

Chapter 4: Superscalar Organization

Modern Processor Design: Fundamentals of Superscalar Processors

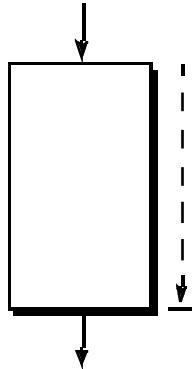
Mikko H. Lipasti

Lecture notes based in part on slides created by
John Shen, Mark Hill, David Wood, Guri Sohi,
and Jim Smith

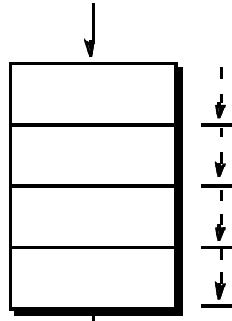
Limitations of Scalar Pipelines

- Scalar upper bound on throughput
 - IPC ≤ 1 or CPI ≥ 1
- Inefficient unified pipeline
 - Long latency for each instruction
- Rigid pipeline stall policy
 - One stalled instruction stalls all newer instructions

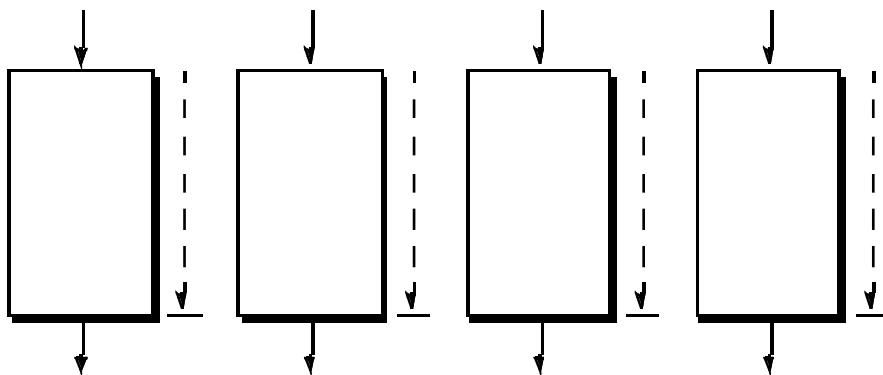
Parallel Pipelines



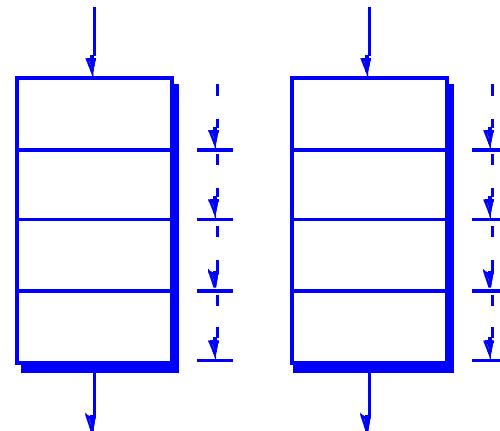
(a) No Parallelism



(b) Temporal Parallelism

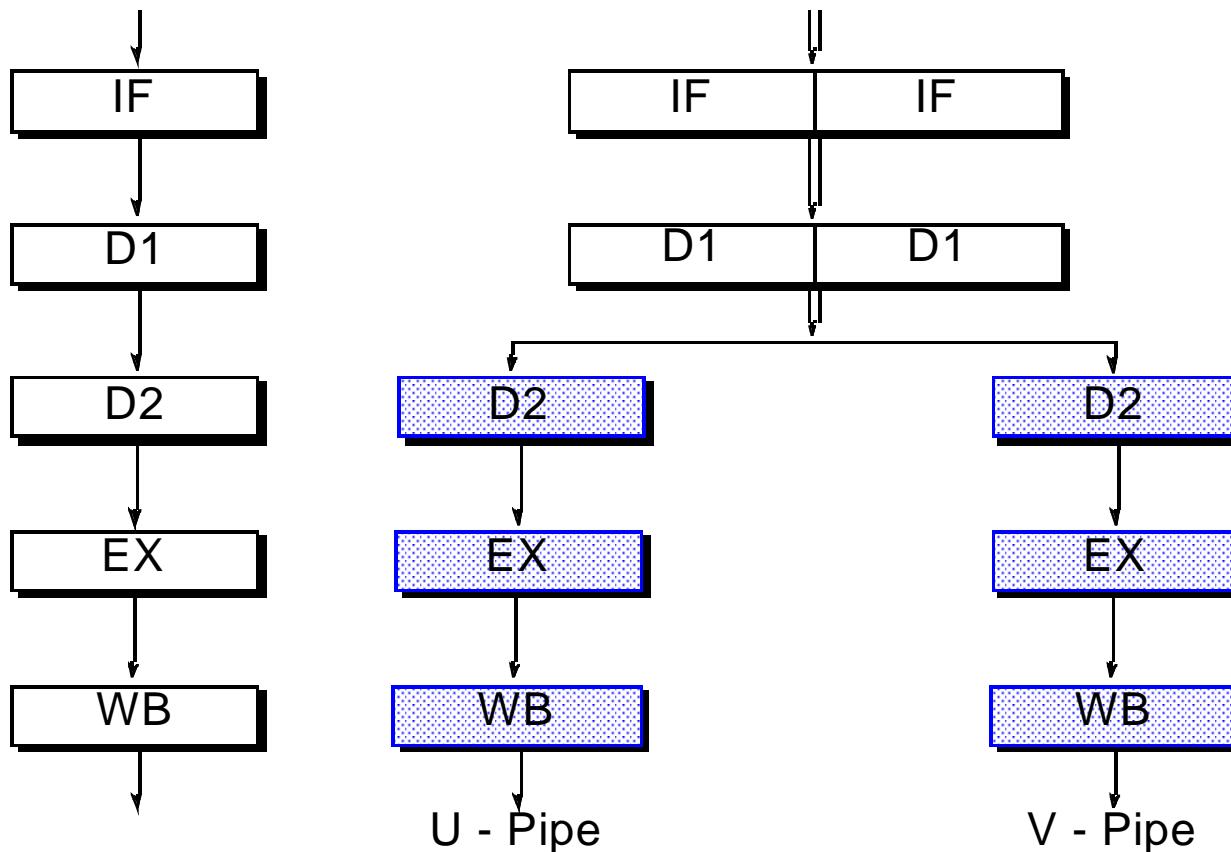


(c) Spatial Parallelism

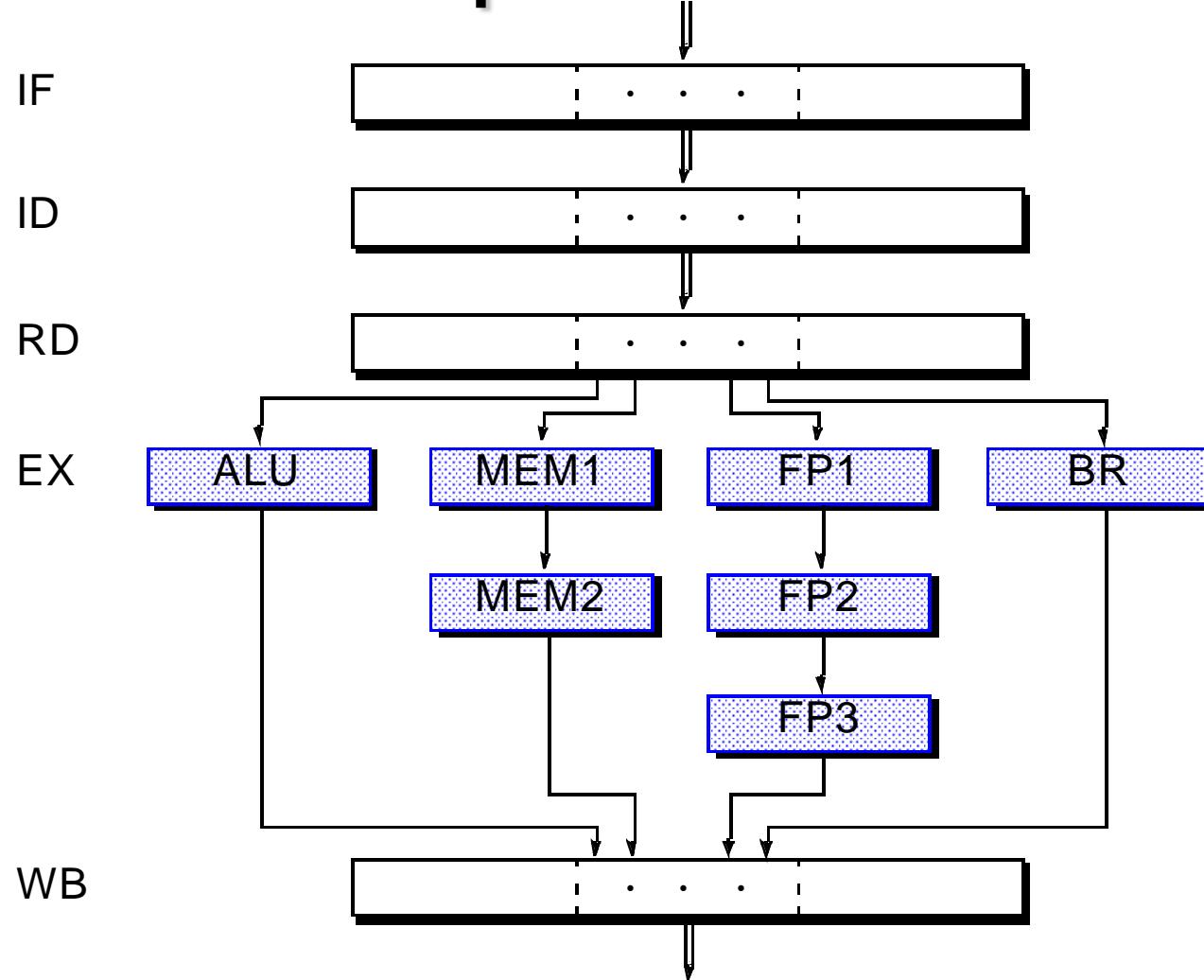


(d) Parallel Pipeline

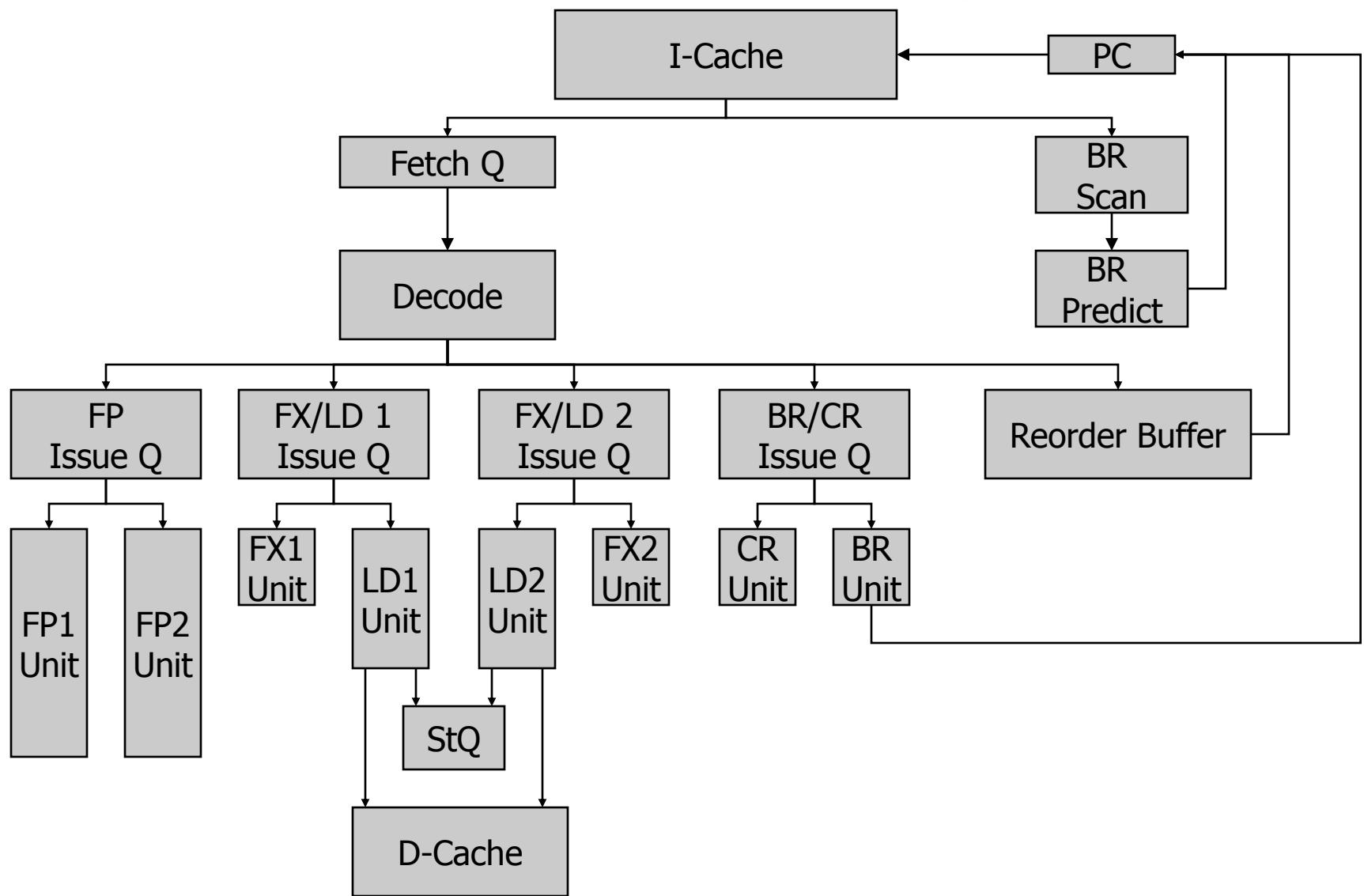
Intel Pentium Parallel Pipeline



Diversified Pipelines

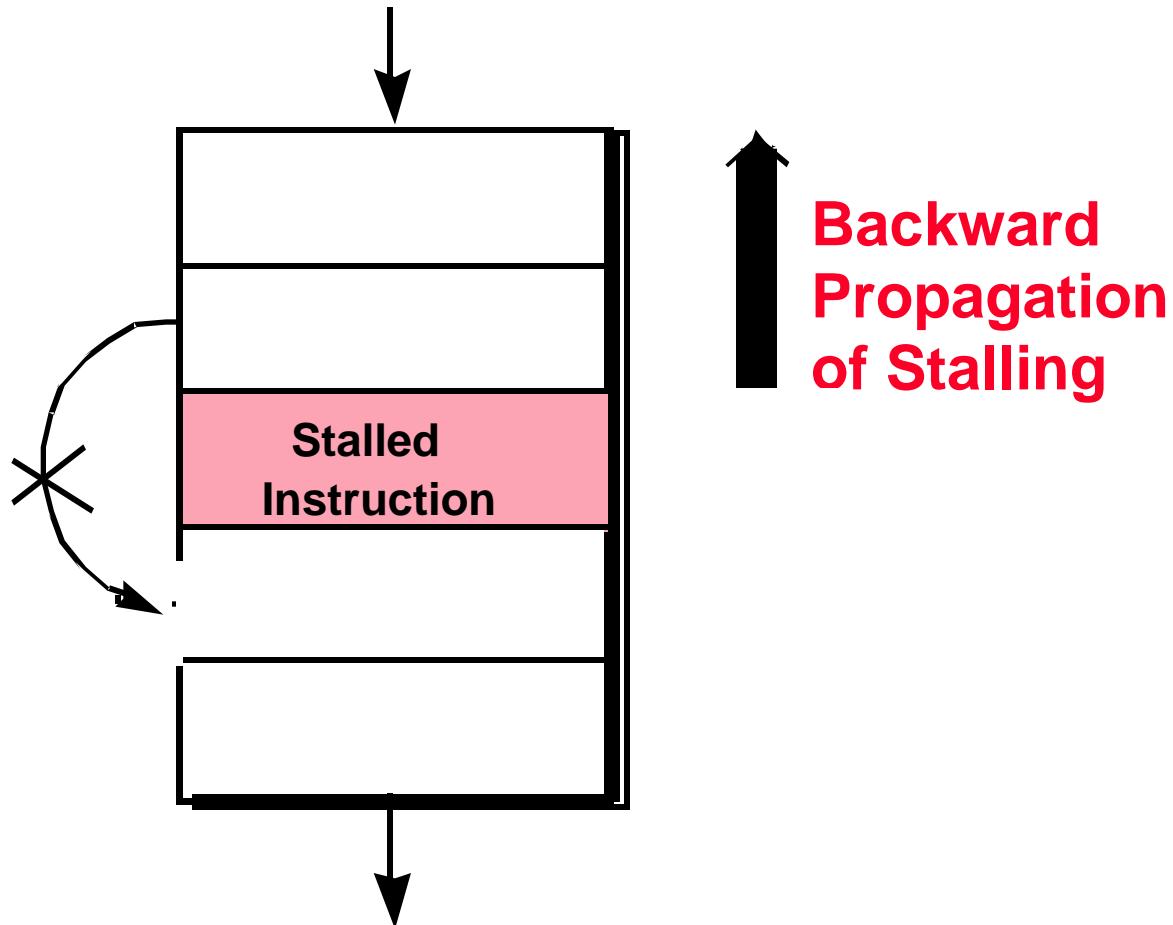


Power4 Diversified Pipelines

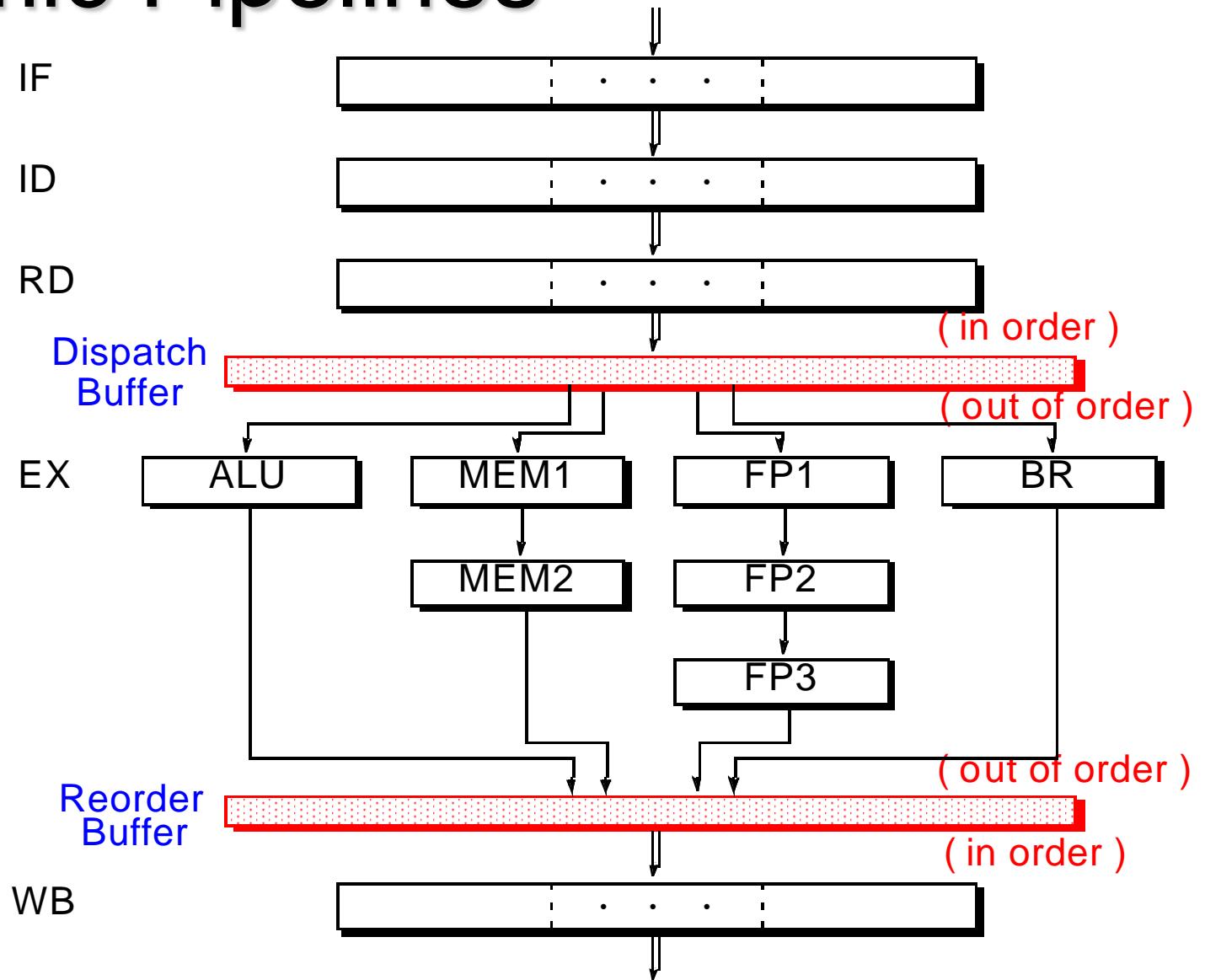


Rigid Pipeline Stall Policy

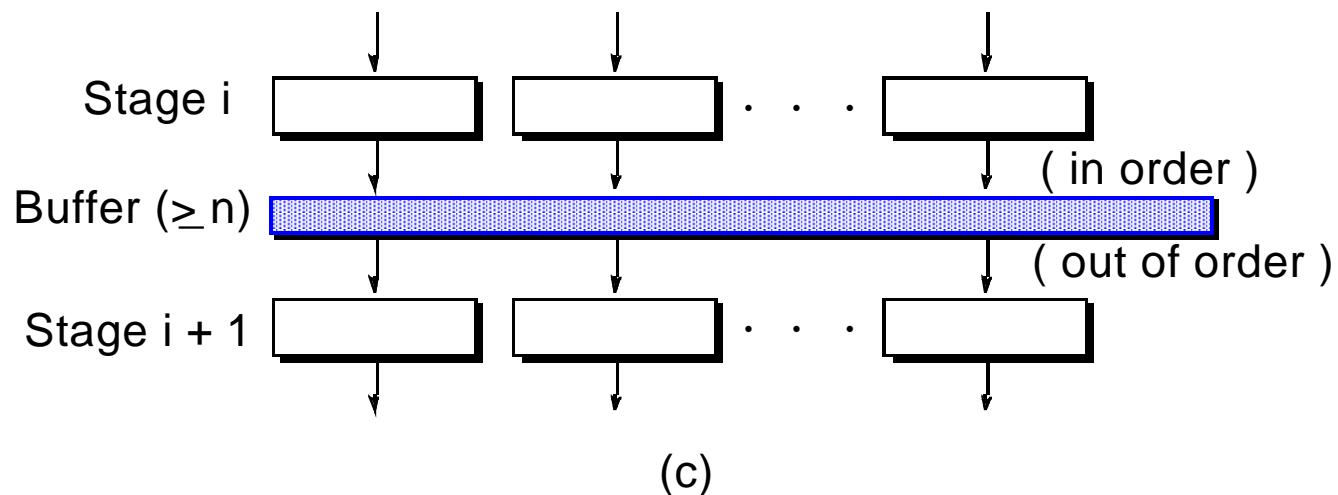
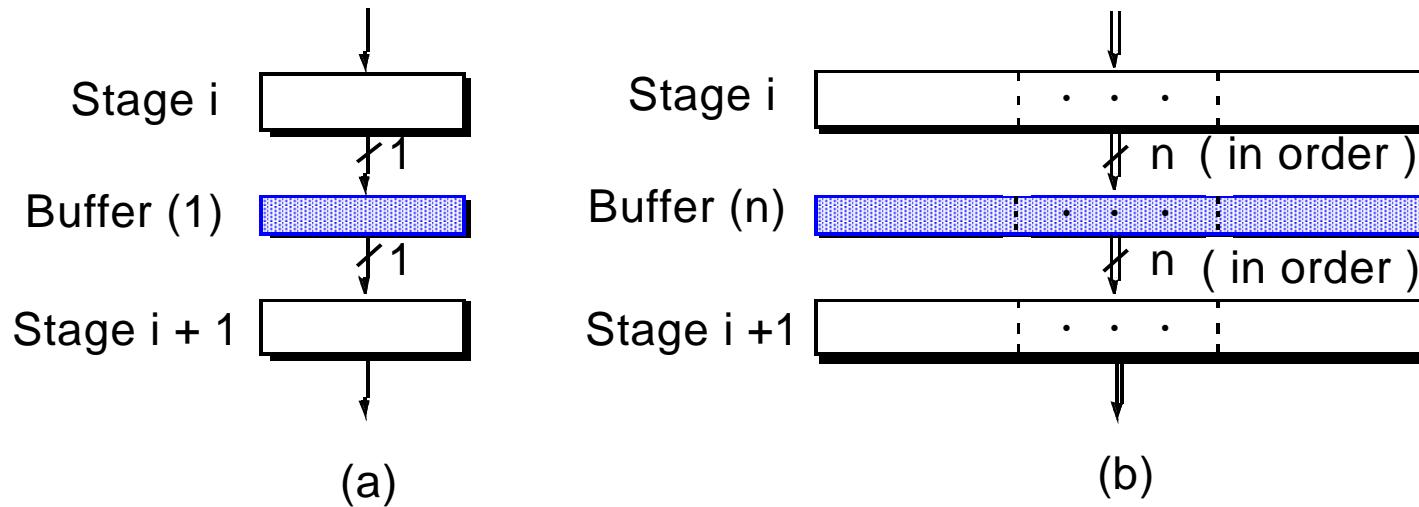
Bypassing
of Stalled
Instruction
Not Allowed



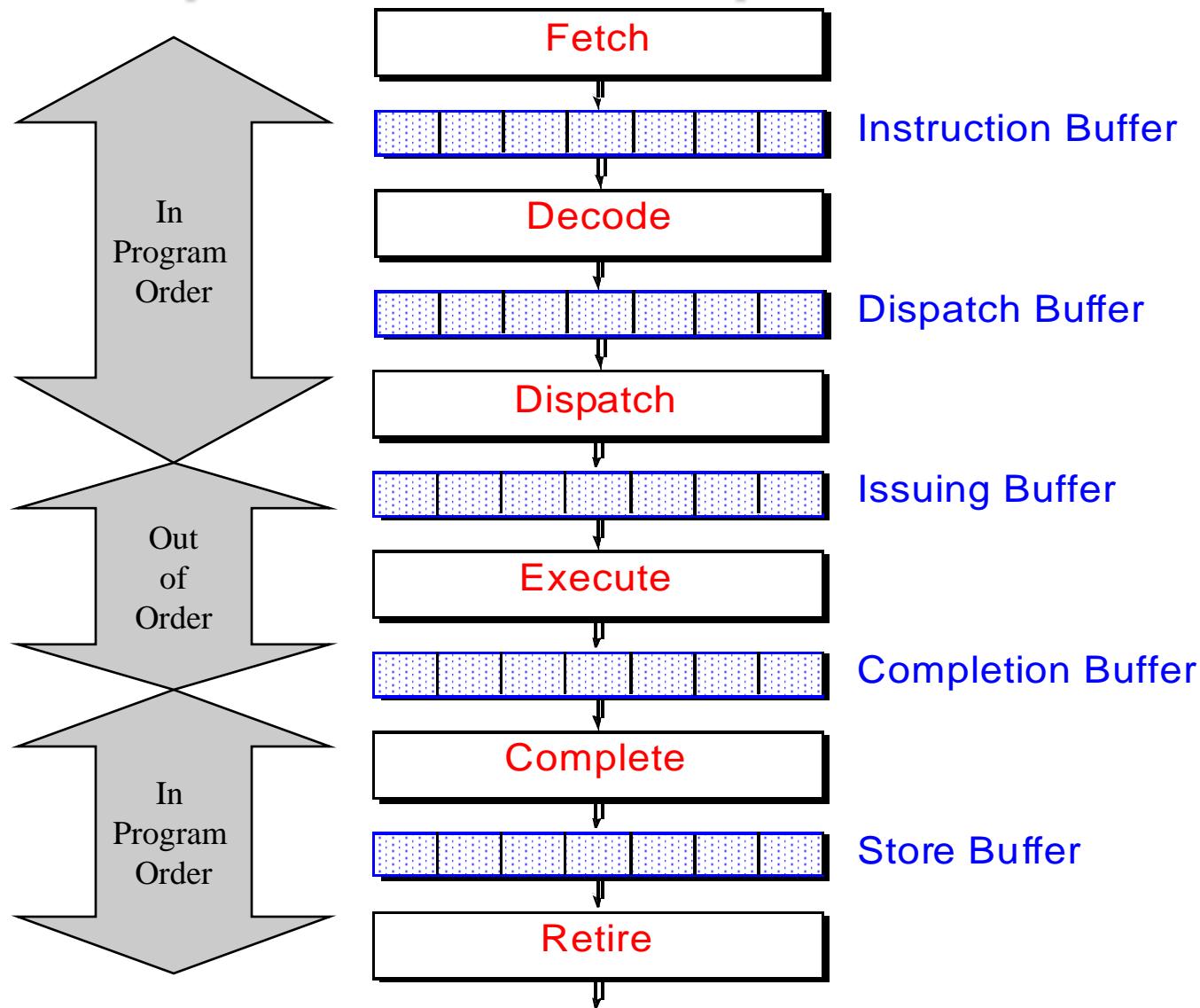
Dynamic Pipelines



Interstage Buffers



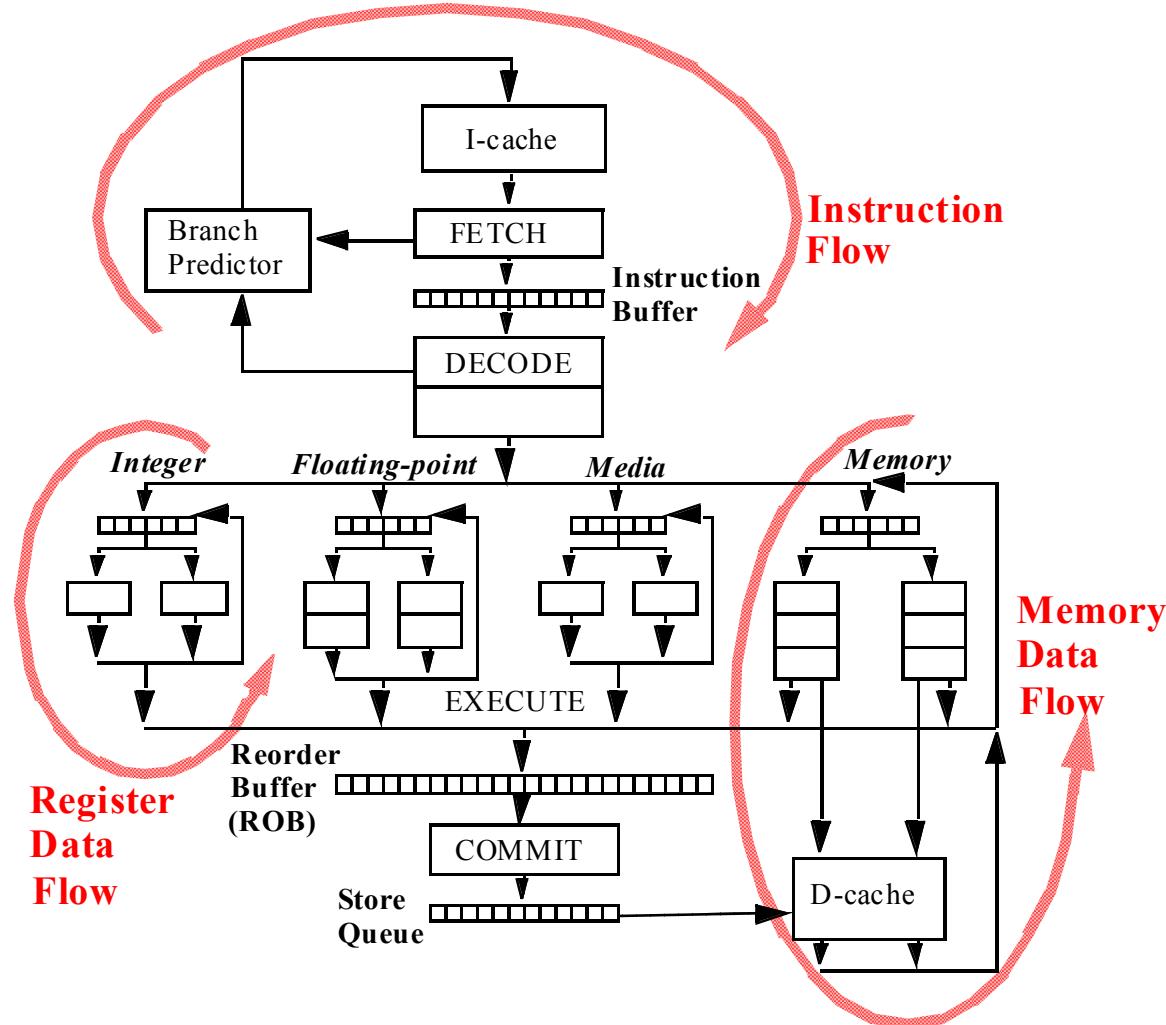
Superscalar Pipeline Stages



Limitations of Scalar Pipelines

- Scalar upper bound on throughput
 - IPC ≤ 1 or CPI ≥ 1
 - Solution: wide (superscalar) pipeline
- Inefficient unified pipeline
 - Long latency for each instruction
 - Solution: diversified, specialized pipelines
- Rigid pipeline stall policy
 - One stalled instruction stalls all newer instructions
 - Solution: Out-of-order execution, distributed execution pipelines

Impediments to High IPC

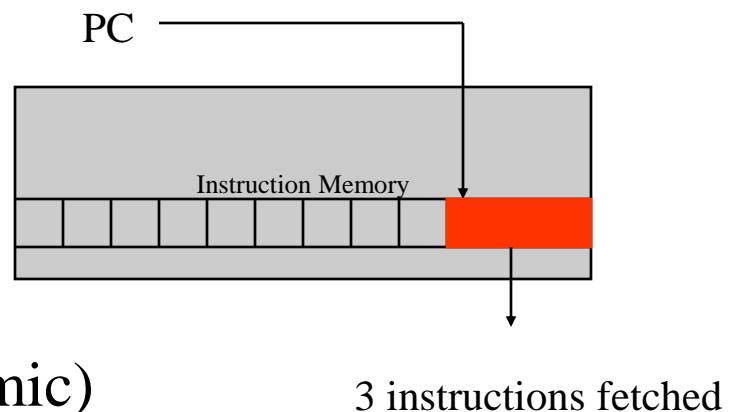


Superscalar Pipeline Design

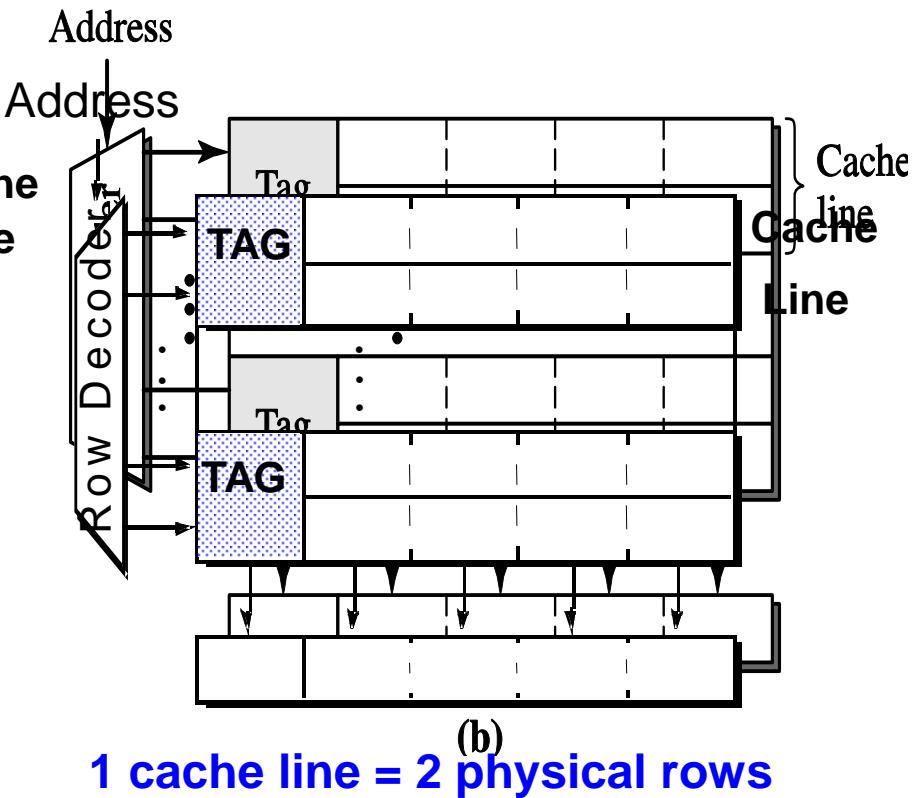
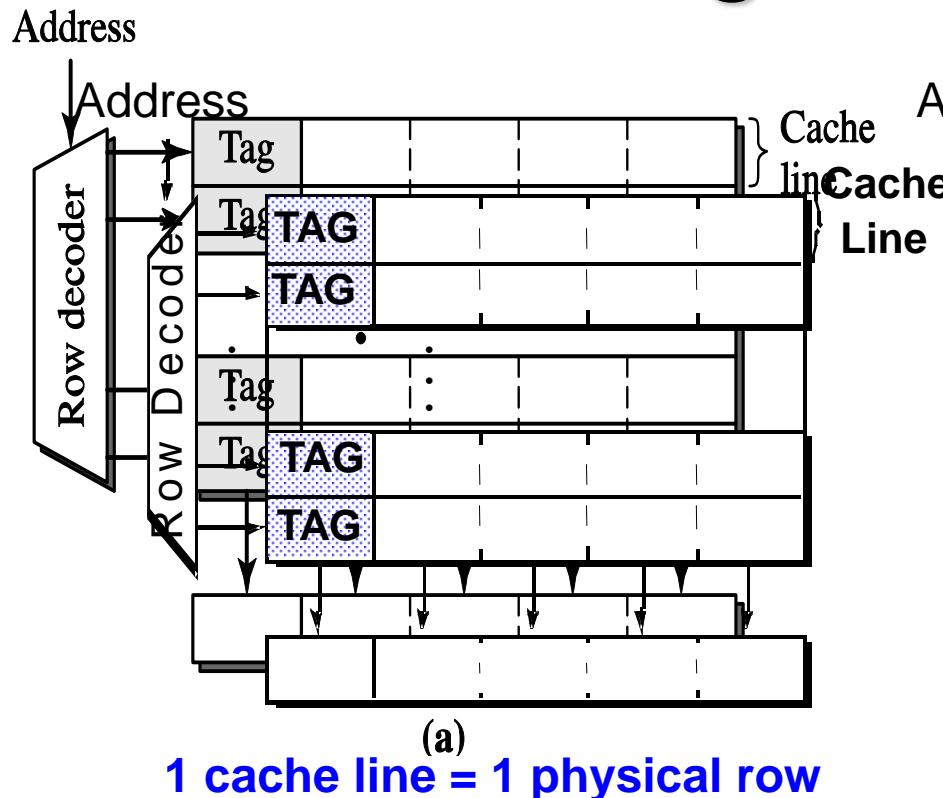
- Instruction Fetching Issues
- Instruction Decoding Issues
- Instruction Dispatching Issues
- Instruction Execution Issues
- Instruction Completion & Retiring Issues

Instruction Flow

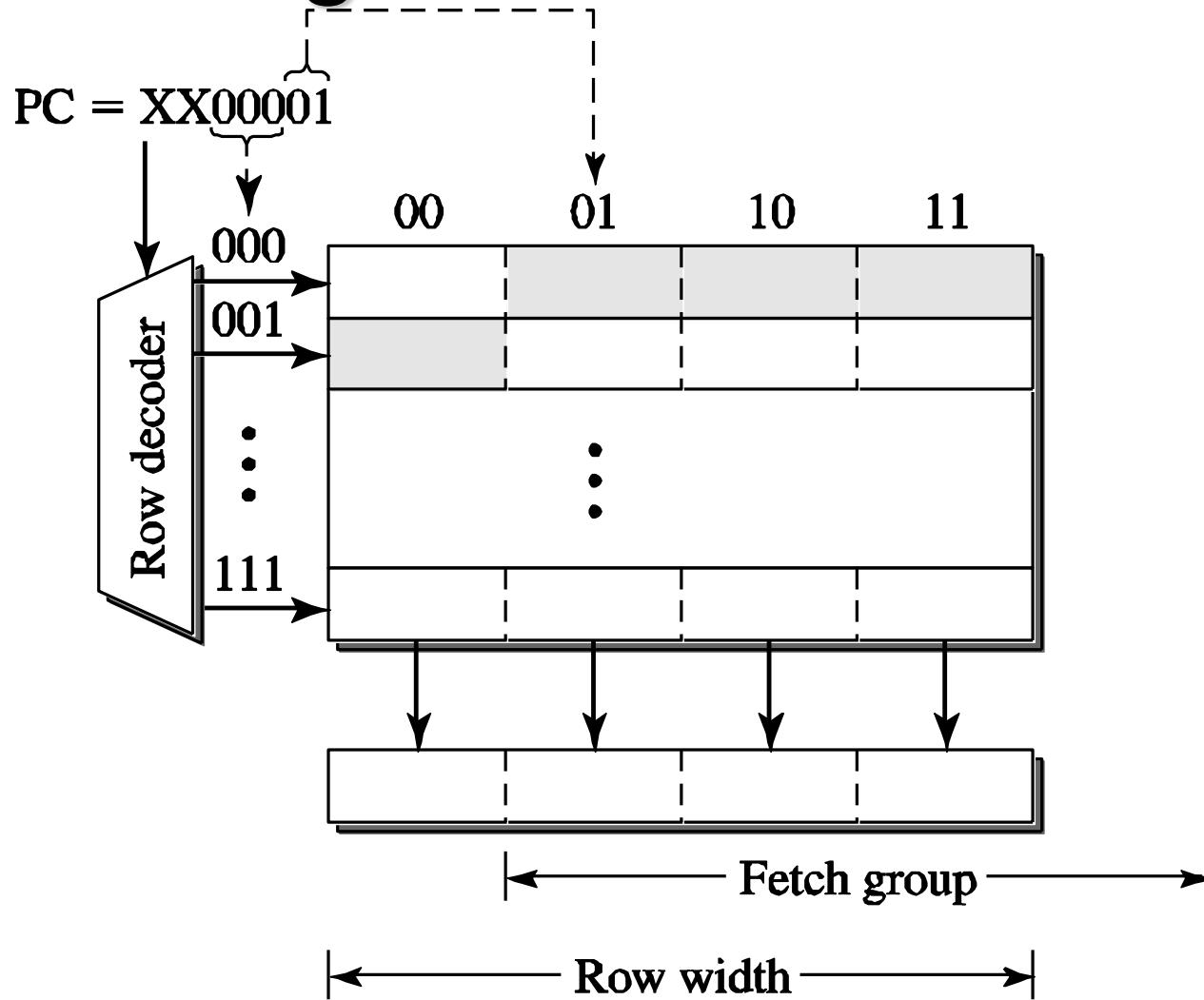
- Objective: Fetch multiple instructions per cycle
- Challenges:
 - Branches: control dependences
 - Branch target misalignment
 - Instruction cache misses
- Solutions
 - Code alignment (static vs.dynamic)
 - Prediction/speculation



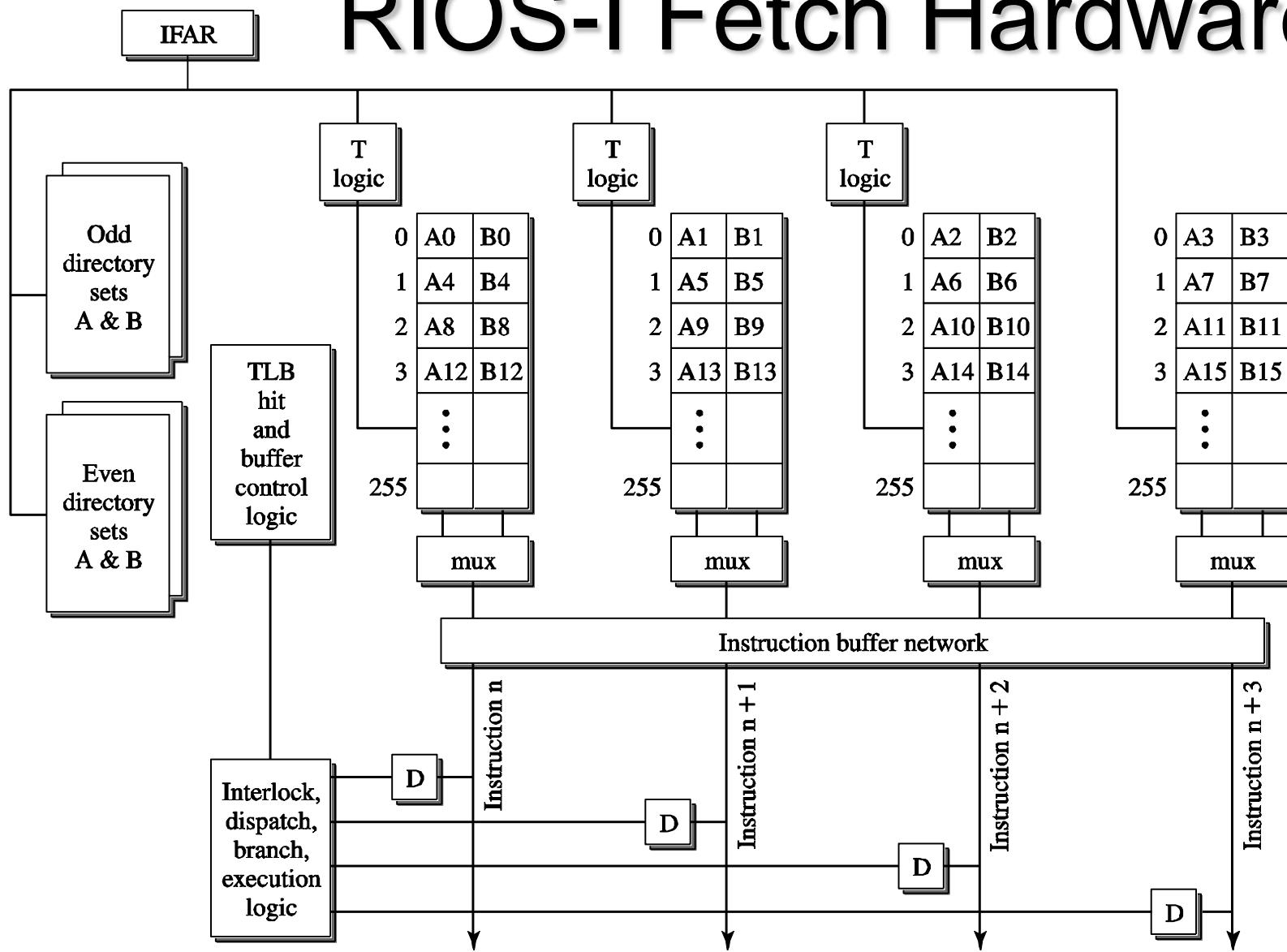
I-Cache Organization



Fetch Alignment



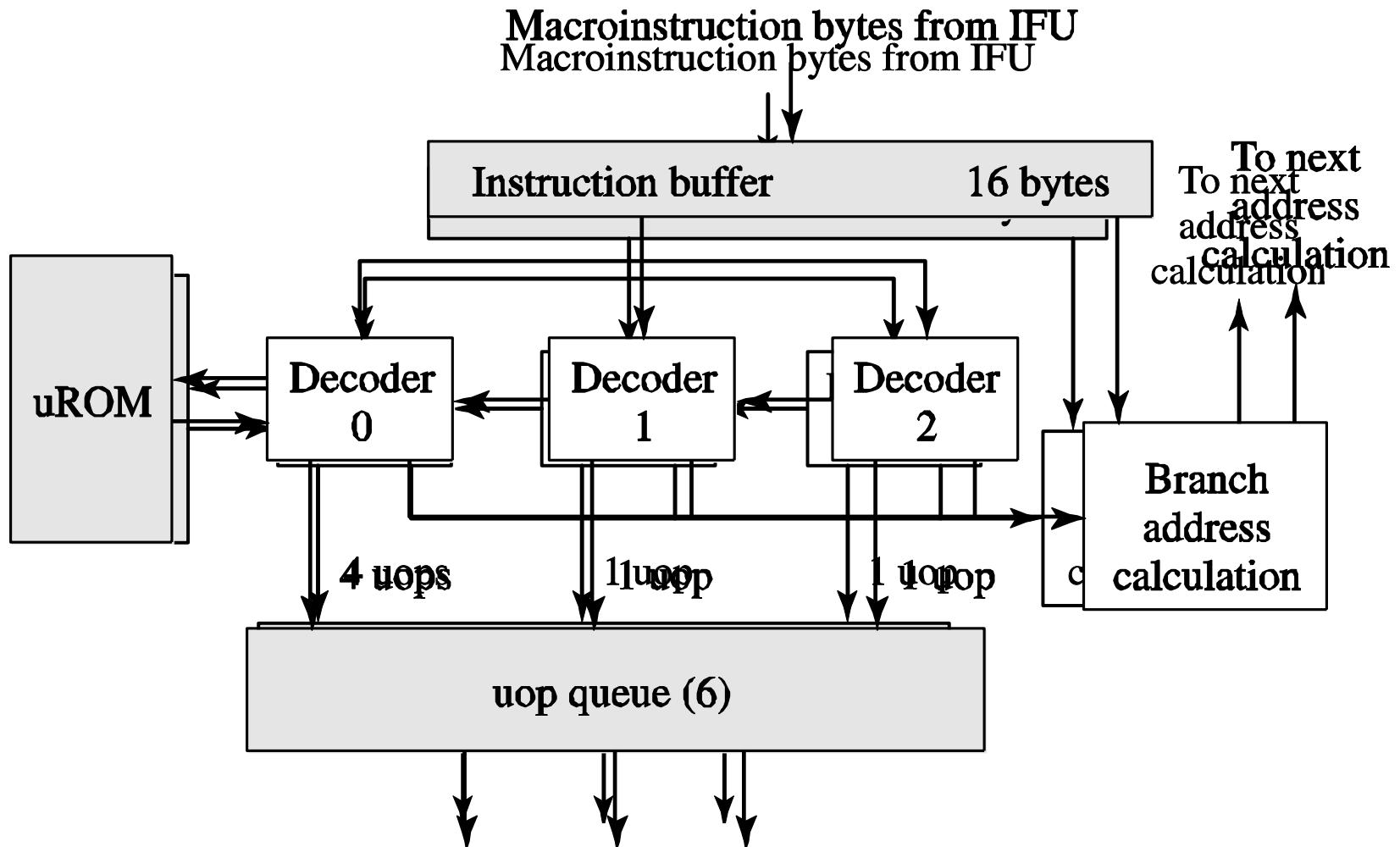
RIOS-I Fetch Hardware



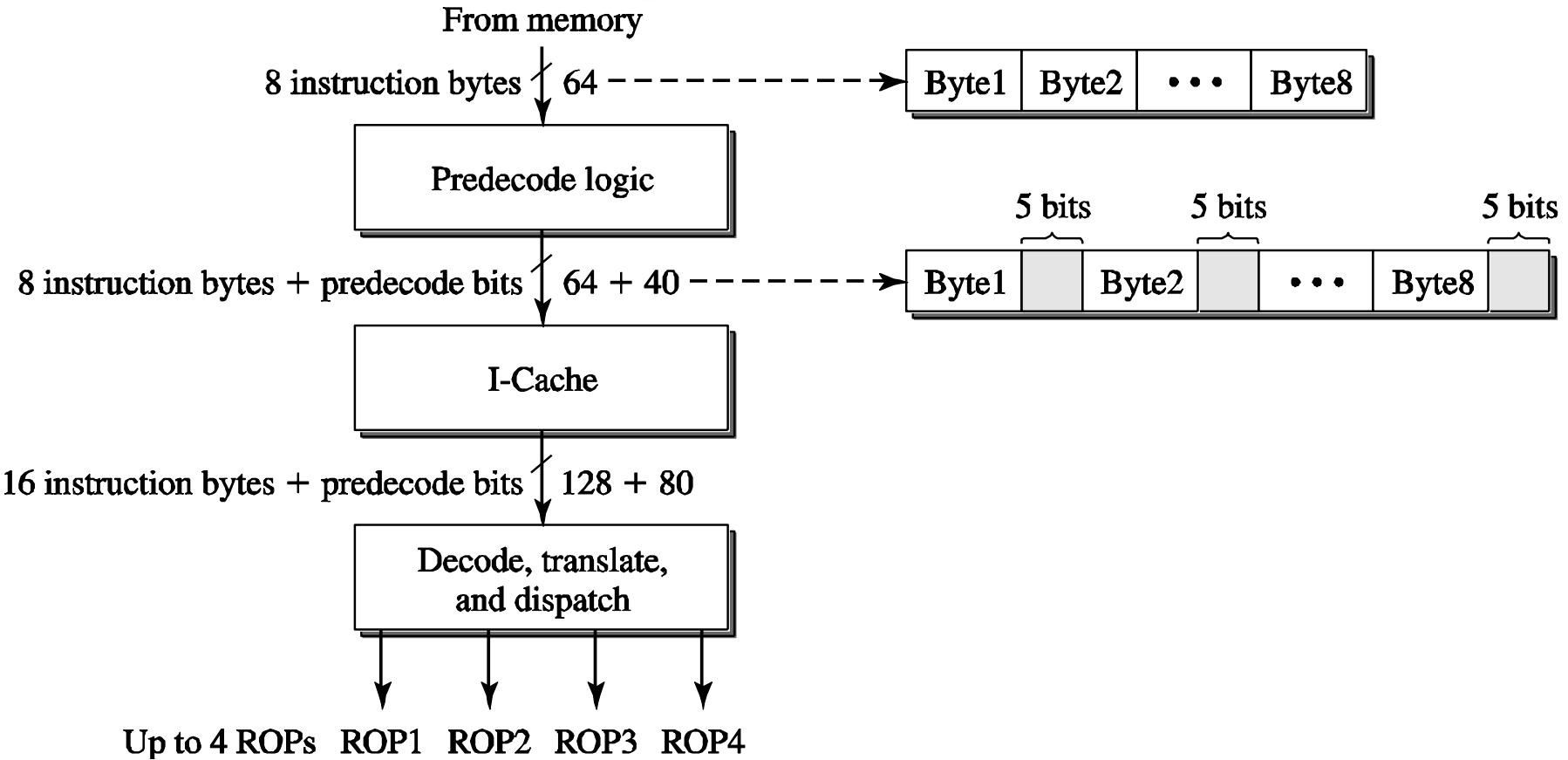
Issues in Decoding

- Primary Tasks
 - Identify individual instructions (!)
 - Determine instruction types
 - Determine dependences between instructions
- Two important factors
 - Instruction set architecture
 - Pipeline width

Pentium Pro Fetch/Decode



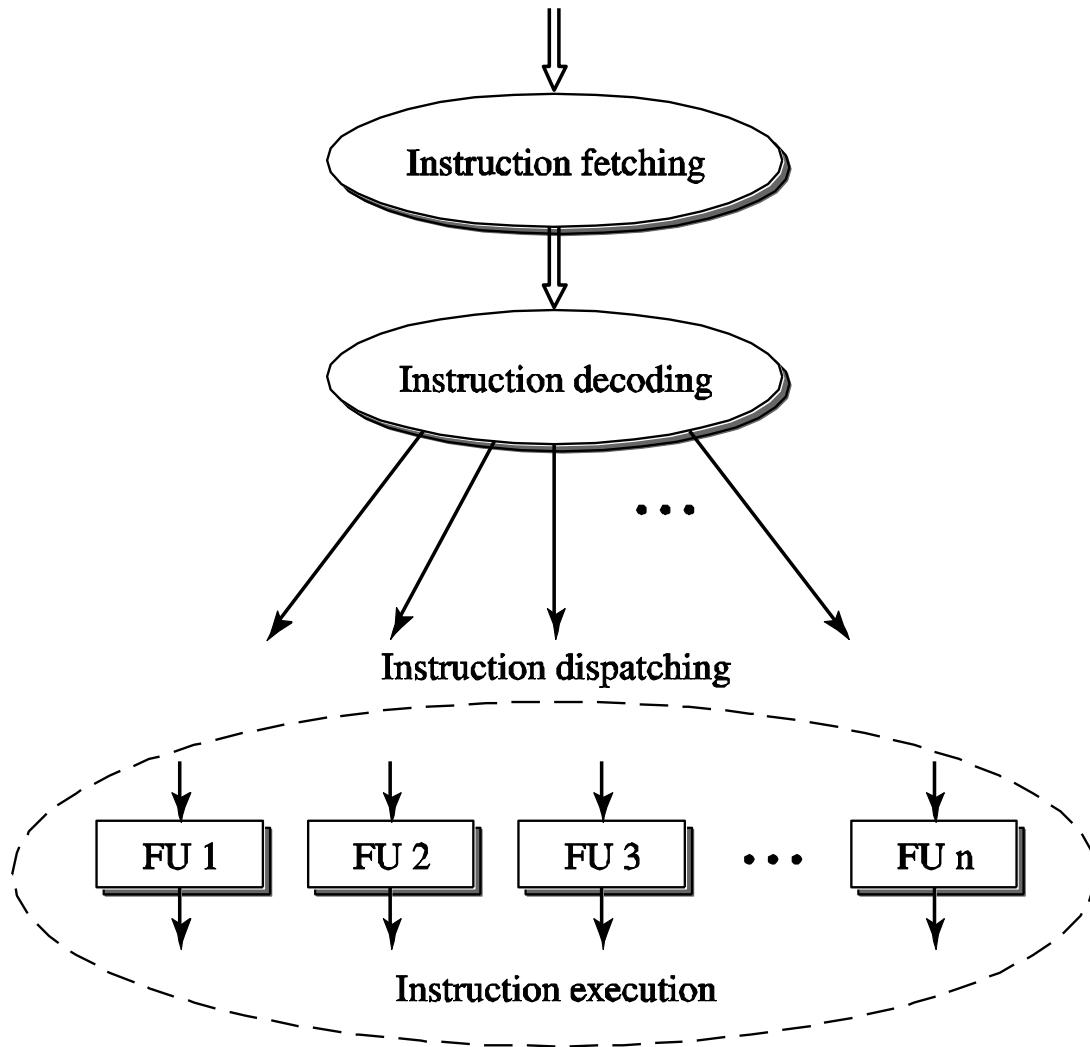
Predecoding in the AMD K5



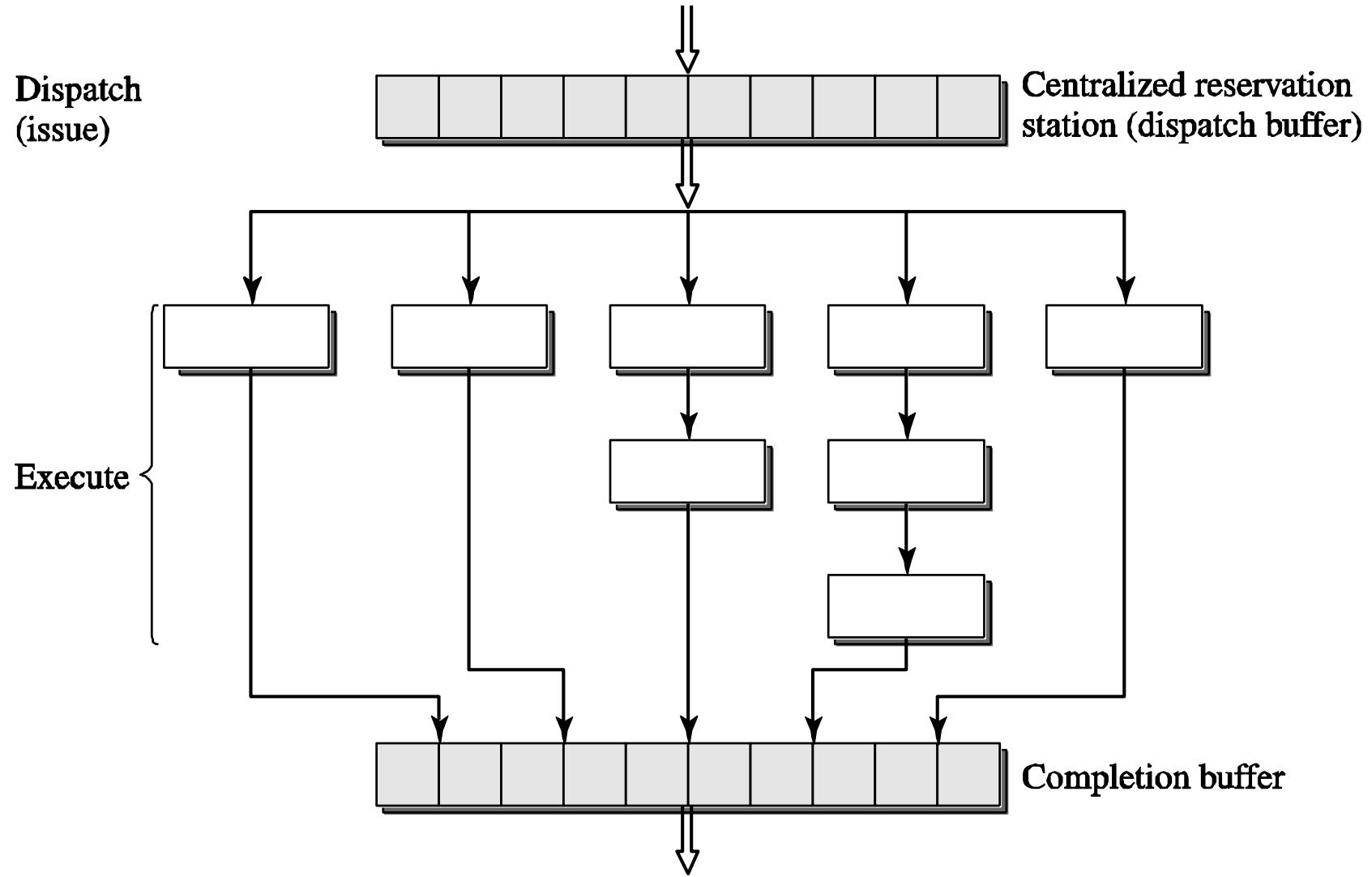
Instruction Dispatch and Issue

- Parallel pipeline
 - Centralized instruction fetch
 - Centralized instruction decode
- Diversified pipeline
 - Distributed instruction execution

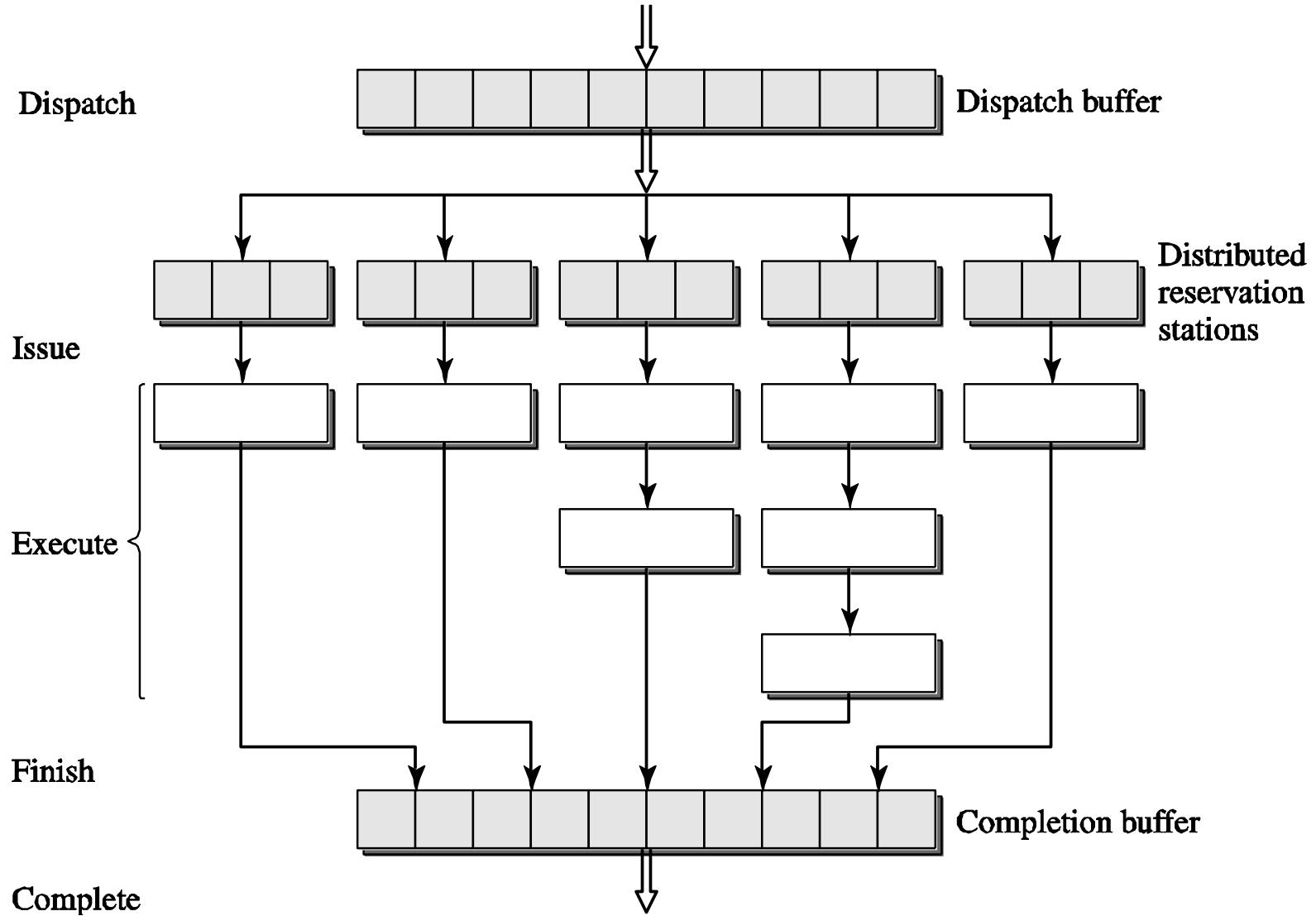
Necessity of Instruction Dispatch



Centralized Reservation Station



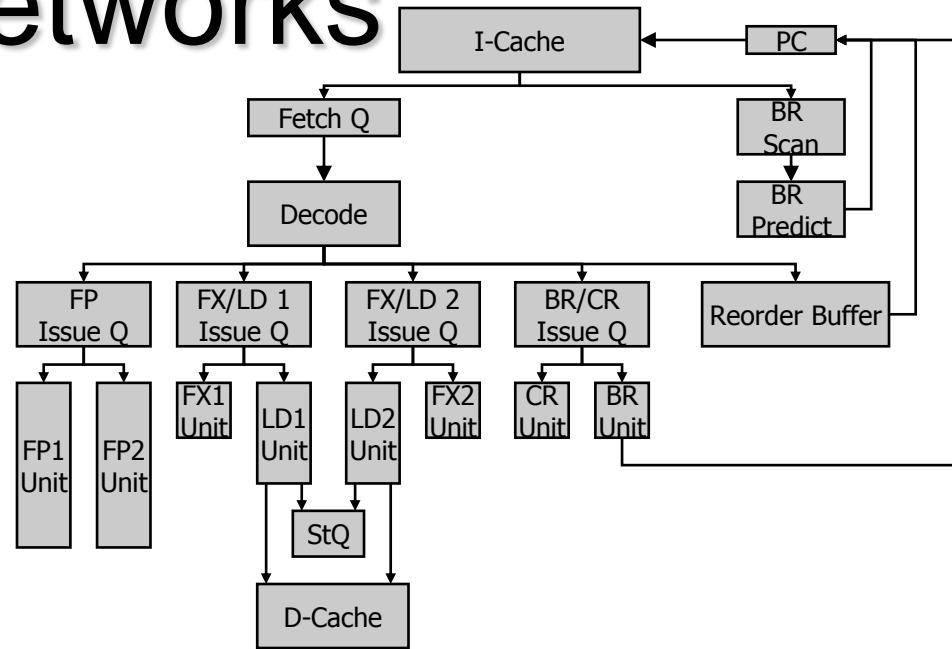
Distributed Reservation Station



Issues in Instruction Execution

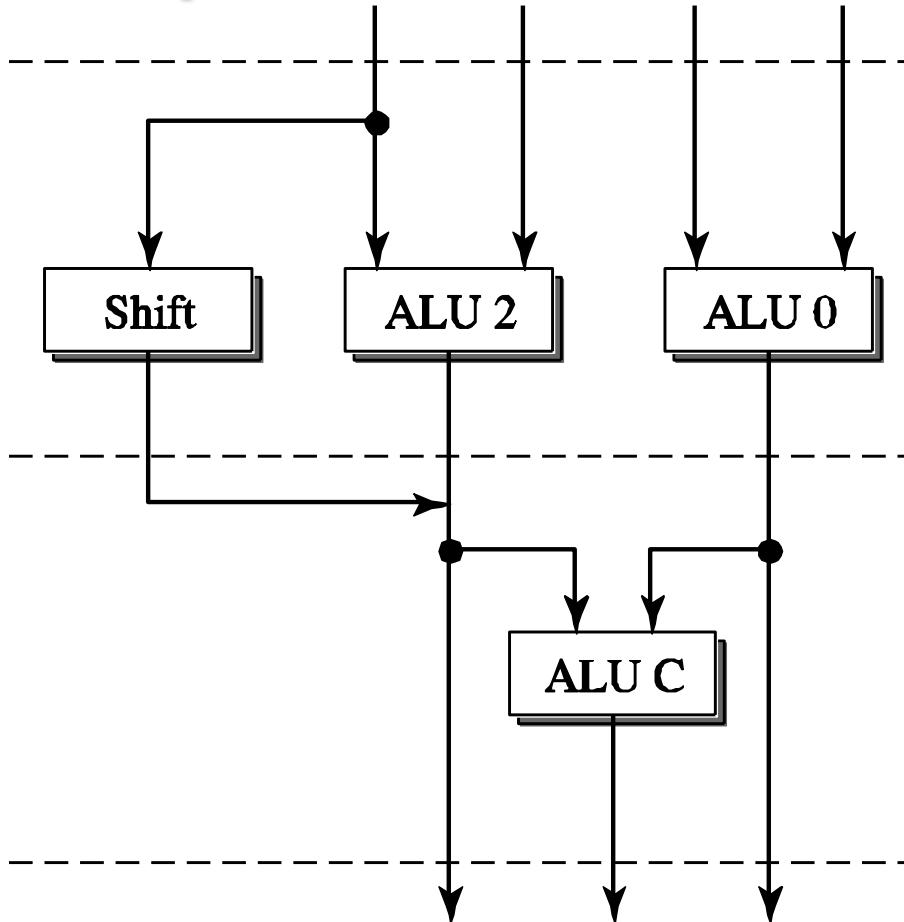
- Current trends
 - More parallelism ← bypassing very challenging
 - Deeper pipelines
 - More diversity
- Functional unit types
 - Integer
 - Floating point
 - Load/store ← most difficult to make parallel
 - Branch
 - Specialized units (media)

Bypass Networks

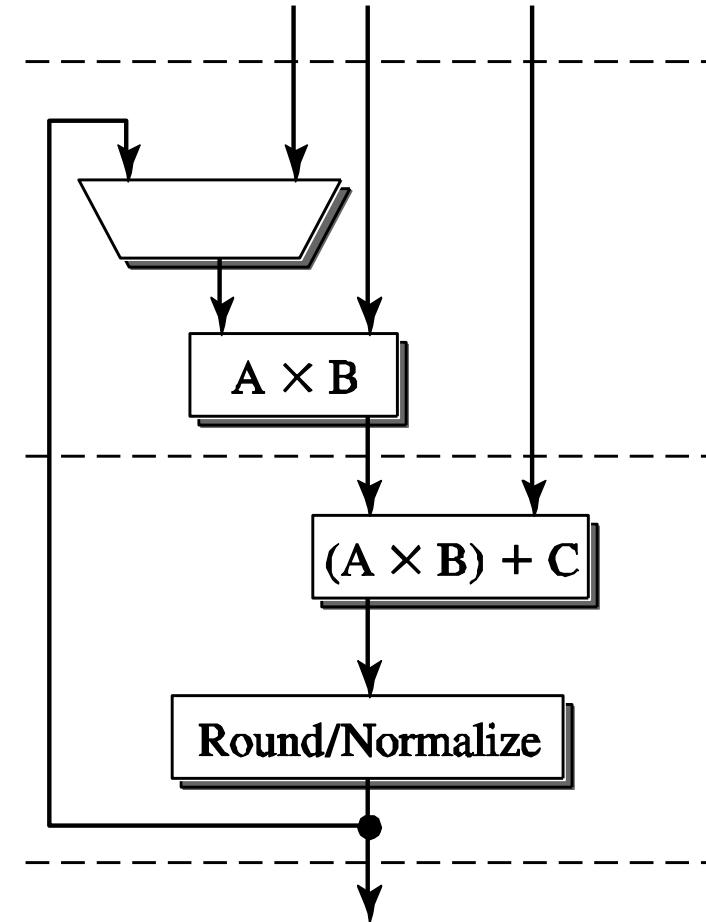


- $O(n^2)$ interconnect from/to FU inputs and outputs
- Associative tag-match to find operands
- Solutions (hurt IPC, help cycle time)
 - Use RF only (Power4) with no bypass network
 - Decompose into clusters (21264)

Specialized units



(a)



(b)

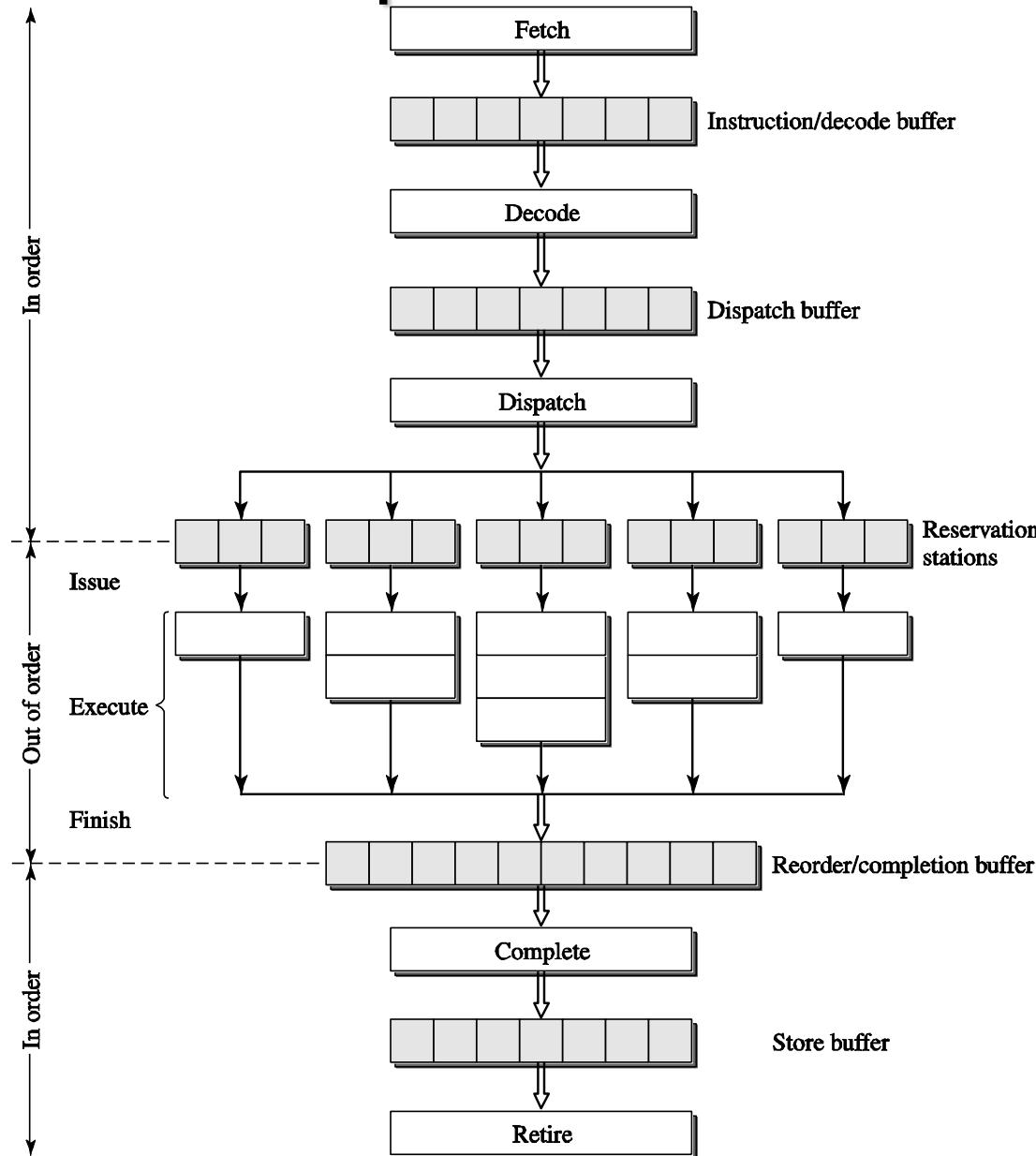
New Instruction Types

- Subword parallel vector extensions
 - Media data (pixels, quantized datum) often 1-2 bytes
 - Several operands packed in single 32/64b register
 - {a,b,c,d} and {e,f,g,h} stored in two 32b registers
 - Vector instructions operate on 4/8 operands in parallel
 - New instructions, e.g. motion estimation
$$me = |a - e| + |b - f| + |c - g| + |d - h|$$
- Substantial throughput improvement
 - Usually requires hand-coding of critical loops

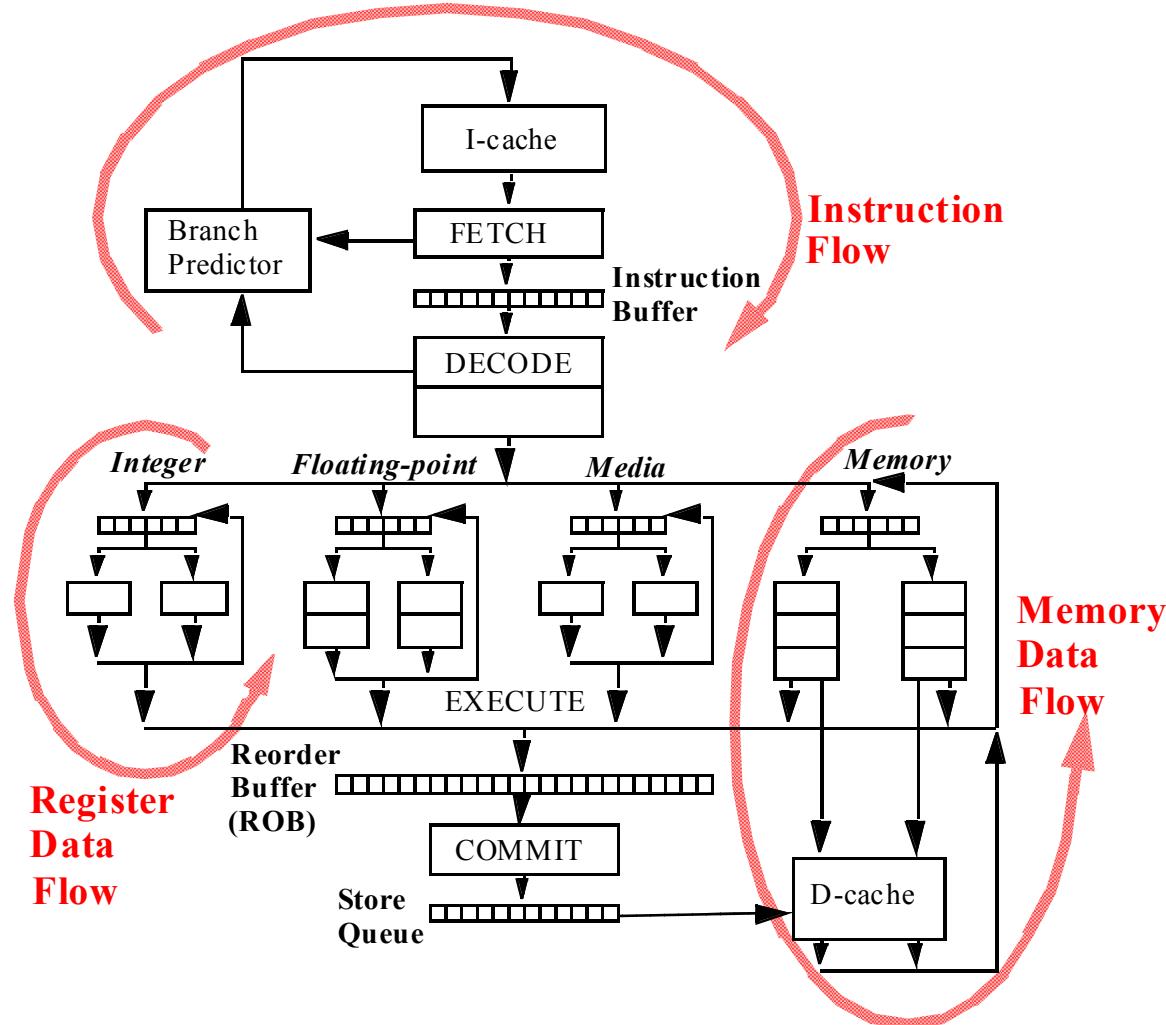
Issues in Completion/Retirement

- Out-of-order execution
 - ALU instructions
 - Load/store instructions
- In-order completion/retirement
 - Precise exceptions
 - Memory coherence and consistency
- Solutions
 - Reorder buffer
 - Store buffer
 - Load queue snooping (later)

A Dynamic Superscalar Processor



Impediments to High IPC



Superscalar Summary

- Instruction flow
 - Branches, jumps, calls: predict target, direction
 - Fetch alignment
 - Instruction cache misses
- Register data flow
 - Register renaming: RAW/WAR/WAW
- Memory data flow
 - In-order stores: WAR/WAW
 - Store queue: RAW
 - Data cache misses