Processing Infrastructures for Active Networks

ECE 697J

November 5th, 2002

Introduction

- We have completed software issues
  - Code deployment
  - Processing environments
  - Safety and security
  - Resource reservation
  - Etc.
- Processing characteristics:
  - Can be complex
  - Should support high link speeds
- What kind of hardware is necessary?
  - Workstation CPUs unsuitable
  - “Network processors” are designed for programmable networks

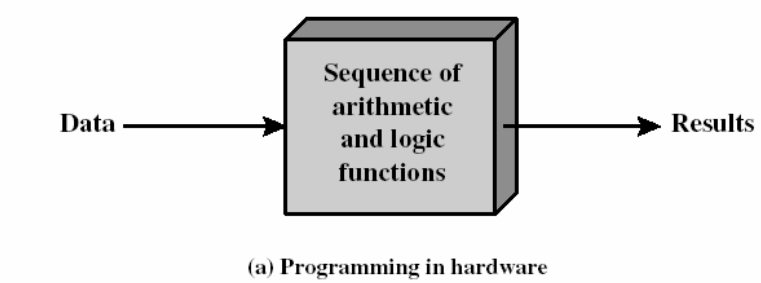
Discussion #1

- Describe the characteristics of processing packets
- Contrast them to processing on workstations

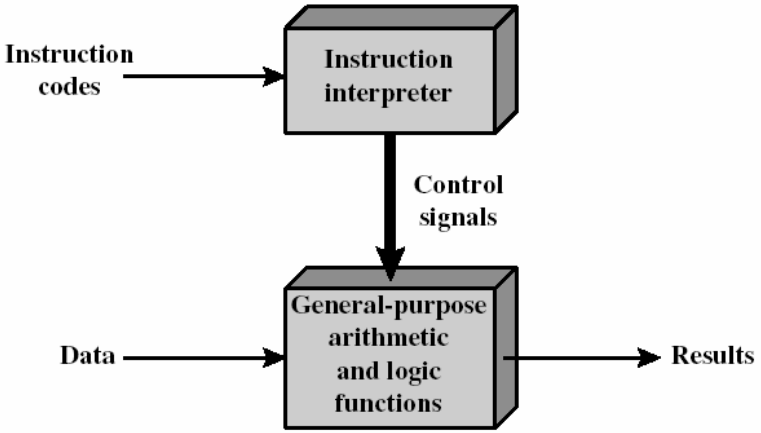
Demands on NPs

- Why is processing packets different from “normal” programs?
  - I/O centric vs. processing centric
  - Real-time vs. best-effort
  - Many simple tasks vs. few complex tasks
  - State: per-flow vs. per program
  - Limited resources (area, power) vs. high-end
- Impacts
  - Processor architecture
  - System architecture
  - Operating system
- What does a workstation system / processor look like?

General-Purpose Processing



(a) Programming in hardware



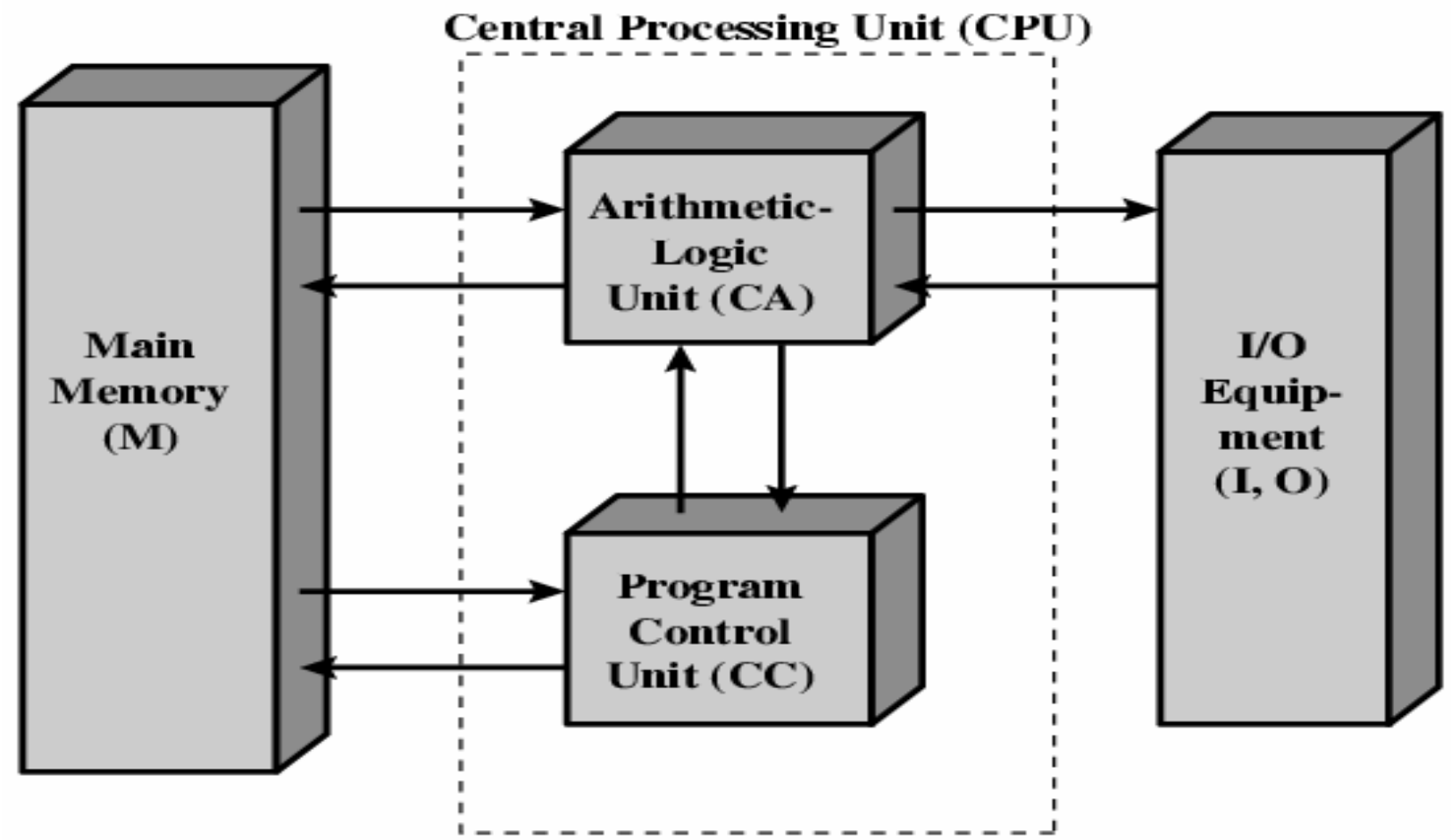
(b) Programming in software

All figures from Stallings: "Computer Organiztion & Architecture" 6th edition

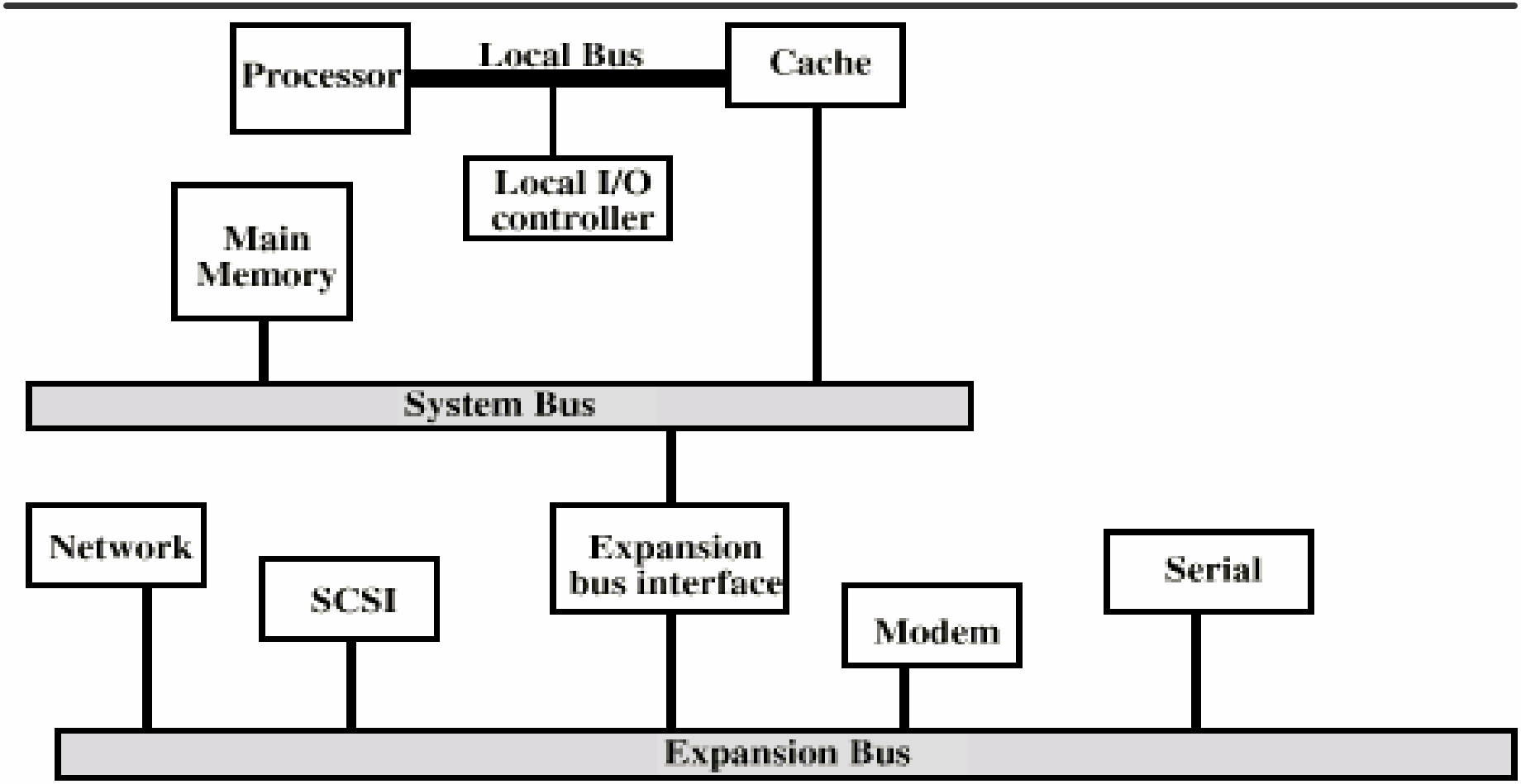
Software

- User program
  - Usually written in higher-level language (C)
  - Machine-independent
- Compiled to object code by compiler
  - Program expressed in “instruction set” of machine
  - Architecture dependent
  - “assembly” is human-readable version of it
- Binary code is executed on hardware
- Operating System supervises execution

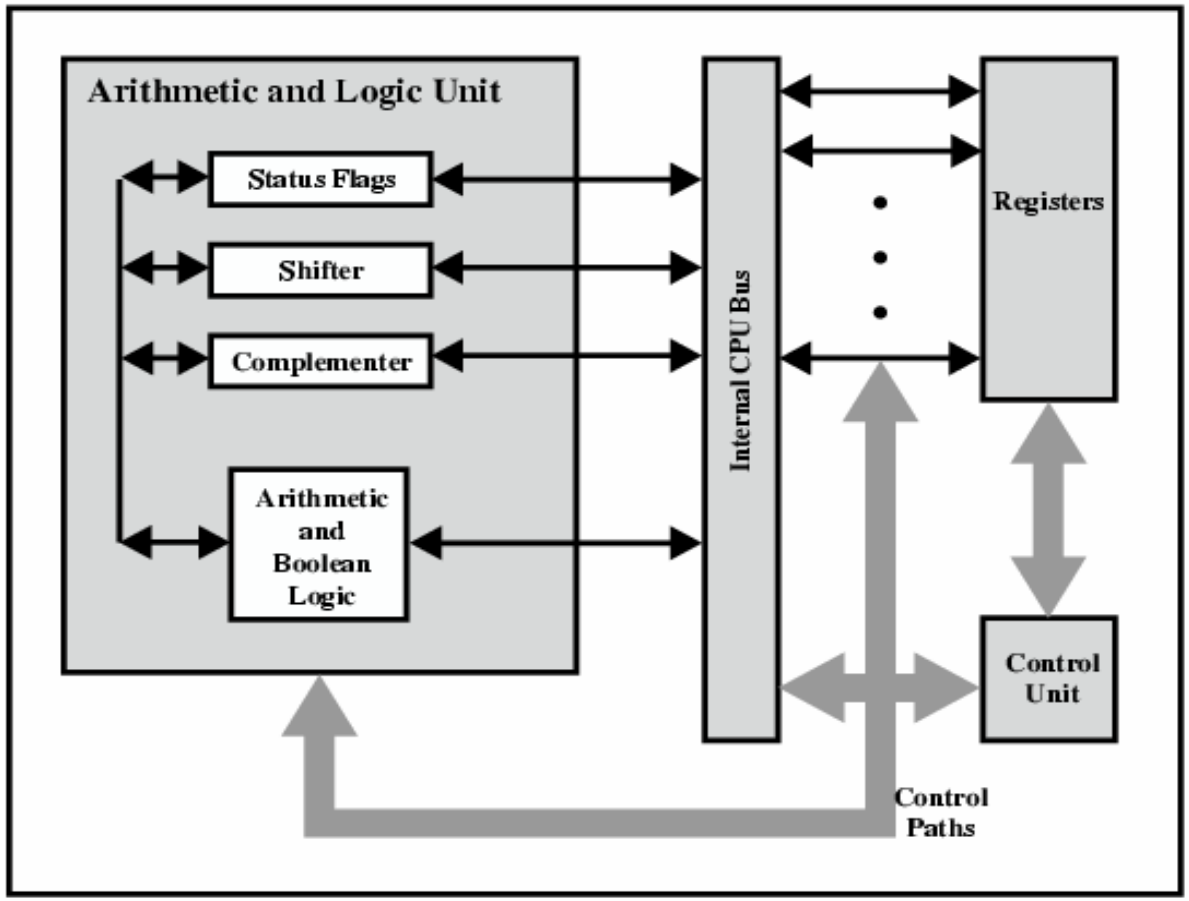
Conceptual Processor



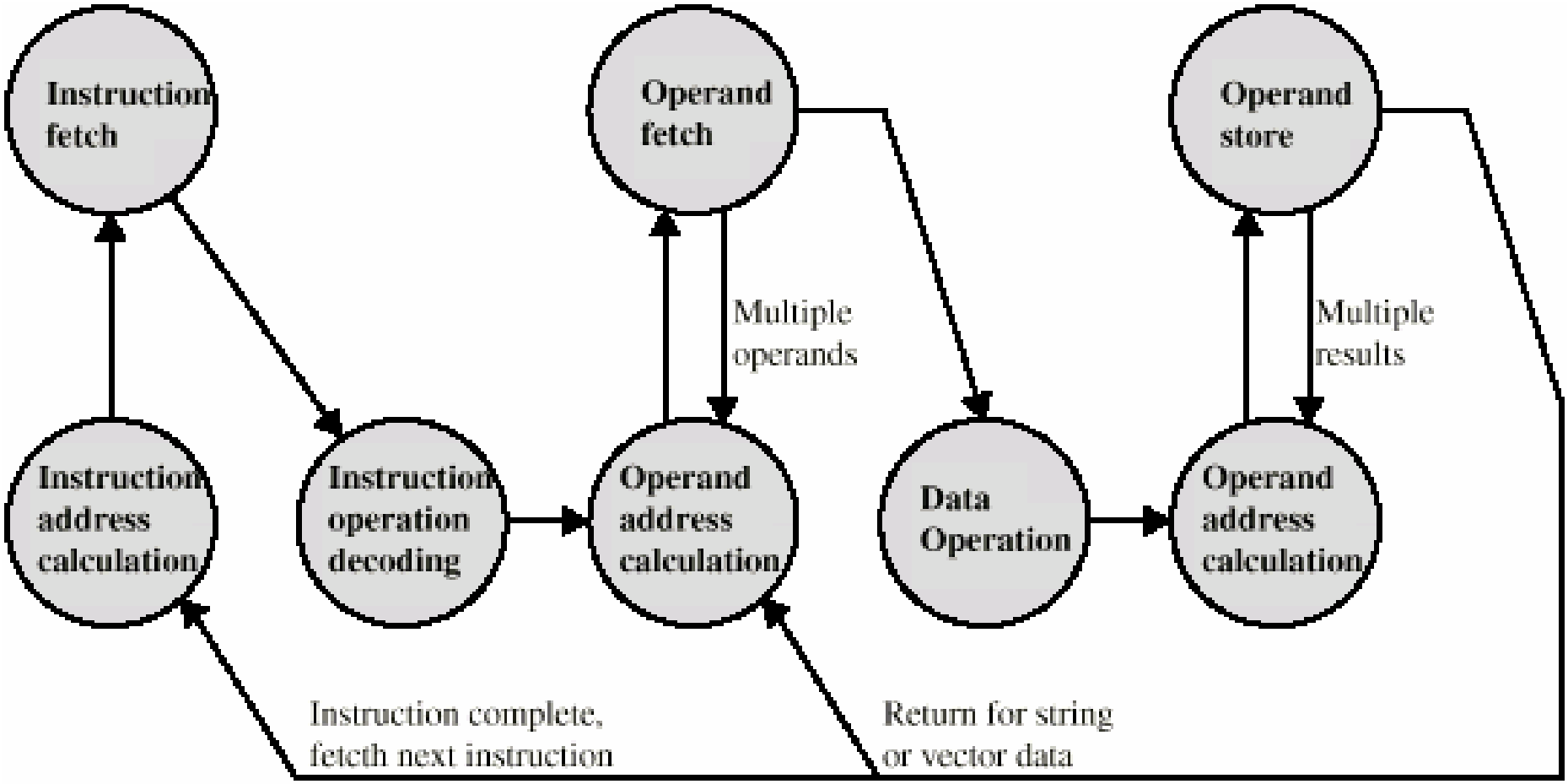
Workstation System Architecture



Processor Architecture



Basic Processing Cycle



Instruction Set

- Instruction Set defines basic operations
  - Data transfer (LOAD, STORE, …)
  - Arithmetic/Logic (ADD, MUL, OR, …)
  - Flow control (JUMP, BRANCH, …)
  - Floating point (ADD, MUL, DIV, …)
- ISA impacts performance
  - CISC vs. RISC
  - RISC is simpler, faster, needs more complex compiler
- What are the characteristics of an RISC instruction set?

CISC vs. RISC

	Complex Instruction Set (CISC)Computer			Reduced Instruction Set (RISC) Computer		Superscalar		
Characteristic	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000	PowerPC	Ultra SPARC	MIPS R10000
Year developed	1973	1978	1989	1987	1991	1993	1996	1996
Number of instructions	208	303	235	69	94	225		
Instruction size (bytes)	2–6	2–57	1–11	4	4	4	4	4
Addressing modes	4	22	11	1	1	2	1	1
Number of general-purpose registers	16	16	8	40 - 520	32	32	40 - 520	32
Control memory size (Kbits)	420	480	246	—	—	—	—	—
Cache size (KBytes)	64	64	8	32	128	16-32	32	64

Discussion #2

- How can performance of a processor be improved?
- How does this depend on
  - Technology?
  - Program characteristics?
  - Use of coprocessors?
  - Other conditions (e.g., data, O/S, compiler, …)

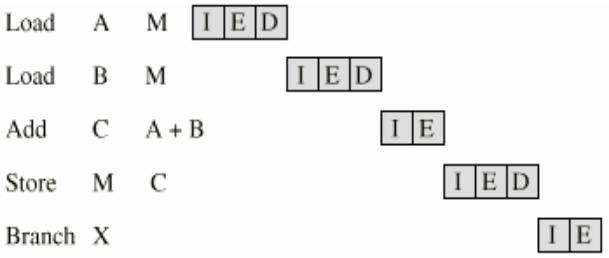
Exploiting parallelism

- We can get lower delay by doing things in parallel
- What levels of parallelism are there?

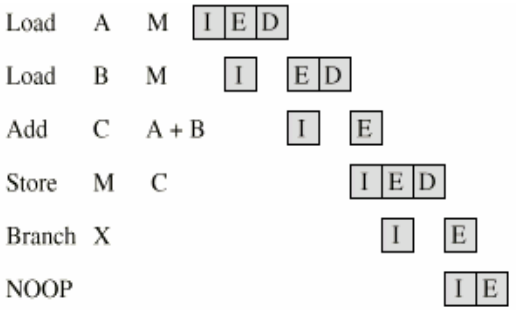
Exploiting parallelism

- We can get lower delay by doing things in parallel
- What levels of parallelism are there?
- Sub-instruction-level parallelism
  - Pipelining
- Instruction-level parallelism
  - Very large instruction word processors (VLIW)
  - Superscalar
- Thread-level parallelism
  - Multi-threading
  - Simultaneous multi-threading (SMT)
- Processor-level parallelism
  - Chip multiprocessors (CMP)

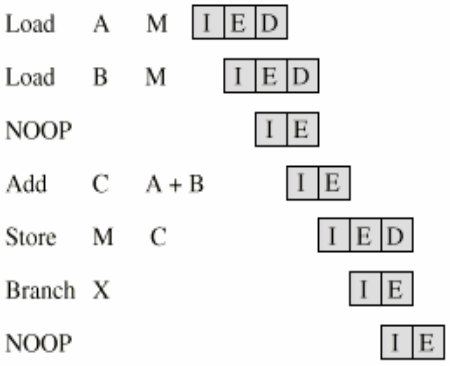
Pipelining



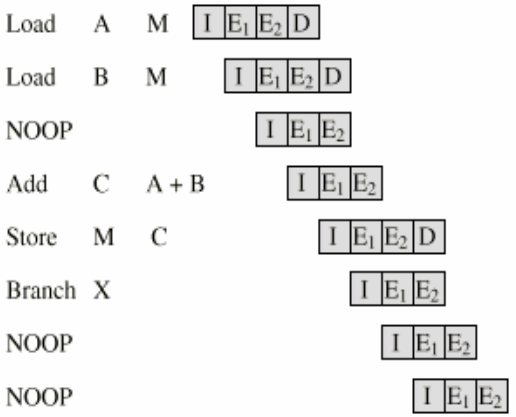
(a) Sequential execution



(b) Two-way pipelined timing



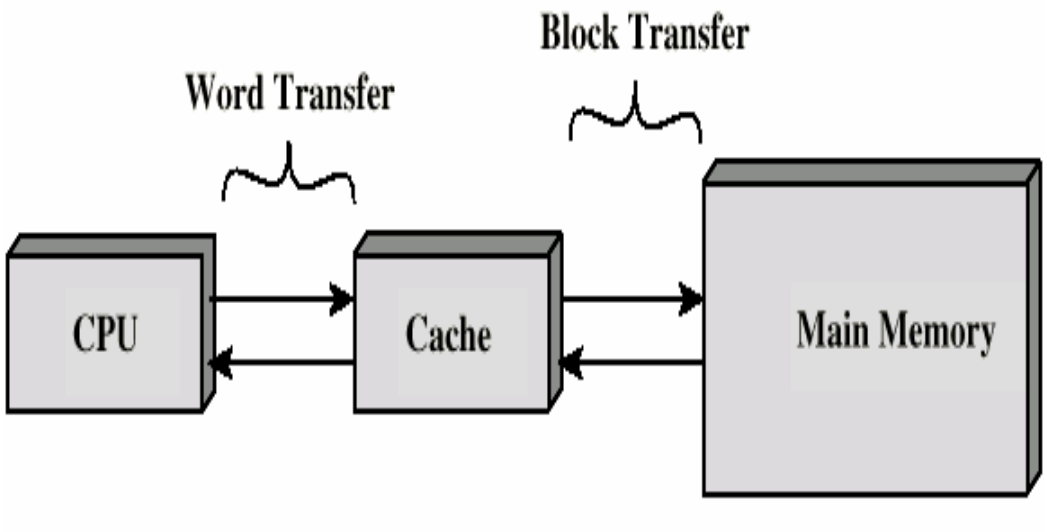
(c) Three-way pipelined timing



(d) Four-way pipelined timing

Pipeline Stalls

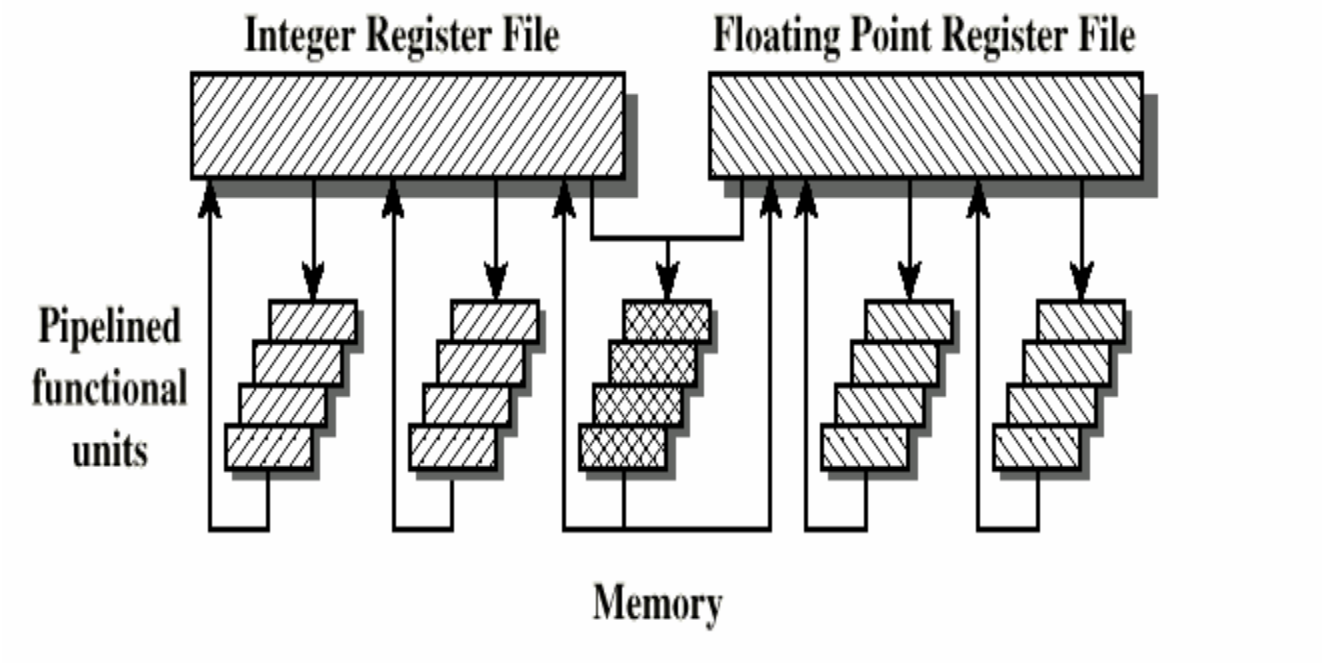
- Limitations of pipeline:
  - Change in control flow (branch misprediction)
  - Data unavailable (cache miss)
- Causes pipeline “stalls”



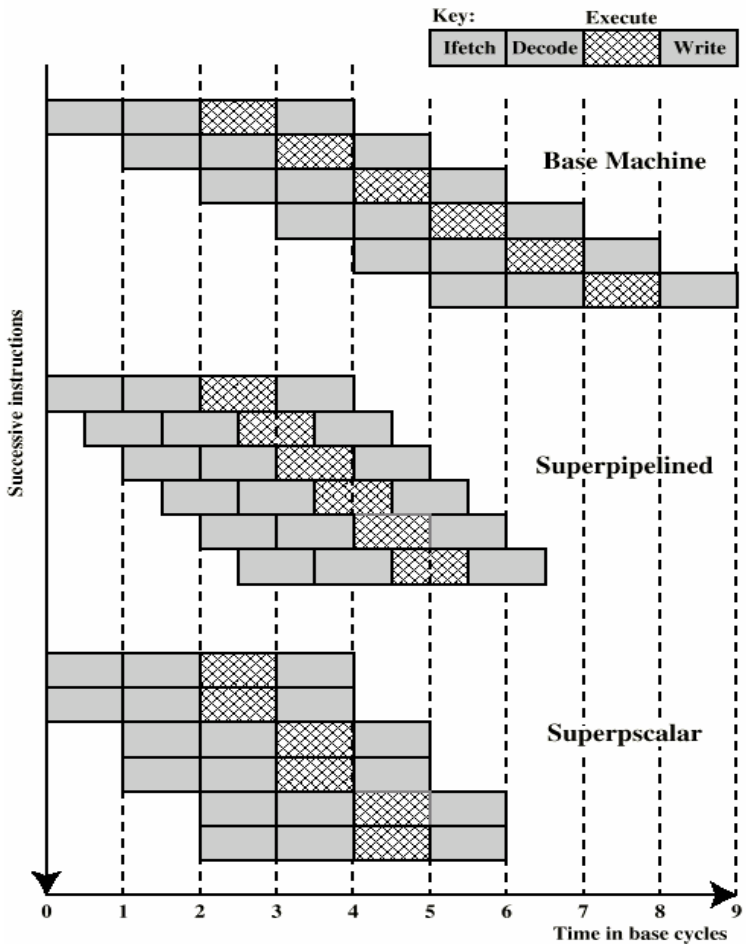
Instruction Level Parallelism

- Multiple functional units for execution on parallel
- Question: how to schedule the instructions to units?
  - Static scheduling: VLIW
  - Dynamic scheduling: Superscalar
- What are the limitations?
  - Dependencies

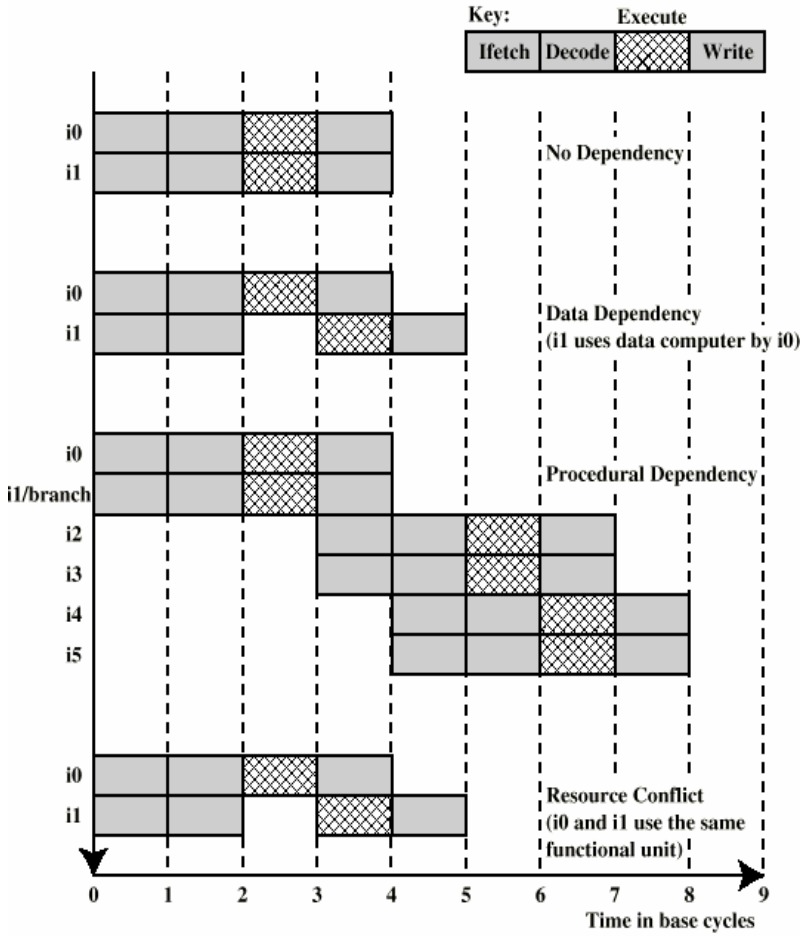
Superscalar



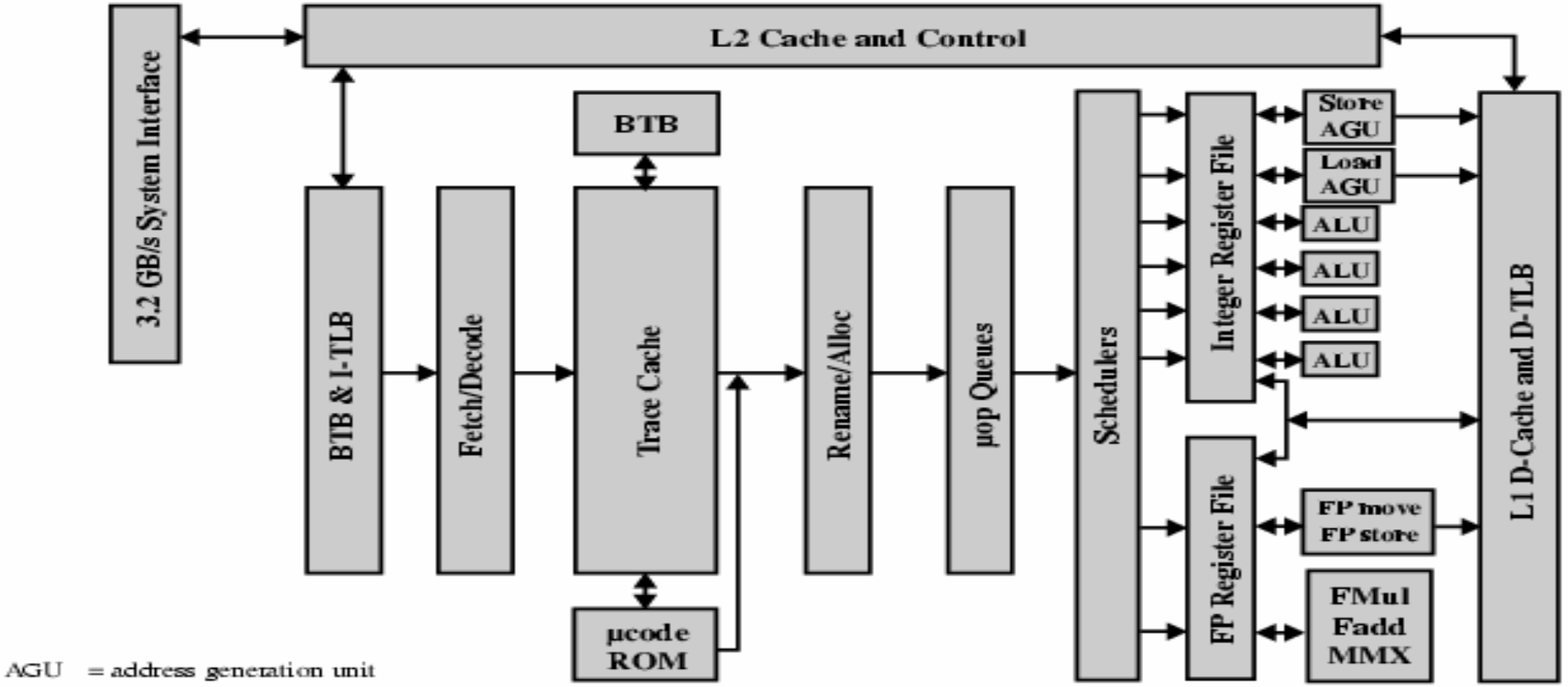
Superscalar vs. Pipelining



Dependencies Reduce Efficiency



P4



AGU = address generation unit
BTB = branch target buffer
D-TLB = data translation lookaside buffer
I-TLB = instruction translation lookaside buffer

Thread-level parallelism

- Memory stalls, branch mispredictions cause pipeline stalls
- Processor can support multiple “threads”
  - Switch between threads when one is stalled
- One thread at a time: multi-threading
- Multiple threads: SMT
- Cost
  - Additional registers
  - Context switching logic
  - Possible cache pollution

Chip Multiprocessors

- Embedded technology makes embedded systems possible:
  - Multiple processor cores
  - Memory
  - I/O
- “System-on-a-Chip” (SOC) can have multiple processors
  - On-chip interconnect for efficient communication

Discussion #3

- Which architecture features are useful for network processors?

Next Lecture

- Look at technology trends
  - Networking demands
  - CMOS technology improvements
- Impact on SOC architectures
- Why it makes sense to build network processors