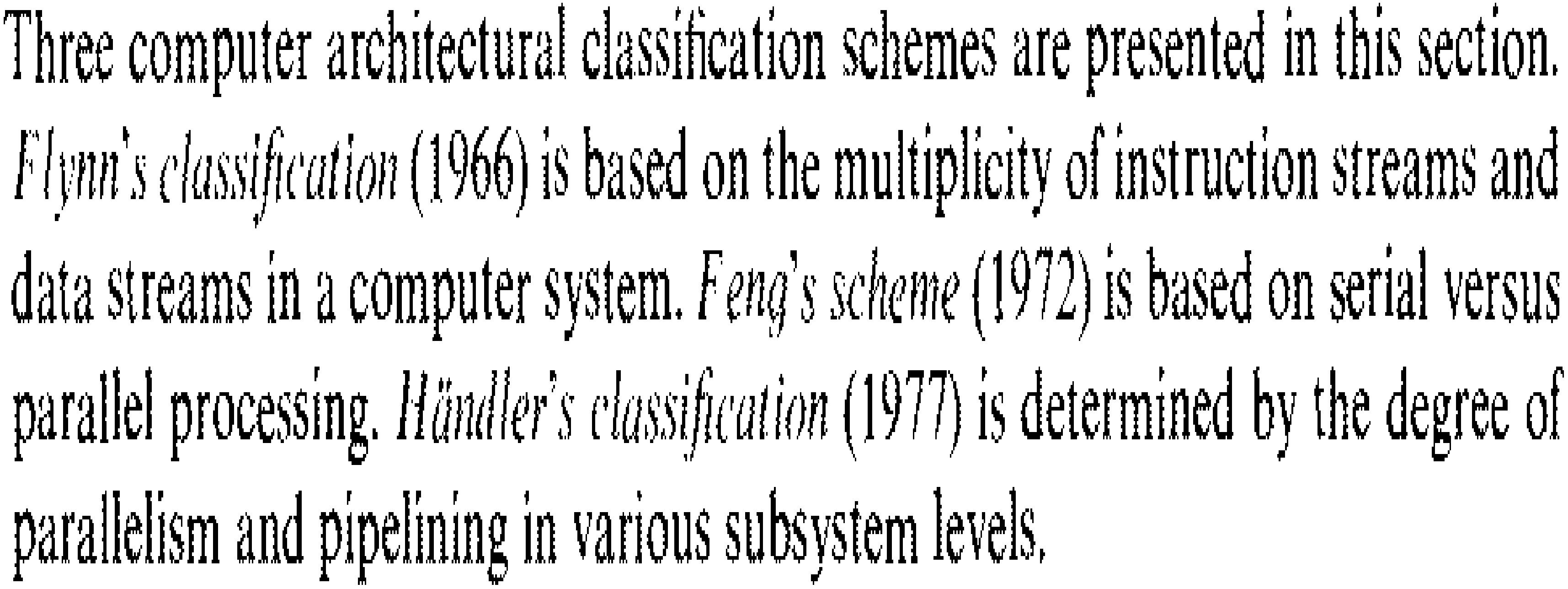
Introduction

Click to add text

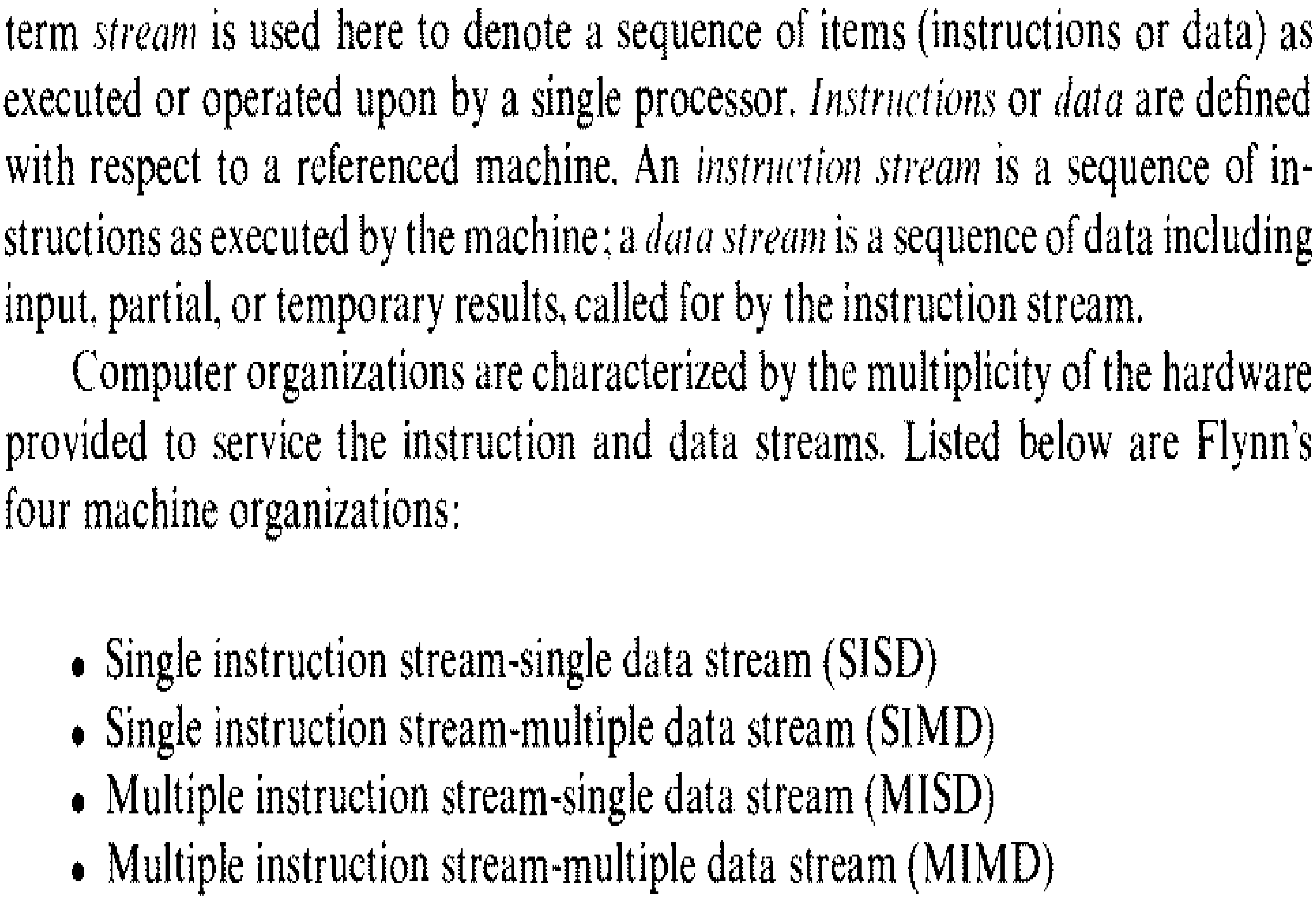
Unit 1

Architectural classification schemes



Three computer architectural classification schemes are presented in this section. *Flynn's classification* (1966) is based on the multiplicity of instruction streams and data streams in a computer system. *Feng's scheme* (1972) is based on serial versus parallel processing. *Händler's classification* (1977) is determined by the degree of parallelism and pipelining in various subsystem levels.

Flynn’s classification



term *stream* is used here to denote a sequence of items (instructions or data) as executed or operated upon by a single processor. *Instructions* or *data* are defined with respect to a referenced machine. An *instruction stream* is a sequence of instructions as executed by the machine; a *data stream* is a sequence of data including input, partial, or temporary results, called for by the instruction stream.

Computer organizations are characterized by the multiplicity of the hardware provided to service the instruction and data streams. Listed below are Flynn’s four machine organizations:

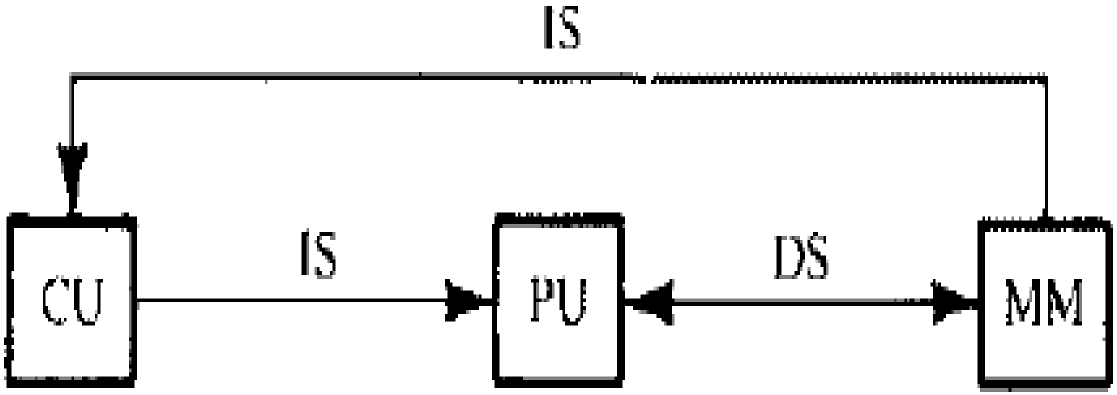
- Single instruction stream-single data stream (SISD)
- Single instruction stream-multiple data stream (SIMD)
- Multiple instruction stream-single data stream (MISD)
- Multiple instruction stream-multiple data stream (MIMD)

s

needed in the illustration. Both instructions and data are fetched from the *memory modules*. Instructions are decoded by the *control unit*, which sends the decoded instruction stream to the *processor units* for execution. Data streams flow between the processors and the memory bidirectionally. Multiple memory modules may

SISD Computer Organization

- Instructions are executed sequentially but may be overlapped in their execution stages(piplining)
- Most SISD uniprocessor systems are pipelined.

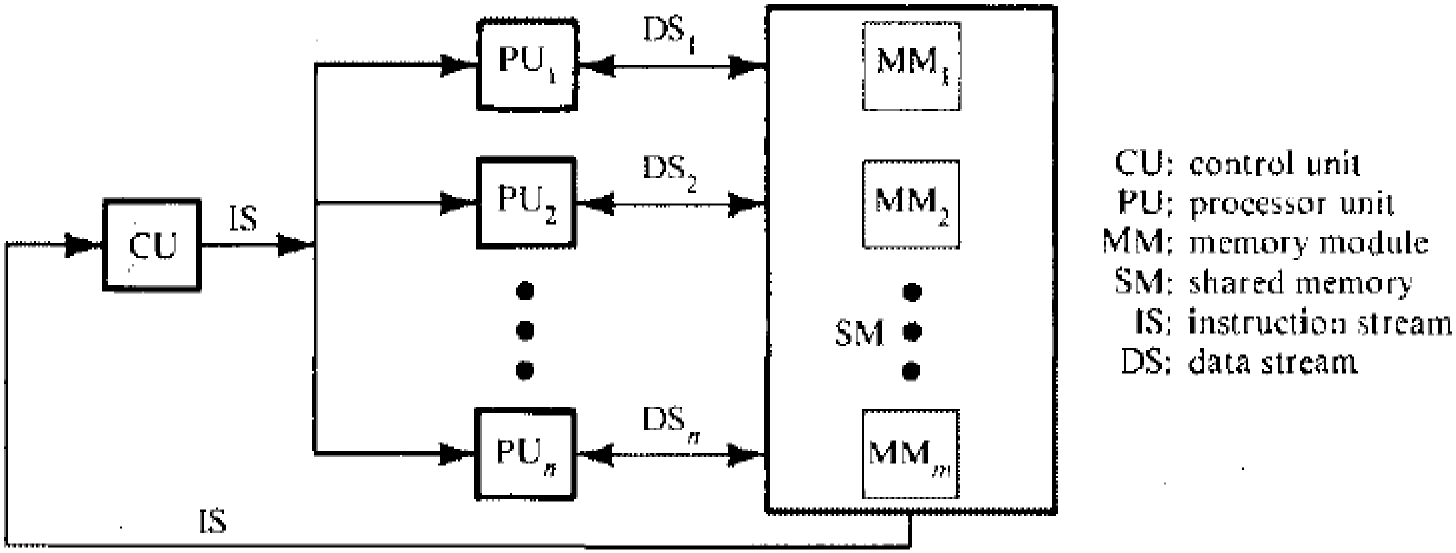


(a) SISD computer

CU: control unit
PU: processor unit
MM: memory module
SM: shared memory
IS: instruction stream
DS: data stream

SIMD Computer Organization

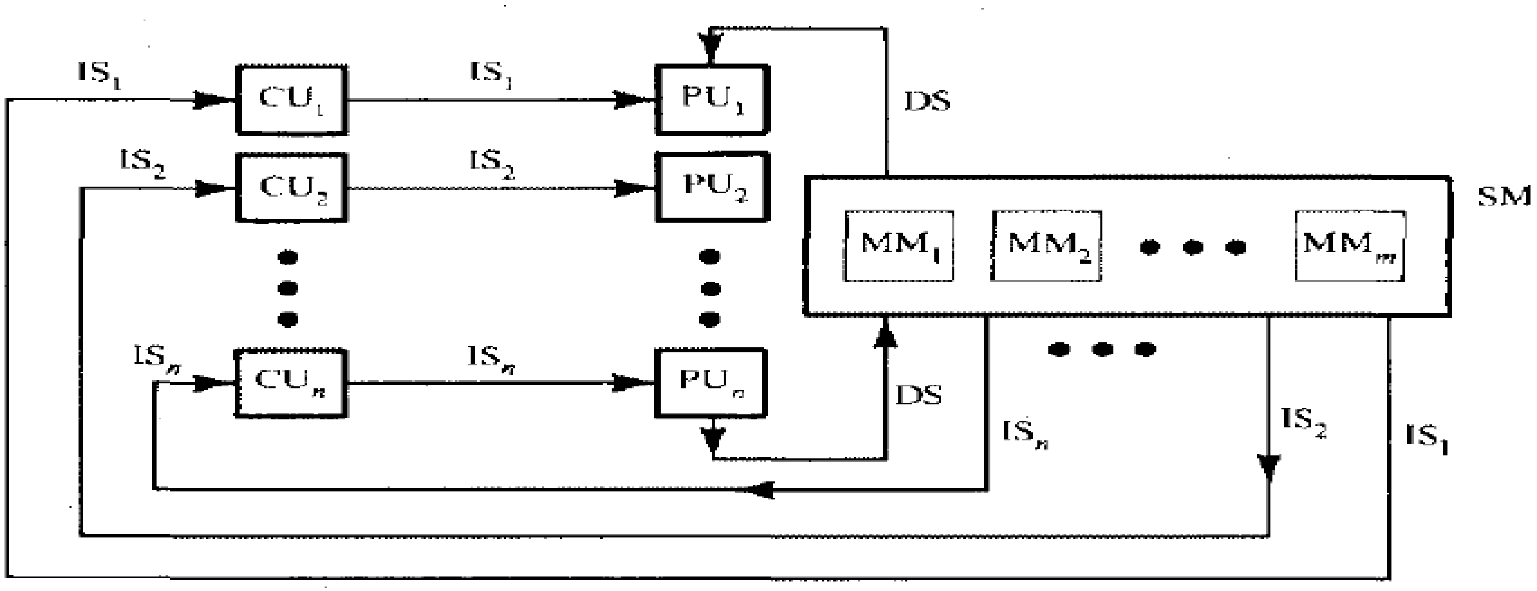
- This class corresponds to array processors.
- There are multiple processing elements supervised by the same control unit.
- All Pes receive the same instruction broadcast from the control unit but operate on different data sets from distinct data streams.
- Shared memory subsystem may contain multiple modules.



(*b*) SIMD computer

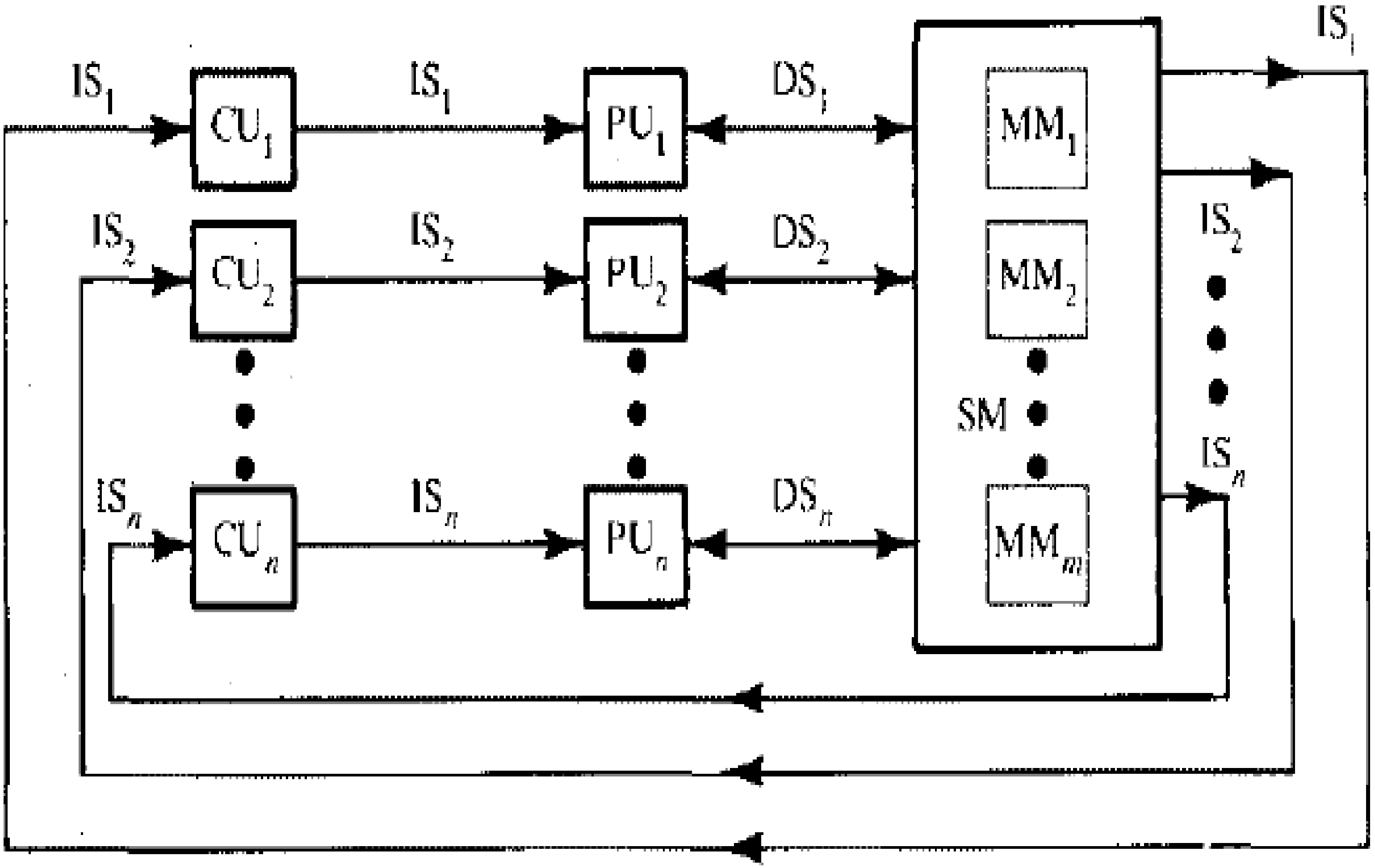
MISD Computer Organization

Figure 1.16c. There are *n* processor units, each receiving distinct instructions operating over the same data stream and its derivatives. The results (output) of one processor become the input (operands) of the next processor in the macropipe.



(c) MISD computer

MIMD Computer Organization

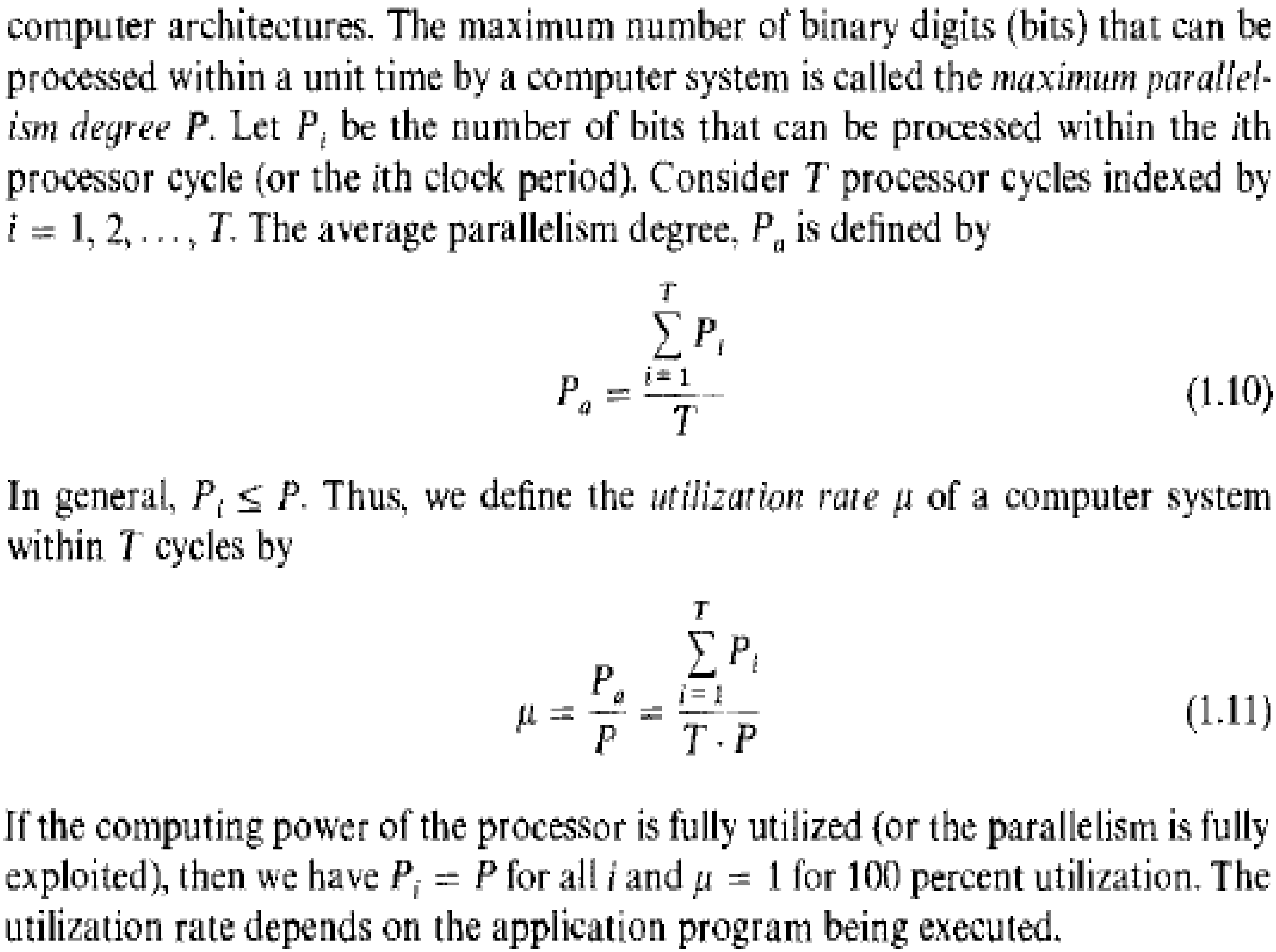


(*d*) MIMD computer

Table 1.3 Flynn's computer system classification

Computer class	Computer system models (chapters where the system is quoted or described)
SISD (uses one functional unit)	IBM 701 (1); IBM 1620 (1); IBM 7090 (1); PDP VAX11/780 (1).
SISD (with multiple functional units)	IBM 360/91 (3); IBM 370/168UP (1); CDC 6600 (1); CDC Star-100 (4); TI-ASC (4); FPS AP-120B (4); FPS-164 (4); IBM 3838 (4); Cray-1 (4); CDC Cyber-205 (4); Fujitsu VP-200 (4); CDC-NASF (4); Fujitsu FACOM-230/75 (4).
SIMD (word-slice processing)	Illiac-IV (6); PEPE (1); BSP (6)
SIMD (bit-slice processing)	STARAN (1); MPP (6); DAP (1).
MIMD (loosely coupled)	IBM 370/168 MP (9); Univac 1100/80 (9); Tandem/16 (9); IBM 3081/3084 (9); C.m* (9)
MIMD (tightly coupled)	Burroughs D-825 (9); C.mmp (9); Cray-2 (9). S-1 (9); Cray-X MP (9); Denelcor HEP (9)

Fengs classification



computer architectures. The maximum number of binary digits (bits) that can be processed within a unit time by a computer system is called the *maximum parallelism degree P*. Let P_i be the number of bits that can be processed within the ith processor cycle (or the ith clock period). Consider T processor cycles indexed by $i = 1, 2, \ldots, T$. The average parallelism degree, P_a is defined by

$$P_a = \frac{\sum_{i=1}^{T} P_i}{T} \qquad (1.10)$$

In general, $P_i \leq P$. Thus, we define the *utilization rate* μ of a computer system within T cycles by

$$\mu = \frac{P_a}{P} = \frac{\sum_{i=1}^{T} P_i}{T \cdot P} \qquad (1.11)$$

If the computing power of the processor is fully utilized (or the parallelism is fully exploited), then we have $P_i = P$ for all i and $\mu = 1$ for 100 percent utilization. The utilization rate depends on the application program being executed.

A bit slice is a string of bits one from each of the words

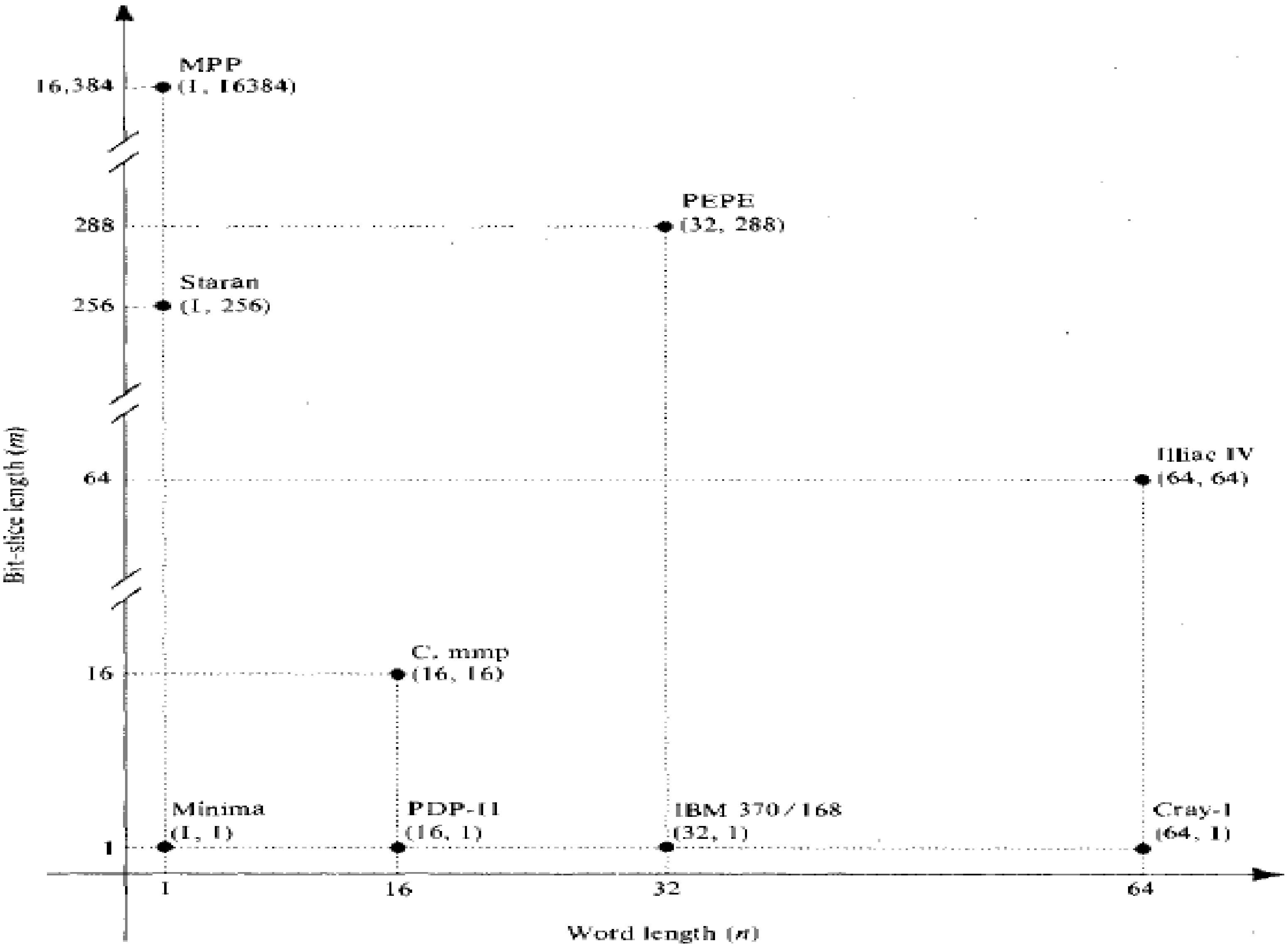
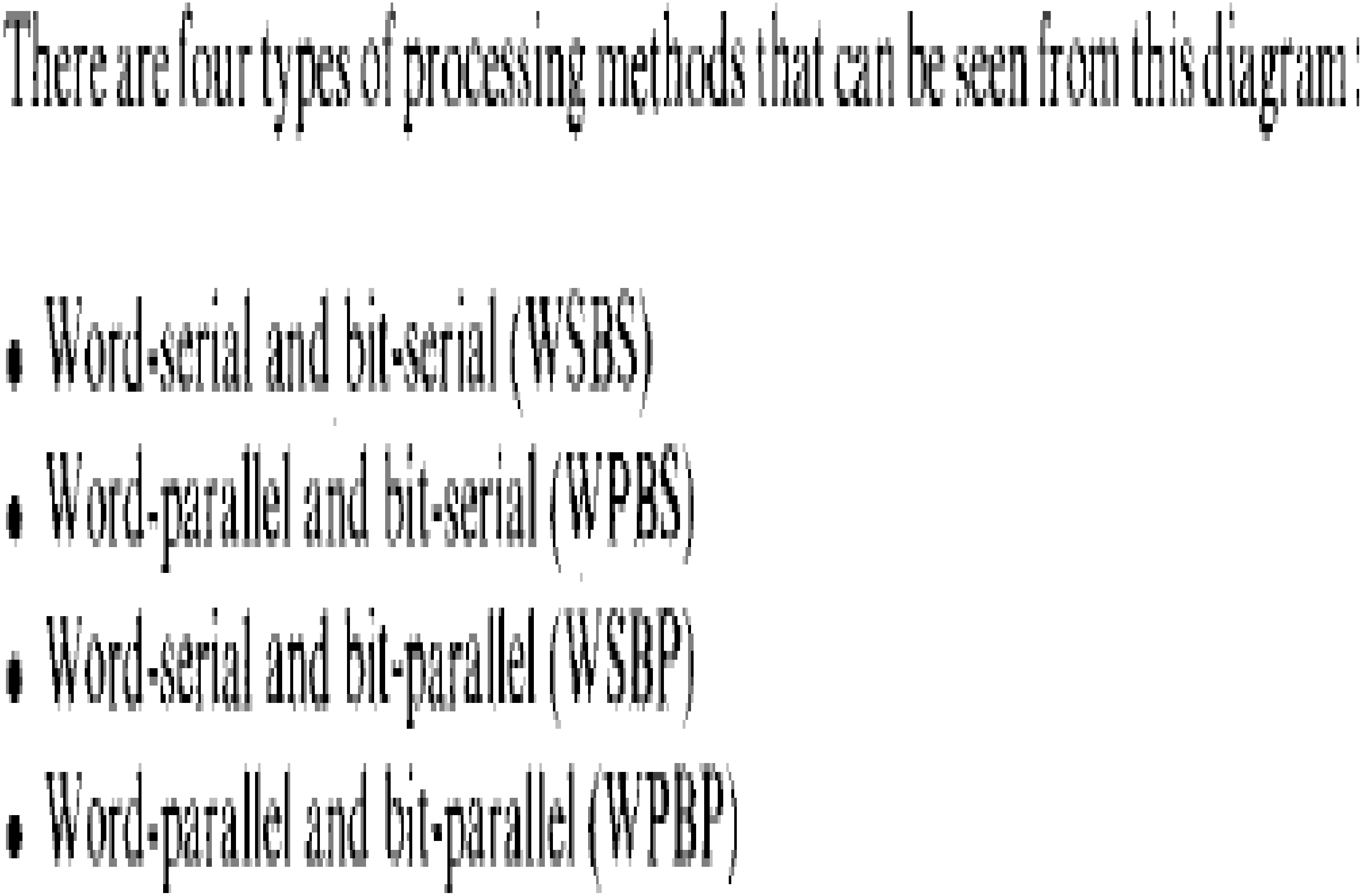


Figure 1.17 Feng's classification of computer systems in terms of parallelism exhibited by word length and bit-slice length.

s



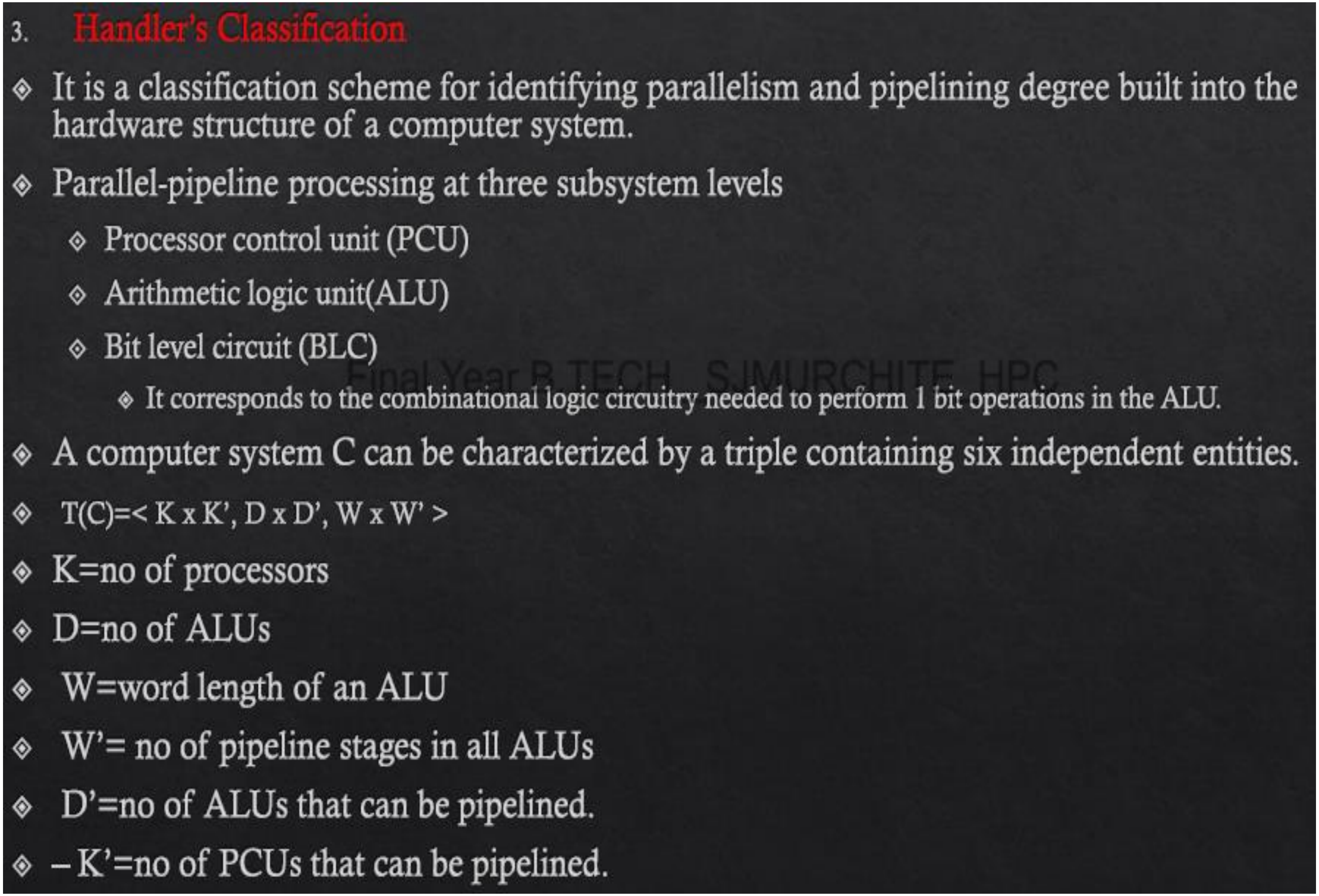
There are four types of processing methods that can be seen from this diagram :

- Word-serial and bit-serial (WSBS)
- Word-parallel and bit-serial (WPBS)
- Word-serial and bit-parallel (WSBP)
- Word-parallel and bit-parallel (WPBP)

s

- WSBS called bit serial processing because one bit is processed at a time
- WPBS called bit slice processing because m bit slice is processed at a time.
- WSBP called word slice processing because one word of n bits is processed at a time.
- WPBP is known as fully parallel processing in which an array of n.m bits processed at one time, the fastest processing mode.

Parallelism Versus Pipelining



3. Handler's Classification

- It is a classification scheme for identifying parallelism and pipelining degree built into the hardware structure of a computer system.
- Parallel-pipeline processing at three subsystem levels
  - Processor control unit (PCU)
  - Arithmetic logic unit(ALU)
  - Bit level circuit (BLC)
    - It corresponds to the 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' >
- K=no of processors
- D=no of ALUs
- W=word length of an ALU
- W'= no of pipeline stages in all ALUs
- D'=no of ALUs that can be pipelined.
- – K'=no of PCUs that can be pipelined.

Several real computer examples are used to clarify the above parametric descriptions. The Texas Instrument's Advanced Scientific Computer (TI-ASC) has one controller controlling four arithmetic pipelines, each has 64-bit word lengths and eight stages. Thus, we have

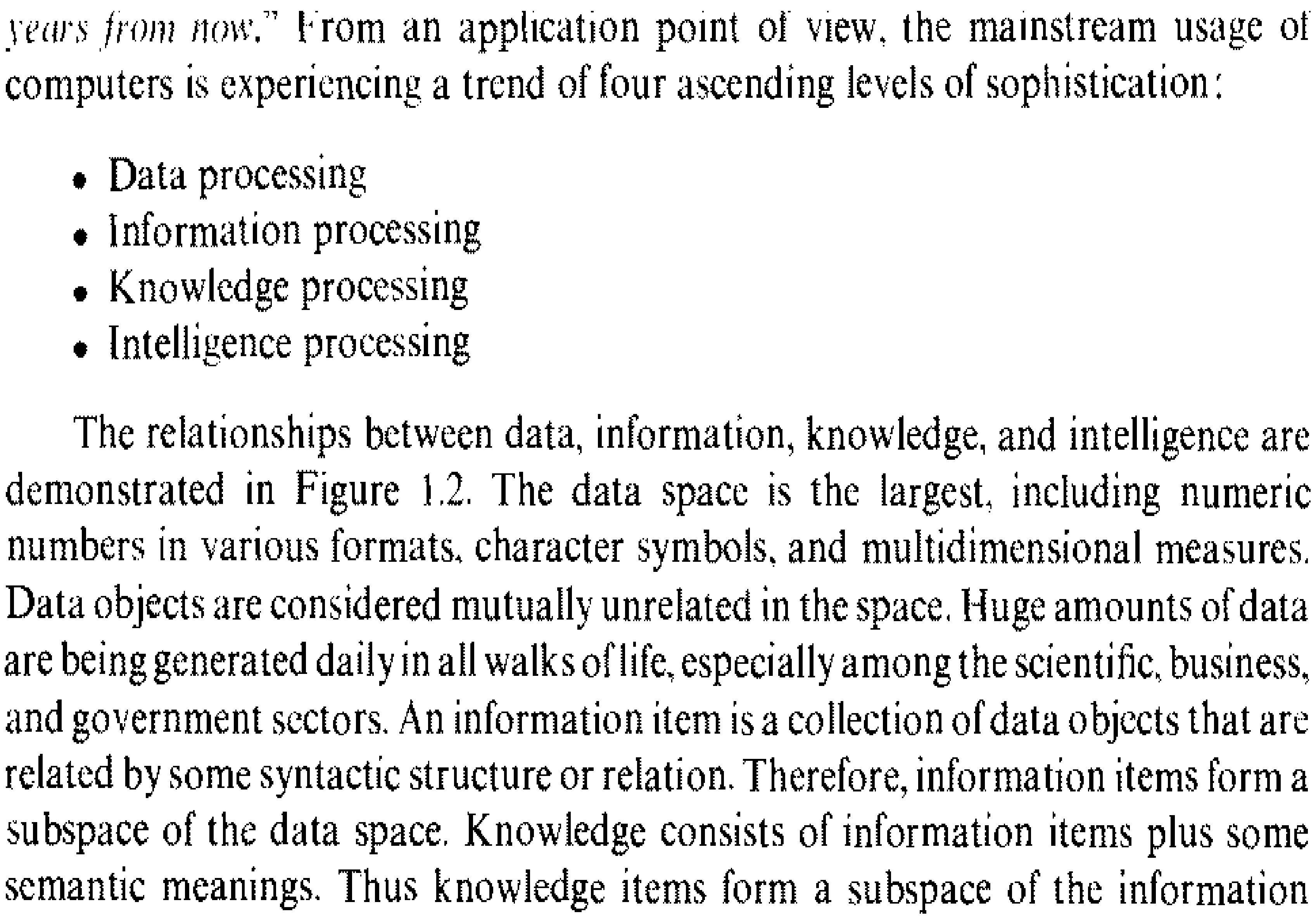
$$T(\text{ASC}) = \langle 1 \times 1, 4 \times 1, 64 \times 8 \rangle = \langle 1, 4, 64 \times 8 \rangle \tag{1.14}$$

Whenever the second entity, K', D', or W', equals 1, we drop it, since pipelining of one stage or of one unit is meaningless.

Another example is the Control Data 6600, which has a CPU with an ALU that has 10 specialized hardware functions, each of a word length of 60 bits. Up to 10 of these functions can be linked into a longer pipeline. Furthermore, the CDC-6600 has 10 peripheral I/O processors which can operate in parallel. Each I/O processor has one ALU with a word length of 12 bits. Thus, we specify 6600 in two parts, using the operator × to link them:

$$\begin{aligned} T(\text{CDC 6600}) &= T(\text{central processor}) \times T(I/O \text{ processors}) \\ &= \langle 1, 1 \times 10, 60 \rangle \times \langle 10, 1, 12 \rangle \end{aligned} \tag{1.15}$$

Trends Towards parallel processing



years from now." From an application point of view, the mainstream usage of computers is experiencing a trend of four ascending levels of sophistication :

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

The relationships between data, information, knowledge, and intelligence are demonstrated in Figure 1.2. The data space is the largest, including numeric numbers in various formats, character symbols, and multidimensional measures. Data objects are considered mutually unrelated in the space. Huge amounts of data are being generated daily in all walks of life, especially among the scientific, business, and government sectors. An information item is a collection of data objects that are related by some syntactic structure or relation. Therefore, information items form a subspace of the data space. Knowledge consists of information items plus some semantic meanings. Thus knowledge items form a subspace of the information

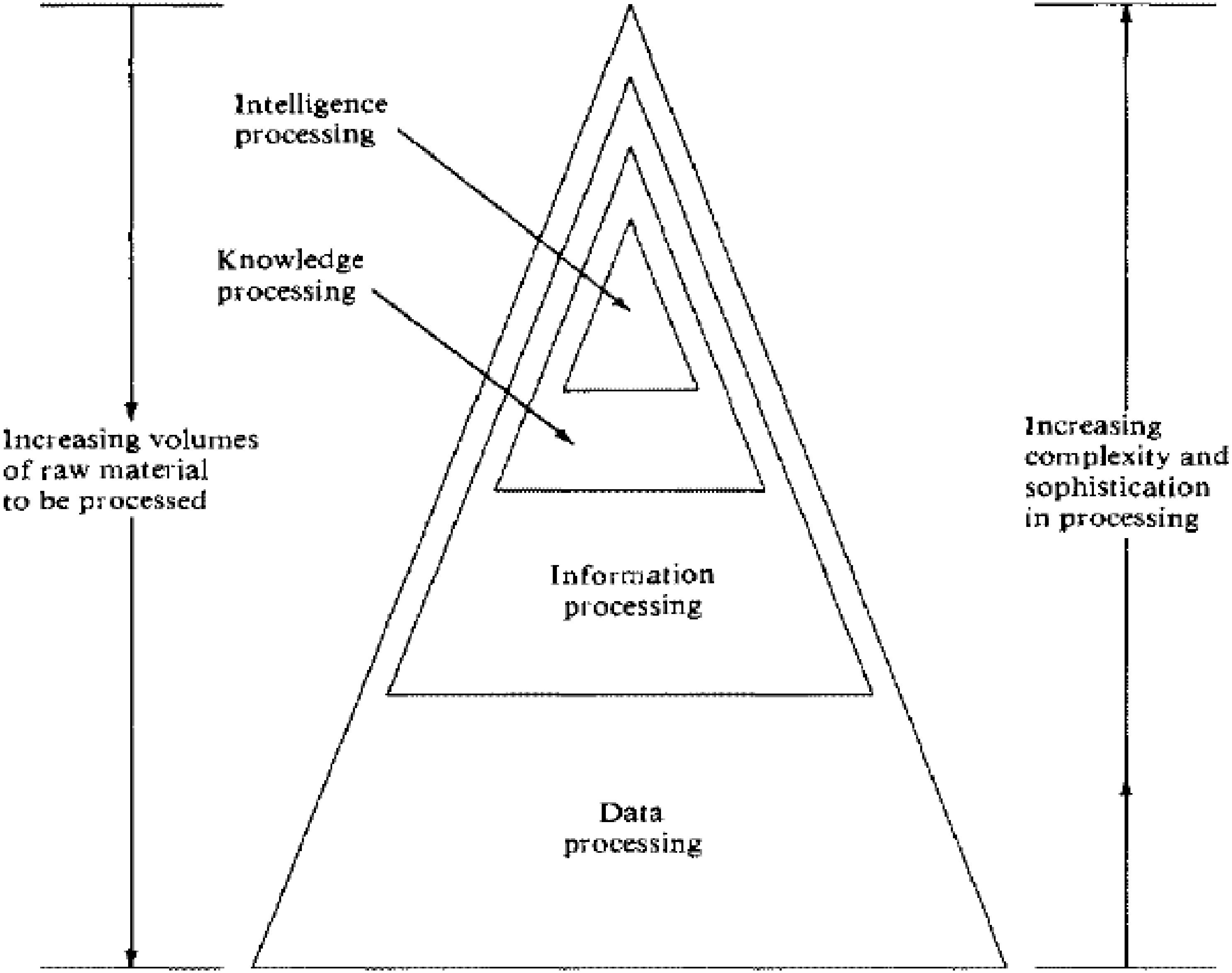
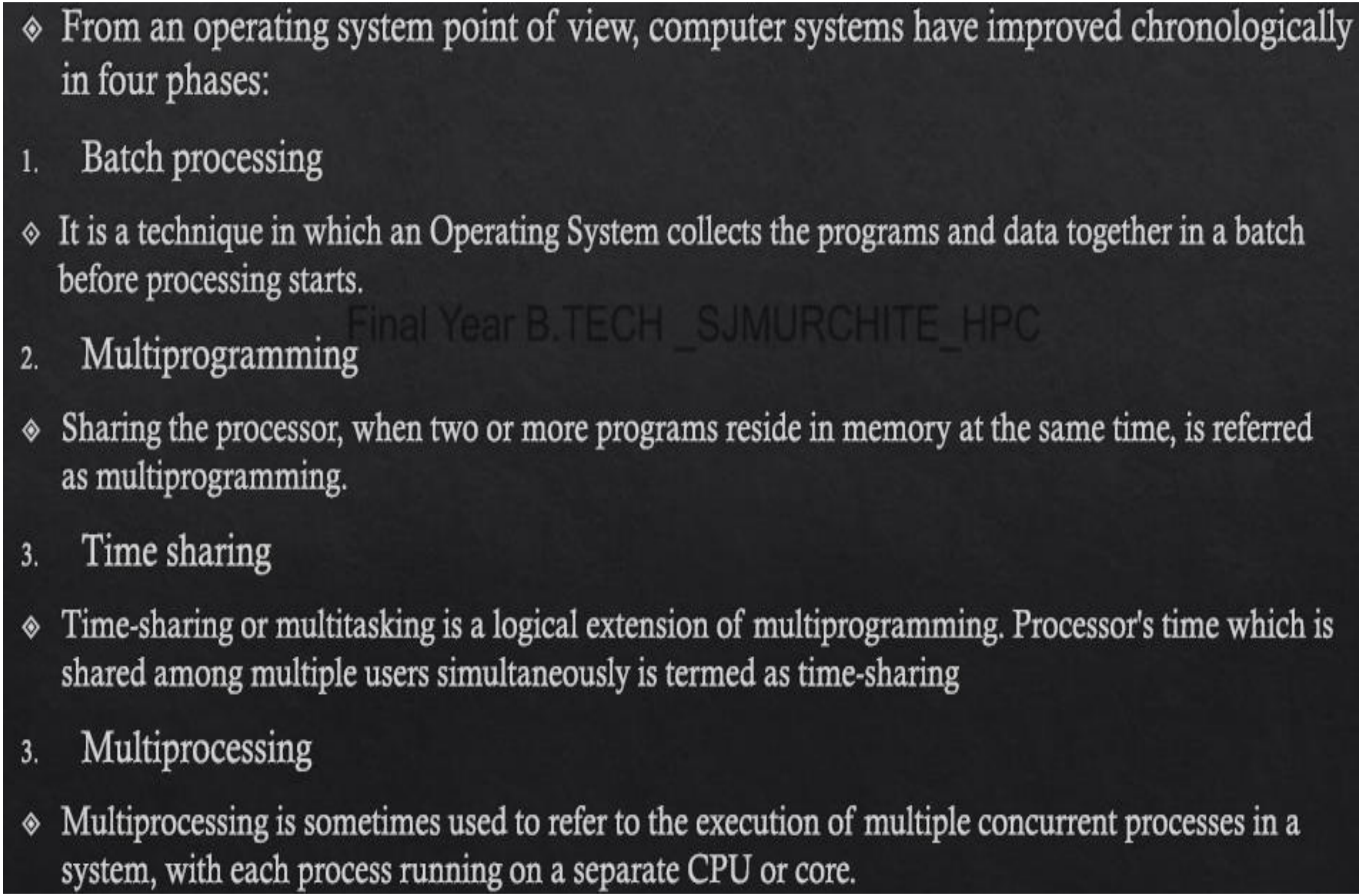

Intelligence processing
Knowledge processing
Increasing volumes of raw material to be processed
Information processing
Data processing
Increasing complexity and sophistication in processing

Figure 1.2 The spaces of data, information, knowledge, and intelligence from the viewpoint of computer processing.

space. Finally, intelligence is derived from a collection of knowledge items. The intelligence space is represented by the innermost and highest triangle in the Venn diagram.

Computer usage started with *data processing*, which 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 (mainly number crunching) to *information processing*. Most of today's computing is still confined within these two processing levels. 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*. For example, the

Future trends towards parallel processing



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

1. Batch processing

- It is a technique in which an Operating System collects the programs and data together in a batch before processing starts.

2. Multiprogramming

- Sharing the processor, when two or more programs reside in memory at the same time, is referred as multiprogramming.

3. Time sharing

- Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing

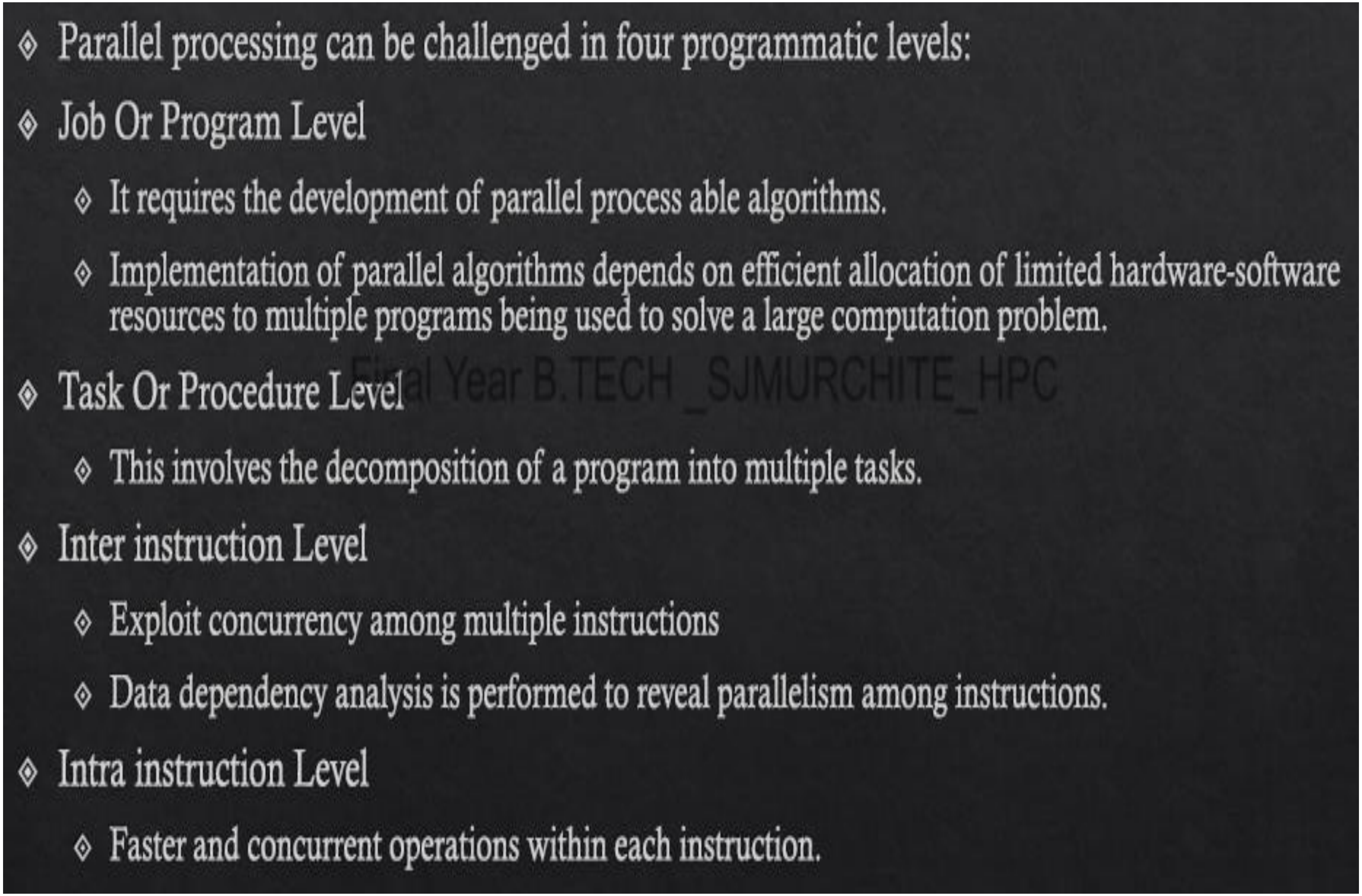
3. Multiprocessing

- Multiprocessing is sometimes used to refer to the execution of multiple concurrent processes in a system, with each process running on a separate CPU or core.

In these four operating modes, the degree of parallelism increases sharply from phase to phase. The general trend is to emphasize parallel processing of information. In what follows, the term *information* is used with an extended meaning to include data, information, knowledge, and intelligence. We formally define *parallel processinq* as follows:

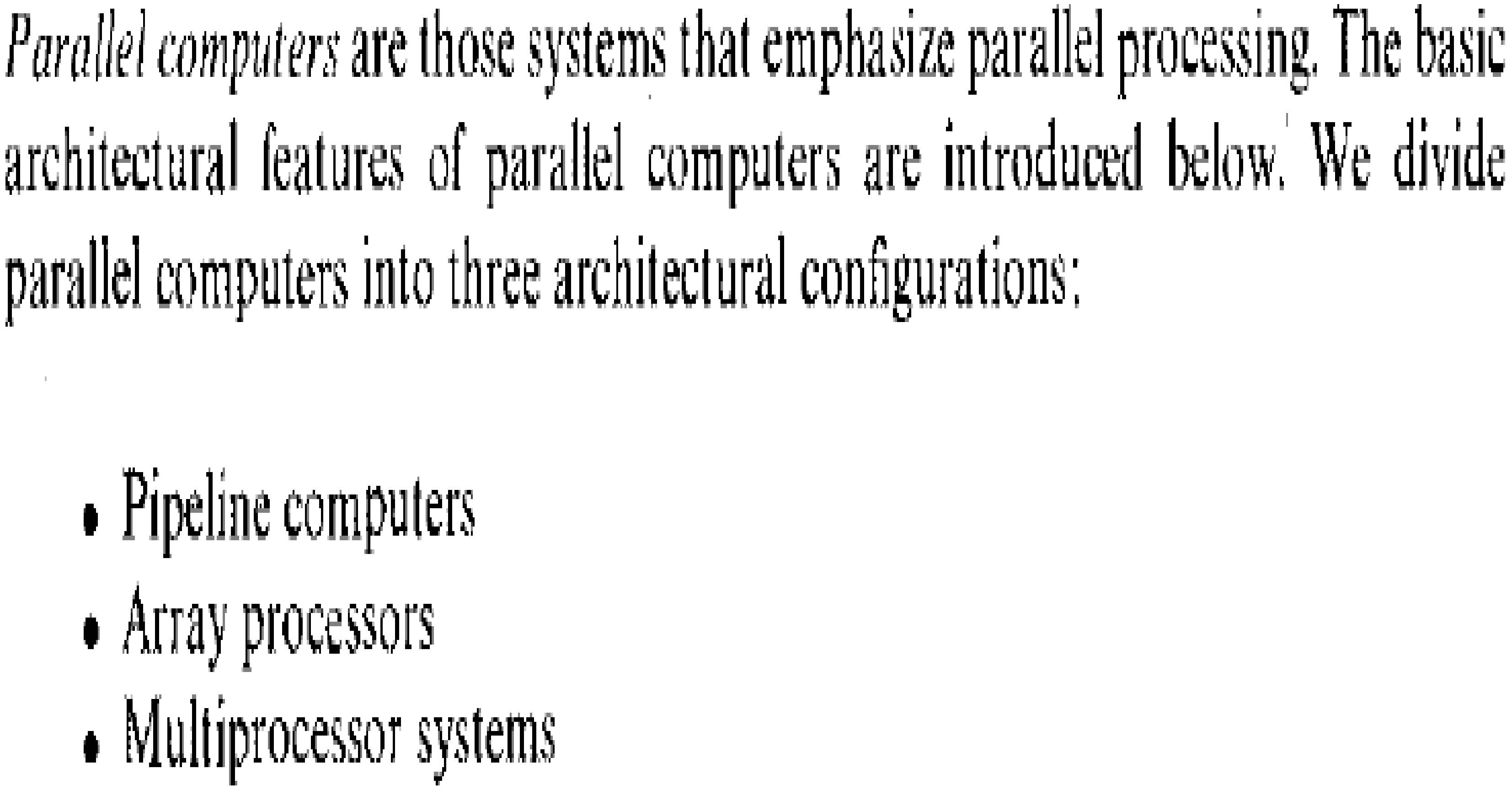
Definition Parallel processing is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process. Concurrency implies parallelism, simultaneity, and pipelining. Parallel events may occur in multiple resources during the same time interval; simultaneous events may occur at the same time instant; and pipelined events may occur in overlapped time spans. These concurrent events are attainable in a computer system at various processing levels. Parallel processing demands concurrent execution of many programs in the computer. It is in contrast to

Future trends towards parallel processing



- Parallel processing can be challenged in four programmatic levels:
- Job Or Program Level
  - It requires the development of parallel process able algorithms.
  - Implementation of parallel algorithms depends on efficient allocation of limited hardware-software resources to multiple programs being used to solve a large computation problem.
- Task Or Procedure Level
  - This involves the decomposition of a program into multiple tasks.
- Inter instruction Level
  - Exploit concurrency among multiple instructions
  - Data dependency analysis is performed to reveal parallelism among instructions.
- Intra instruction Level
  - Faster and concurrent operations within each instruction.

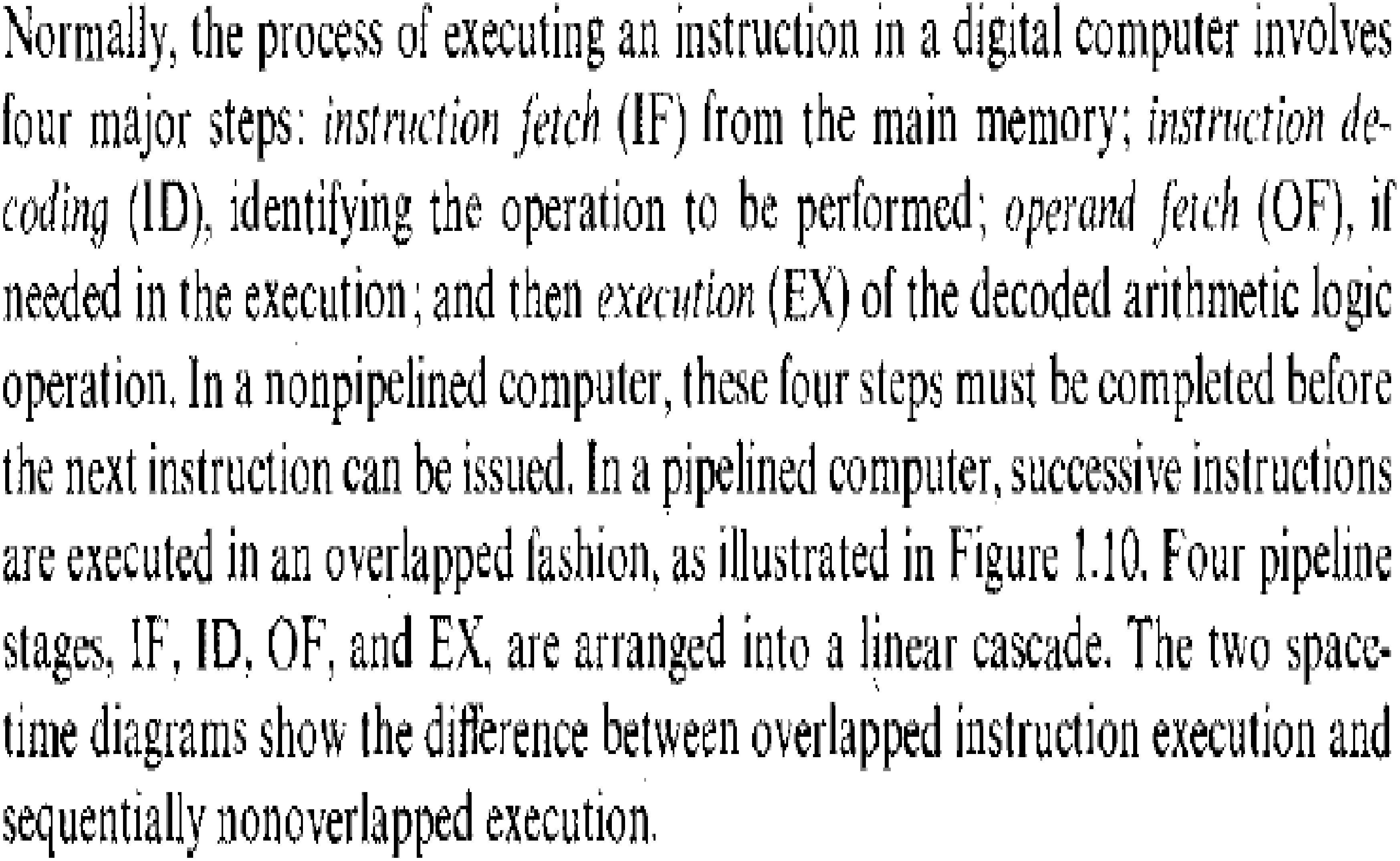
Parallel computer structures



Parallel computers are those systems that emphasize parallel processing. The basic architectural features of parallel computers are introduced below. We divide parallel computers into three architectural configurations:

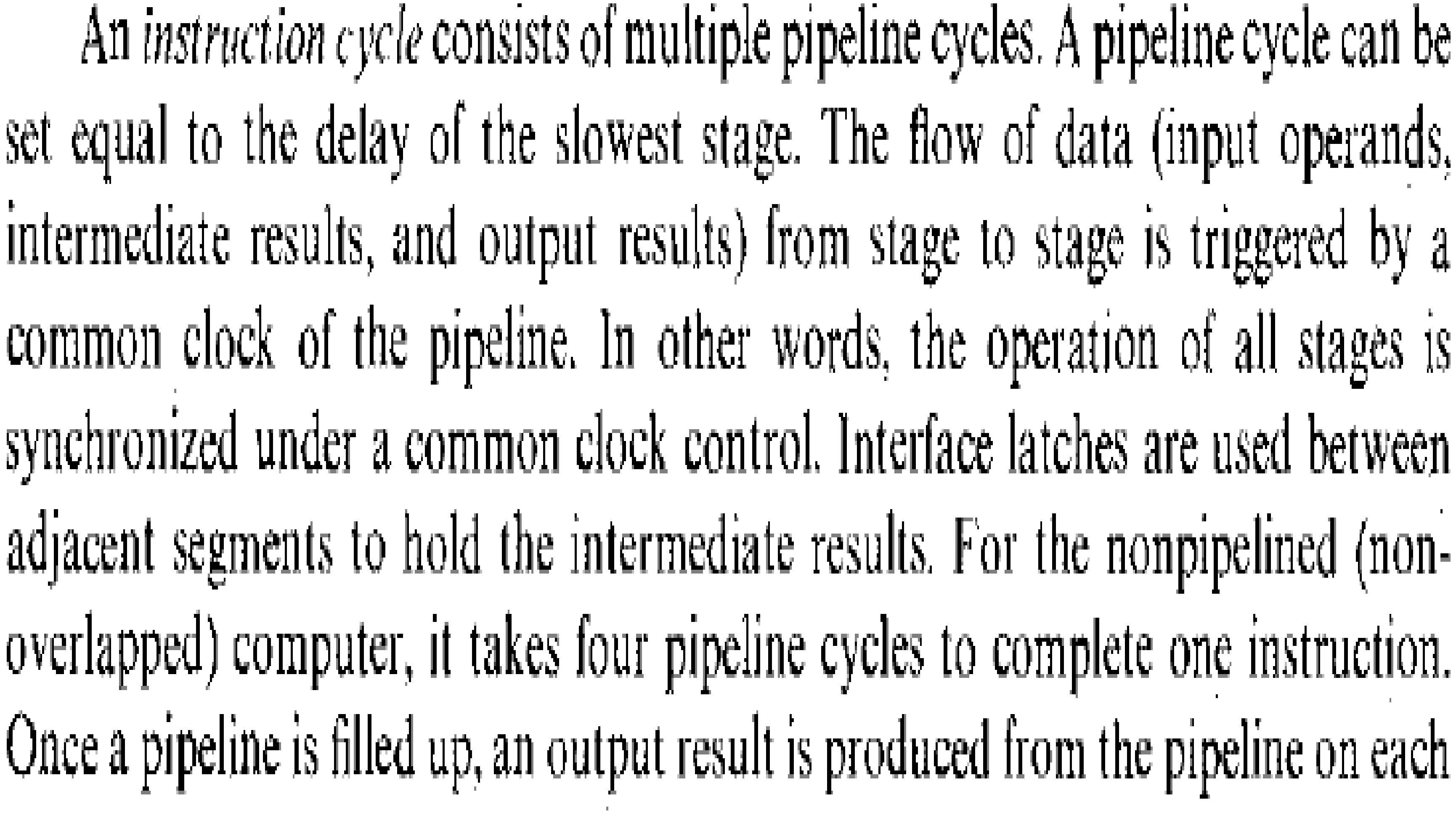
- Pipeline computers
- Array processors
- Multiprocessor systems

Pipeline Computers



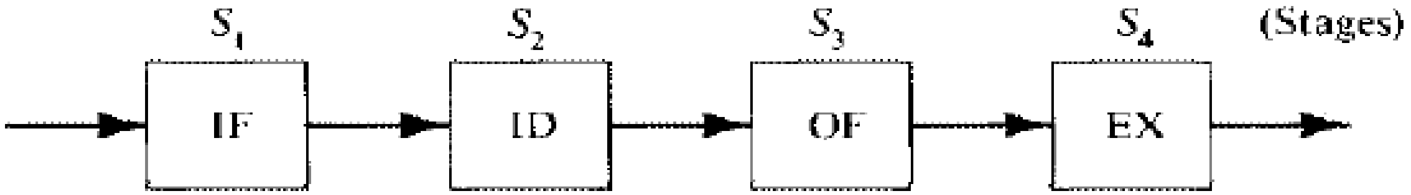
Normally, the process of executing an instruction in a digital computer involves four major steps: *instruction fetch* (IF) from the main memory; *instruction decoding* (ID), identifying the operation to be performed; *operand fetch* (OF), if needed in the execution; and then *execution* (EX) of the decoded arithmetic logic operation. In a nonpipelined computer, these four steps must be completed before the next instruction can be issued. In a pipelined computer, successive instructions are executed in an overlapped fashion, as illustrated in Figure 1.10. Four pipeline stages, IF, ID, OF, and EX, are arranged into a linear cascade. The two space-time diagrams show the difference between overlapped instruction execution and sequentially nonoverlapped execution.

s

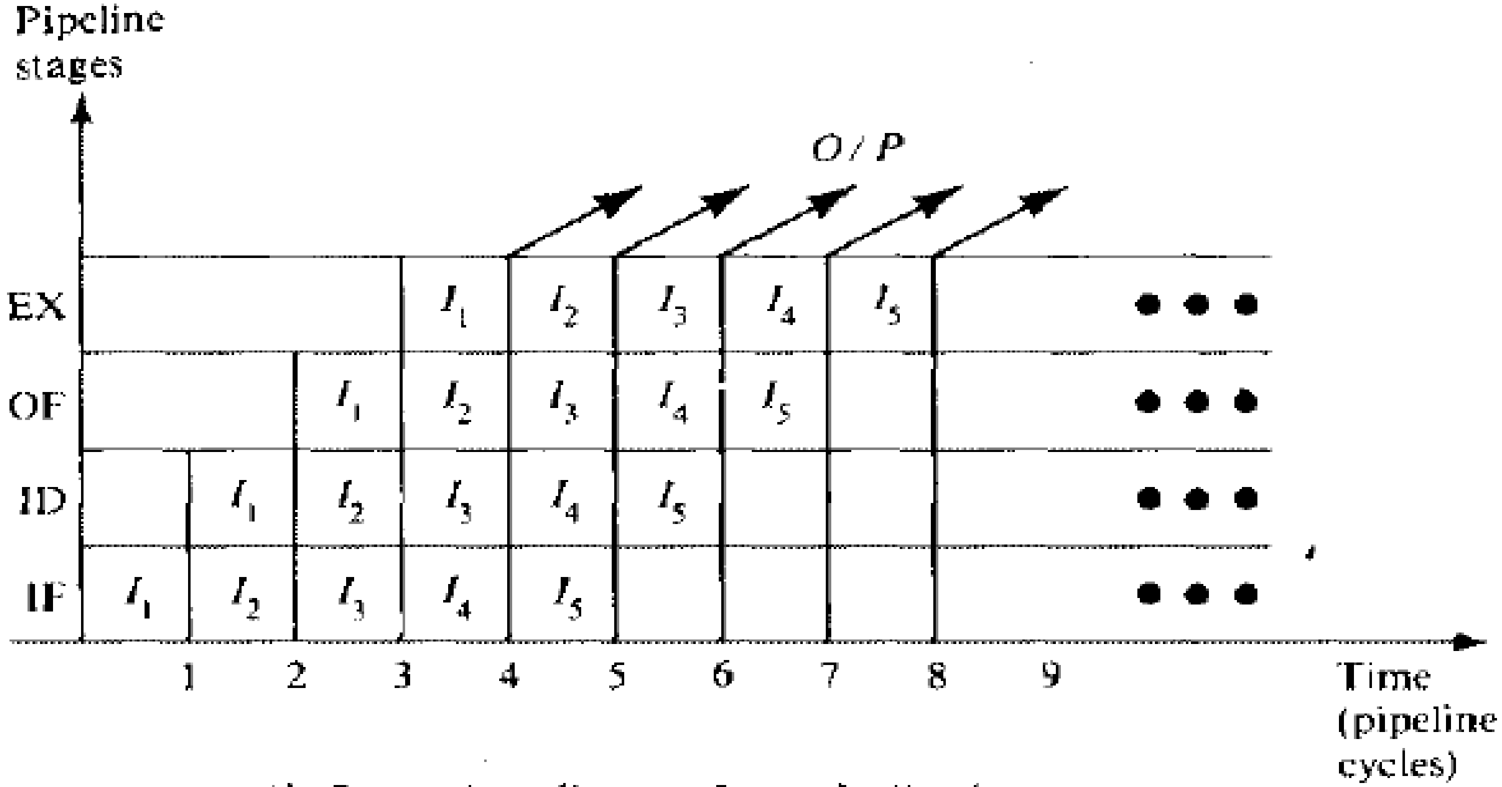


An *instruction cycle* consists of multiple pipeline cycles. A pipeline cycle can be set equal to the delay of the slowest stage. The flow of data (input operands, intermediate results, and output results) from stage to stage is triggered by a common clock of the pipeline. In other words, the operation of all stages is synchronized under a common clock control. Interface latches are used between adjacent segments to hold the intermediate results. For the nonpipelined (non-overlapped) 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

s



(*a*) A pipelined processor



EX				I_1	I_2	I_3	I_4	I_5	• • •
OF			I_1	I_2	I_3	I_4	I_5		• • •
ID		I_1	I_2	I_3	I_4	I_5			• • •
IF	I_1	I_2	I_3	I_4	I_5				• • •

(*b*) Space-time diagram for a pipelined processor

Array Computers

An *array processor* is a synchronous parallel computer with multiple arithmetic logic units, called *processing elements* (PE), that can operate in parallel in a lock-step fashion. By replication of ALUs, one can achieve the spatial parallelism. The PEs are synchronized to perform the same function at the same time. An appropriate data-routing mechanism must be established among the PEs. A typical

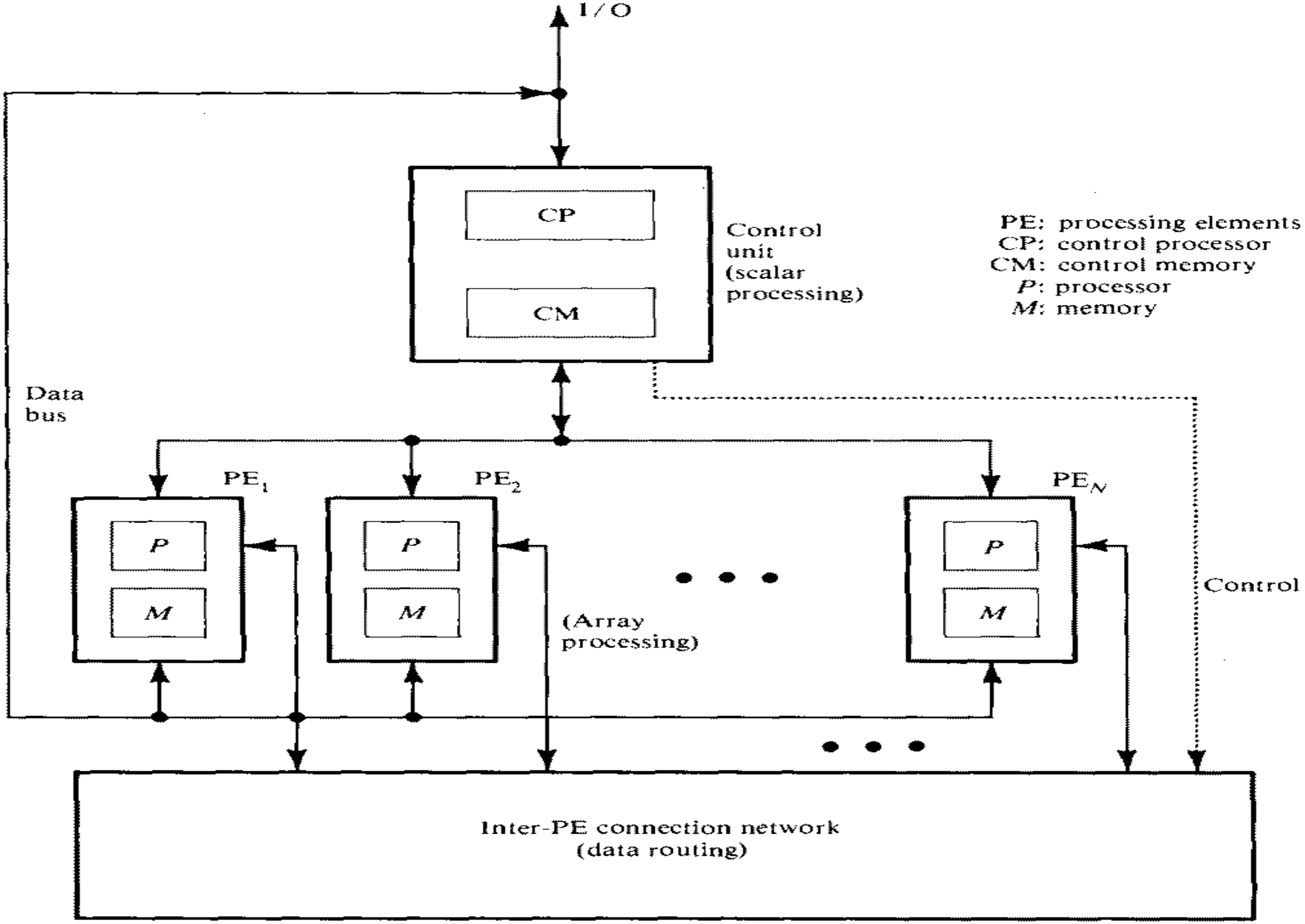
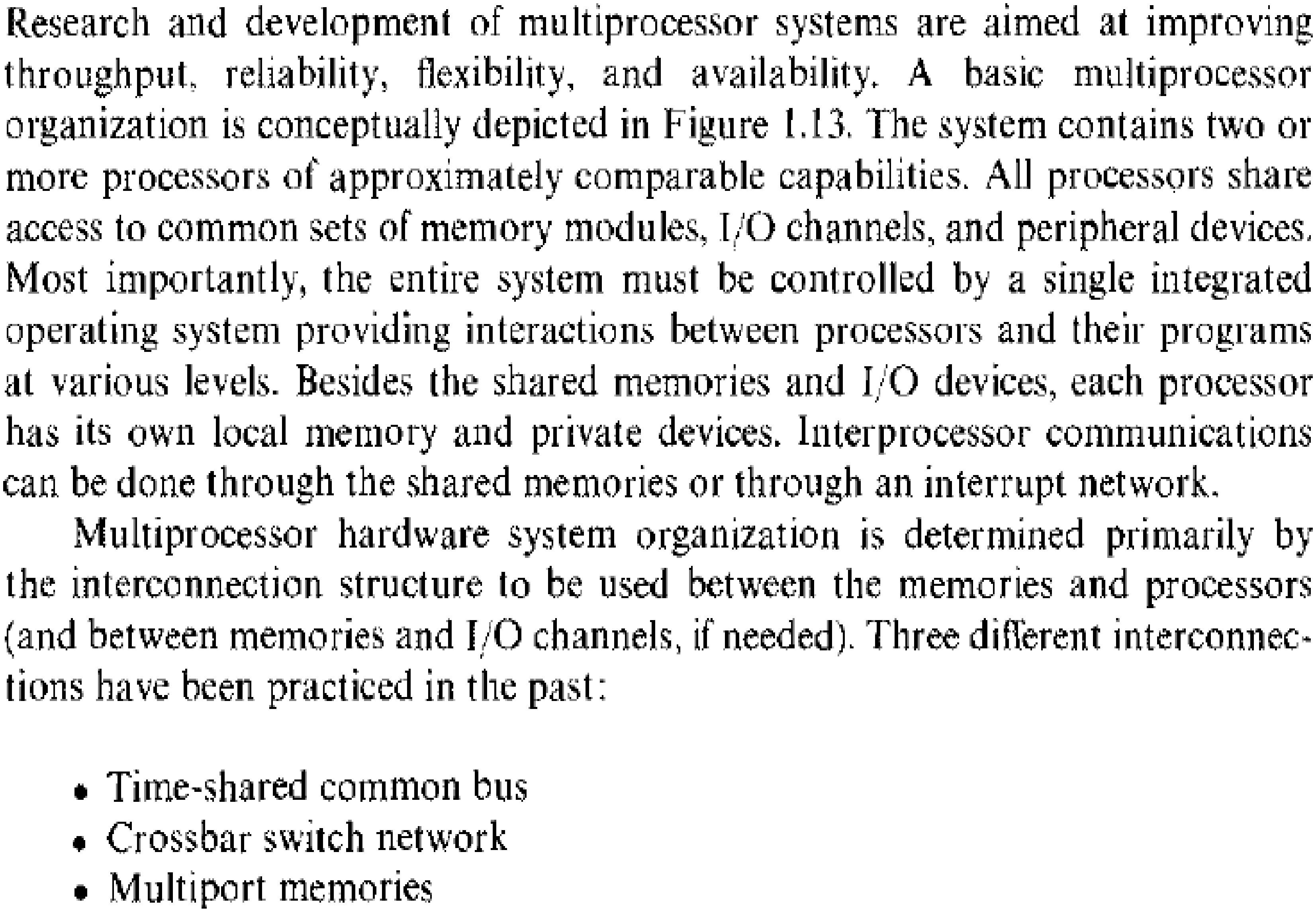


Figure 1.12 Functional structure of an SIMD array processor with concurrent scalar processing in the con;rol unit.

s

array processor is depicted in Figure 1.12. Scalar and control-type instructions are directly executed in the *control unit* (CU). Each PE consists of an ALU with registers and a local memory. The PEs are interconnected by a data-routing network. The interconnection pattern to be established for specific computation is under program control from the CU. Vector instructions are broadcast to the PEs for distributed execution over different component operands fetched directly from the local memories. Instruction fetch (from local memories or from the control memory) and decode is done by the control unit. The PEs are passive devices without instruction decoding capabilities.

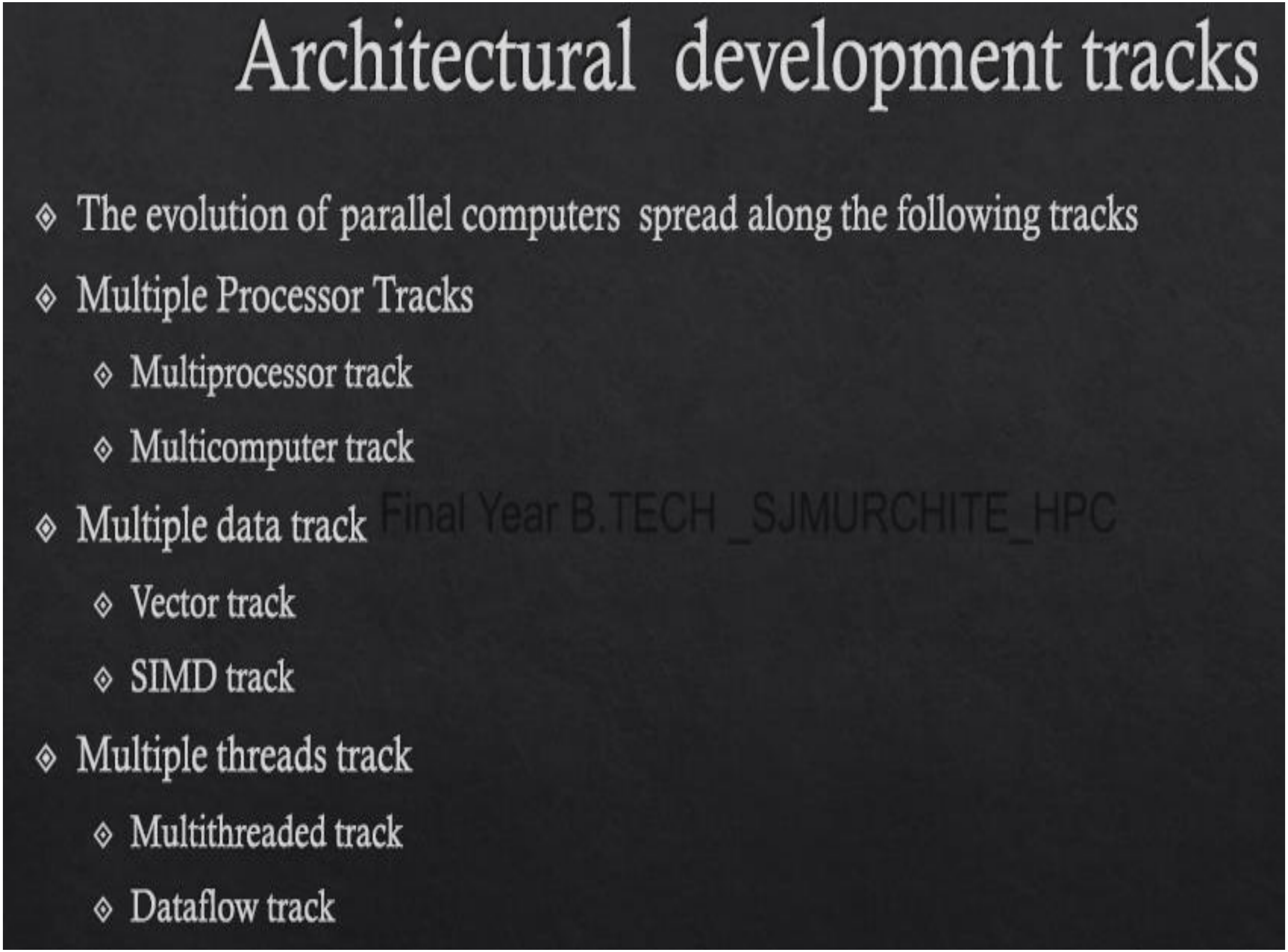
Multiprocessor system



Research and development of multiprocessor systems are aimed at improving throughput, reliability, flexibility, and availability. A basic multiprocessor organization is conceptually depicted in Figure 1.13. The system contains two or more processors of approximately comparable capabilities. All processors share access to common sets of memory modules, I/O channels, and peripheral devices. Most importantly, the entire system must be controlled by a single integrated operating system providing interactions between processors and their programs at various levels. Besides the shared memories and I/O devices, each processor has its own local memory and private devices. Interprocessor communications can be done through the shared memories or through an interrupt network.

Multiprocessor hardware system organization is determined primarily by the interconnection structure to be used between the memories and processors (and between memories and I/O channels, if needed). Three different interconnections have been practiced in the past:

- Time-shared common bus
- Crossbar switch network
- Multiport memories



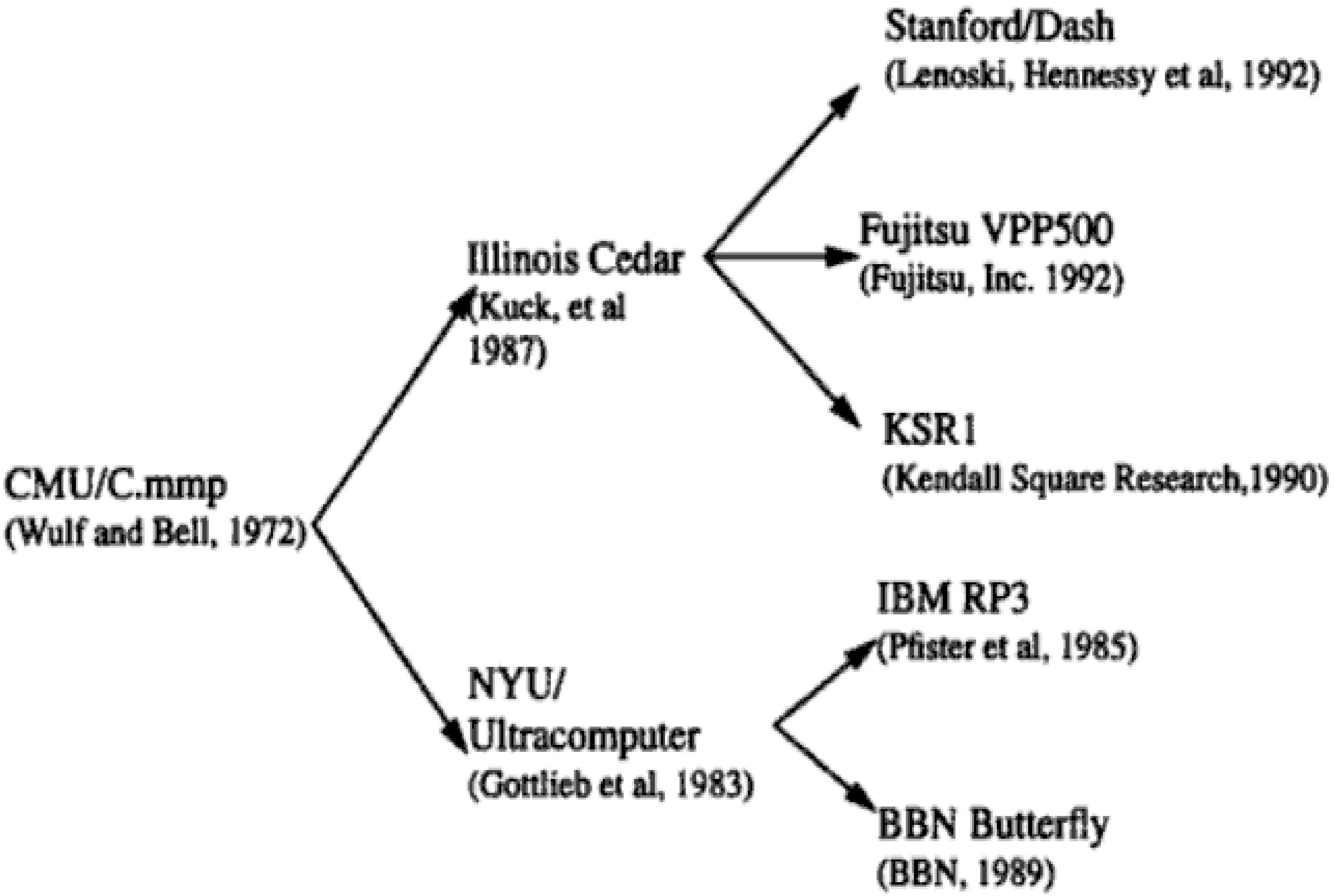
Architectural development tracks

- The evolution of parallel computers spread along the following tracks
- Multiple Processor Tracks
  - Multiprocessor track
  - Multicomputer track
- Multiple data track
  - Vector track
  - SIMD track
- Multiple threads track
  - Multithreaded track
  - Dataflow track

- Provides insights for a new architectural development.
- Distinguished by similarity in computational models and technological bases.
- **Multiprocessor track**
- multiple processor system can be shared memory or distributed memory.
- **Shared memory track**
- It shows track of multiprocessor development employing a single address space in the entire system
- The track started with C.mmp system developed at mellon university.
- c. Mmp was a UMA multiprocessor.
- The c.mmp project pioneered shared memory multiprocessor development not only in the cross architecture but in multiprocessor operating system development.

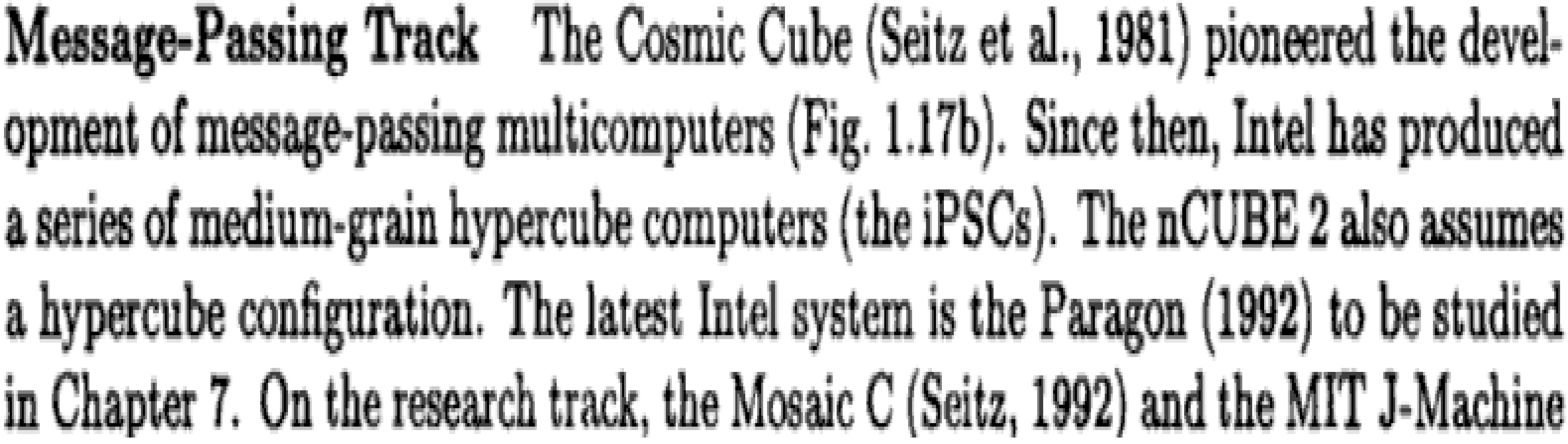
s

- Illinois Cedar project and NYO ultra computer project both were developed with a single address space.
- Both uses multi stage network as system inter connect.
- Achievement in cedar project are used in parallel compilers and performance benchmarking experiments.
- Stanford Dash is a NUMA multiprocessor with distributed memory forming a global address space. cache coherence is enforced with distributed directories.

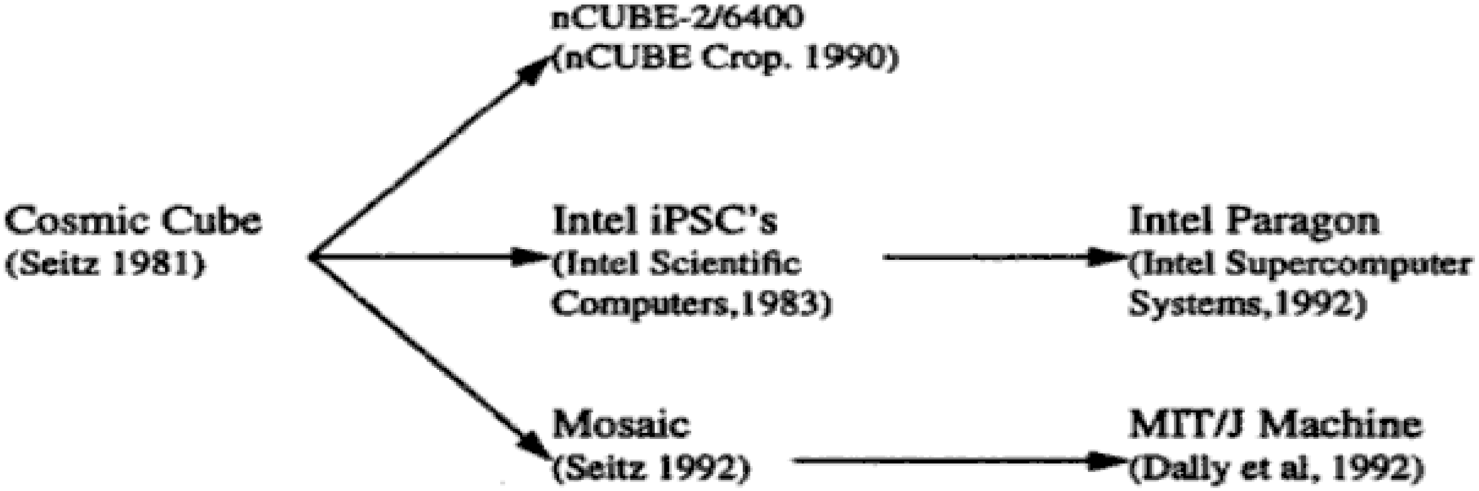


(a) Shared-memory track

Message passing track

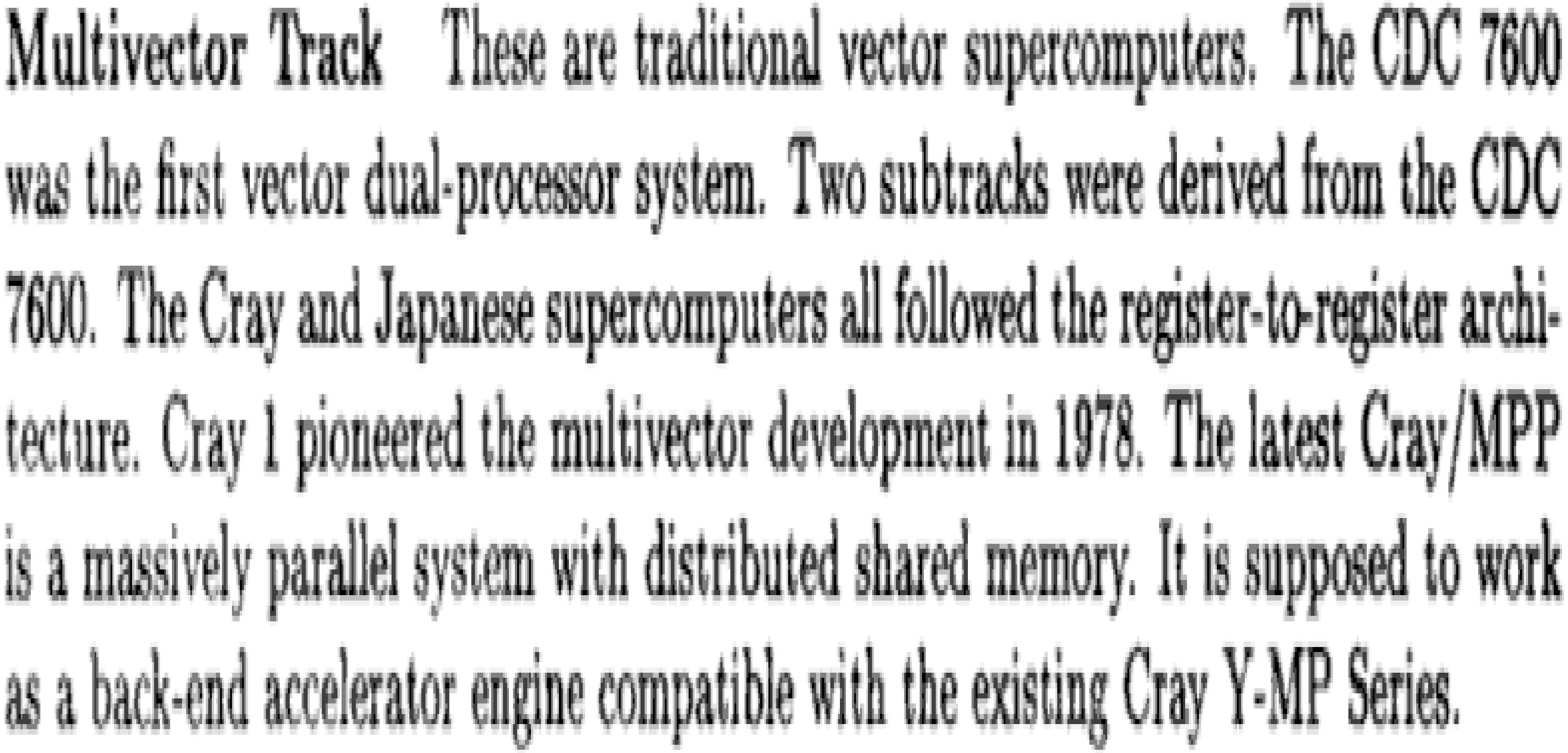


Message-Passing Track The Cosmic Cube (Seitz et al., 1981) pioneered the development of message-passing multicomputers (Fig. 1.17b). Since then, Intel has produced a series of medium-grain hypercube computers (the iPSCs). The nCUBE 2 also assumes a hypercube configuration. The latest Intel system is the Paragon (1992) to be studied in Chapter 7. On the research track, the Mosaic C (Seitz, 1992) and the MIT J-Machine



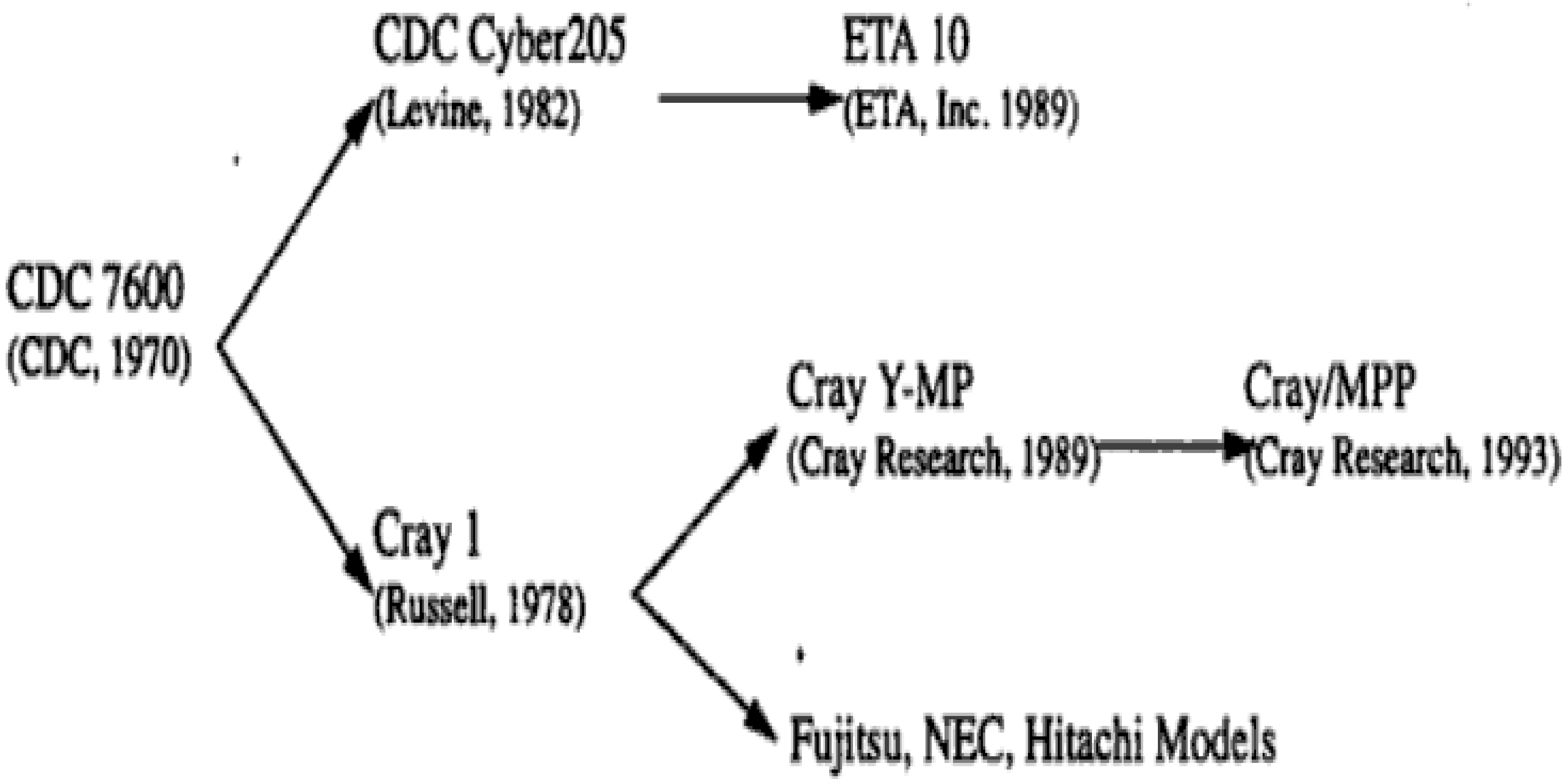
(b) Message-passing track

Multivector and SIMD Tracks



Multivector Track These are traditional vector supercomputers. The CDC 7600 was the first vector dual-processor system. Two subtracks were derived from the CDC 7600. The Cray and Japanese supercomputers all followed the register-to-register architecture. Cray 1 pioneered the multivector development in 1978. The latest Cray/MPP is a massively parallel system with distributed shared memory. It is supposed to work as a back-end accelerator engine compatible with the existing Cray Y-MP Series.

s



CDC 7600
(CDC, 1970)

CDC Cyber205
(Levine, 1982)

ETA 10
(ETA, Inc. 1989)

Cray 1
(Russell, 1978)

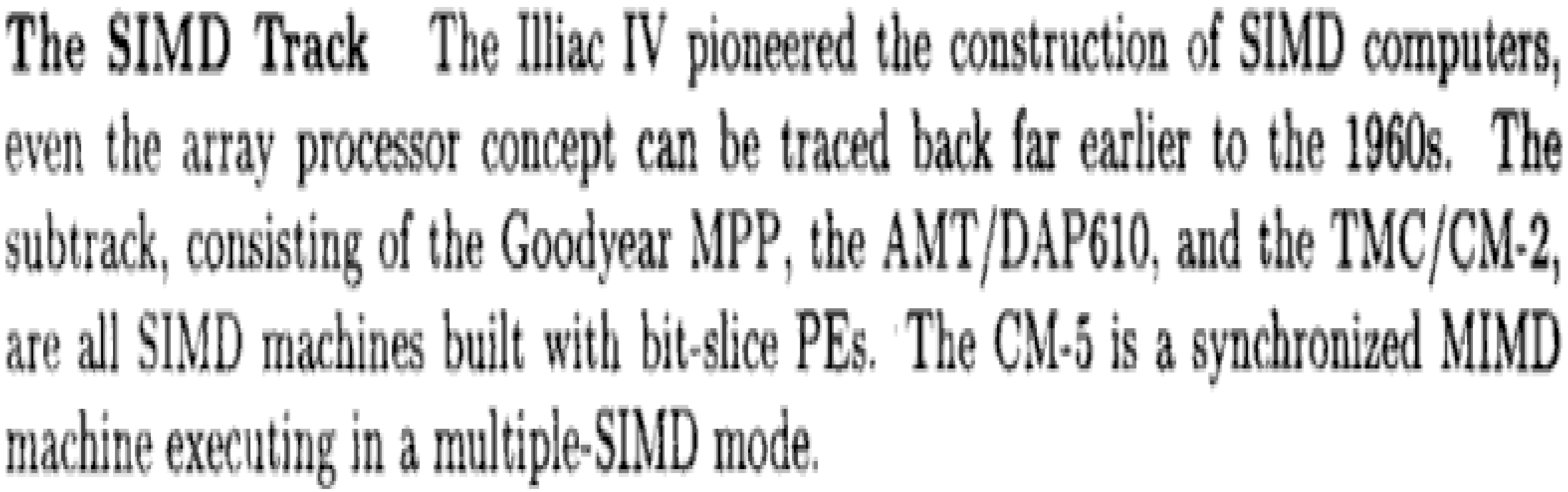
Cray Y-MP
(Cray Research, 1989)

Cray/MPP
(Cray Research, 1993)

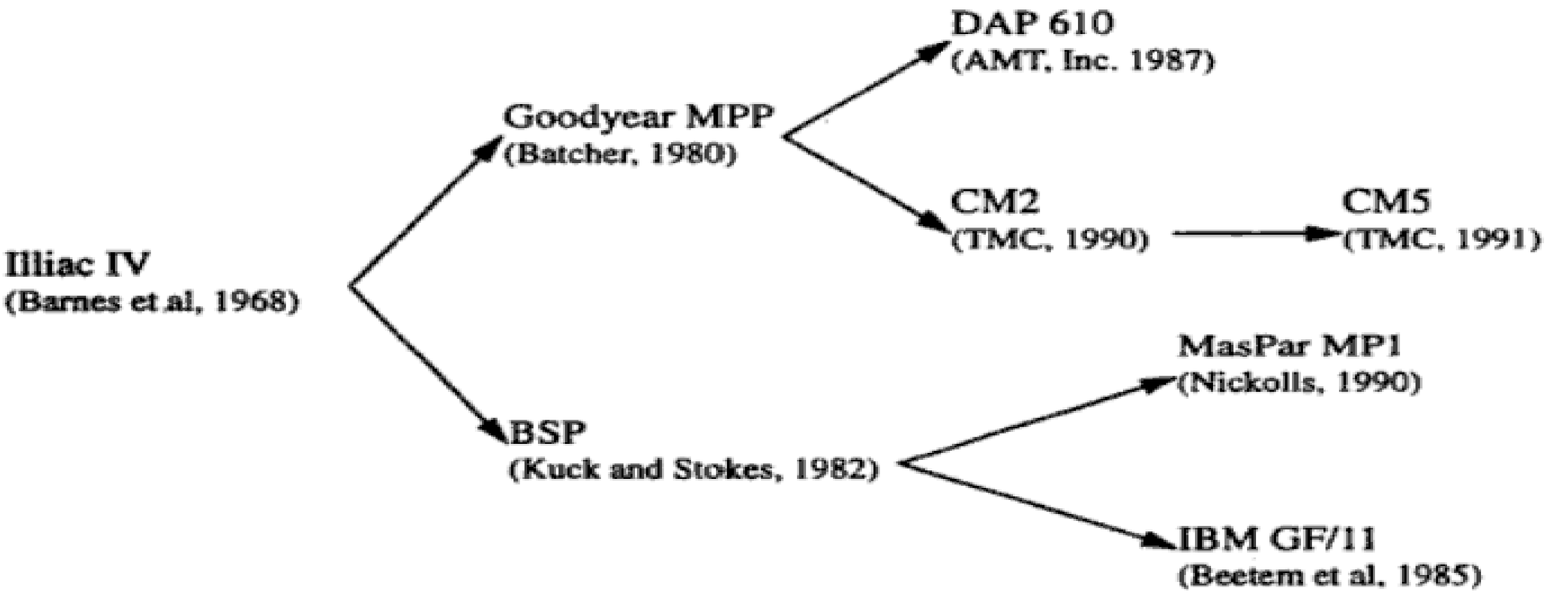
Fujitsu, NEC, Hitachi Models

(a) Multivector track

SIMD Track

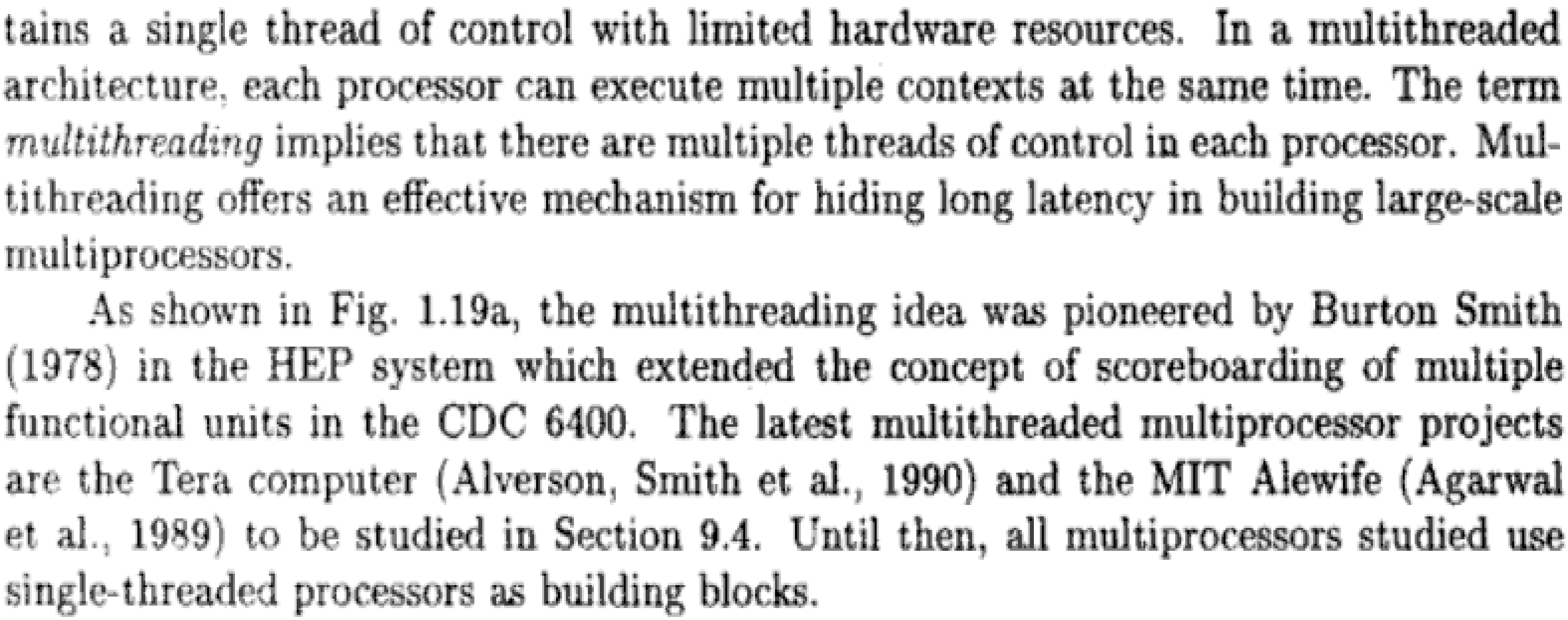


The SIMD Track The Illiac IV pioneered the construction of SIMD computers, even the array processor concept can be traced back far earlier to the 1960s. The subtrack, consisting of the Goodyear MPP, the AMT/DAP610, and the TMC/CM-2, are all SIMD machines built with bit-slice PEs. The CM-5 is a synchronized MIMD machine executing in a multiple-SIMD mode.



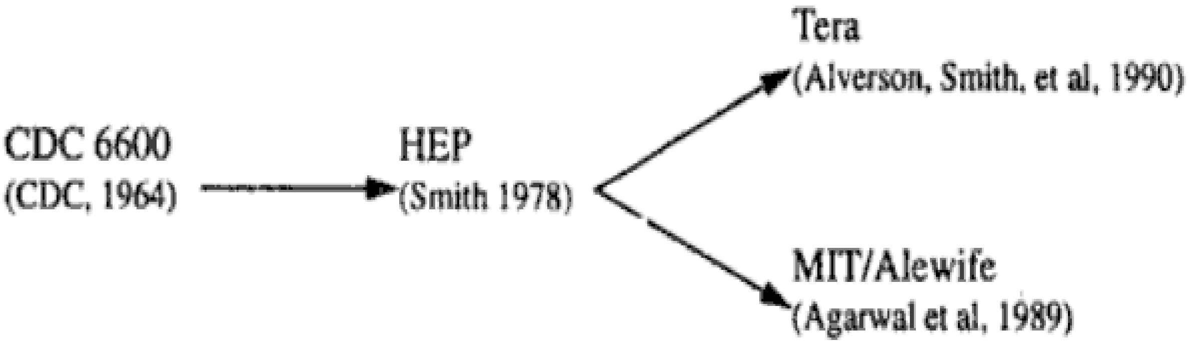
(b) SIMD track

Multithreaded and Dataflow track



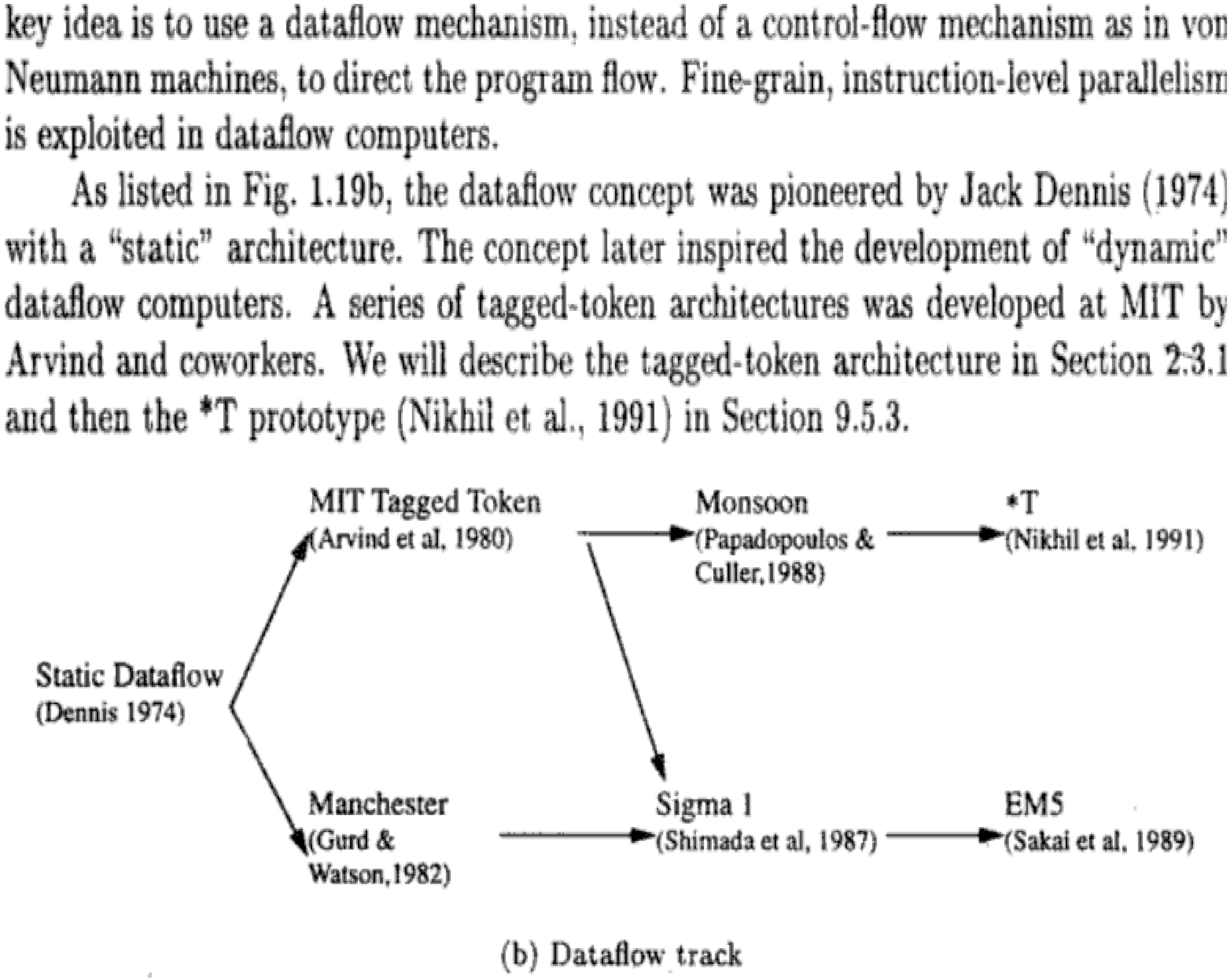
tains a single thread of control with limited hardware resources. In a multithreaded architecture, each processor can execute multiple contexts at the same time. The term *multithreading* implies that there are multiple threads of control in each processor. Multithreading offers an effective mechanism for hiding long latency in building large-scale multiprocessors.

As shown in Fig. 1.19a, the multithreading idea was pioneered by Burton Smith (1978) in the HEP system which extended the concept of scoreboarding of multiple functional units in the CDC 6400. The latest multithreaded multiprocessor projects are the Tera computer (Alverson, Smith et al., 1990) and the MIT Alewife (Agarwal et al., 1989) to be studied in Section 9.4. Until then, all multiprocessors studied use single-threaded processors as building blocks.



(a) Multithreaded track

Data flow track



key idea is to use a dataflow mechanism, instead of a control-flow mechanism as in von Neumann machines, to direct the program flow. Fine-grain, instruction-level parallelism is exploited in dataflow computers.

As listed in Fig. 1.19b, the dataflow concept was pioneered by Jack Dennis (1974) with a "static" architecture. The concept later inspired the development of "dynamic" dataflow computers. A series of tagged-token architectures was developed at MIT by Arvind and coworkers. We will describe the tagged-token architecture in Section 2.3.1 and then the *T prototype (Nikhil et al., 1991) in Section 9.5.3.

(b) Dataflow track

Tutorial 1

1. Write short note on

    a) Pipeline computer

    b) Array processor

    c) Multiprocessor

2. Explain Flynn’s classification scheme

3. Explain Trends Towards parallel processing