Page No. 
Date

NAME:- AFNAN ATTAR PRN: F19112003 CLASS:- B.E. COMP II
SUBJECT : HPC CLASS ASSIGNMENT NO.:- 02

Q.1) Mention some reason for motivation of parallelism.

Ans Some reasons for motivation of parallelism are as follows:-

A) Computational Power Argument:
- Moore's law states that circuit complexity doubles every eighteen months.
- This empirical relationship has remain resilient over the years for both microprocessors and DRAMS.
- It is possible to fabricate devices with a very large transistor count, however, translating it into useful OPS (operations per second) is the critical part.
- A logical recourse to this problem is to use implicit and explicit parallelism.

B) Memory/Disk Speed Argument:
- Speed of computation not only depends on processor but also access time of memory.
- Gaps in performance difference between processors and memory present a bottleneck.
- To reduce this gap we use memory caches and we can also use parallelism.
- Parallelism reduces this bottleneck by:-
  i) Aggregating larger caches (ii) Higher aggregate bandwidth

C) Data Communication Argument:-
- Consider mining of large commercial datasets over a low network bandwidth.
- Even if computing power is available it is infeasible to collect data without parallelism.

Page No.
Date

- Hence motivation for parallelism comes in due to infeasibility of centralized approach.

Q2) Describe dichotomy of parallel computing.

Ans 1. The dichotomy of parallel computing from a programmer's perspective are ways of expressing parallel tasks and mechanism for specifying interaction between these tasks.

2. The former is known as Control Structure and the latter as communication model.

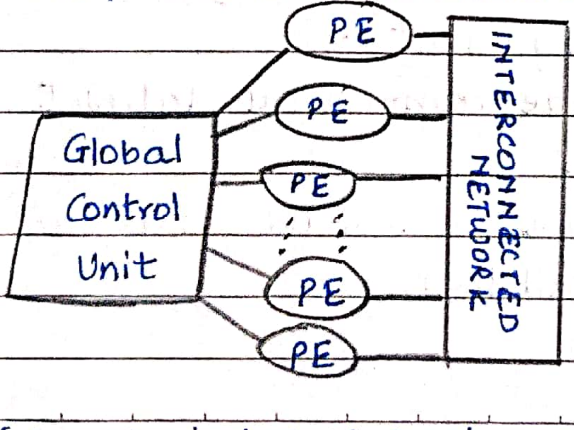
- Control Structure :-

1. Parallel tasks can be specified at various levels of granularity i) Each program in a set of programs can be viewed as one parallel task. (ii) Individual instructions within a program can be viewed as parallel tasks.

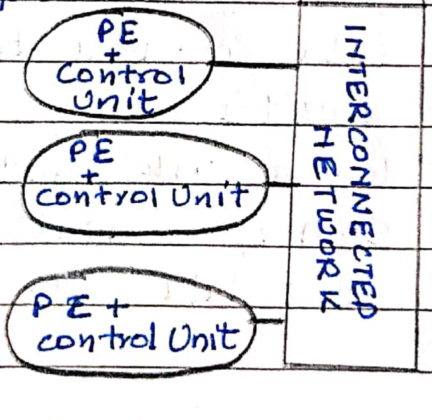
2. Between (i) and (ii) lie a range of models for specifying the control structure of programs and the corresponding architectural support for them.

3. In architectures refered to as single instruction stream, multiple data stream (SIMD), a single control unit dispatches instructions to each processing unit.

4. Computers in which each processing element is capable of executing a different program independent of the other processing element are called multiple instruction stream, multiple data stream (MIMD) computers.



(a) A typical SIMD Architecture



(b) A typical MIMD architecture

Page No.
Date

- Communication Model:-
- There are two primary tasks/forms of data exchange between parallel tasks:- accessing a shared data space and exchanging messages.

i) Shared-Address-Space Platforms:-

1. The "shared-adress-space" view of a parallel platform supports a common data space that is accessible to all processors.
2. If the time taken by processor to access any memory word in the system is the same, the platform is classified as a uniform memory access (UMA) multicomputer.
3. If the time taken to access certain memory words is longer than others, the platform is called non-uniform memory access (NUMA) multicomputer.

ii) Messaging Platforms:-

1. The logical machine view of a messaging platform consists of p processing nodes, each with its own exclusive address space.
2. Each of these processors can be single processors or a shared-address-space multipurpose.
3. Since interactions are accomplished by sending and receiving messages, the basic operations in this programming paradigm are send and receive.

---

Q3) Difference between SIMD & MIMD.

Ans

| | SIMD | MIMD |
|---|---|---|
| 1. | Stands for Single Instruction Multiple data. | Stands for multiple instruction multiple data. |
| 2. | Requires small or less memory. | Requires large or more memory |

|  |  |  |
|---|---|---|
| 3. | Less expensive. | More expensive. |
| 4. | Single decoder. | Multiple decoder. |
| 5. | Has latent/Tacit synchronization | Has acurate/explicit synchronization. |
| 6. | It is synchronous programming. | It is asynchronous programming. |
| 7. | Has less complexity. | Has more complexity. |
| 8. | Less efficient in performance. | More efficient in performance. |

Q4) What is meant by Implicit parallelism?

Ans 1. Implicit parallelism allows programmers to write their programs without any concern about exploitation of parallelism.

2. Exploitation of parallelism is automatically performed by the compiler or the runtime system.

3. In this way parallelism is transparent to programmer maintaining complexity of software development at the same level of standard sequential programming.

4. However, extracting parallelism implicitly is not an easy task, especially in the case of imperative programming languages.

5. The situation is brighter for declarative languages due to the following reasons:-

i) They are referentially transparent, i.e variables are immutable

ii) Operational semantics are based on some form of non-determinism

iii) Eager evaluation schemes allow dataflow like computation.

6. Due to this efforts in parallelizing declarative programming languages have been successful.

7. Some disadvantages are :-

Page No.

Date

i) Typically the system lacks knowledge of various components of a computation, and may exploit very fined grained parallelism leading to slow-downs instead of speed-ups.

ii) The system may attempt to parallelize code that is only apparantely parallel, being instead inherently sequential. This may lead to large amount of synchronization points and lead to inefficient execution.

Q5) Write short note on N-wide superscaler architecture.

Ans 1. Superscaler architecture is called as N-wide architecture if it supports to fetch and dispatch of n instructions in every cycle.

2. Common instructions (arthimetic, load/store, conditional branch) can be initated and executed independently in seperate pipelines.

3. Processors with more than one pipeline are called super pipelined processors.

4. The ability of processor to issue multiple instructions in the same cycle is called as superscaler execution.

5. Superscaler architecture helps exploit power of Instruction Level Parallelism [ILP].

6. Superscaler comes with following advantages:-

i) Execute instructions concurrently and independtly in seperate pipelines.

ii) Improve throughput of concurrent pipelines by allowing out of order execution.

7. Superscaler Architecture :-

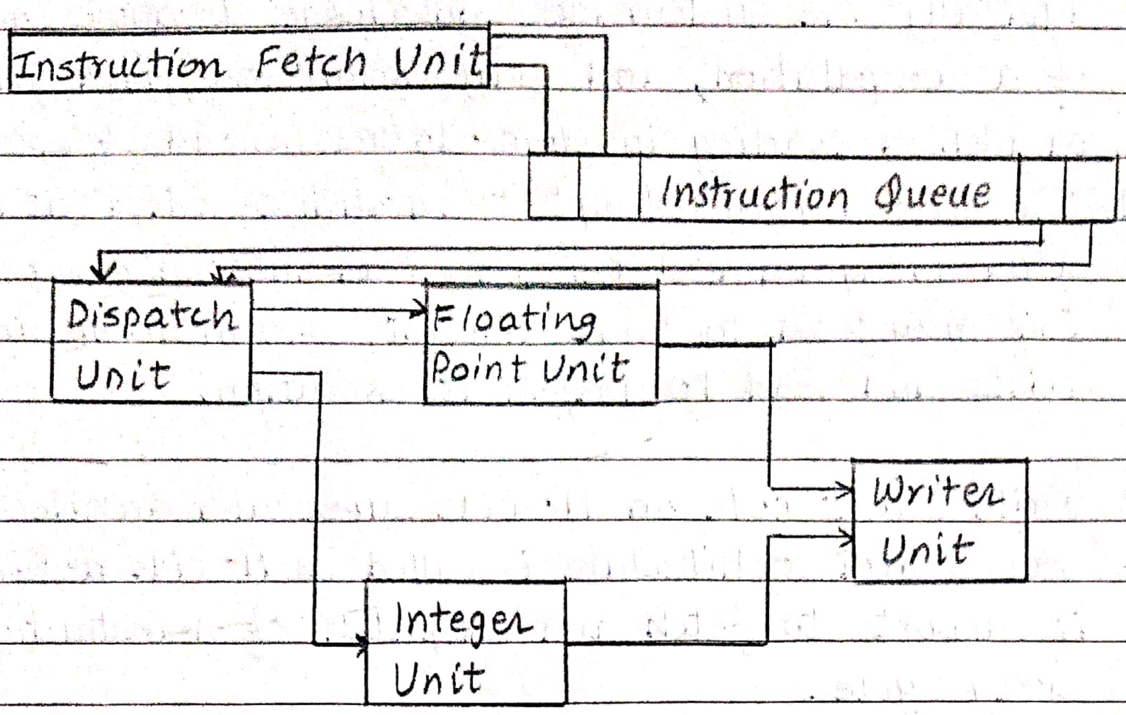


Fig: Superscaler Architecture
(N-Wide)