Superscalar Architecture

Introduction

Computer System Architecture

IIT Tirupati

24th March, 2020

jtt@iittp.ac.in

Instruction-Level Parallelism

- When exploiting instruction-level parallelism, goal is to maximize IPC
  - Pipeline IPC =
    - Ideal pipeline, IPC = 1
    - In reality we have stalls due to hazards
      - Structural stalls
      - Data hazard stalls
      - Control stalls
    - Therefore, IPC < 1

- Parallelism with basic block is limited
  - Typical size of basic block = 3-6 instructions
  - Must optimize across branches

Limitations of Scalar Pipelines

1. Scalar upper bound on throughput
   - IPC <= 1 or CPI >= 1
   - Solution: wide (superscalar) pipeline
   - (Also exploited in VLIW architecture, this will not be covered in this course, you can read for yourself)
2. Inefficient unified pipeline
   - Long latency for each instruction
3. Rigid pipeline stall policy
   - One stalled instruction stalls all newer instructions

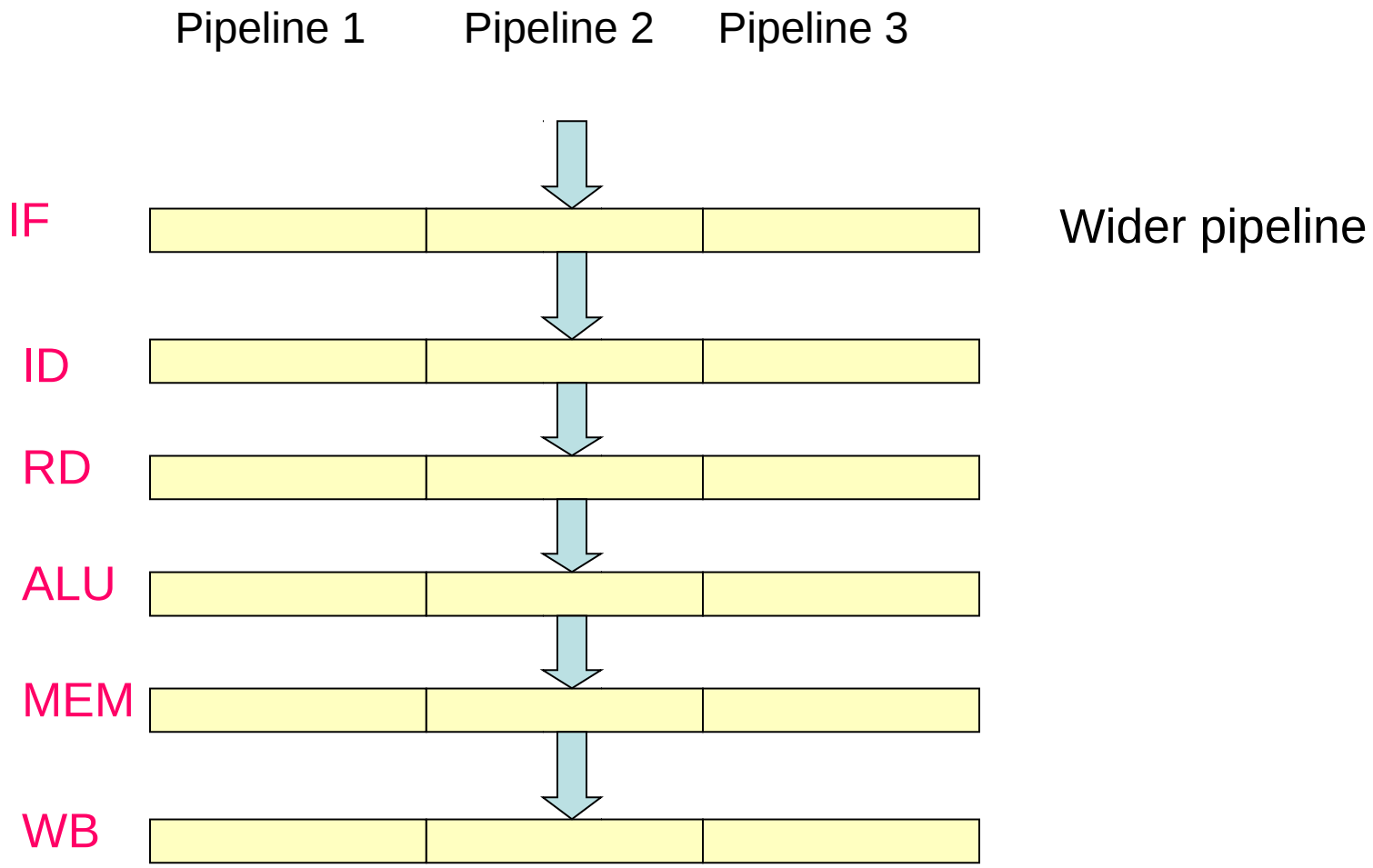
The super scalar architecture addresses above three problems!

Superscalar Architecture

The solution to the first problem, throughput limit

- Wide pipeline
- Instructions are not independent
- Superscalar architecture is natural descendant of pipelined scalar RISC
- Superscalar techniques largely concern the processor organization, independent of the ISA and the other architectural features
- Thus, possibility to develop a processor code compatible with an existing architecture

Superscalar Pipelines



Ideally all the pipeline should be filled in all cycles! However, the instruction dependency is a big challenge (hurdle).

Limitations of Scalar Pipelines

1. Scalar upper bound on throughput
   - IPC <= 1 or CPI >= 1
   - Solution: wide (superscalar) pipeline
   - (Also exploited in VLIW architecture, this will not be covered in this course, you can read for yourself)
2. Inefficient unified pipeline
   - Long latency for each instruction
   - Solutions: Diversified and specialized pipeline

Both of the above solutions depends on

ILP: Instruction level parallelism and
MLP: machine level parallelism

Instruction Level Parallelism

- Instruction parallelism of a program is a measure of the average number of instructions that a superscalar processor might be able to execute at the same time

- Mostly, ILP is determined by the number of true dependencies and the number of branches in relation to other instructions

Machine Level Parallelism

- Machine parallelism of a processor is a measure of the ability of processor to take advantage of the ILP (ILP in hardware)
- Determined by the number of instructions that can be fetched and executed at the same time
- A challenge in the design of superscalar processor is to achieve good balance between instruction parallelism and machine parallelism

Superscalar Pipelines (Diversified)

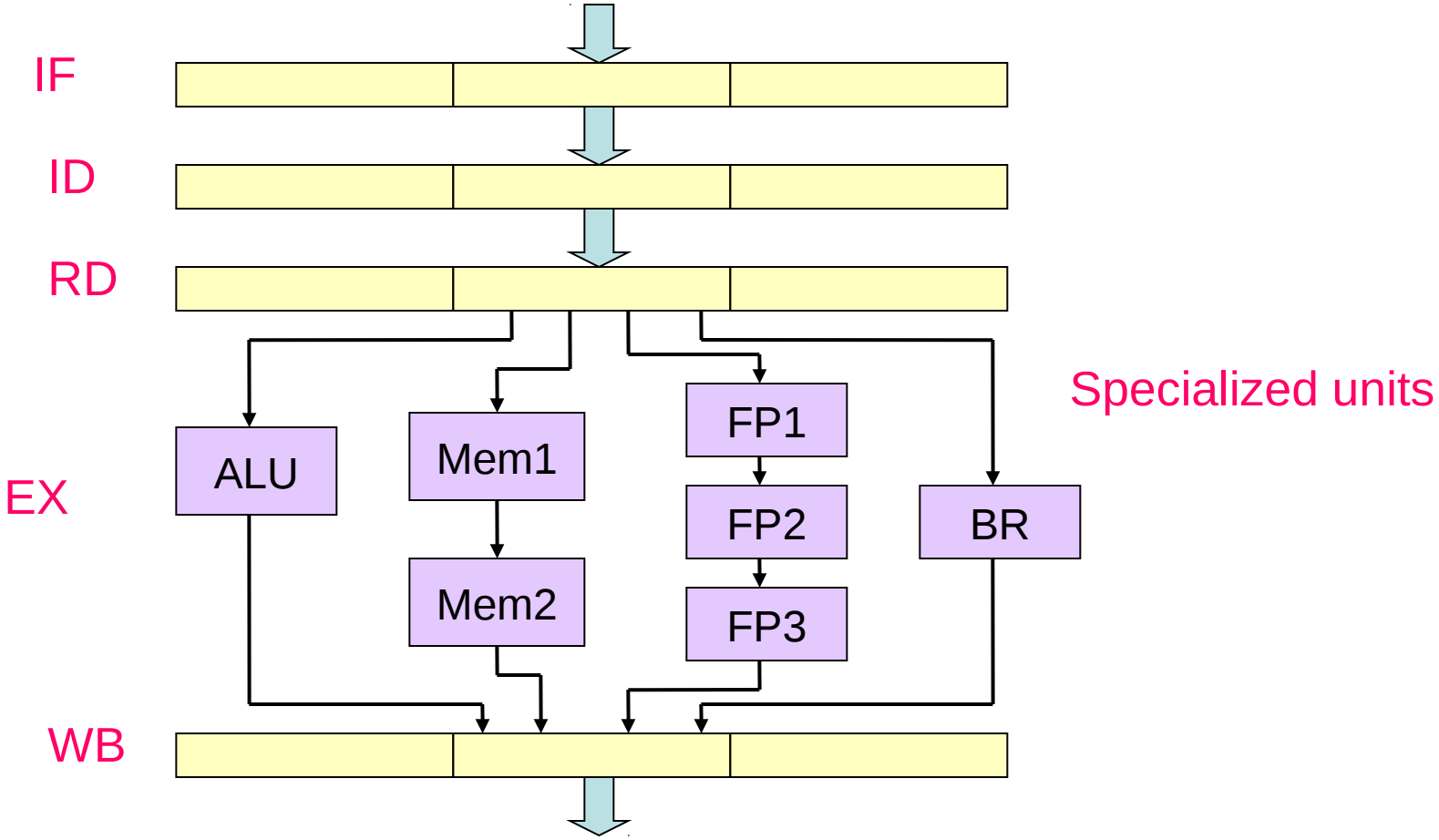
Diversified Pipelines

- Each pipeline can be customized for particular instruction type
- Each instruction type incurs only necessary latency
- Certainly less expensive than identical copies
- If all inter-instruction dependencies are resolved then there is no stall after instruction issue

Require special consideration

- Number (how many?) and Mix (which type?) of functional units

Superscalar Pipelines (Diversified)



Limitations of Scalar Pipelines

1. Scalar upper bound on throughput
   - IPC <= 1 or CPI >= 1
   - Solution: wide (superscalar) pipeline
   - (Also exploited in VLIW architecture, this will not be covered in this course, you can read for yourself)
2. Inefficient unified pipeline
   - Long latency for each instruction
   - Solutions: Diversified and specialized unit
3. Rigid pipeline stall policy
   - One stalled instruction stalls all newer instructions
   - Solutions: Out-of-order execution

The super scalar architecture addresses above three problems!

Superscalar Pipelines

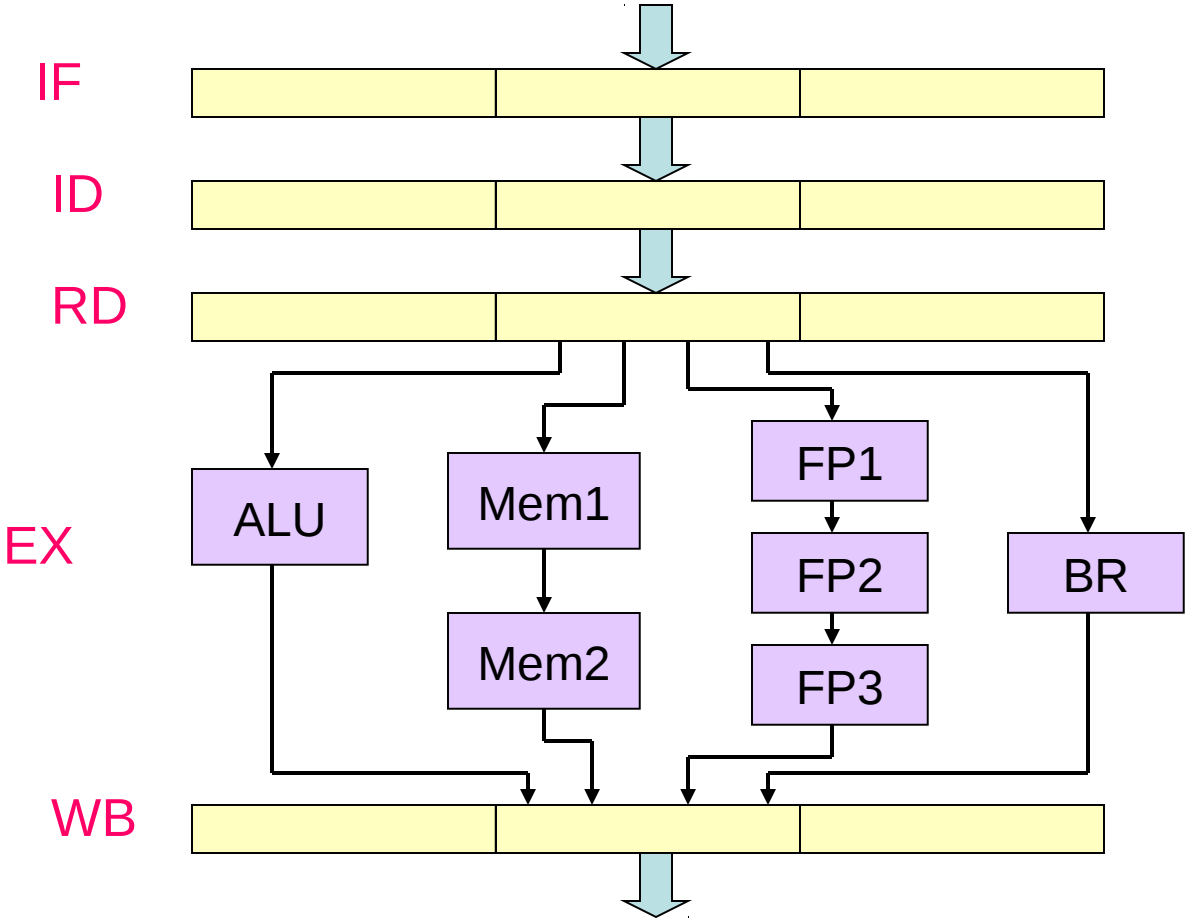
Dynamic Pipelines:

Superscalar pipeline are also called dynamic pipeline for the reason that executes instruction out-of-order

Features:

1. Alleviate the limitations of pipelined implementation
2. Use diversified pipelines
3. Temporal machine parallelism

Superscalar Pipelines



In program order
(in order)
IF | ID | RD

Out-of-order
execution

In program order
(in order)
WB

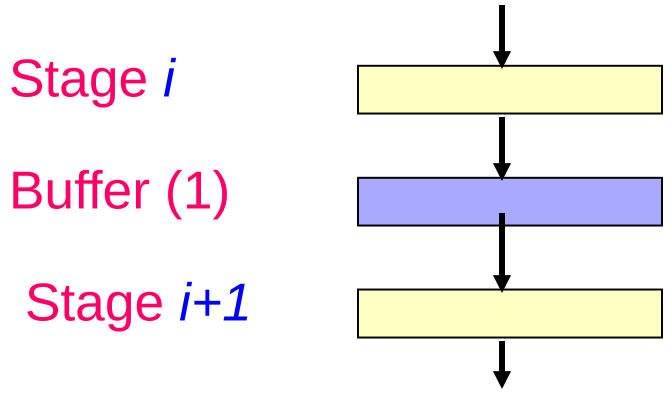
Superscalar Pipelines

Dynamic Pipelines implementation issues:

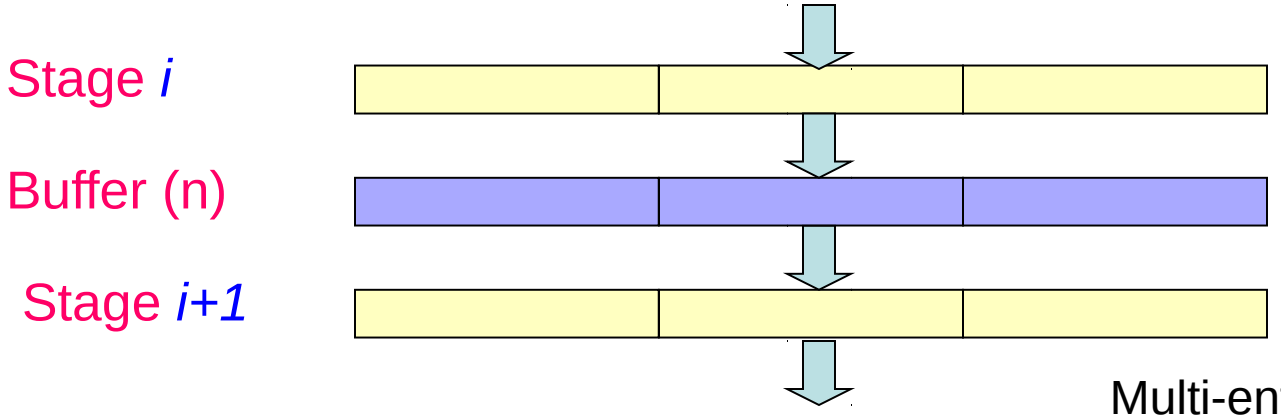
- Buffers are needed
  - Multi-entry buffers
  - Every entry is hardwired to one read port and one write port
  - Complex multi-entry buffers
  - Minimize stalls

Superscalar Pipelines

Interpipeline-Stage Buffers:



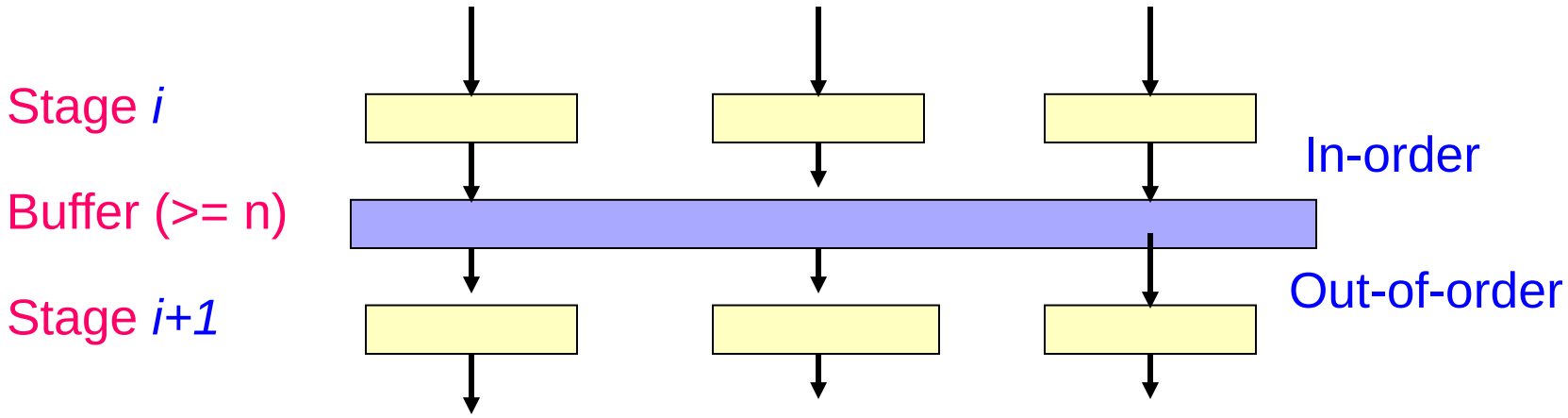
Single entry



Multi-entry Buffer

Superscalar Pipelines

Interpipeline-Stage Buffers:



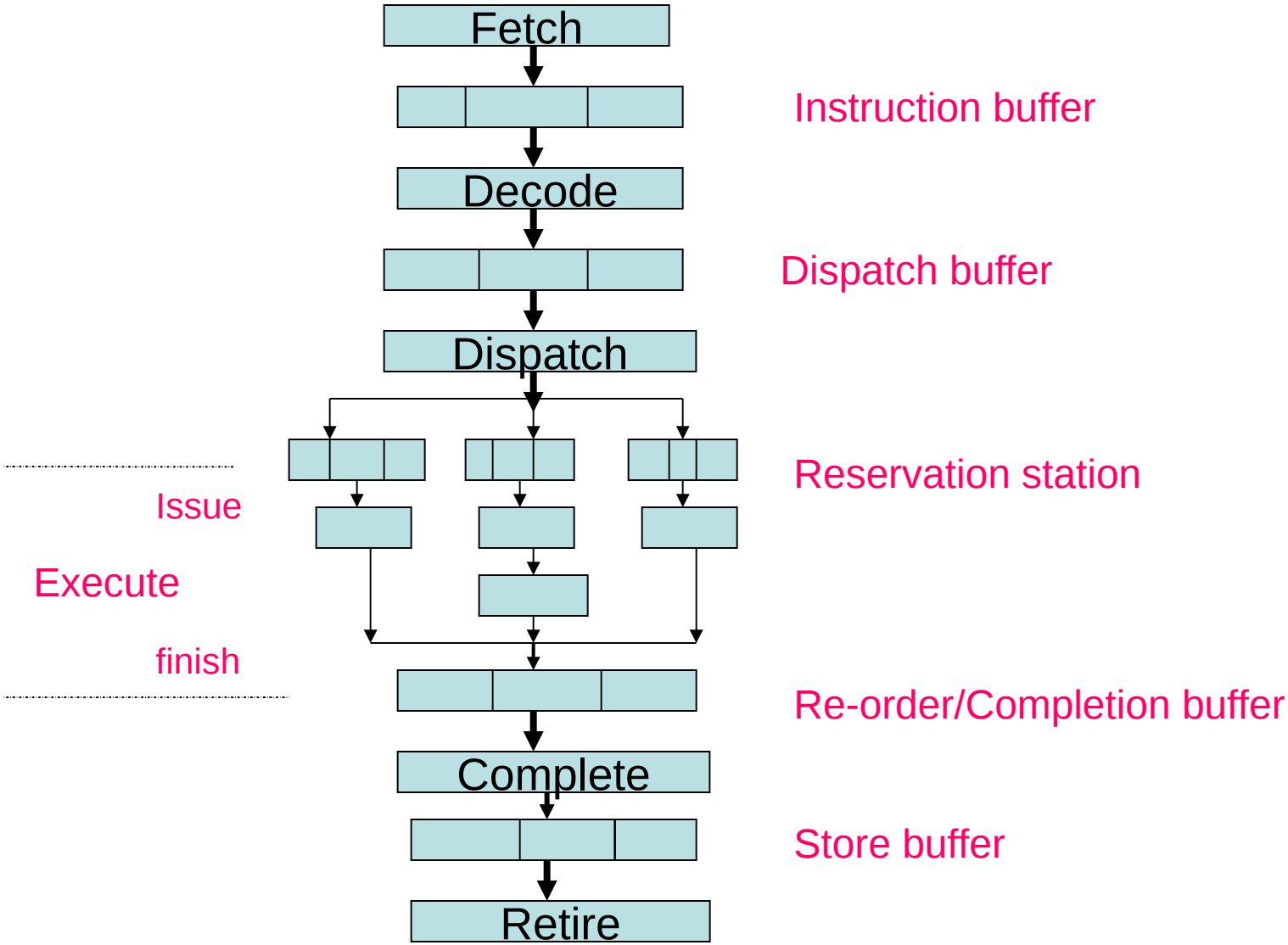
- Trailing instructions are allowed to bypass stalled instruction
- Out-of-order execution

Instruction Issue & MLP

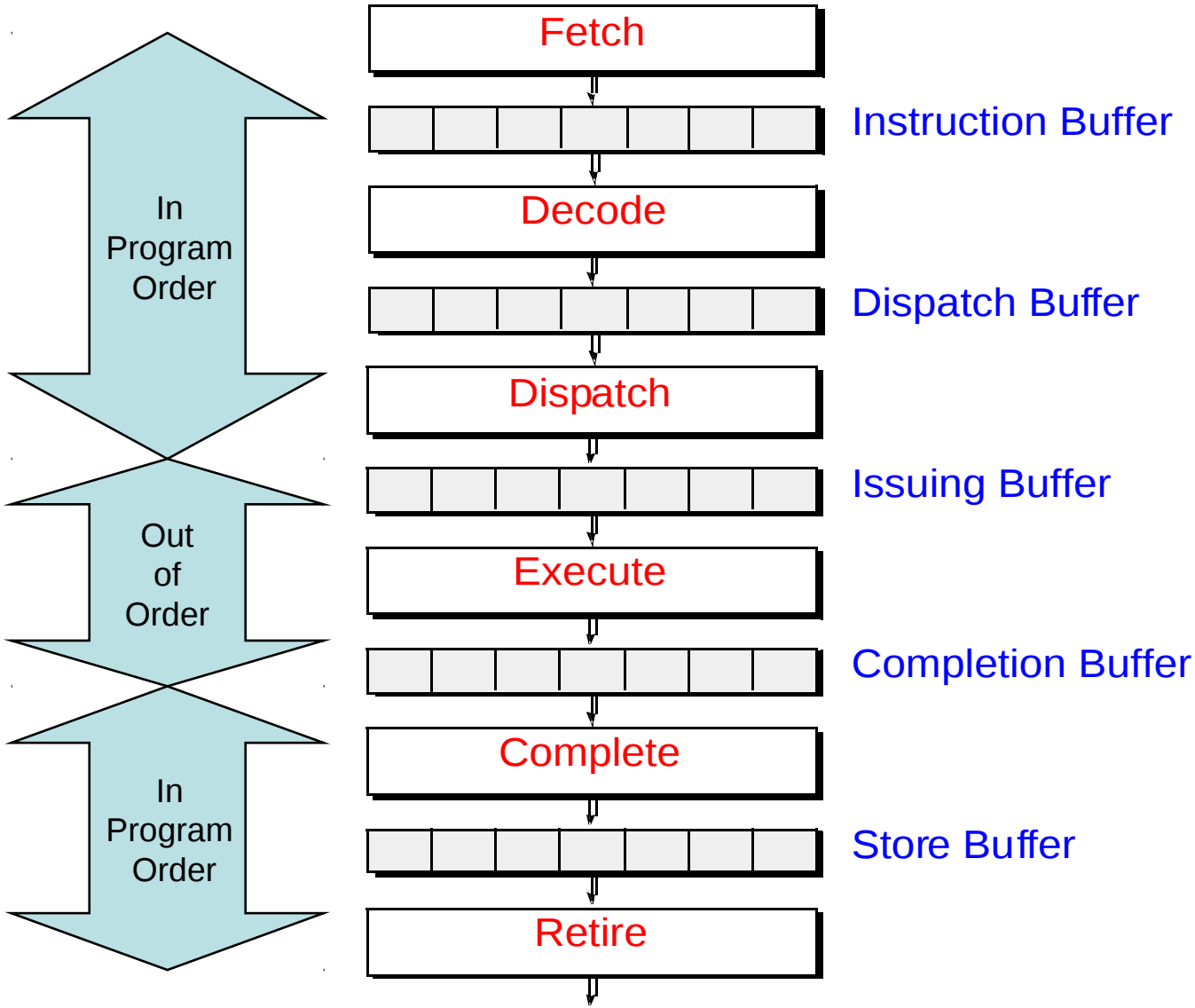
- ILP is not necessarily exploited by widening the pipelines and adding more resources
- Processor policies towards fetching decoding, and executing instruction have significant effect on its ability to discover instructions which can be executed concurrently
- Instruction issue is refer to the process of initiating instruction execution
- Instruction issue policy limits or enhances performance because it determines the processor's look ahead capability

Super-scalar Architecture

The modern processor design



Superscalar Pipeline Stages



Express Way:

Analogy to Superscalar pipeline

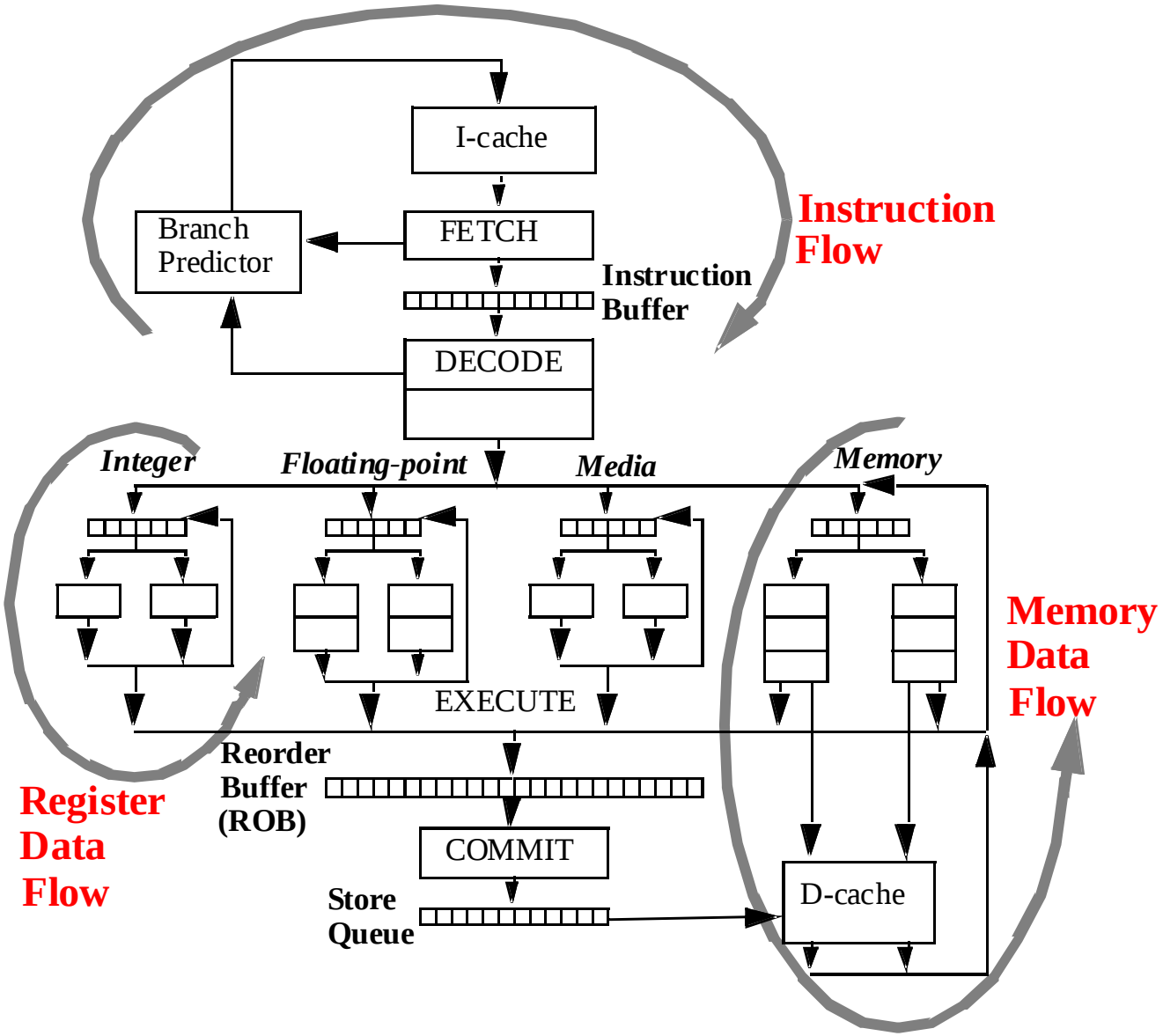
The multi-lane system

Bad Traffic:

Analogy to single pipeline

The common mono-lane system

Superscalar Challenges



Next Lecture:

Dynamic Scheduling