EAST WEST UNIVERSITY

excellence in education

Computer Architecture

Course Code: CSE360

Lecture 14

Reduced Instruction Set Computers

The Next Step - RISC

- Reduced Instruction Set Computer (RISC)

- Key features
  - Large number of general purpose registers
  - or use of compiler technology to optimize register use
  - Limited and simple instruction set
  - Emphasis on optimising the instruction pipeline

Driving force for CISC

Complex Instruction Set Computer

- Software costs far exceed hardware costs
- Increasingly complex high level languages
- Semantic gap (the difference between operations provided in HLLs and those provided in computer architecture)
- Leads to:
  - Large instruction sets
  - More addressing modes
  - Hardware implementations of HLL (high-level languages) statements

Intention of CISC

- Ease the task of compiler writing
- Improve execution efficiency
  - Complex sequences of operations can be implemented in microcode
- Provide support for more complex HLLs

RISC Characteristics

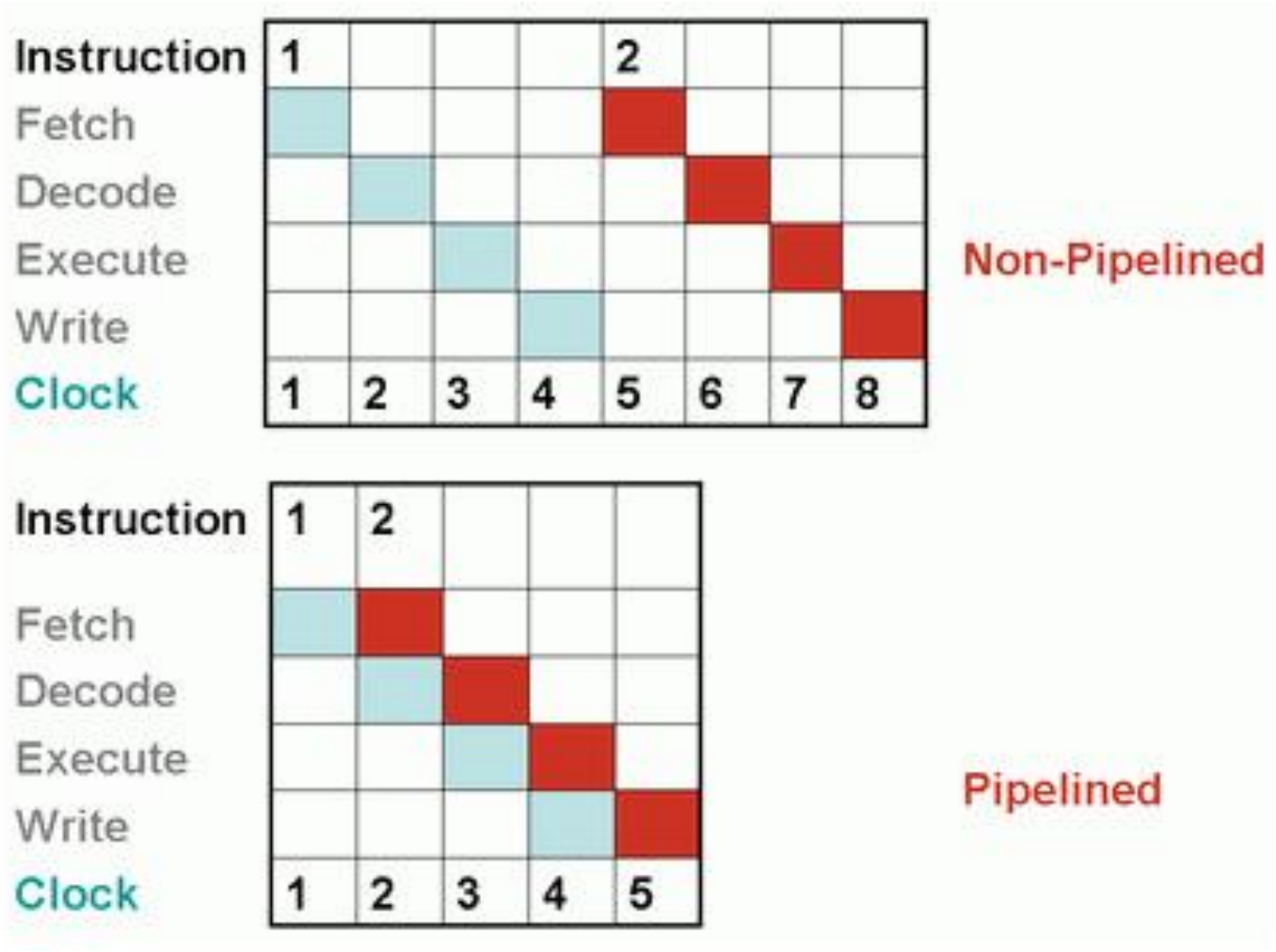
- One instruction per cycle
- Register to register operations
- Few, simple addressing modes
- Few, simple instruction formats
- Hardwired design (no microcode)
- Fixed instruction format
- More compile time/effort

RISC Pipelining

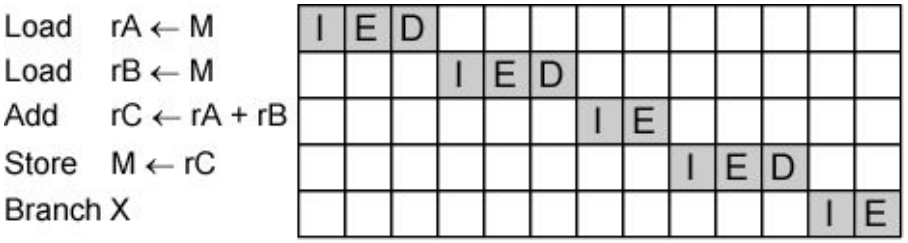
- Most instructions are register to register
- Two phases of execution
  - I: Instruction fetch
  - E: Execute
    - ALU operation with register input and output
- For load and store
  - I: Instruction fetch
  - E: Execute
    - Calculate memory address
  - D: Memory
    - Register to memory or memory to register operation

NOOP (No operation)

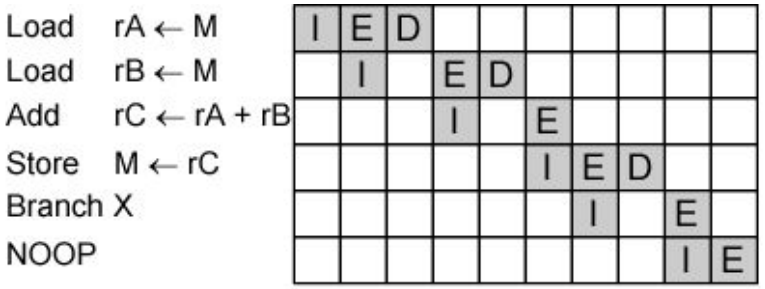
RISC Pipelining



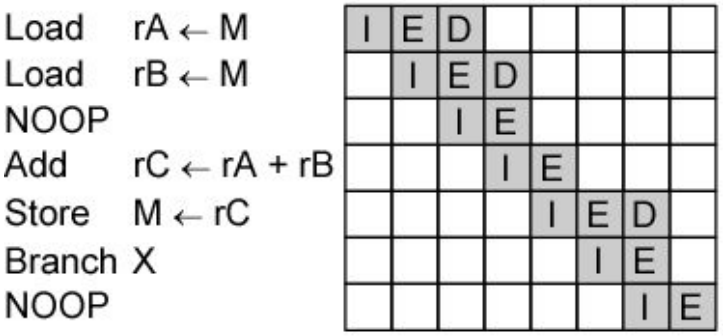
Effects of Pipelining



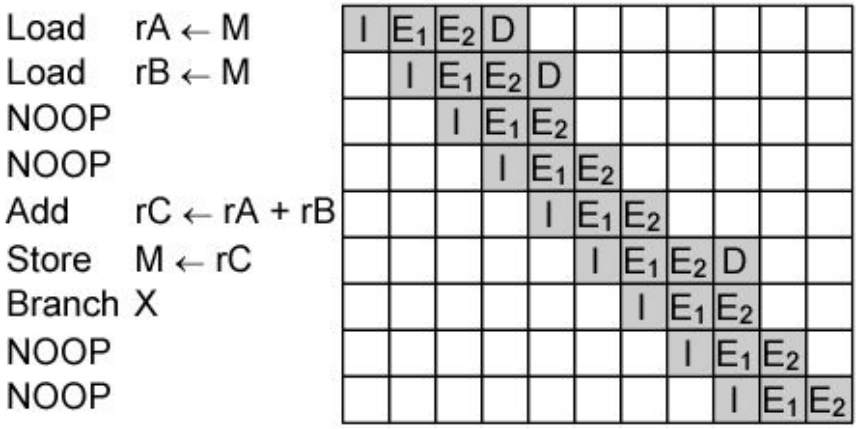
(a) Sequential execution



(b) Two-stage pipelined timing



(c) Three-stage pipelined timing



(d) Four-stage pipelined timing

Superscalar Processors

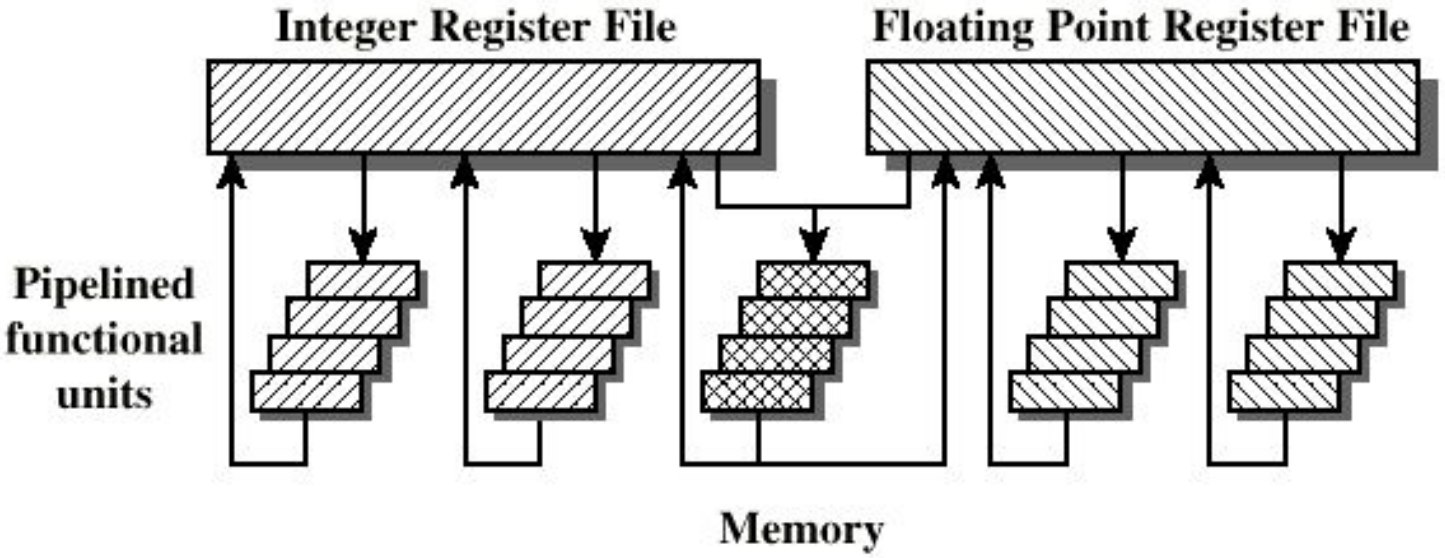
What is Superscalar?

- A superscalar processor is one in which multiple independent instruction pipelines are used.
- Each pipeline consists of multiple stages, so that each pipeline can handle multiple instructions at time.
- Multiple pipelines introduce a new level of parallelism, enabling multiple instructions to be processed at a time
- Common instructions (arithmetic, load/store, conditional branch) can be initiated and executed independently
- Equally applicable to RISC & CISC
- In practice usually RISC

Why Superscalar?

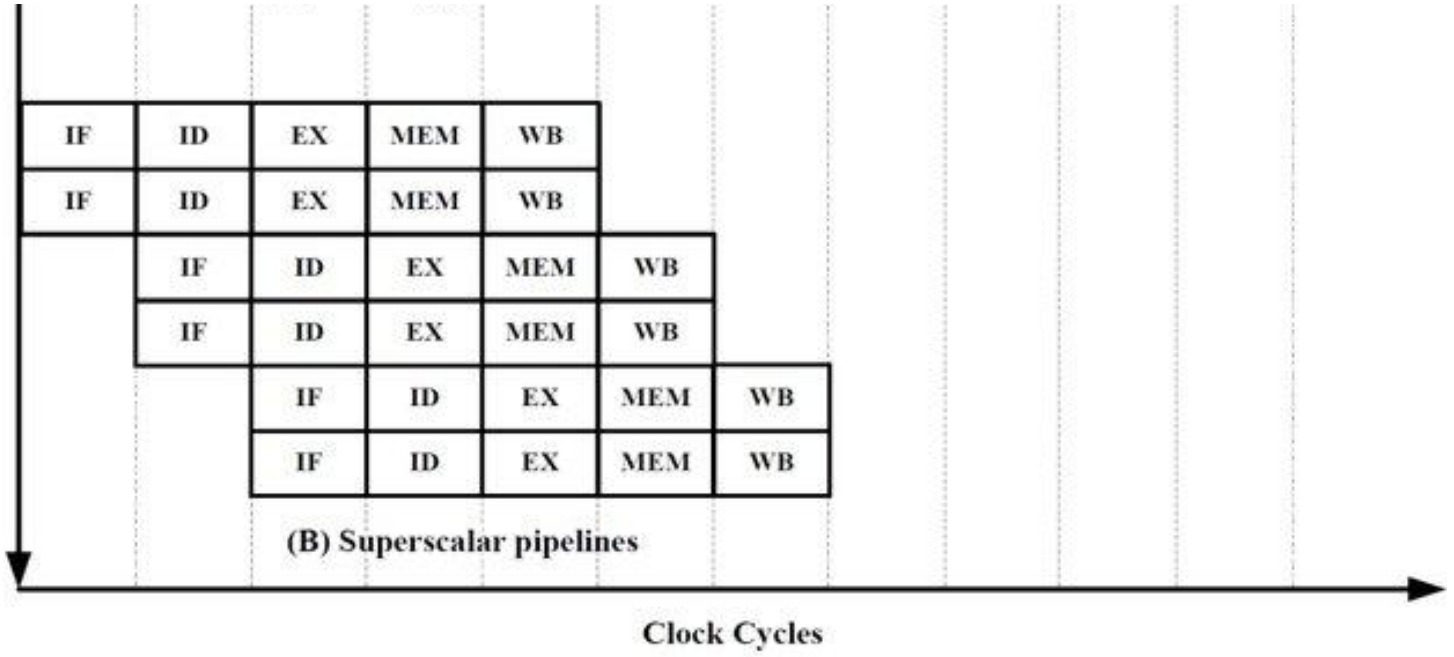
- The term superscalar refers to a machine that is designed to improve the performance of the execution of scalar instruction
- Most operations are on scalar quantities (see RISC notes)

General Superscalar Organization



- Figure shows superscalar organization
- There are multiple functional units, each of which is implemented as a pipeline, which support parallel execution of several instructions
- e.g. two integer, two floating point, and one memory (either load or store) operations can be executing at the same time

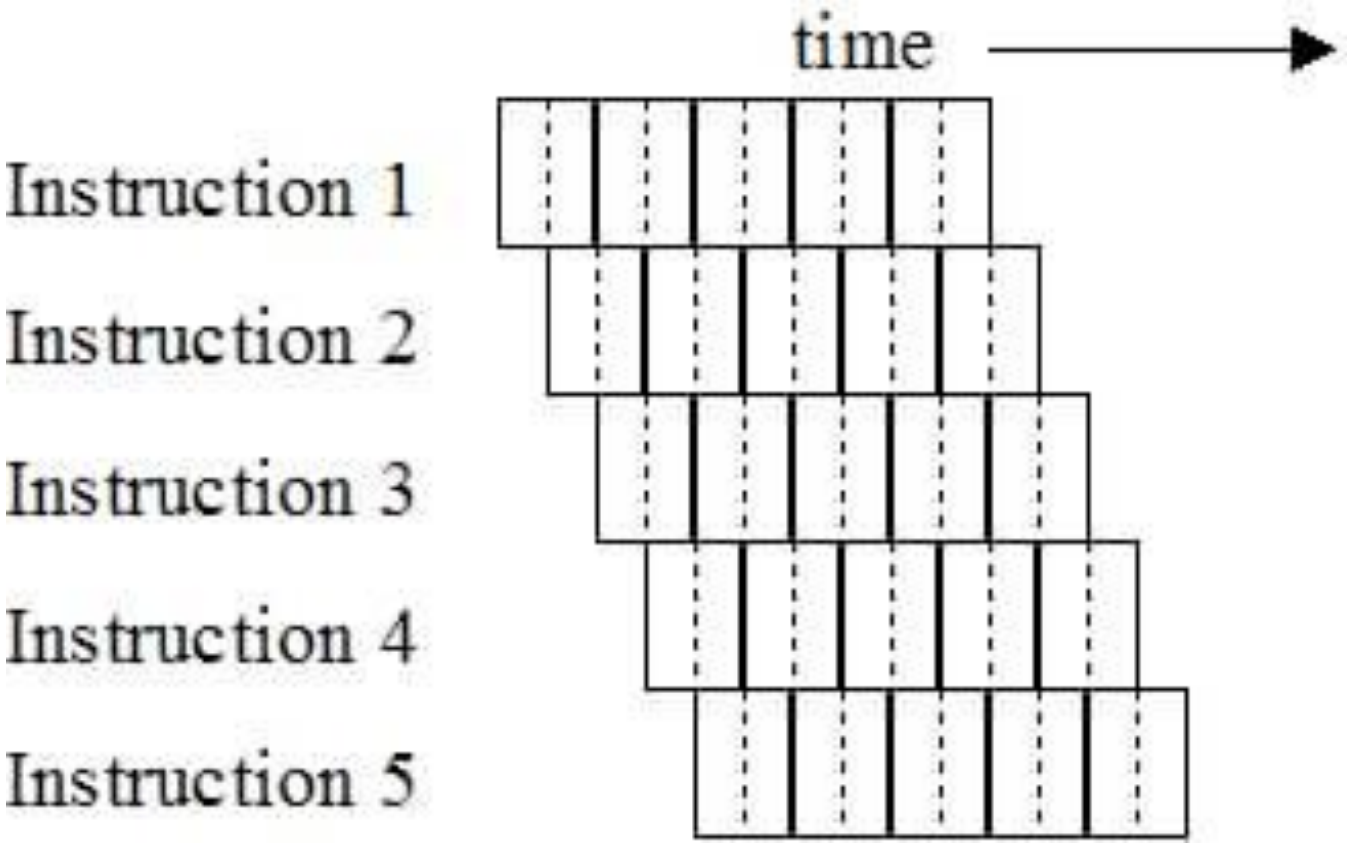
General Superscalar Organization



Superpipelined

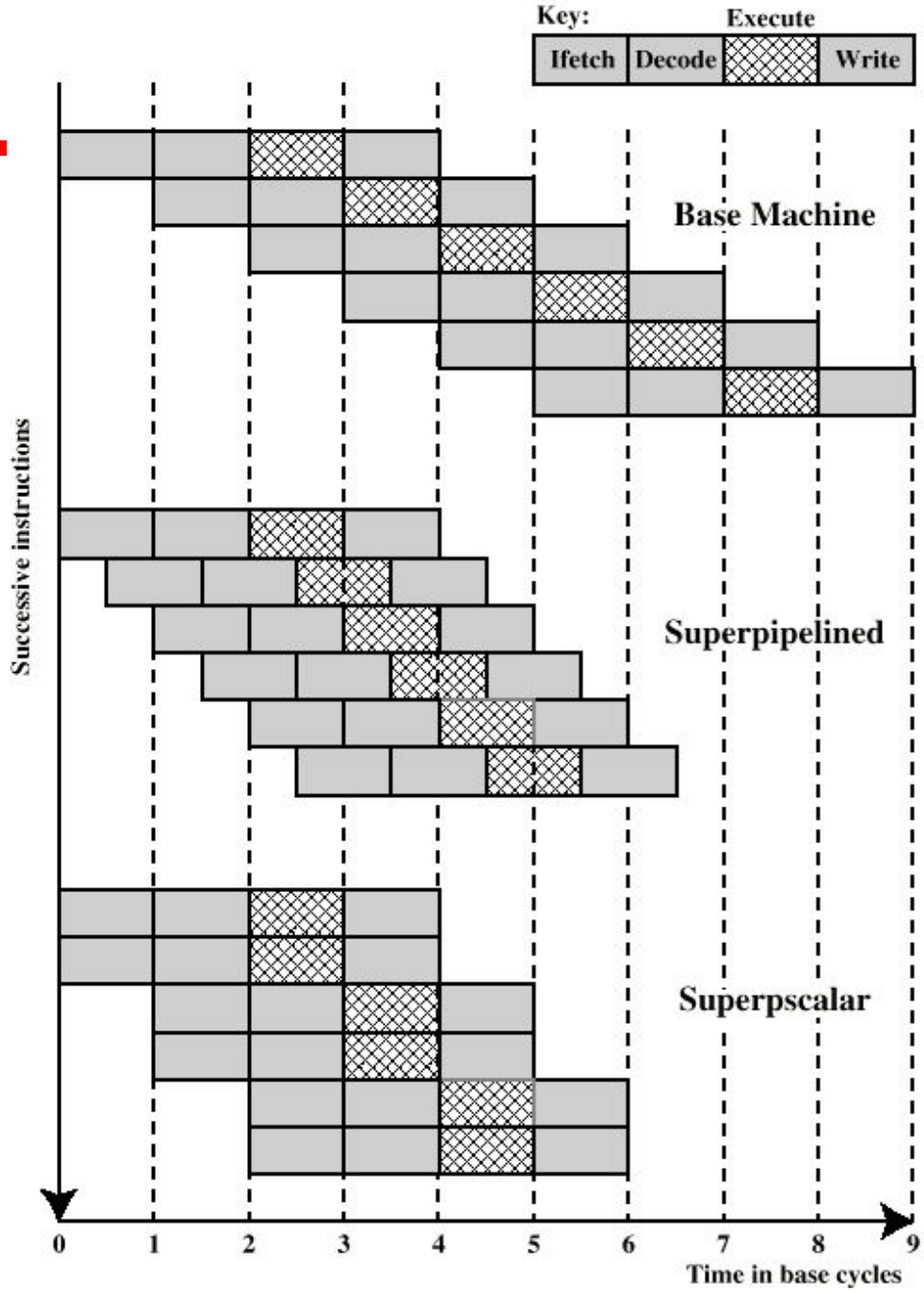
- Superpipelining exploits the fact that many pipeline stages performs tasks that require less than half a clock cycle
- Thus, double internal clock speed gets two tasks per external clock cycle

Superpipelined



Superscalar v Superpipeline

- The pipeline has four stages: instruction fetch, operation decode, operation execution, and result wire back
- In 4-stage pipeline, several instructions are executing concurrently, only one instruction is in its execution state at any one time
- Superpipelined implementation capable of performing two pipeline stages per clock cycle (the function performed in each stage can be split into two non-overlapping parts and each can execute in half a clock cycle)
- Superscalar implementation capable of executing two instances of each stage in parallel.
- The superlined processor falls behind the superscalar processor



Parallel Processing

Multiple Processor Organization

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

Single Instruction, Single Data Stream - SISD

- A single processor
- Executes single instruction stream
- Operate data stored in single memory
- Uni-processor falls into this category

Single Instruction, Multiple Data Stream - SIMD

- Single machine instruction
- Controls simultaneous execution
- Number of processing elements
- Lockstep basis
- Each processing element has associated data memory
- **Each instruction executed on different set of data by different processors**
- Vector and array processors

Multiple Instruction, Single Data Stream - MISD

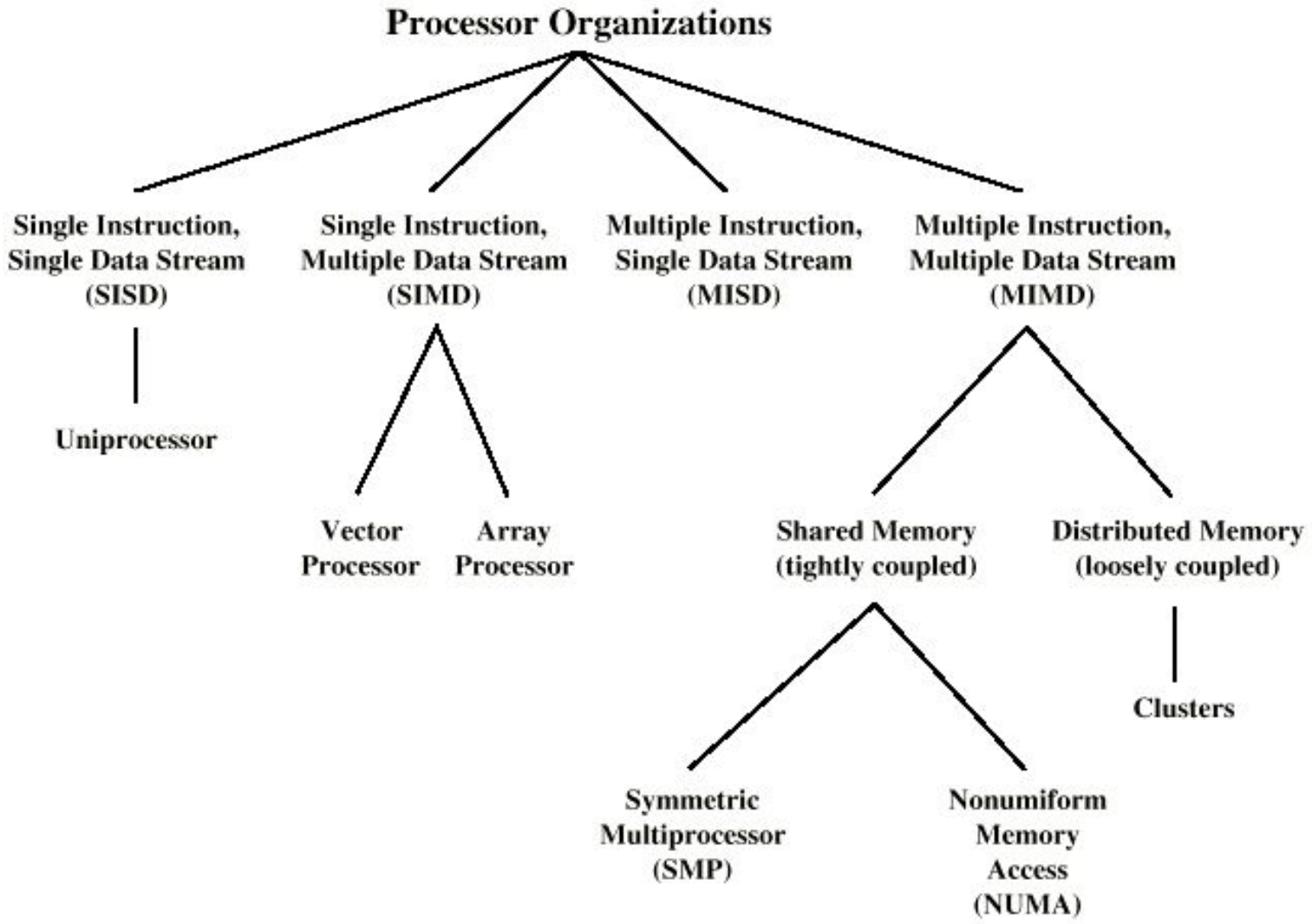
- **Sequence of data**
- Transmitted to set of processors
- Each processor executes **different instruction sequence**
- Never been implemented

Multiple Instruction, Multiple Data Stream- MIMD

- Set of processors
- Simultaneously execute different instruction sequences
- Different sets of data
- SMPs (symmetric multiprocessors), clusters, and NUMA (non-uniform memory access) systems

- MIMD can be subdivided by means in which the processors communicate

Taxonomy of Parallel Processor Architectures



MIMD - Overview

- General purpose processors
- Each can process all instructions necessary
- MIMD can be subdivided by means in which the processors communicate

Tightly Coupled - SMP

- Processors share memory
- Communicate via that shared memory
- Symmetric Multiprocessor (SMP)
  - Share single memory or pool
  - Shared bus to access memory
  - Memory access time to given area of memory is approximately the same for each processor

Tightly Coupled - NUMA

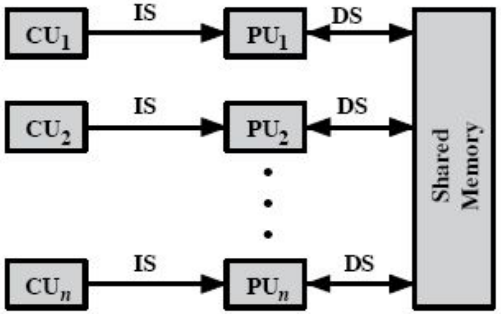
- Nonuniform memory access
- Access times to different regions of memroy may differ

Loosely Coupled - Clusters

- Collection of independent uniprocessors or SMPs
- Interconnected to form a cluster
- Communication via fixed path or network connections

Parallel Organizations

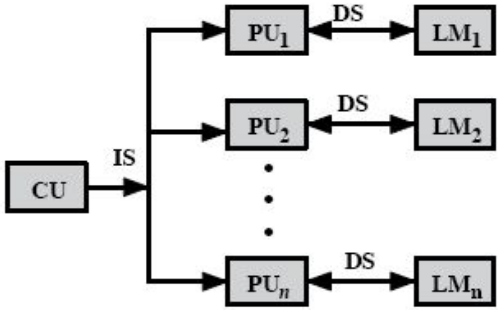
(a) SISD



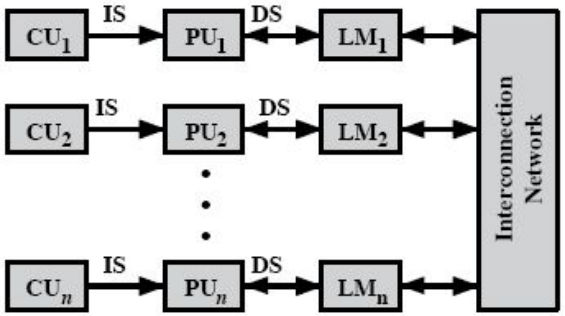
(c) MIMD (with shared memory)

CU = control unit
IS = instruction stream
PU = processing unit
DS = data stream
MU = memory unit
LM = local memory

SISD = single instruction, single data stream
SIMD = single instruction, multiple data stream
MIMD = multiple instruction, multiple data stream



(b) SIMD (with distributed memory)



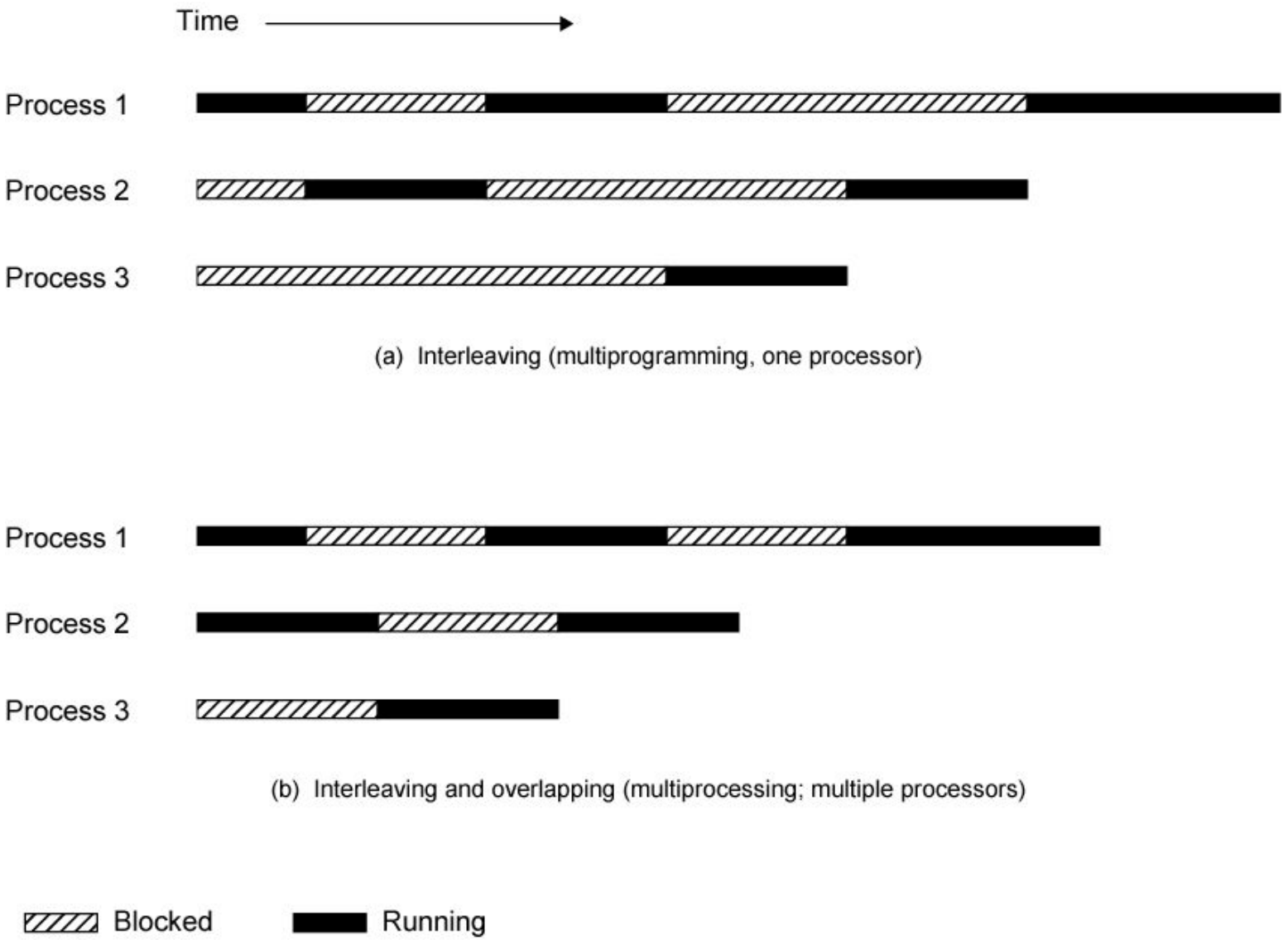
(d) MIMD (with distributed memory)

- (b) With an SIMD, there is still a single control unit, feeding a single instruction unit to multiple PUs (for multiple data)
- (b) Each PU may have its own dedicated memory or there may be a shared memory
- Finally, with the MIMD, there are multiple control units, each feeding a separate instruction stream to its own PU
- The MIMD may be shared-memory multiprocessor (c) or a distributed-memory multicomputer (d)

Symmetric Multiprocessors

- A stand alone computer with the following characteristics
  - Two or more similar processors of comparable capacity
  - Processors share same memory and I/O
  - Processors are connected by a bus or other internal connection
  - Memory access time is approximately the same for each processor
  - All processors share access to I/O
    - Either through same channels or different channels giving paths to same devices
  - All processors can perform the same functions (hence symmetric)
  - System controlled by integrated operating system
    - providing interaction between processors
    - Interaction at job, task, file and data element levels

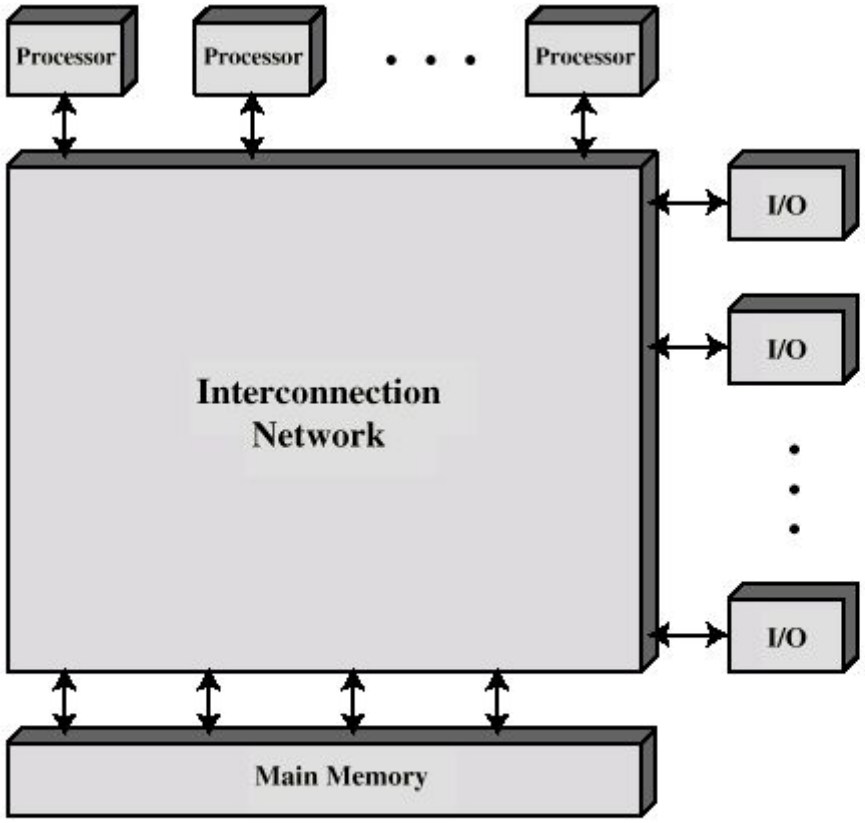
Multiprogramming and Multiprocessing



SMP Advantages

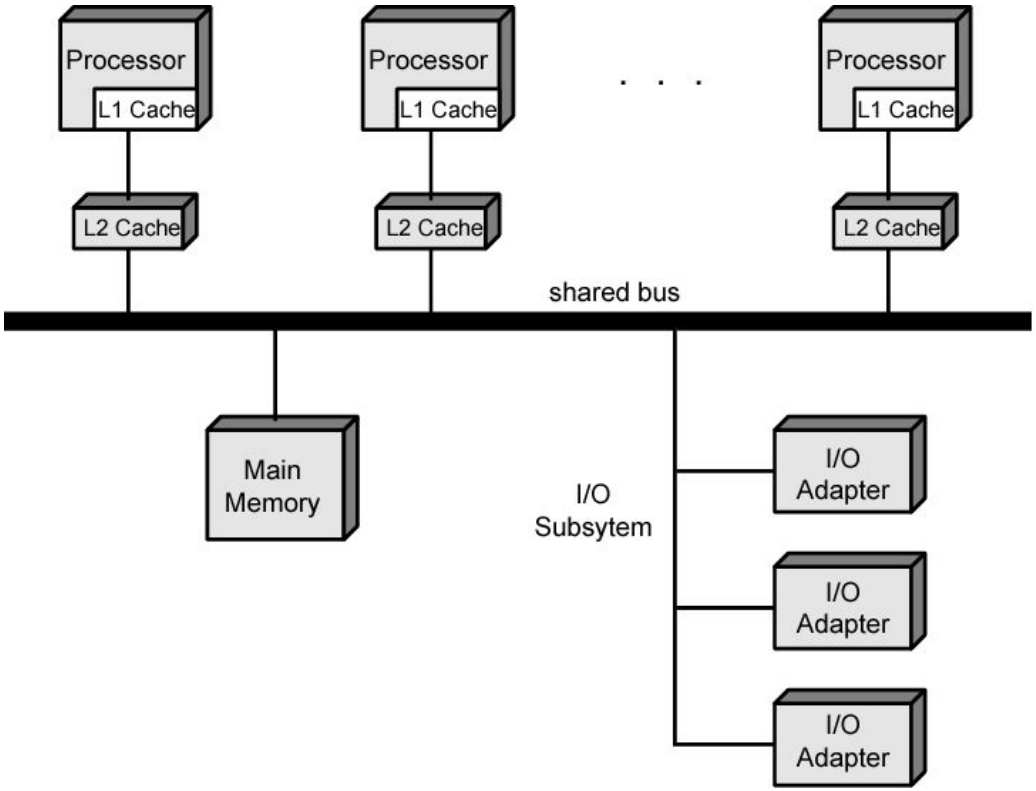
- **Performance**
  - If some work can be done in parallel, then a system with multiple processors yield greater performance than one with a single processor of the same type (see figure)
- **Availability**
  - Since all processors can perform the same functions, failure of a single processor does not halt the system
- **Incremental growth**
  - User can enhance performance by adding additional processors
- **Scaling**
  - Vendors can offer range of products based on number of processors

Block Diagram of Tightly Coupled Multiprocessor



- There are two or more processors
- Each processor is self-contained, including a control unit, ALU, registers, and one or more level of cache
- Each processor has access to a shared main memory and I/O devices through interconnection mechanism
- The processors can communicate with each other through memory

Symmetric Multiprocessor Organization



- The time shared bus is the simplest mechanism for constructing a multiprocessor system
- The structure and interfaces are basically the same as for a single processor system that uses a bus interconnection
- The bus consists of control, address, data lines

Multithreading and Chip Multiprocessors

- Instruction stream divided into smaller streams (threads)
- Executed in parallel
- Wide variety of multithreading designs

- The most important measure of performance for a processor is the rate at which it executes instruction. This can be expressed as,

  MIPS rate = $f \times IPC$

  where f is the processor clock frequency in MHz, and IPC (instructions per cycle).