William Stallings Computer Organization and Architecture 8th Edition

Chapter 17

Parallel Processing

Multiprocessor Configurations

1. Flynn's classification,
2. Parallel processing concepts,
3. Introduction to pipeline processing and pipeline hazards,
4. design issues of pipeline architecture,
5. Instruction pipelining

Parallel Processing Concepts

- To fulfil increasing demands for higher performance
- Need to process data concurrently to achieve better throughput
- Parallelism
  - Multiple functional units-several ALU's
  - Multiple processors-several processors operating concurrently

Flynn's Classification

- Based on
  - Internal organization of processors
  - Interconnection Structures

  Instruction Stream

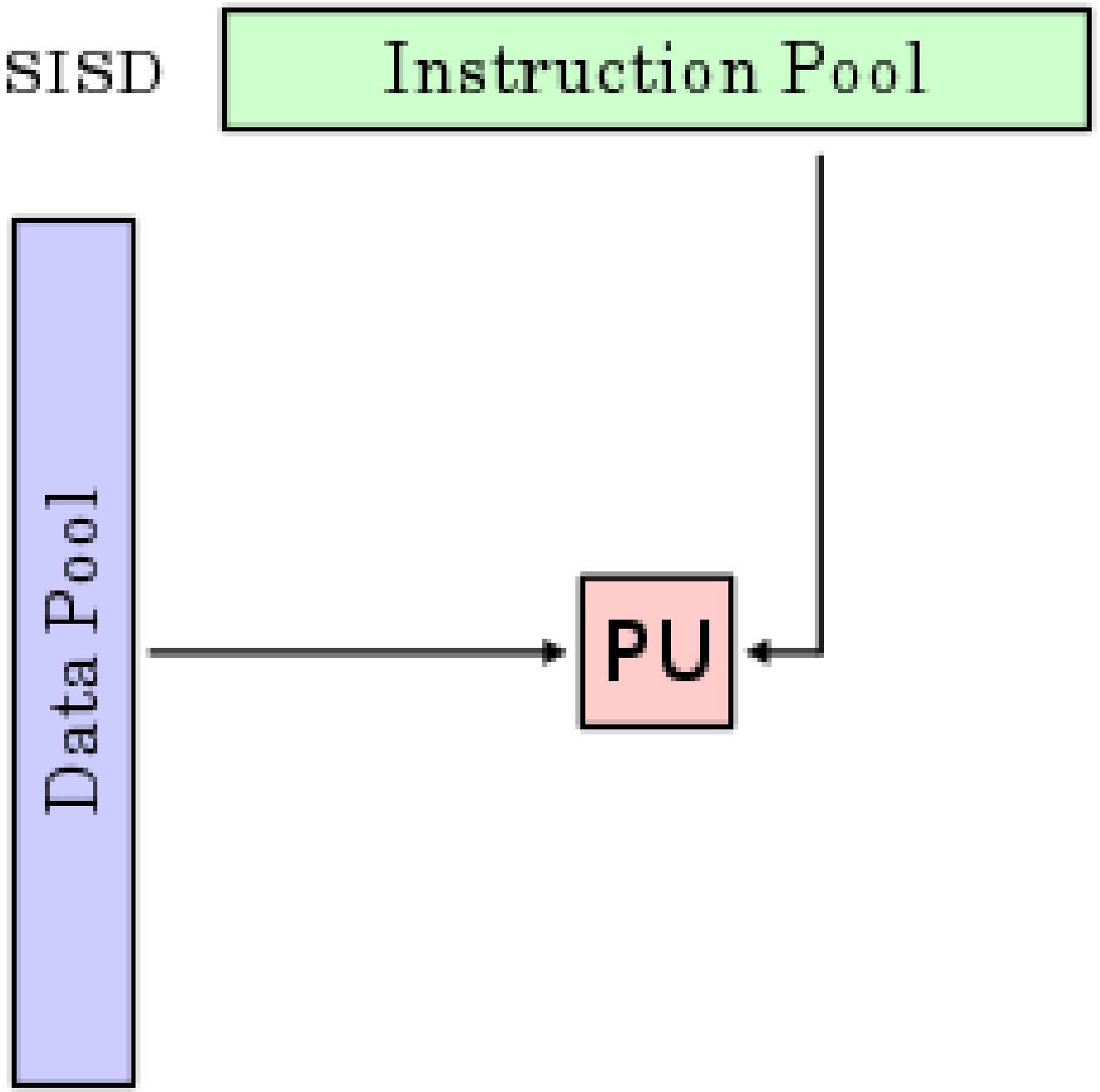
  Instruction from main memory to processor

  Data Stream

  Operands flowing to and from the processor

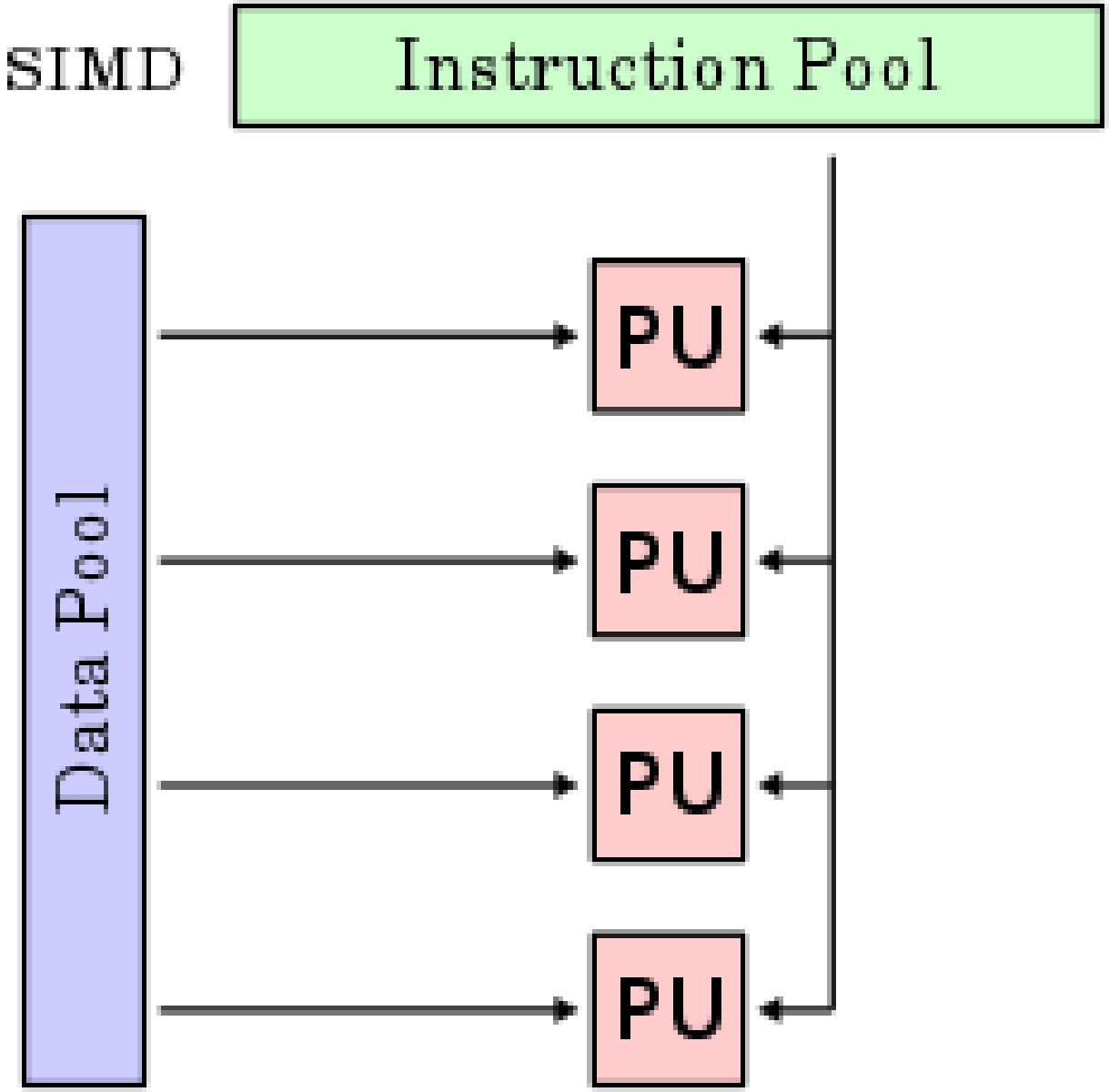
FLYNN's CLASSIFICATION
Multiple Processor Organization

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


SISD
Instruction Pool
Data Pool
PU

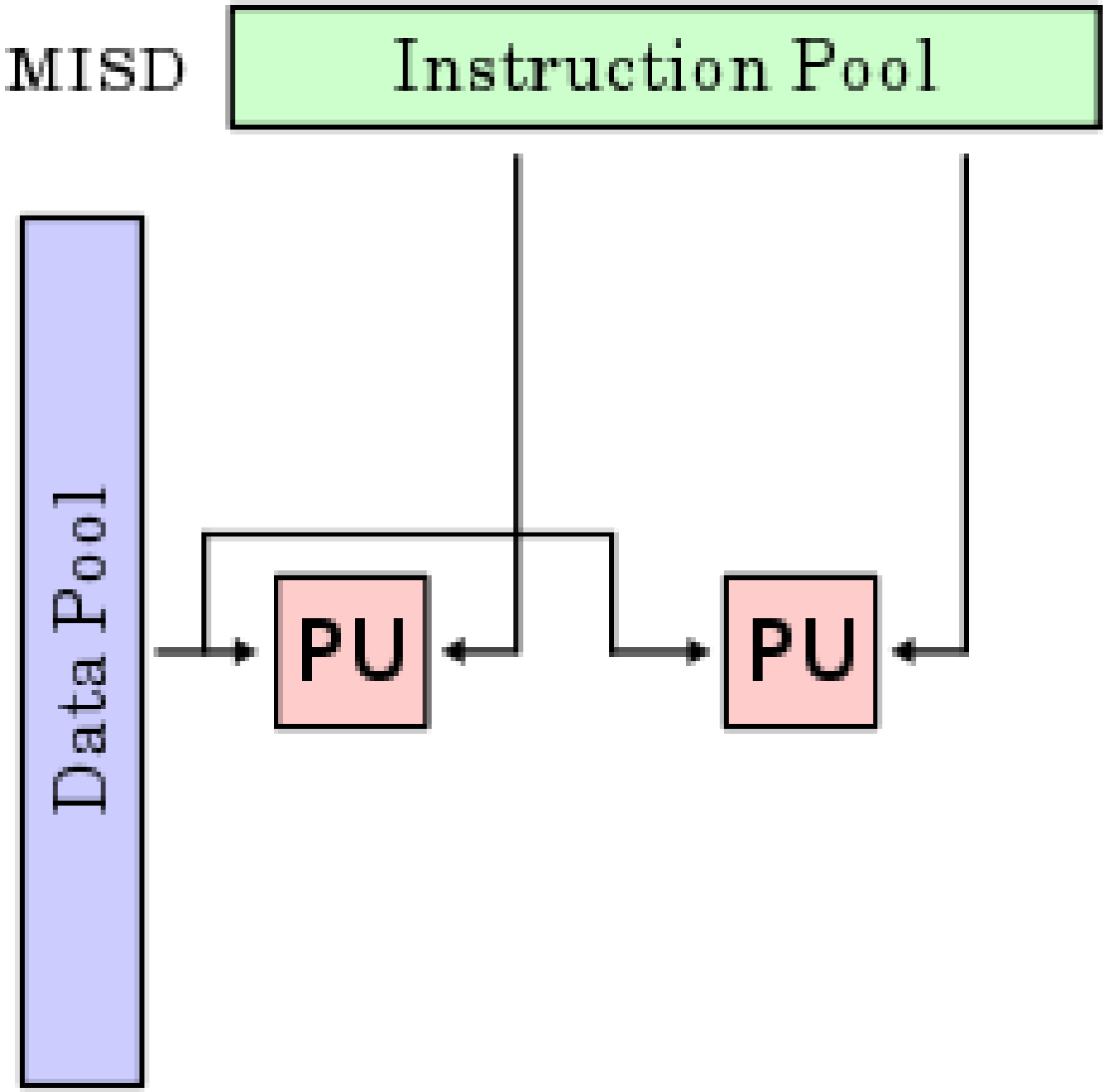
Single Instruction, Single Data Stream - SISD

- Single processor
- Single instruction stream
- Data stored in single memory
- Uni-processor


SIMD
Instruction Pool
Data Pool
PU
PU
PU
PU

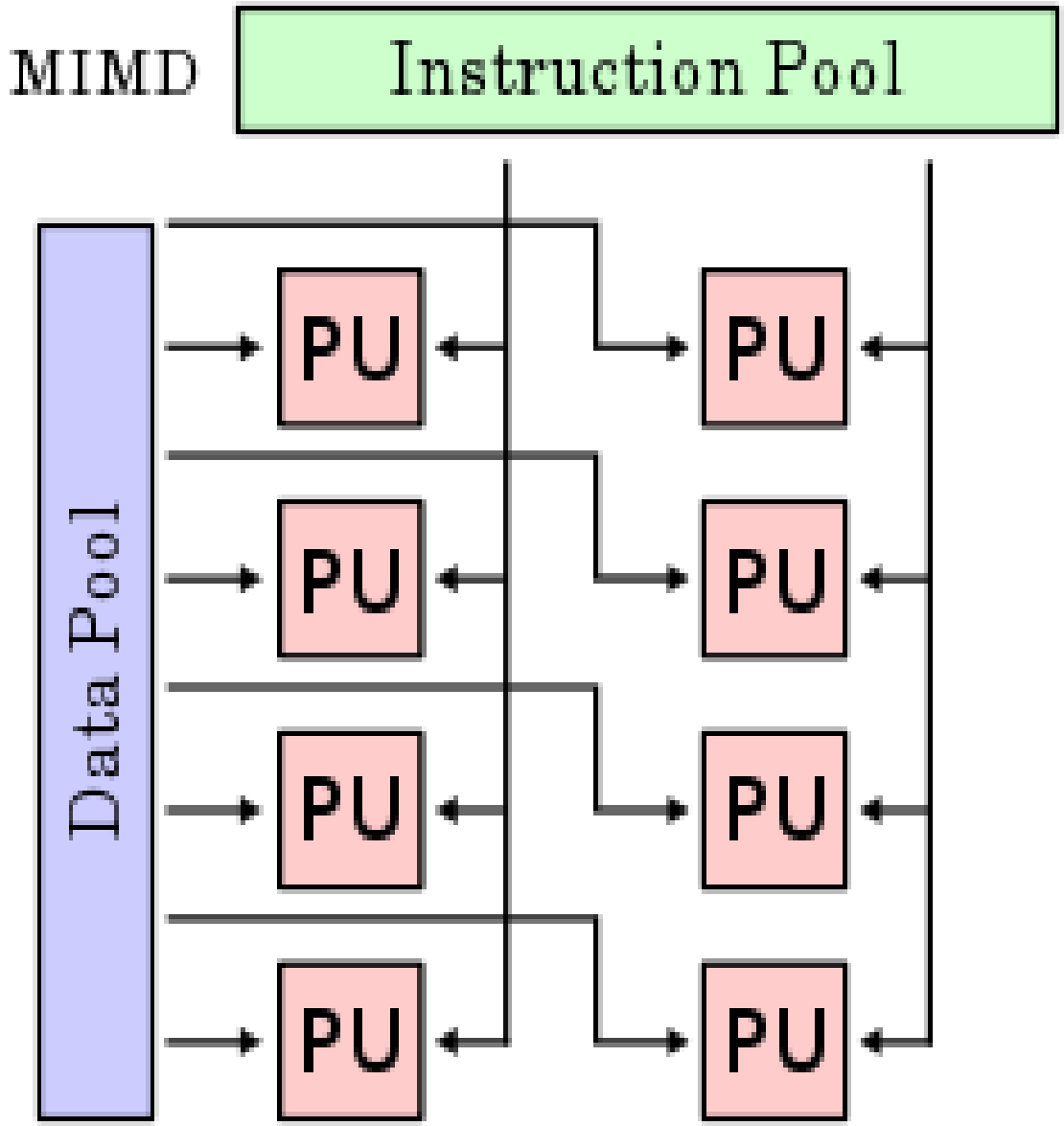
Single Instruction, Multiple Data Stream - SIMD

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


MISD
Instruction Pool
Data Pool
PU
PU

Multiple Instruction, Single Data Stream - MISD

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


MIMD
Instruction Pool
Data Pool
PU
PU
PU
PU
PU
PU
PU
PU

Multiple Instruction, Multiple Data Stream- MIMD

- Set of processors
- Simultaneously execute different instruction sequences
- Different sets of data
- SMPs, NUMA systems

Introduction to pipeline processing and pipeline hazards

- Pipelining- Temporal overlapping of processing
- Subdivide input task(process) into a sequence of subtasks
- Each executed by specialized hardware that operates concurrently with other stages of pipeline

Pipelining allows the next **instructions** to be fetched while the processor is performing arithmetic operations

Holds them in a buffer close to the processor until each **instruction** operation can be performed.

The staging of **instruction** fetching is continuous.

SIX STAGE OF INSTRUCTION PIPELINING

- **Fetch Instruction(FI)**

  Read the next expected instruction into a buffer
- **Decode Instruction(DI)**

  Determine the opcode and the operand specifiers.
- **Calculate Operands(CO)**

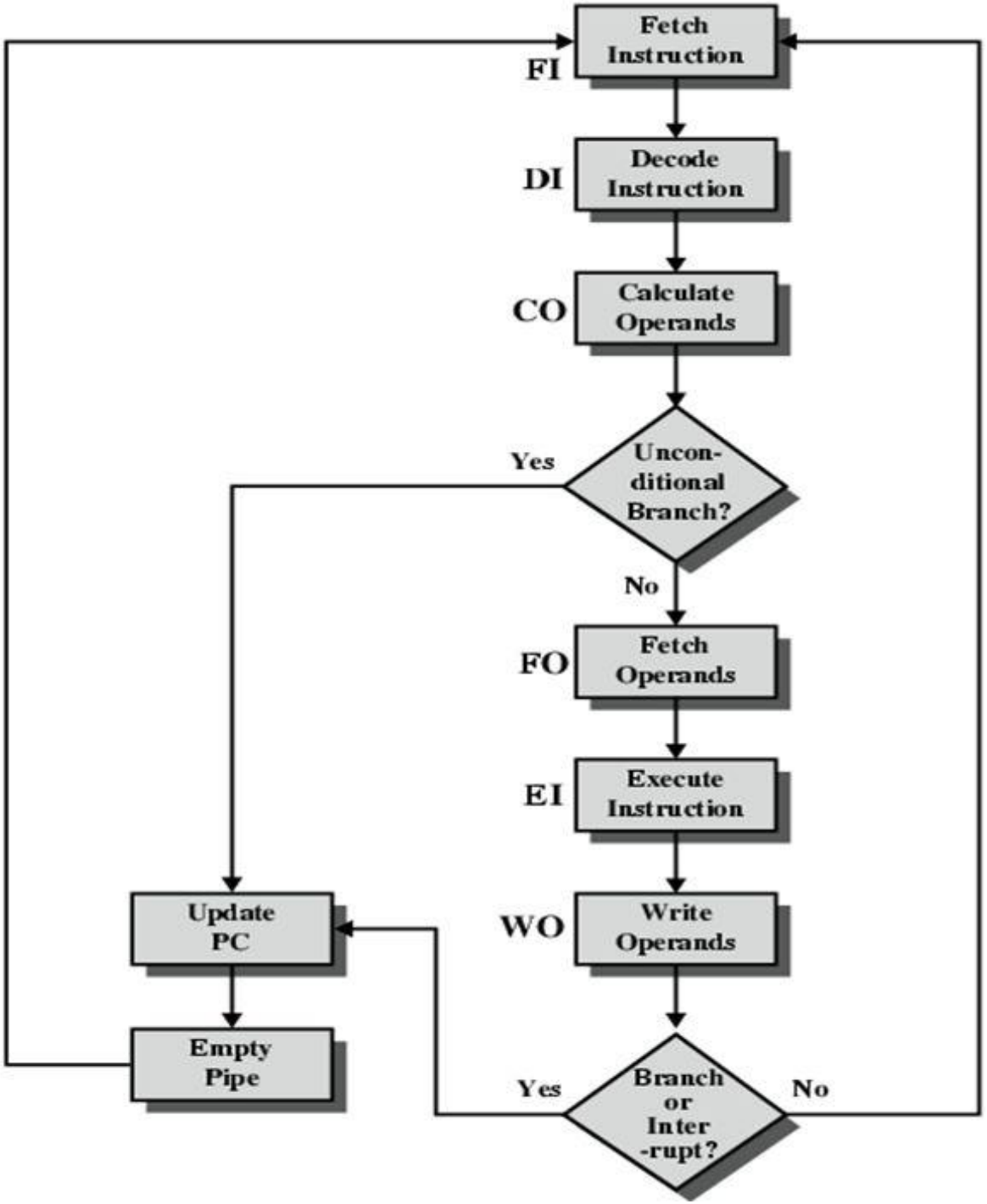
  Calculate the effective address of each source operand
- **Fetch Operands(FO)**

  Fetch each operand from memory. Operands in regist need not be fetched.
- **Execute Instruction(EI)**

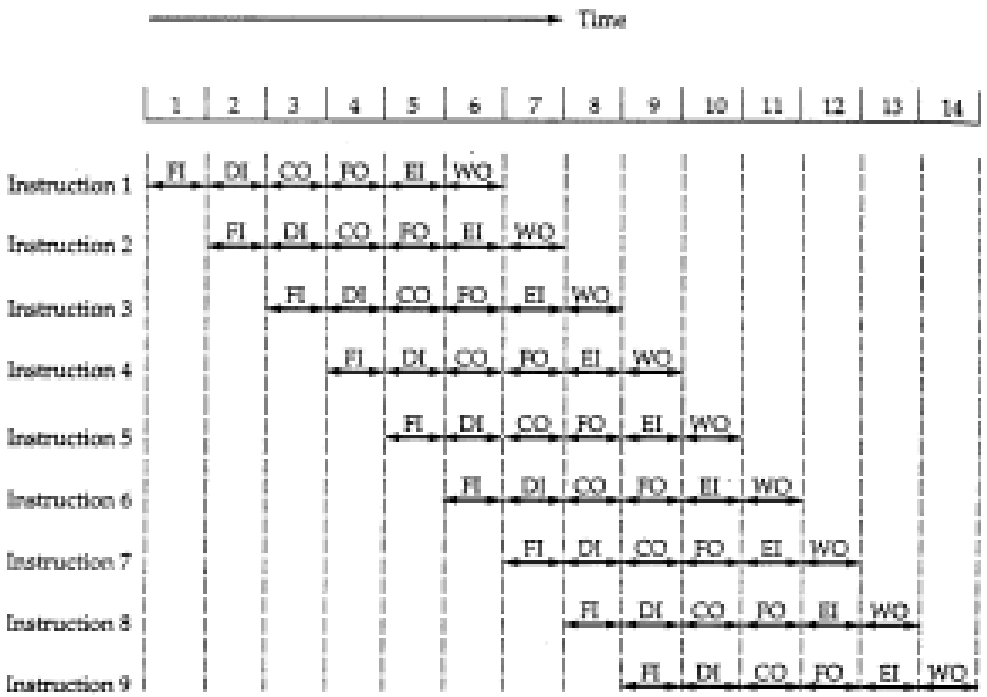
  Perform the indicated operation and store the result
- **Write Operand(WO)**

  Store the result in memory.

Six Stage Instruction Pipeline



Timing diagram for 6 stage instruction pipeline



Pipeline hazards

- Any reason that causes the pipeline to stall is called hazard or conflict
- Resource usage (inter instruction dependencies)
- Job scheduling problems

3 types of Hazards

- RESOURCE CONFLICTS
  - insufficient resources
- DATA or DATA DEPENDENCY CONFLICTS
  - instruction in pipeline depend on result of previous instruction which still in pipeline yet to be completed
- BRANCH DIFFICULTIES-Contents of PC get changed

RESOURCE HAZARD

- Stalling pipeline causes Structural Hazard
- Commonly need Memory Access
  - Use separate cache for **instruction** and **data**

DATA HAZARD

- RAW
- WAW
- WAR
- RAR?

- Rearrange the pipeline- pipeline scheduling

BRANCH HAZARDS

- Flush Pipeline
- Delayed branching
- Conditional branching

Design issues of pipeline architecture

- **Instruction Pipeline design**
  - Instruction Issuing
    - In order issuing, out of order issuing or re order issuing
- **Arithmetic Pipeline design**
  - Fixed point , floating point , integer arithmetic

Principles of designing pipelined processors

- **Proper data buffering** to avoid congestion and smooth pipeline operations
- Instruction **dependence** relationship
- **Logic hazards** should be detected and resolved
- Avoid Collisions and structural hazards by proper **sequencing**
- **Reconfiguration** of the pipeline should be possible