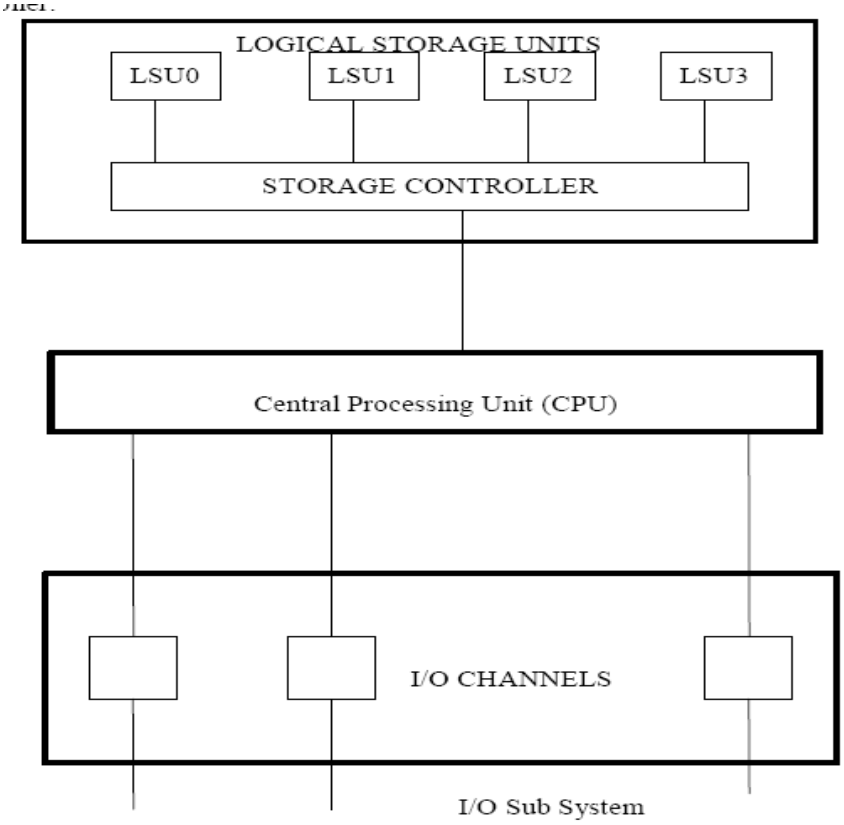

Figure 2.1 The System Architecture of the mainframe IBM System

The CPU contains the instruction decoding and execution units as well as a cache. Main memory is divided into four units, referred to as **logical storage units** that are four-way interleaved. The storage controller provides multiport connections between the CPU and the four

LSUs.Peripherals are connected to the system via high speed I/O channels which operate asynchronously with the CPU.

b) Parallel Processing Mechanism

A number of parallel processing mechanisms have been developed in uniprocessor computers. We identify them in the following six categories:

- Multiplicity of functional units
- Parallelism and pipelining within the CPU
- Overlapped CPU and I/O operations
- Use of a hierarchical memory system
- Multiprogramming and time sharing
- Multiplicity of functional units

Multiplicity of Functional Units

Use of multiple processing elements under one controller

•Many of the ALU functions can be distributed to multiple specialized units

•These multiple Functional Units are independent of each other

Example:

•IBM 360/91

–2 parallel execution units

•Fixed point arithmetic

•Floating point arithmetic(2 Functional units)

–Floating point add-sub

–Floating point multiply-div

The early computer has only one ALU in its CPU and hence performing a long sequence of ALU instructions takes more amount of time. The CDC-6600 has 10 functional units built into its CPU.These 10 units are independent of each other and may operate simultaneously. A score board is used to keep track of the availability of the functional units and registers being demanded. With 10 functional units and 24 registers available, the instruction issue rate can be significantly increased.

System Architecture of the CDC-6600 computer

Fig:

Another good example of a multifunction uniprocessor is the IBM 360/91 which has 2 parallel execution units. One for fixed point arithmetic and the other for floating point arithmetic. Within the floating point E-unit are two functional units: one for floating point add- subtract and other for floating point multiply – divide. IBM 360/91 is a highly pipelined, multifunction scientific uniprocessor.

Parallelism and Pipelining Within the CPU

Parallelism & pipelining within theCPU

•Parallelism is provided by building parallel adders in almost all ALUs

•Pipelining

–Each task is divided into subtasks which canbe executed in parallel

Overlapped CPU and I/O Operations

I/O operations can be performed simultaneously with the CPUcomputations by using

–separate I/O controllers –I/O channels –I/O processors

Use of Hierarchical Memory System

Speed of CPU = 1000 times speed of Main memory

•hierarchical memory structure is used to close up the speed gap

–Cache memory –Virtual memory –Parallel memories for array processors

The CPU is 1000 times faster than memory access. A hierarchical memory system can be used to close up the speed gap. Computer memory hierarchy is conceptually illustrated in fig below:

Fig:

The hierarchical order listed is

- registers
- Cache

- Main Memory
- Magnetic Disk
- Magnetic Tape
-

The inner most level is the register files directly addressable by ALU.Cache memory can be used to serve as a buffer between the CPU and the main memory. Virtual memory space can be established with the use of disks and tapes at the outer levels.

Balancing Of Subsystem Bandwidth

Balancing bandwidth between mainmemory and CPU
•Balancing bandwidth between mainmemory and I/O
CPU is the fastest unit in computer. The bandwidth of a system is defined as the number of operations performed per unit time. In case of main memory the memory bandwidth is measured by the number of words that can be accessed per unit time.

Bandwidth Balancing Between CPU and Memory

The speed gap between the CPU and the main memory can be closed up by using fast cache memory between them. A block of memory words is moved from the main memory into the cache so that immediate instructions can be available most of the time from the cache.

Bandwidth Balancing Between Memory and I/O Devices

Input-output channels with different speeds can be used between the slow I/O devices and the main memory. The I/O channels perform buffering and multiplexing functions to transfer the data from multiple disks into the main memory by stealing cycles from the CPU.

Multiprogramming &**Time-sharing**

Multiprogramming

–Mix the execution of various types of programs (I/o bound, CPU bound)
–The interleaving of CPU and I/O operationsacross several programs
Time-sharing OS is used to avoid high-priority programs occupying the CPU for long
–Fixed or variable time-slices are used
–Creates a concept of virtual processors

Parallel Computer Structures

Parallel computers are those systems that emphasize parallel processing. We divide parallel computers into three architectural configurations:

- Pipeline computers
- Array processors
- Multiprocessors

- ⇨ In a pipelined computer successive instructions are executed in an overlapped fashion. In a non pipelined computer these four steps must be completed before the next instructions can be issued.
- ⇨ An array processor is a synchronous parallel computer with multiple arithmetic logic units called processing elements (PE) that can operate in parallel in lock step fashion.By replication one can achieve spatial parallelism. The PEs is synchronized to perform the same function at the same time.
- ⇨ A basic multiprocessor contains two or more processors of comparable capabilities. All processors share access to common sets of memory modules, I/O channels and peripheral devices.

Pipeline Computers

The process of executing an instruction in a digital computer involves 4 major steps

- Instruction fetch
- Instruction decoding
- Operand fetch
- Execution

In a pipelined computer successive instructions are **executed in an overlapped fashion.** In a non pipelined computer these four steps **must be completed before the next instructions can be issued.**

- Instruction fetch : Instruction is fetched from the main memory
- Instruction decoding: Identifying the operation to be performed.
- Operand Fetch: If any operands is needed is fetched.

- Execution : Execution of the Arithmetic and logical operation

An instruction cycle consists of multiple pipeline cycles. The flow of data (input operands, intermediate results and output results) from stage to stage is triggered by a common clock of the pipeline. The operations of all stages are triggered by a common clock of the pipeline.

For non pipelined computer, it takes four pipeline cycles to complete one instruction. Once a pipeline is filled up, an output result is produced from the pipeline on each cycle. The instruction cycle has been effectively reduced to 1/4th of the original cycle time by such overlapped execution.

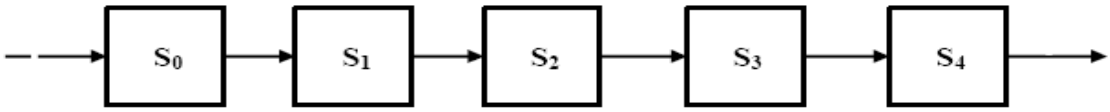

Figure 3.1 A pipelined Processor

Time	0	1	2	3	4	5	6	7	8	9	10	11 ...
I0	I01	I02	I03	I04	I05							
I1		I11	I12	I13	I14	I15						
I2			I21	I22	I23	I24	I25					
I3				I31	I32	I33	I34	I35				
I4					I41	I42	I43	I44	I45			
I5						I51	I52	I53	I54	I55		
I6							I61	I62	I63	I64	I65	
I7								I71	I72	I73	I74	I75
I8									I81	I82	I83	I84 ...
...										...		
Completion					I0	I1	I2	I3	I4	I5	I6	I7...

Figure 3.2 Space Diagram for a Pipelined Processor

Array Processors

- An array processor is a synchronous parallel computer with multiple arithmetic logic units called processing elements (PE) that can operate in parallel in lock step.
- The PEs is synchronized to perform the same function at the same time.
- Scalar and control type of instructions are directly executed in the control unit (CU).
- Each PE consists of an ALU registers and a local memory.

The PEs is interconnected by a data routing network. Vector instructions are broadcasted to the PEs for distributed execution over different component operands fetched directly from local memories. Array processors designed with associative memories are called as associative processors.

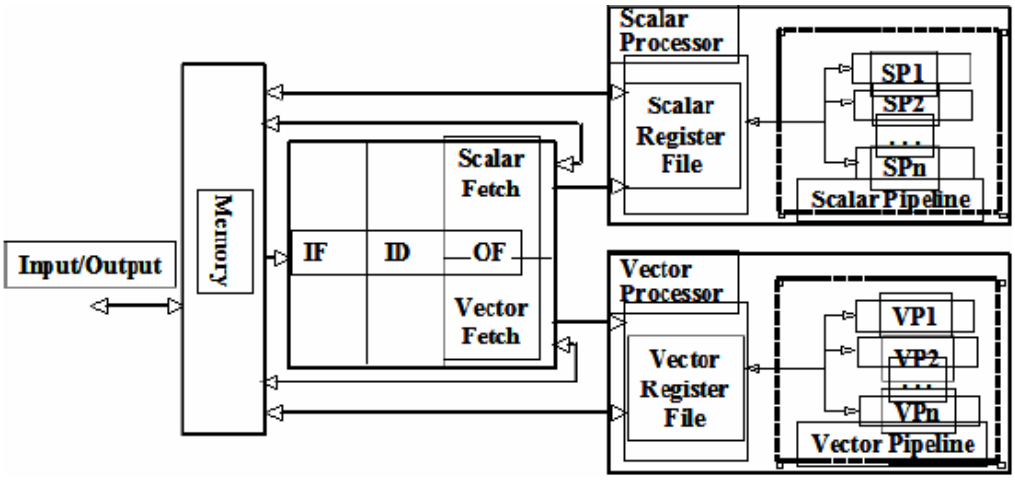

Figure 3.3 Functional structure of a modern pipeline computer with scalar and vector capabilities

Functional Structure of SIMD array processor with concurrent processing in the control unit

Fig: Refer Text

Multiprocessor Systems

A basic multiprocessor contains two or more processors of comparable capabilities. All processors share access to common sets of memory modules, I/O channels and peripheral devices. The entire system must be controlled by a single integrated operating system providing interactions between processors and their programs at various levels.

Multiprocessor hardware system organization is determined by the interconnection structure to be used between the memories and processors. Three different interconnection are

- Time shared Common bus
- Cross Bar switch network
- Multiport memories

Functional Structure of MIMD multiprocessor system
Fig:Refer Text

Architectural Classification Schemes

Introduction

The Flynn's classification scheme is based on the multiplicity of instruction streams and data streams in a computer system. Flynn's scheme is based on serial versus parallel processing. Handler's classification is determined by the degree of parallelism and pipelining in various subsystem levels.

Architectural Classification Schemes

1. Flynn's Classification

The most popular taxonomy of computer architecture was defined by Flynn in 1966.Flynn's classification scheme is based on the notion of a stream of information. Two types of information flow into a processor: instructions and data. The instruction stream is defined as the sequence of instructions performed by the processing unit. The data stream is defined as the data traffic exchanged between the memory and the processing unit.

According to Flynn's classification, either of the instruction or data streams can be single or multiple.

Computer architecture can be classified into the following four distinct categories:

- single-instruction single-data streams (SISD);
- single-instruction multiple-data streams (SIMD);
- multiple-instruction single-data streams (MISD); and
- Multiple-instruction multiple-data streams (MIMD).

General Notes:

Conventional single-processor von Neumann computers are classified as SISD systems. Parallel computers are either SIMD or MIMD. When there is only one control unit and all processors

execute the same instruction in a synchronized fashion, the parallel machine is classified as SIMD.

In a MIMD machine, each processor has its own control unit and can execute different instructions on different data. In the MISD category, the same stream of data flows through a linear array of processors executing different instruction streams. In practice, there is no viable MISD machine; however, some authors have considered pipelined machines (and perhaps systolic-array computers) as examples for MISD. An extension of Flynn's taxonomy was introduced by D. J. Kuck in 1978. In his classification, Kuck extended the instruction stream further to single (scalar and array) and multiple (scalar and array) streams. The data stream in Kuck's classification is called the execution stream and is also extended to include single (scalar and array) and multiple (scalar and array) streams. The combination of these streams results in a total of 16 categories of architectures.

SISD Architecture

- A serial (non-parallel) computer
- Single instruction: only one instruction stream is being acted on by the CPU during any
- one clock cycle
- Single data: only one data stream is being used as input during any one clock cycle
- Deterministic execution
- This is the oldest and until recently, the most prevalent form of computer
- Examples: most PCs, single CPU workstations and mainframes

Fig:

SIMD Architecture

- A type of parallel computer

- Single instruction: All processing units execute the same instruction at any given clock cycle
- Multiple data: Each processing unit can operate on a different data element
- This type of machine typically has an instruction dispatcher, a very high-bandwidth internal network, and a very large array of very small-capacity instruction units.
- Best suited for specialized problems characterized by a high degree of regularity, such as image processing.
- Synchronous (lockstep) and deterministic execution
- Two varieties: Processor Arrays and Vector Pipelines
- Examples:
    - Processor Arrays: Connection Machine CM-2, Maspar MP-1, MP-2
    - Vector Pipelines: IBM 9000, Cray C90, Fujitsu VP, NEC SX-2, Hitachi S820

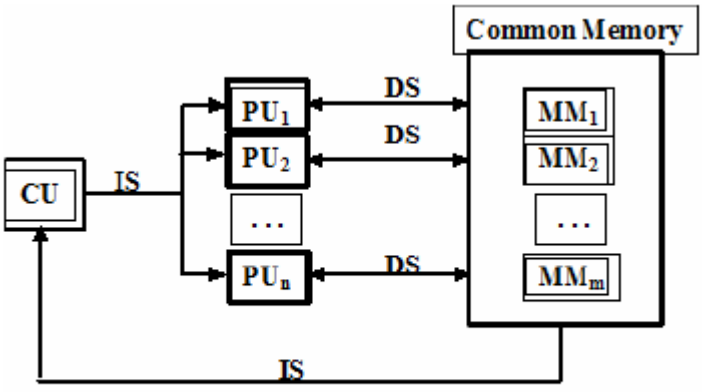

Figure 4.2 SIMD COMPUTER

CU-control unit

PU-processor unit

MM-memory module

SM-Shared memory

IS-instruction stream

DS-data stream

MISD Architecture

There are n processor units, each receiving distinct instructions operating over the same data streams and its derivatives. The output of one processor becomes input of the other in the macro pipe. No real embodiment of this class exists.

- A single data stream is fed into multiple processing units.
- Each processing unit operates on the data independently via independent instruction streams.
- Few actual examples of this class of parallel computer have ever existed. One is the experimental Carnegie-Mellon C.mmp computer (1971).
- Some conceivable uses might be:
    - multiple frequency filters operating on a single signal stream
    - Multiple cryptography algorithms attempting to crack a single coded message.

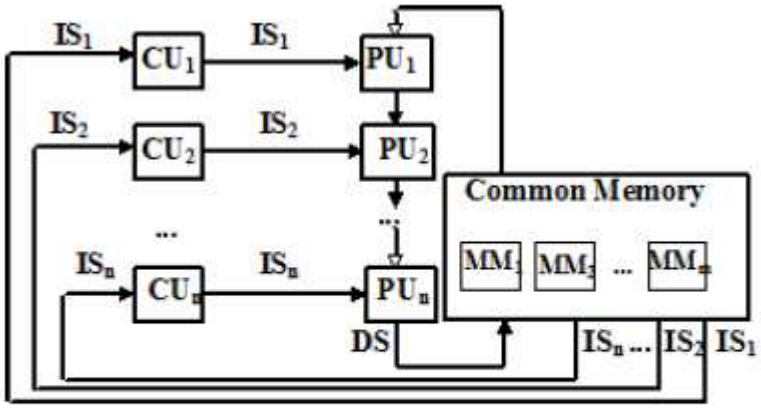

Figure 4.3 MISD COMPUTER

MIMD Architecture

Multiple-instruction multiple-data streams (MIMD) parallel architectures are made of multiple processors and multiple memory modules connected together via some interconnection network. They fall into two broad categories: shared memory or message passing. Processors exchange information through their central shared memory in shared memory systems, and exchange information through their interconnection network in message passing systems.

- Currently, the most common type of parallel computer. Most modern computers fall into this category.
- Multiple Instruction: every processor may be executing a different instruction stream
- Multiple Data: every processor may be working with a different data stream
- Execution can be synchronous or asynchronous, deterministic or non-deterministic
- Examples: most current supercomputers, networked parallel computer "grids" and multiprocessor SMP computers - including some types of PCs.

A shared memory system typically accomplishes interprocessor coordination through a global memory shared by all processors. These are typically server systems that communicate through a bus and cache memory controller.

A message passing system (also referred to as distributed memory) typically combines the local memory and processor at each node of the interconnection network. There is no global memory, so it is necessary to move data from one local memory to another by means of message passing.

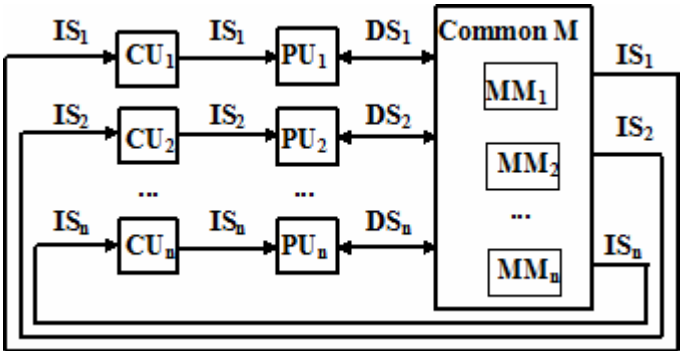

Fig: MIMD Computer

Feng's Classification

Tse-yun Feng suggested the use of degree of parallelism to classify various computer architectures.

Serial versus Parallel Processing

The maximum number of binary digits that can be processed within a unit time by a computer system is called the maximum parallelism degree P.A bit slice is a string of bits one from each of the words at the same vertical position.

There are 4 types of methods under above classification

- Word Serial and Bit Serial (WSBS)
- Word Parallel and Bit Serial (WPBS)
- Word Serial and Bit Parallel(WSBP)
- Word Parallel and Bit Parallel (WPBP)

    - ⇨ WSBS has been called bit parallel processing because one bit is processed at a time.
    - ⇨ WPBS has been called bit slice processing because m-bit slice is processes at a time.
    - ⇨ WSBP is found in most existing computers and has been called as Word Slice processing because one word of n bit processed at a time.
    - ⇨ WPBP is known as fully parallel processing in which an array on n x m bits is processes at one time.

Handler's Classification

Parallelism versus Pipelining

Wolfgang Handler has proposed a classification scheme for identifying the parallelism degree and pipelining degree built into the hardware structure of a computer system. He considers at three subsystem levels:

- Processor Control Unit (PCU)
- Arithmetic Logic Unit (ALU)
- Bit Level Circuit (BLC)

Each PCU corresponds to one processor or one CPU. The ALU is equivalent to Processor Element (PE). The BLC corresponds to combinational logic circuitry needed to perform 1 bit operations in the ALU.

A computer system C can be characterized by a triple containing six independent entities

T(C) = <K x K', D x D', W x W' >

Where K = the number of processors (PCUs) within the computer

D = the number of ALUs under the control of one CPU

W = the word length of an ALU or of an PE

W' = the number of pipeline stages in all ALUs or in a PE

D' = the number of ALUs that can be pipelined

K' = the number of PCUs that can be pipelined

Amdhals law

Amdahl's law, also known as **Amdahl's argument**, is used to find the maximum expected improvement to an overall system when only part of the system is improved. It is often used in parallel computing to predict the theoretical maximum speedup using multiple processors. The law is named after computer architect Gene Amdahl, and was presented at the AFIPS Spring Joint Computer Conference in 1967.

The speedup of a program using multiple processors in parallel computing is limited by the time needed for the sequential fraction of the program. For example, if a program needs 20 hours using a single processor core, and a particular portion of the program which takes one hour to execute cannot be parallelized, while the remaining 19 hours (95%) of execution time can be parallelized, then regardless of how many processors are devoted to a parallelized execution of this program, the minimum execution time cannot be less than that critical one hour. Hence the speedup is limited to at most 20×.

Amdahl's law is a model for the relationship between the expected speedup of parallelized implementations of an algorithm relative to the serial algorithm, under the assumption that the problem size remains the same when parallelized. For example, if for a given problem size a parallelized implementation of an algorithm can run 12% of the algorithm's operations arbitrarily quickly (while the remaining 88% of the operations are not parallelizable), Amdahl's law states

that the maximum speedup of the parallelized version is 1/(1 – 0.12) = 1.136 times as fast as the non-parallelized implementation.

More technically, the law is concerned with the speedup achievable from an improvement to a computation that affects a proportion *P* of that computation where the improvement has a speedup of *S*. (For example, if 30% of the computation may be the subject of a speed up, *P* will be 0.3; if the improvement makes the portion affected twice as fast, *S* will be 2.) Amdahl's law states that the overall speedup of applying the improvement will be:

To see how this formula was derived, assume that the running time of the old computation was 1, for some unit of time. The running time of the new computation will be the length of time the unimproved fraction takes (which is $1 - P$), plus the length of time the improved fraction takes. The length of time for the improved part of the computation is the length of the improved part's former running time divided by the speedup, making the length of time of the improved part P/S. The final speedup is computed by dividing the old running time by the new running time, which is what the above formula does.

Here's another example. We are given a sequential task which is split into four consecutive parts: P1, P2, P3 and P4 with the percentages of runtime being 11%, 18%, 23% and 48% respectively. Then we are told that P1 is not sped up, so S1 = 1, while P2 is sped up 5×, P3 is sped up 20×, and P4 is sped up 1.6×. By using the formula P1/S1 + P2/S2 + P3/S3 + P4/S4, we find the new sequential running time is:

or a little less than $\frac{1}{2}$ the original running time. Using the formula $(P1/S1 + P2/S2 + P3/S3 + P4/S4)^{-1}$, the overall speed boost is 1 / 0.4575 = 2.186, or a little more than double the original speed. Notice how the 20× and 5× speedup don't have much effect on the overall speed when P1 (11%) is not sped up, and P4 (48%) is sped up only 1.6 times.