List and Explain parallel processing mechanism in uniprocessor computers

- Parallel processing is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process.
- Parallel Processing demands concurrent execution of many programs in the computer.

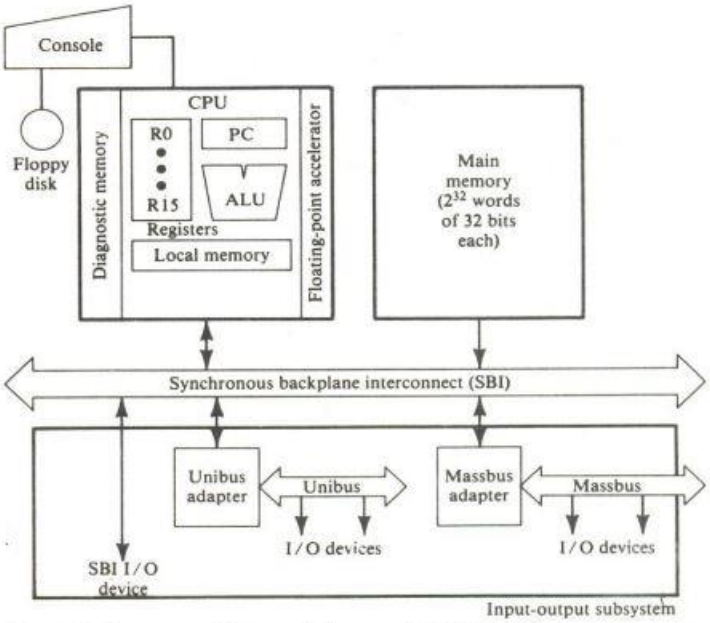
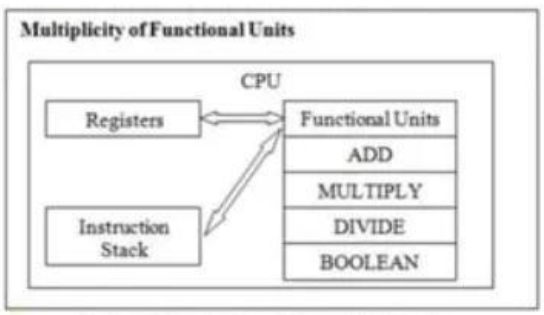


Figure 1.3 The system architecture of the supermini VAX-11/780 uniprocessor system (Courtesy of Digital Equipment Corporation).

- A number of parallel processing mechanism have been developed in uniprocessor computers. We identify them in the following six categories , they are :
  1. Multiplicity of functional units
  2. Parallelism and pipelining within the CPU
  3. Overlapped CPU and I/O operations
  4. Use of a hierarchical memory system.
  5. Balancing of subsystem bandwidths
  6. Multiprogramming and time sharing

Multiplicity of functional units :

- In earlier computers, the CPU consists of only one arithmetic logic unit which used to perform only one function at a time.
- This slows down the execution of the long sequence of arithmetic instructions.
- To overcome this the functional units of the CPU can be increased to perform parallel and simultaneous arithmetic operations.



Parallelism and Pipelining within CPU :

- Parallel adders can be implemented using techniques such as carry-look ahead and carry-save.
- A parallel adder is a digital circuit that adds two binary numbers, where the length of one bit is larger as compared to the length of another bit and the adder operates on equivalent pairs of bits parallelly.
- The multiplier can be recoded to eliminate more complex calculations. • Various instruction execution phases are pipelined and to overcome the situation of overlapped instruction execution the techniques like instruction pre-fetch and data buffers are used.

Overlapped CPU and I/O Operation :

- To execute I/O operation parallel to the CPU operation we can use I/O controllers or I/O processors.
- For direct information transfer between the I/O device and the main memory, direct memory access (DMA) can be used.
- The DMA is conducted on a cycle stealing basis, which is apparent to the CPU.

Use Hierarchical Memory System

- The processing speed of the CPU is 1000 times faster than the memory accessing speed which results in slowing the processing speed.
- To overcome this speed gap hierarchical memory system can be used.
- The faster accessible memory structure is registered in CPU, then cache memory which buffers the data between CPU and main memory.

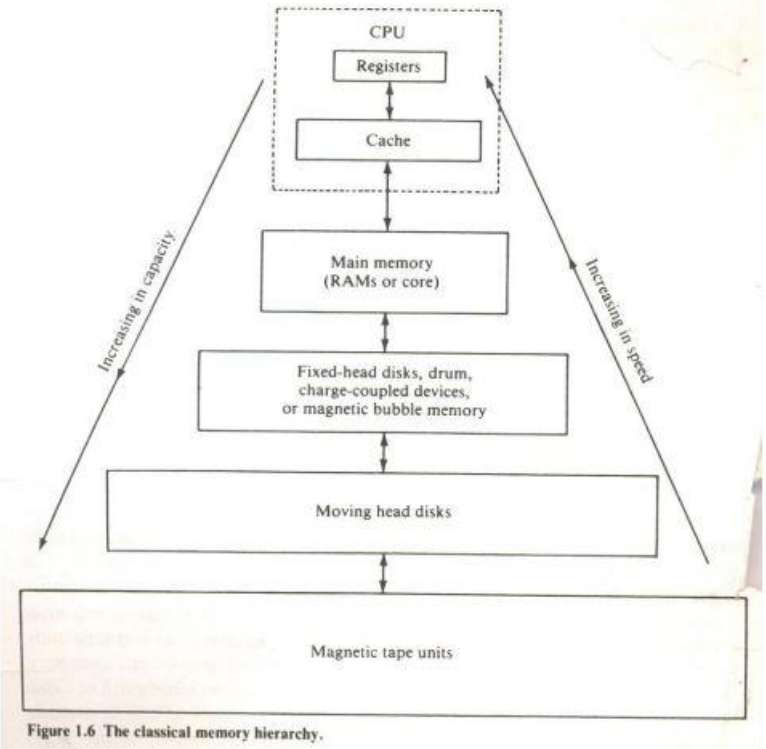


Figure 1.6 The classical memory hierarchy.

Balancing of Subsystem Bandwidth :

- The processing and accessing time of CPU, main memory, and I/O devices are different.
- If arrange the processing time of these units in descending order the order would be:

  Td >Tm>Tp

  where Td is the processing time of the device,
  Tm is the processing time of the main memory, and
  Tp is the processing time of the central processing unit.
- The processing time of the I/O devices is greater as compared to the main memory and processing unit. CPU is the fastest unit.
- To put a balance between the speed of CPU and memory a fast cache memory can be used which buffers the information between memory and CPU.
- To balance the bandwidth between memory and I/O devices, input-output channels with different speeds can be used between main memory and I/O devices.

Multi programming:

- There may be multiple processes active in computers and some of them may be competing for memory, some for I/O devices and some for CPU.
- So, to establish a balance between these processes, program interleaving must be practiced.
- This will boost resource utilization by overlapping the I/O and CPU operations.
- Program interleaving can be understood as when a process P, is engaged in I/O operation the process scheduler can switch CPU to operate process P2.
- This led process P, and P_2 to execute simultaneously.
- This interleaving between CPU and I/O devices is called multiprogramming.

Time-sharing:

- Multiprogramming is based on the concept of time-sharing.
- The CPU time is shared among multiple programs.
- Sometimes a high-priority program can engage the CPU for a long period starving the other processes in the computer.
- The concept of timesharing assigns a fixed or variable time slice of CPUs to multiple processes in the computer.
- This provides an equal opportunity to all the processes in the computer.

Explain functional structure SIMD array Processor

- SIMD ('Single Instruction and Multiple Data Stream') processors is a computer with several processing units which operate in parallel.
- These processing units perform the same operation in synchronizing under the supervision of the common control unit (CCU).
- The SIMD processor includes a set of identical PEs (processing elements) where each PES has a local memory.

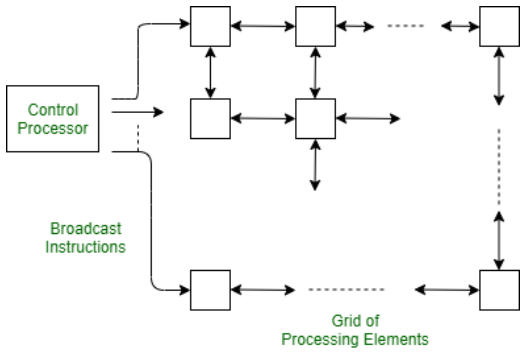


Figure - SIMD Array Processor

1. Control Processor

- The control processor is responsible for broadcasting instructions to all the processing elements.
- It synchronizes the operations across the grid, ensuring that each processing element performs the same operation simultaneously but on different data points.
- This centralized control allows for a high degree of parallelism in processing.

2. Grid of Processing Elements

- The array processor is composed of a grid of processing elements.
- Each element in the grid can represent a specific point in space for solving problems like temperature distribution across a plane.
- This setup is particularly suitable for two-dimensional problems where each processing element represents a discrete spatial point.

3. Broadcasting Instructions

- The control processor sends instructions to all processing elements via a broadcast mechanism.
- These instructions include data movement commands (up, down, left, right) and arithmetic operations that each element needs to execute.
- This ensures uniformity in the execution of tasks across the entire array.

4. Interconnection Network

- Each processing element can exchange values with its neighboring elements.
- The interconnection network facilitates data transfer between adjacent PEs. The network allows these values to move seamlessly between elements.

5. Local Memory and Registers

- Each processing element has local memory and registers for storing data and intermediate results.
- This local storage capability allows each PE to independently maintain its data without needing constant access to shared memory resources, thus enhancing processing speed and efficiency.
- **Network Register**: Each PE contains a special network register used for moving values to and from its neighbors, which is essential for the iterative calculation processes.

6. Arithmetic Logic Unit (ALU)

- Each processing element is equipped with an ALU that performs arithmetic operations like addition, subtraction, multiplication, etc.
- The ALU processes instructions broadcasted by the control processor, enabling each PE to compute results based on its local data and values received from its neighbors.

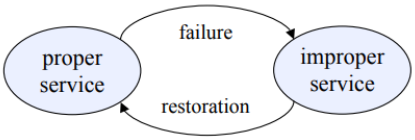
Advantages

The advantages of an SIMD array processor include the following :

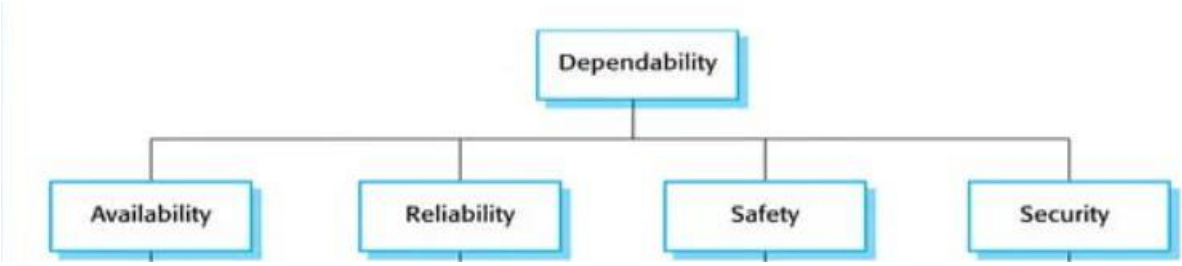
- Array processors improve the whole instruction processing speed.
- These processors run asynchronously from the host CPU the overall capacity of the system is improved.
- These processors include their own local memory that provides extra memory to systems. So this is an important consideration for the systems through a limited address space or physical memory.
- These processors simply perform computations on a huge array of data.
- These are extremely powerful tools that help in handling troubles with a high amount of parallelism.

What is dependability ? State different measures of dependability. How MTBF is measured ?

- Dependability is the ability of a system to deliver a specified service. System service is classified as proper if it is delivered as specified; otherwise it is improper.



- Dependability is a measure of a system's availability, reliability, maintainability, and in some cases, other characteristics such as durability, safety and security.
- Dependability aims to ensure that computer systems continue to operate correctly even in the presence of faults, whether these faults are due to hardware failures, software bugs, or external factors such as power loss.
- Dependability is typically measured using several key attributes:



- Two measures of dependability : 1)Module reliability. 2)Module availability.

Module reliability :

- It is a measure of the continuous service accomplishment (or, equivalently, of the time to failure) from a reference initial instant.
- Hence, the mean time to failure (MTTF) is a reliability measure.
- The reciprocal of MTTF is a rate of failures, generally reported as failures per billion hours of operation, or FIT (for failures in time).
- Thus, an MTTF of 1,000,000 hours equals 10^9⁄ 10^6 or 1000 FIT.

Module availability :

- Module availability is a measure of the service accomplishment with respect to the alternation between the two states of accomplishment and interruption.
- For non redundant systems with repair, module availability is ,

$$\text{Module availability} = \frac{\text{MTTF}}{(\text{MTTF} + \text{MTTR})}$$

Mean time between failures (MTBF) :

- Mean Time Between Failures (MTBF) is a key metric used to quantify the reliability of a system or component.
- It is the average time available for a system or component to perform its normal operations between failures.
- MTBF= (SUM of operational time/total number of failures)
- It measures the average time a system operates before experiencing a failure.

- This metric is particularly useful for understanding the expected reliability and performance of machinery, electronics, and other repairable components.

Calculating MTBF

MTBF is calculated by taking the total operational time of an asset and dividing it by the number of failures that occurred during that period:

MTBF = Total Uptime/Number of Failures

Example Calculation

- Find the Total Uptime: Suppose you have 40 machines that each run for 400 hours, totaling 16,000 hours of operational time.
- Count the Failures: If there are 20 failures during these 16,000 hours, you can use this data to calculate MTBF.

Calculate MTBF:

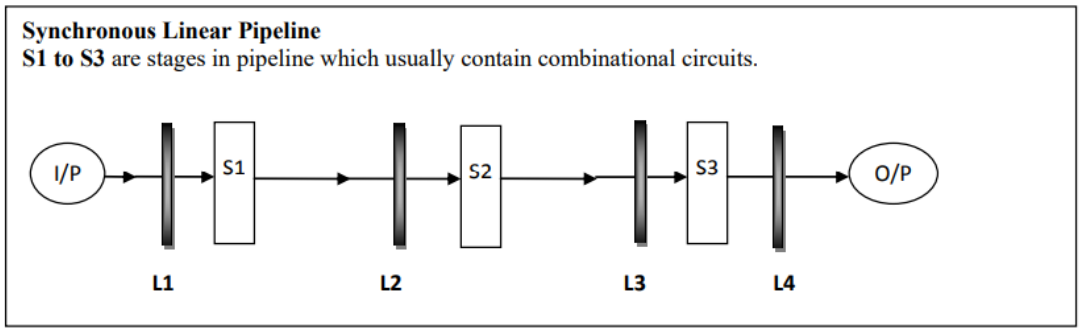
MTBF = 16,000 hours/ 20 failures

= 800 hours

This example indicates that the average time between failures for the group of machines is 800 hours.

Explain with neat diagram basic structure of linear Pipelining.

- Usually there are two categories of pipelines, first is linear pipeline and second is non-linear pipeline.
- In linear pipelining there are no feedback connections from output to input, so output and input is totally independent of output.
- To achieve pipeline, one must subdivide the input task(process) into a sequence of subtasks, each of which can be executed by a specialized hardware stage that operate concurrently with other stages in the pipeline.
- In a basic linear-pipeline processor the pipeline consists of a cascade of processing stages
- The stages are pure combinational circuits performing arithmetic or logic operations over the data stream flowing through the pipe.



- The stages are separated by high speed interface latches (L1, L2, L3…..).
- The latches are fast registers for holding the intermediate results between the stages
- The information flows between adjacent stages(S1, S2, S3) are under the control of a common clock applied to all the latches simultaneously.

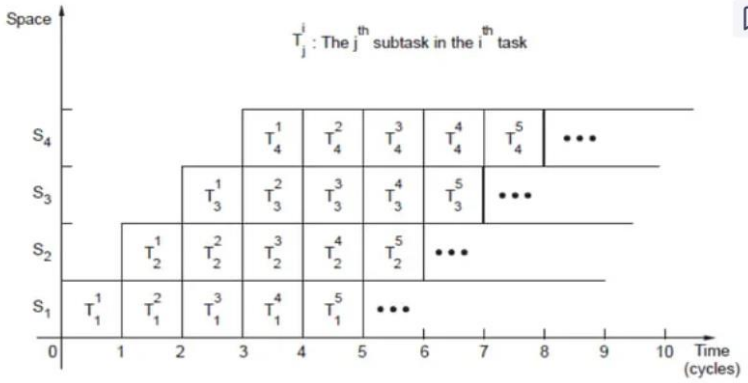


Fig. 3.6.6 (b) The space-time diagram showing the overlapped operations

- Ideally, a linear pipeline with m stages can process n tasks in T_m = m + (n - 1) clock cycles, where m cycles are needed to complete the execution of the first task and the remaining n - 1 tasks require n - 1 cycles. Thus the total time required is
  - T_m = [m + (n - 1)]
- The space-time diagram shown in Fig. 3.6.6 (b) has four stages and five tasks. Therefore, the ideal total time required is

  T_m = [4 + (5 - 1)]
  = 8 clock cycles

- The amount of time required to execute same number of tasks in a non-pipeline processor can be given as: $T_1 = n\,m\tau$.

Clock Period:

- The logic circuitry in each stage Si has a time delay denoted by ti. Let tl be the time delay of each interface latch. The clock period of linear pipeline is defined by

$$t = \max\{t_i\}_{i=1}^{k} + t_1 = t_m + t_1$$

Frequency=i/t

- Ideally, a k-stage pipeline can process n tasks in Tk=k+(n-1) clock periods.

Speedup ratio :

- The speed up ratio is ratio between maximum time taken by non pipeline process over process using pipelining.
- Thus in general if there are n processes and each process is divided into k segments (subprocesses)
- Speed of a k-stage linear pipeline over an equivalent nonpipeline processor is

$$S_k = \frac{T_1}{T_k} = \frac{n.k}{k + (n-1)}$$

- The maximum speedup that a linear pipeline can provide is k, where k is the number of stages in the pipe
- This maximum speedup is never fully achievable because of data dependencies, between instructions, interrupts, program branches, and other factors

Compare adavantages & disadvantages of interleaved memory organization.

- Interleaved memory organization is a technique used in computer systems to improve memory access speed and overall performance by dividing the main memory into multiple modules.
- Each module can be accessed independently and simultaneously, which allows for parallel processing of data.
- This approach is particularly effective in systems that require high-speed access to memory, such as vector processors and pipeline processors, where large amounts of data need to be processed quickly.

Advantages and Disadvantages of Interleaved Memory Organization :

Advantages:

1. **Increased Memory Bandwidth:** Interleaved memory allows simultaneous access to different memory modules. This parallelism increases the memory bandwidth, enabling more words to be accessed per unit time. It significantly improves the performance of systems that process large volumes of data, like vector processors or systems handling multiple threads.
2. **Reduced Memory Access Time:** By spreading memory addresses across multiple modules, interleaved memory minimizes the delay associated with consecutive memory accesses. This is especially beneficial in applications requiring high-speed data retrieval, such as scientific computations and real-time data processing.
3. **Better Utilization of Memory:** Interleaving distributes memory accesses evenly across different modules, preventing bottlenecks in specific memory regions and ensuring more efficient use of available memory resources.
4. **Enhanced Parallel Processing:** In systems with multiple processors, interleaving helps to distribute the load evenly among memory modules, facilitating efficient parallel processing without significant contention or conflicts.

Disadvantages:

1. **Complexity in Memory Management:** Interleaving requires a more complex memory management scheme, with additional circuitry to handle the addressing and synchronization of multiple memory modules. This complexity can increase the overall design and manufacturing costs.
2. **Potential for Memory Conflicts:** Despite the increased bandwidth, interleaved memory can still face contention issues when multiple processors or requests target the same memory module simultaneously. This can lead to conflicts and delays, somewhat negating the benefits of interleaving.
3. **Higher Costs:** The need for additional hardware and sophisticated memory controllers to manage interleaved memory systems can lead to increased costs. This may not be justified for smaller or less demanding applications.
4. **Difficulty in Fault Isolation:** In the case of a memory module failure, identifying and isolating the faulty module can be more challenging in an interleaved setup. This is because the data is spread across multiple modules, making it harder to pinpoint the exact location of the failure.