High Performance Computing

M T W T F S S
Page No.:
Date 29-03-23
YOUVA

Insem
30

Unit I. Introduction to parallel computing

a) What is Parallel computing & Explain Motivating Parallelism.
a) Write short note on Levels of Parallelism.
a) Explain Dichotomy of Parallel computing platforms
a) Explain N-wide superscalar Architecture
a) Explain the communication costs in Parallel Machines.
a) Differentiate Between SIMD and MIMD
a) What is implicit parallelism?.
a) Explain the Pipelining and superscalar Execution & Explain with example
a) Explain example of a two way superscalar implementation of instructions?
a) What is VLIW Design Architecture?. Explain in details with Architecture Design?. Diagram?
a) Explain typical SIMD & MIMD Processors Architecture?. & SIMT Architecture?

a) What is parellel computing ? Explain Motivating parallelism ?

→ - A parallel computer is a set of processors that are able to work cooperatively to solve a computational problem.
- Parallel computing refers to the process of executing several processors an application or computation simutaneously.
- Generally, it is a kind of computing architecture where the large problems break into independent, smaller, usually similar parts that can be processed in one go.
- It is done by multiple CPUs communicating via shared memory, which combines results upon completion.
- parallel computing also helps in faster application processing and task resolution by increasing the available computation power of systems.

Fig: Parallel

① A broke problem is broken into discrete parts that can be solved concurrently
② Each partt is further broken down to a series of instructions.
③ Instructions from each part execute simultaneously on different CPUs

The Types of the levels of Parallelism are.
① Bit-level parallelism
② Instructional Parallelism
③ Data level / loop level parallelism
④ Task / Functional / control level parallelism.

Applications -
① one of the primary applications of parallel computing is Databases and Data mining.
② The real-time simulation of systems is another use of parallel computing
③ The Technologies such as Networked videos and multimedia.
④ science and Engineering
⑤ collaborative work environments
⑥ The concept of parallel computing is used by augmented reality, advanced graphics and virtual reality.

Date:

Motivating Parallelism -

- The part of parallelism in accelerate multiplying speeds has been known for numerous decades.
- Its part in provide multiplicity of data paths and improved admittance to storage elements has been important in profitable applications.
- The scalable show and lower cost of parallel stages is replicated in the wide diversity of application.
- Emerging parallel hardware and software has conventionally been time and effort demanding.
- The emergences of standardized parallel training situations, libraries and hardware have considerably summary time to (parallel) explanation.

Q) Write short note on levels of Parallelism.

→ The Types of the levels of Parallelism are
(1) Data parallelism
(2) Task parallelism
(3) Bit-level parallelism
(4) Instruction-level parallelism

(1) Data parallelism - Data Parallelism means concurrent execution of the same task on each multiple computing core.

(2) Task Parallelism - Task parallelism means concurrent execution of the different task on multiple computing cores.
Task parallelism is the form of parallelism in which the tasks are decomposed into subtasks. Then each sub-task is allocated for execution and the execution of subtasks is performed concurrently by processors.

(3) Bit level parallelism - The form of parallel computing in which every task is dependent on processor word size. In terms of performing a task on large-sized data, it reduces the number of instructions the processorr must execute.

Date:

④ Instruction-level parallelism
In a single CPU clock cycle, the processor decides in the instruction-level parallelism how many instructions are implemented at the same time.
The software approach in instruction-level parallelism functions on static parallelism, where the computer decides which instructions to execute simultaneously.

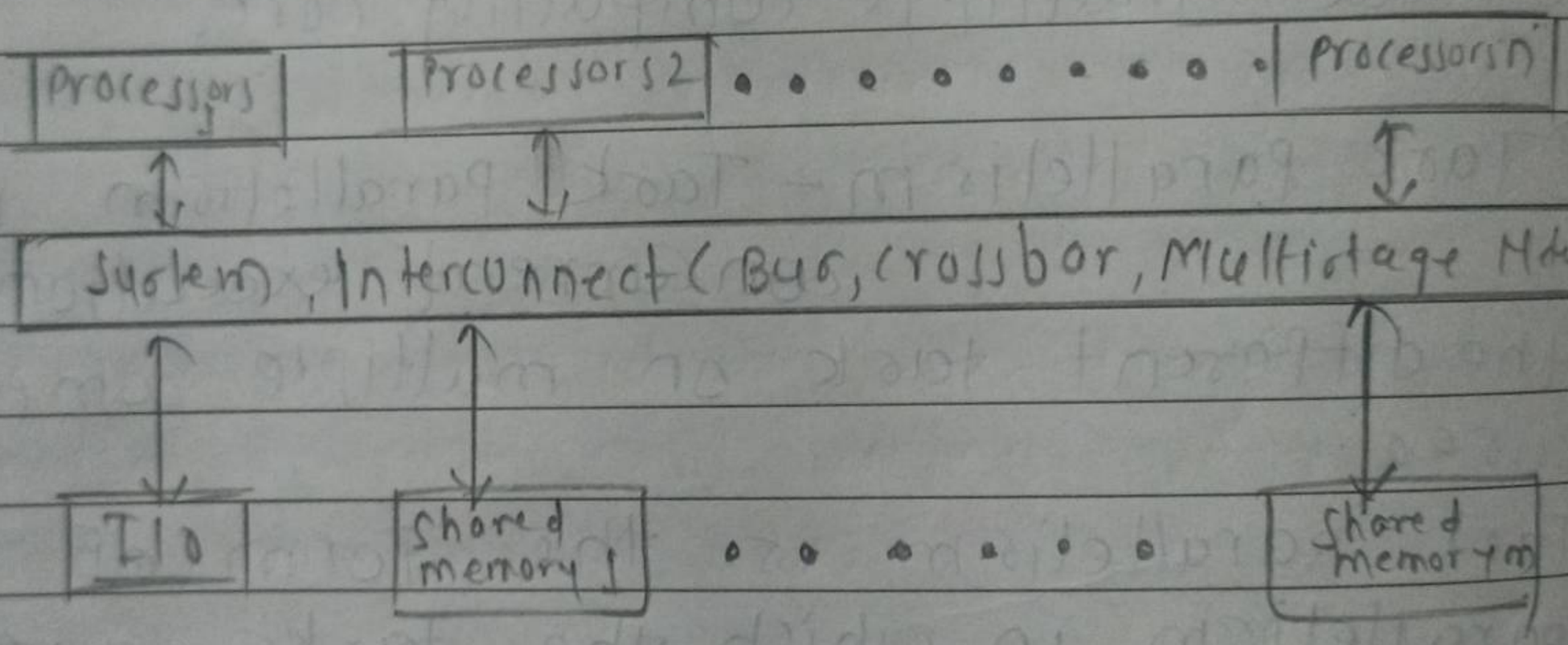


Fig. Parallel computing Architecture

M T W T F S S

Page No.:

Date:

YOUVA

Q) Explain Implicit parallelism in detail.

→ - Implicit parallelism allows programmers to write their programs without any concern about the the exploitation of parallelism

- The compiler, runtime system and the underlying hardware play an important role in exploitting the parallelism implicitly.
- The Parallelism is transparent to the programmer so the programmer will write the standard sequential program without adapting any special parallel constructs
- It is the job of underlying systems to Figure out the parallelism from the sequential code, with the help of different techniques.
- various Implicit parallel mechanisms used in Pipelining and superscalar execution.

Q) Explain VLIW Architecture.

→ - VLIW stand for Very-long Instruction word (VLIW) Architecture.

- It is an appropriate alternative for exploitting instruction-level parallelism (ILP) in programs especially for performing more than one basic instruction at a time.
- These processors include various Funtional units, fetch from the instruction cache a very-long instruction word including various primitive instructions and dispatch the whole VLIW for parallel implementation
- The main goal of VLIW is to remove the complicated instruction scheduling and parallel dispatch that appears in most modern microprocessors.
- A "VLIW processor should be quicker and less costly than a comparable RISC chip

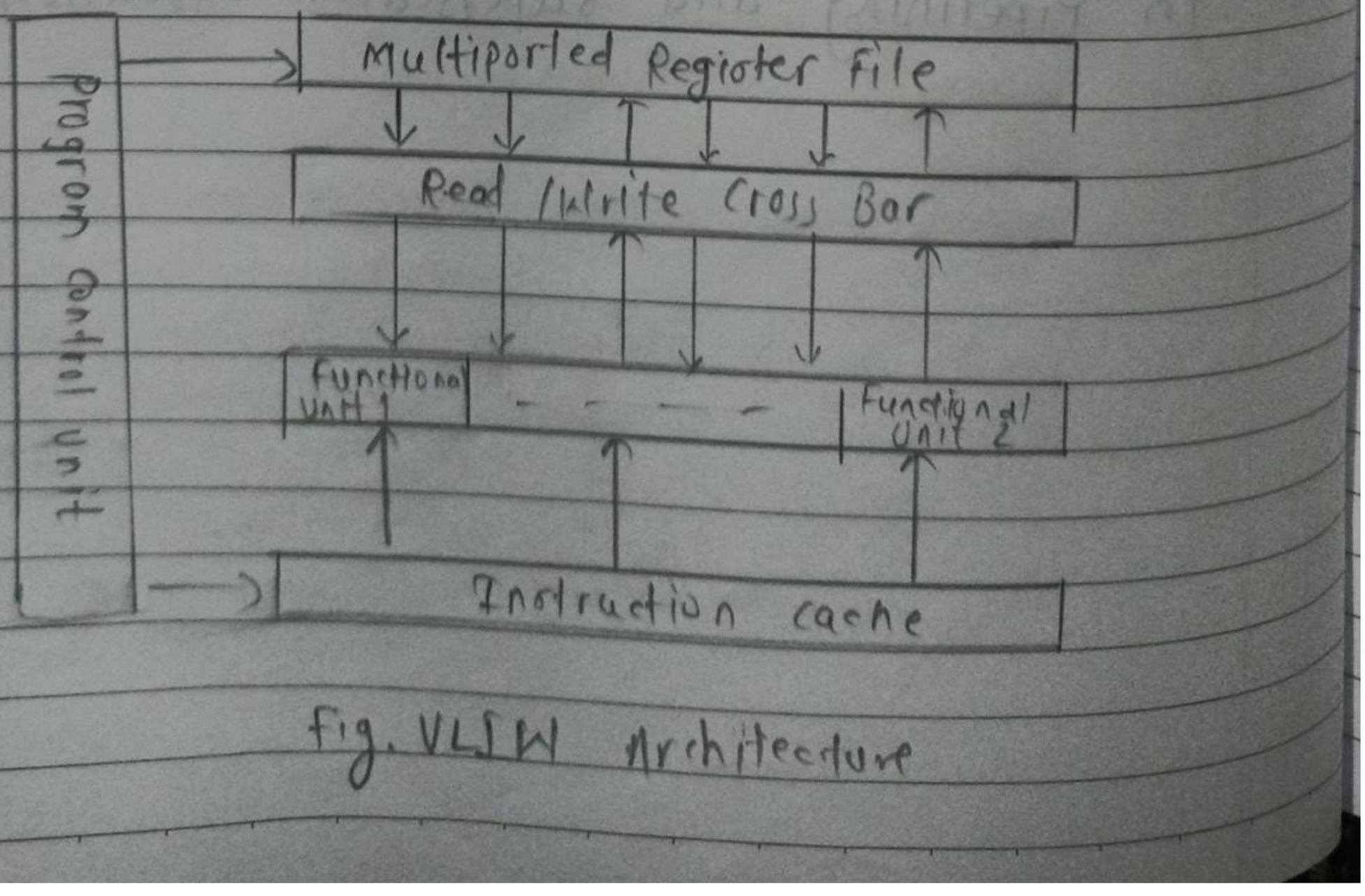


fig. VLIW Architecture

M T W T F S S

Page No.:

Date:

YOUVA

Advantages of VLIW Architecture.

- It can improve performance
- It is used to potentially scalable.
- More implementation units can be inserted and so more instructions can be overflowin into the VLIW instruction

Disadvantages

- It can be used by a new programmer required.
- The program should keep track of instruction scheduling
- It can increased memory use.
- It is used for high power consumption.

Page No.:
Date:
YOUVA

Q) Explain Dichotomy of Parallel computing platforms.

→ A dichotomy is based on the logical and physical organizations of parallel platforms

- physical organisation is the actual hardware organisation of a platform whereas logical organisation refers to a programmer's view of the platform
- From programmer's perspective the two important components of parallel computing are.

① control structure - The various way of expressing parallel tasks is known as control structure.

② Communication model - The mechanisms for specifying interaction between the parallel tasks is called as communication model.

Page No.:
Date:
YOUVA

Q) Explain typical SIMD & MIMD processors architecture &.

→ SIMD and MIMD are types of computer Architectures that are used to improve the performance of certain types of computational tasks.

- Michael flynn classified the computer organization into SIMD and MIMD. where SIMD stands for single Instruction Multiple data, MIMD stands for (multiple Instruction Multiple data)
- SIMD short for single Instruction Multiple data, computer architecture can execute a single instruction on multiple data streams.
- MIMD computer architectures can execute several instructions on multiple data streams

SIMD - is a form of parallel computer architecture that is categorized under Flynn's classification given by Michael flynn.
- In SIMD architecture, a single instruction is applied to several data streams.
- SIMD consists of a single control signal that is used to call several isolated processing units. Therefore all the processing units accept the same instruction from the control unit and use it on separate element of data.
- The SIMD organization uses shared memory unit which is divided into different modules.

Page No.:
Date:
YOUVA

SIMD As a result the memory unit can interact with all the processing units simultaneously. since the SIMD architecture uses a single copy of instruction on multiple data streams it requires less memory.
SIMD architectures are particularly effective for tasks that can be easily parallelized such as Image processing, video encoding and decoding.

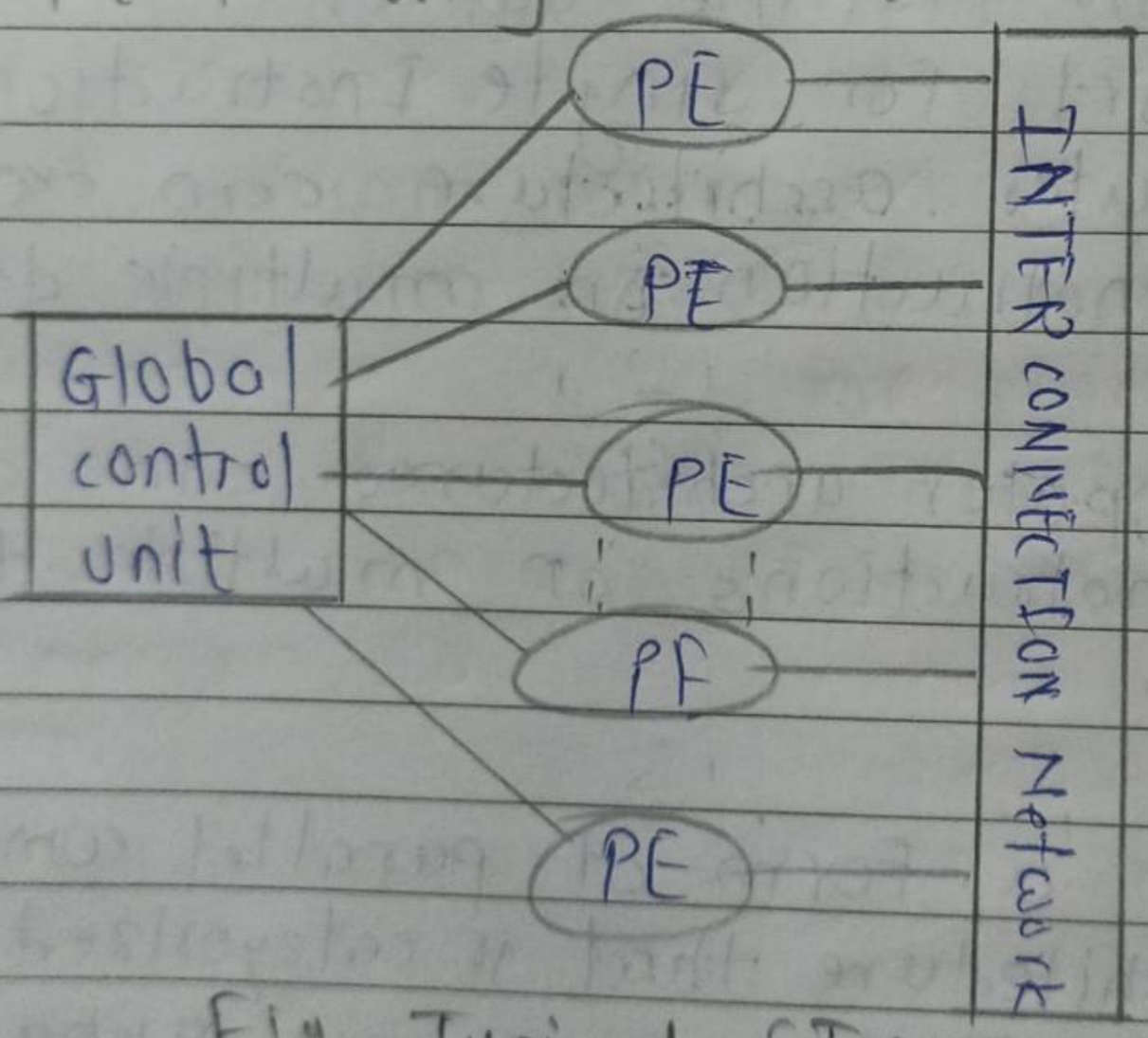


Fig Typical SIMD Architecture.

MIMD - The MIMD architecture consists of multiple instructions and data streams. Therefore MIMD architecture requires multiple processing units
- MIMD systems are considered to have most complex organisation, but they are highly efficient.

- The MIMD architecture does not need any addition control unit which reduces the cost of the system.
- It also provides efficient execution of the conditional statement such as if/else statements.
- MIMD architectures are more flexible and are better suited for tasks that require more complex and varied computation such as general-purpose computing and AI Applications.

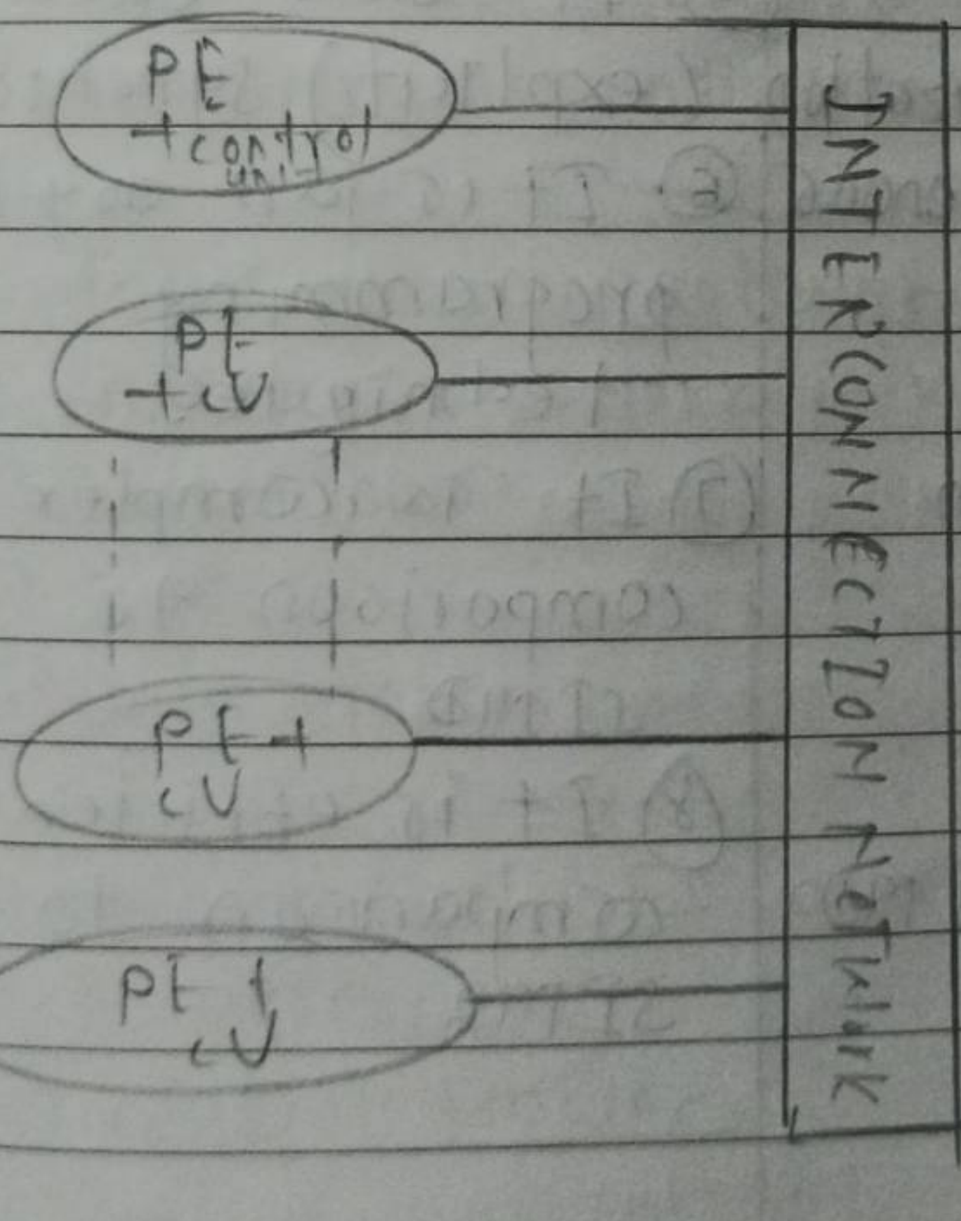


Fig. MIMD Architecture.

Page No.:
Date:

Q) difference between SIMD & MIMD

SIMD	MIMD
① SIMD stands for single Instruction multiple Data.	① MIMD stands for multiple Instruction multiple Data.
② It requires less memory	② It requires more memory.
③ It is inexpensive in comparision to MIMD	③ It is expensive in comparision to SIMD.
④ It has a single decoder.	④ It contains multiple decoder.
⑤ It uses latent (tacit) synchronization	⑤ It uses accurate (explicit) synchronization
⑥ It is a synchronous programming technique.	⑥ It is an asynchronous programming technique
⑦ It is simple in comparison to MIMD.	⑦ It is complex in comparison to SIMD
⑧ It is not as efficient as MIMD in terms of performance.	⑧ It is efficient in comparison to SIMD.

Q) Explain Data flow diagrams

→ - Functional modding is represented through a hierarchy of DfDs.
- The DfD is a graphical representation of a system that shows the inputs to the system, the processing upon the inputs, the outputs of the system as well as the internal data stores.

→ DFD is A data flow diagram is a graph which shows the flow of data values from their sources in objects through processes that transform them to their destinations on other objects".

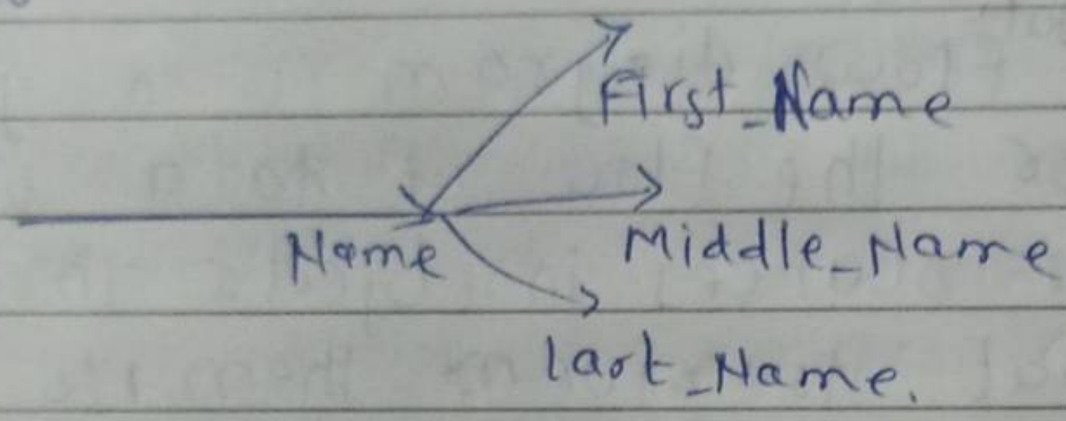
The four main parts of DFD are.
① processes
② Data Flows
③ Actors
④ Data stores.

other parts of DFD are.
① constraints
② control flows.

Features of a DfD
① Processes - processes are the computational activities that transform data values.
- A process may be further divided into smaller components. The lowest level process may be a simple function.

(2) Dataflows -

- Data flows represent the flow of data between two processes.
- It could be between an actor and a process.
- A data flow is represented by a directed arc or an arrow, labeled with name of the data item that it carries.

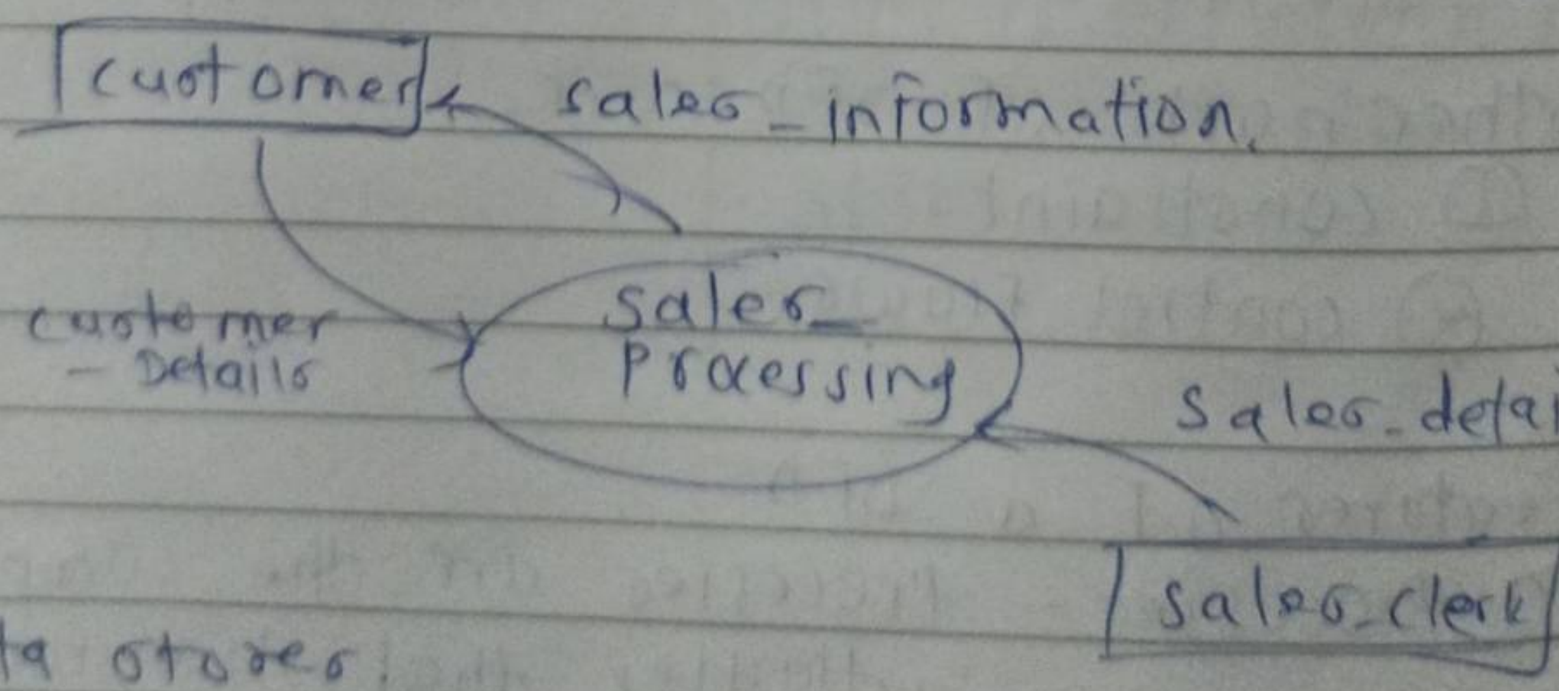


(3) Actors

Actors are the active objects that interact with the system by either producing data and inputting them to the system.

- An Actor is represented by a rectangle.

ex



(4) Data stores:

- Data stores are the passive objects that act as a repository of data.
- A data store is represented by two parallel lines containing the name of the data store.

a) N-wide Superscalar Architectures.

⇒ - The combination of temporal parallelism and data parallelism by issuing several instructions simultaneously in each cycle, to improve the speed of a processor, is called superscalar processing.
- In an n-issue superscalar, n instructions are fetched, decoded, executed and committed per cycle.
-

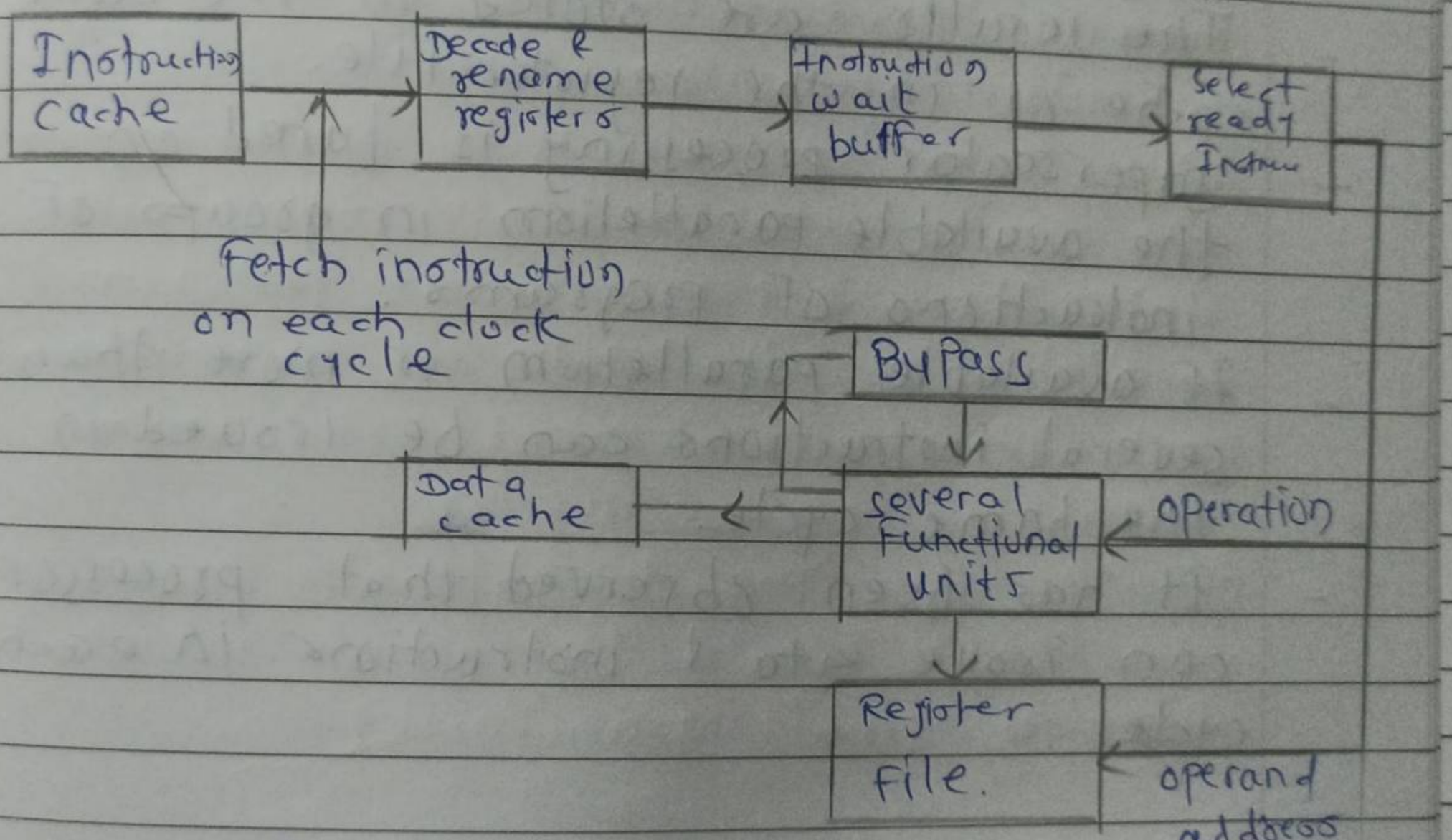


Fig. Superscalar Processor

pipeline stages

- The fetch unit ~~fethes~~ fetches several instructions in each clock cycle from the instruction cache.
- The decode unit decodes the instructions and renames registers so that false dependencies are eliminated and only true dependencies remain.

Date: YOUVA

- The instructions are sent to an instruction buffer where they are kept temporarily till the source operands are identified and become available.
- Ready instructions are dispatched based on the availability of functional units
- The operands are fetched either from the register file or from bypass path from earlier instructions which produce these operands.
- The results are stored in the data cache or in the register file.
- superscalar processing is based on the available parallelism in groups of instructions of programs.
- If available parallelism is more than several instructions can be issued in the same cycle.
- It has been observed that processors can issue 4 to 5 instructions in each cycle.