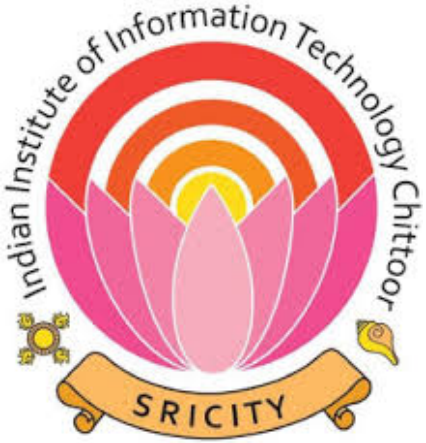
CO: Computer Organization

Day 1

Indian Institute of Information Technology, Sri City

Jan - May - 2018



Computer Organization Syllabus

- Introduction
- Number Systems
- Arithmetic Operations on Integers
- Operations on Floating Point Numbers
- ISA
- MIPS ISA
- Assembly Programs

Mid Sem I

Computer Organization Syllabus

- Memory System
  - Memory Hierarchy
  - Cache Memories
  - RAM Memories
  - Disks
  - Virtual Memories
- Pipelining and Hazards
  - Basics
  - Data Hazards
  - Structural Hazards
  - Control Hazards
- Measuring Performance

Mid Sem II

Computer Organization Syllabus

- Superscalar Processor Architecture
- Simultaneous multi-threading
- Multi-core processors

End Semester Exam

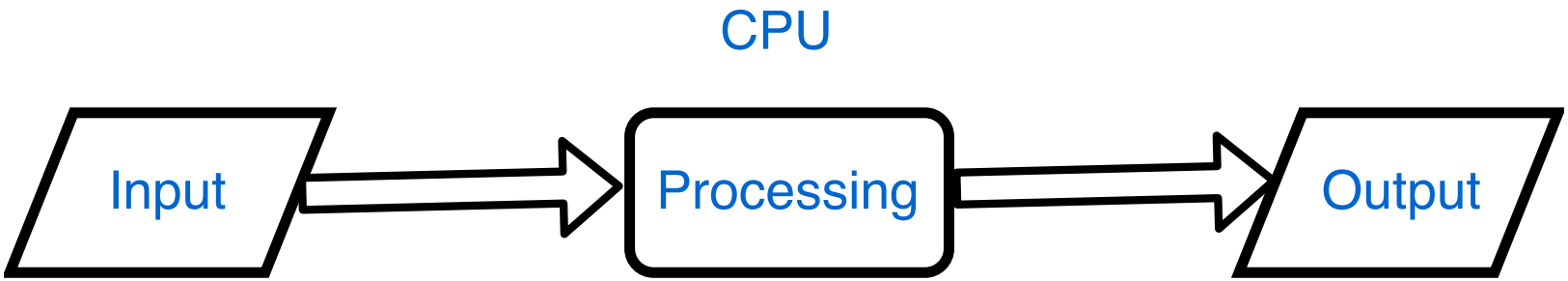
Reference Books:

1. Computer Organization, by C. Hamacher, V. Zvonko, S. Zaky, McGraw Hill Publishing Company
2. Structured Computer Organization by Andrew Tanenbaum, Prentice Hall of India
3. Computer Organization and Architecture: Design for performance, by William Stallings, Prentice Hall of India
4. Computer Architecture and Organization by John Hayes, McGraw-Hil Publishing Company

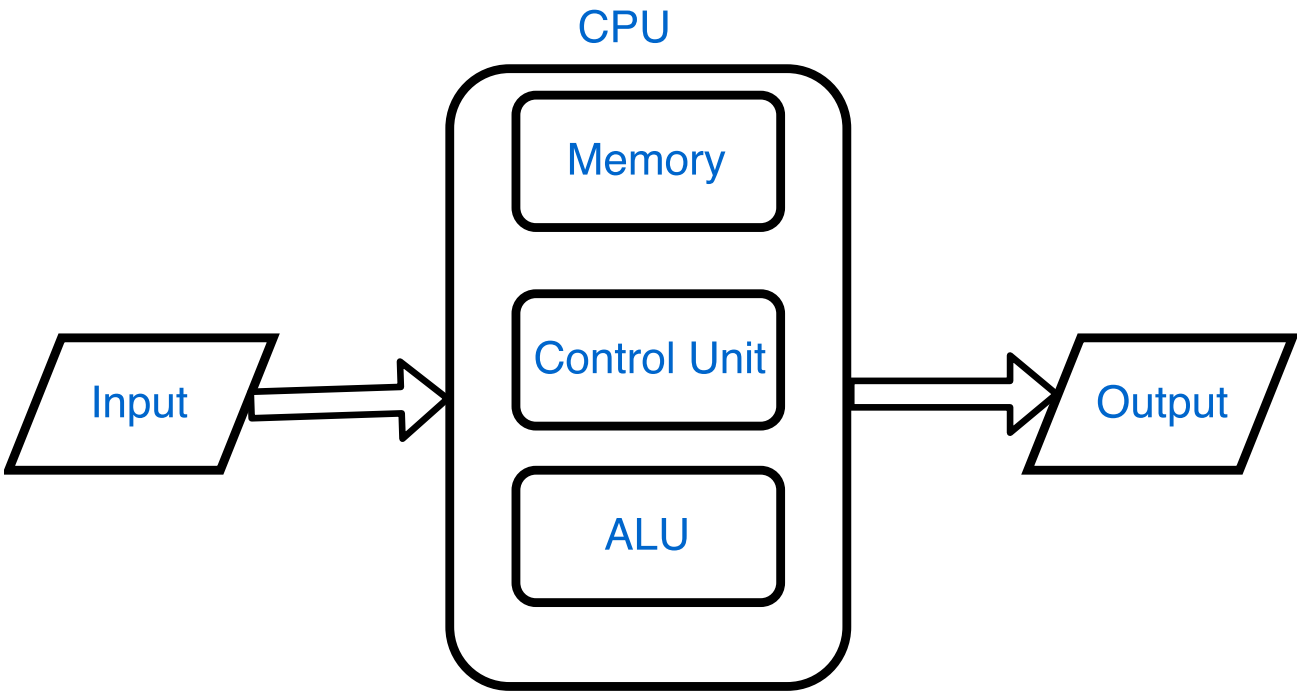
Evaluation

- Short Assignments - 5 X 4 = 20 Marks
- Mid Sem Exams - 2 X 20 = 40 Marks
- End Sem Exam - 1 X 40 = 40 Marks

Block Diagram of a Computer



Block Diagram of a Computer



Types of Data

- Numerical Data
  - Integers
  - Reals
- Character Data
  - char
  - varchar
- Signal Data
  - Audio
  - Video
  - Speech
  - Image

Performance

Let $t_1, t_2, t_3 ... t_n$
be time periods to execute 'n' instructions respectively, then
total time to execute (T)= $\sum_{i=1}^{n} t_i$
where t_i = Number of clock cycles $\times$ Clock cycle time

Performance depends on:

- Type of operation
- Type of operands
- Location of operands
- Type of circuit

Goal: Reduce the time/space/power of an execution.

Amdahl's Law

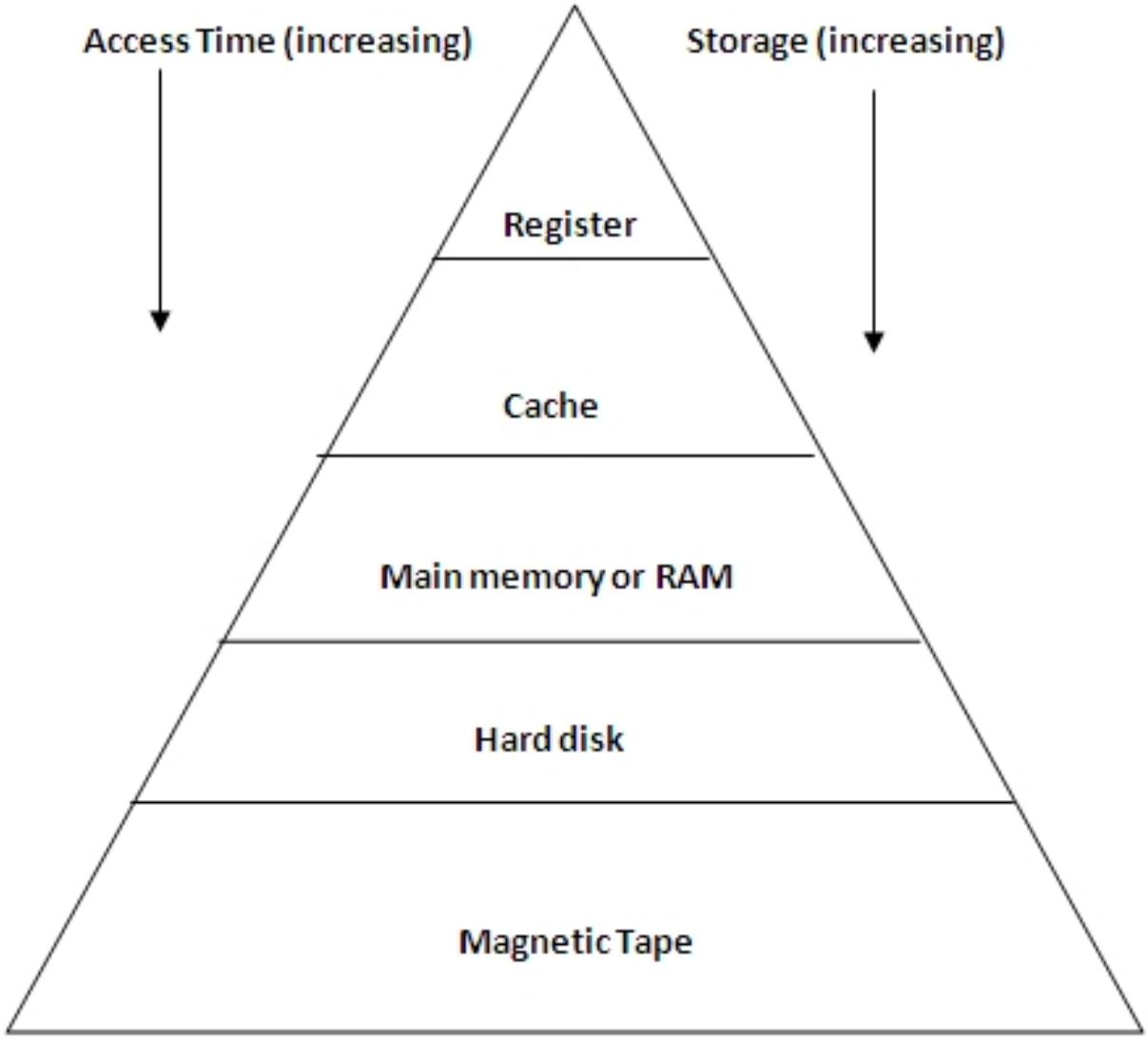
For a fixed workload:
Speed up factor $= \frac{ExecutionTimeBeforeEnhancement}{ExecutionTimeAfterEnhancement}$

Example: Two programs P_1 and P_2 are calculating 15^{th} Fibonacci number, P_1 takes 100 micro seconds and P_2 takes 25 micro seconds. Then what is the speedup of P_2.

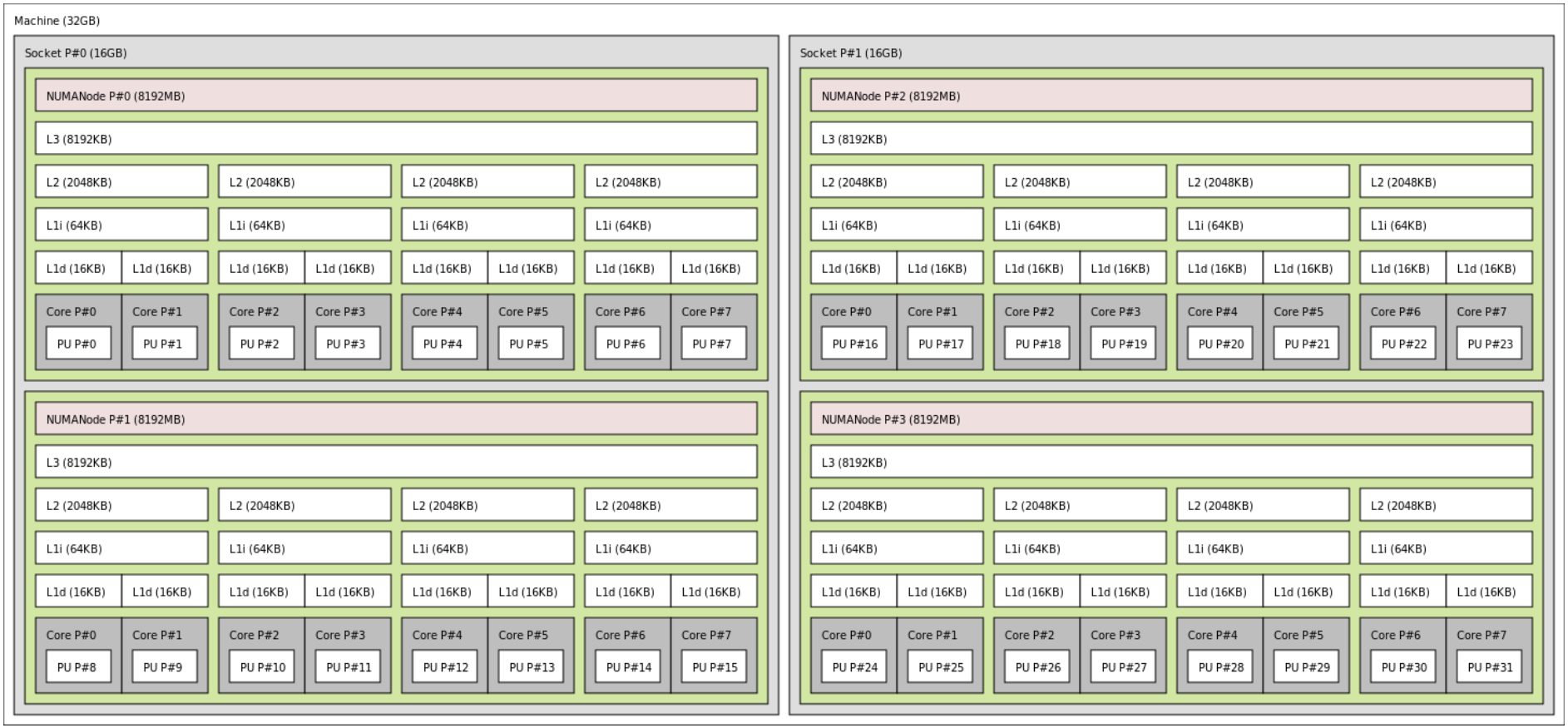
ISA: Instruction Set Architecture

- Instruction Set
  - Add, Sub, Mul, Div, ...
  - AND, OR, ...
- Addressing Modes
  - Register
  - Direct
  - Immediate
- Instruction Formats
  - R-type
  - I-type
  - J-type

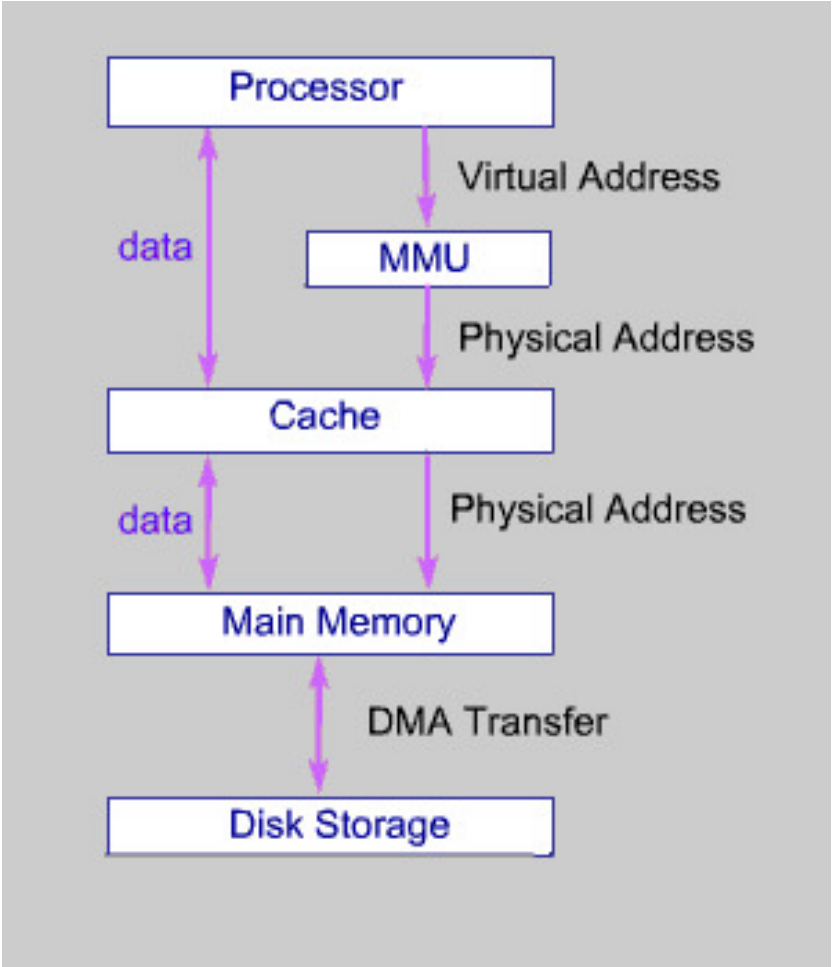
Memory Hierarchy



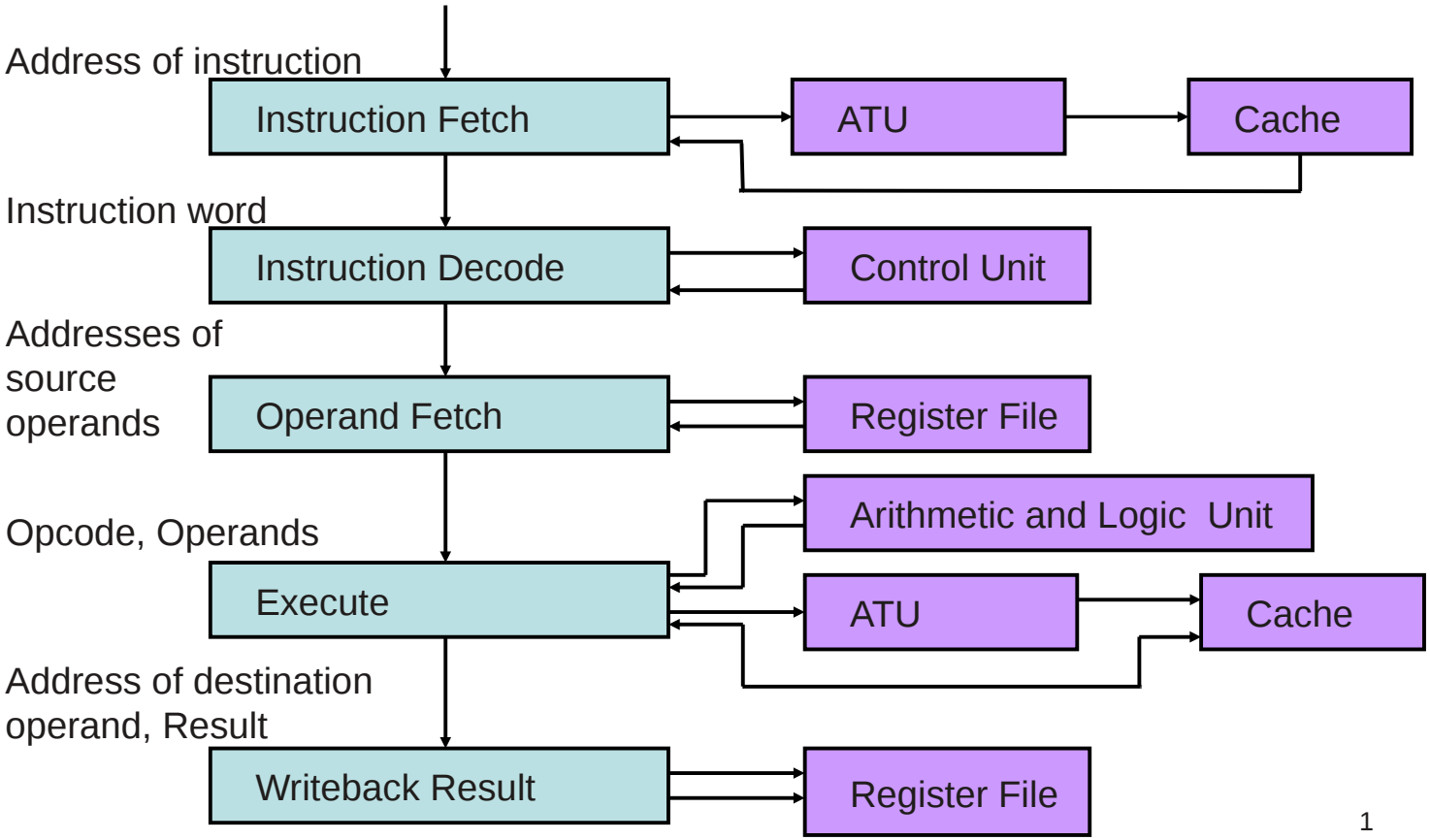
Memory hierarchy of an AMD Bulldozer server



How a Memory Request will be Addressed

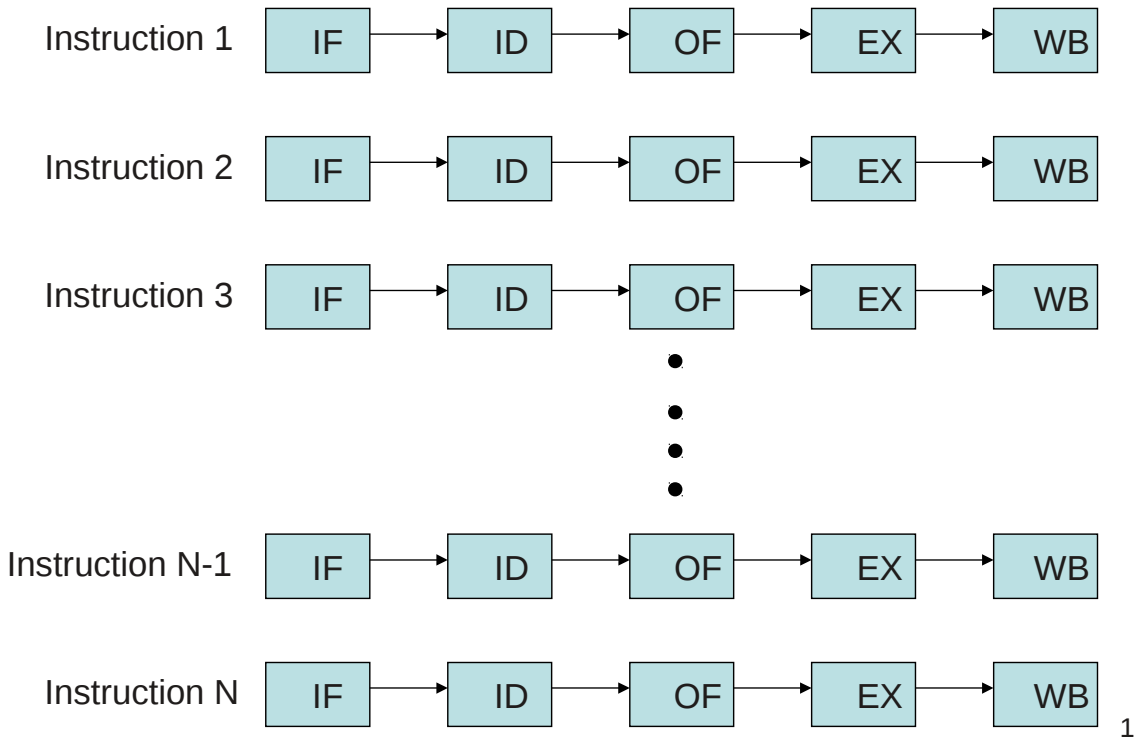


Instruction Cycle in RISC Processors



1

Non-pipelined Execution of Instructions



1

Pipelined Execution of Instructions

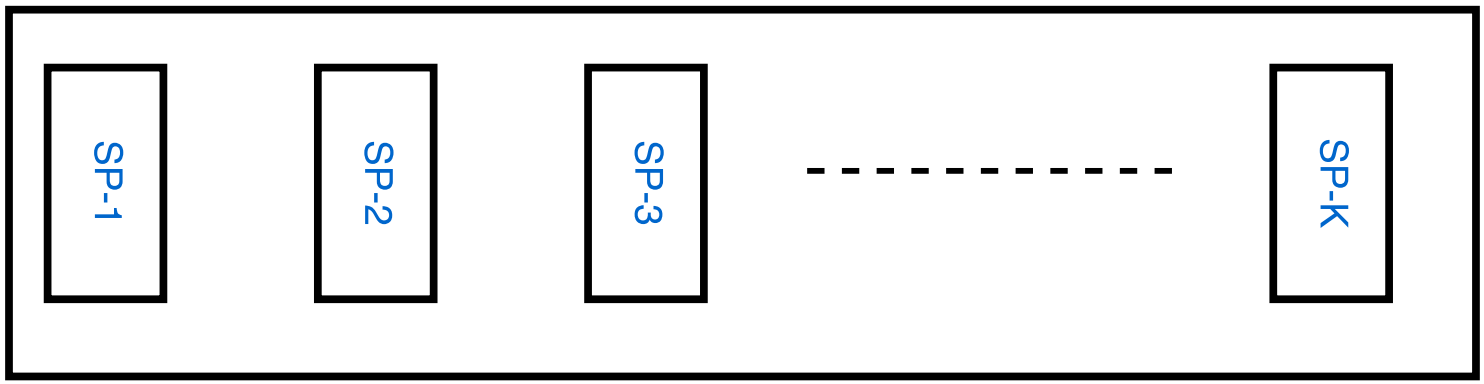
Pipelining is an implementation technique where multiple instruction are overlapped in execution.

Instruction	1	2	3	4	5	6	7	8	9	10
I1	IF	ID	OF	EX	WB					
I2		IF	ID	OF	EX	WB				
I3			IF	ID	OF	EX	WB			
I4				IF	ID	OF	EX	WB		
I5					IF	ID	OF	EX	WB	
I6						IF	ID	OF	EX	WB

- Non-pipelined takes: 5N cycles
- Pipeline takes : 5+N−1 cycles

1

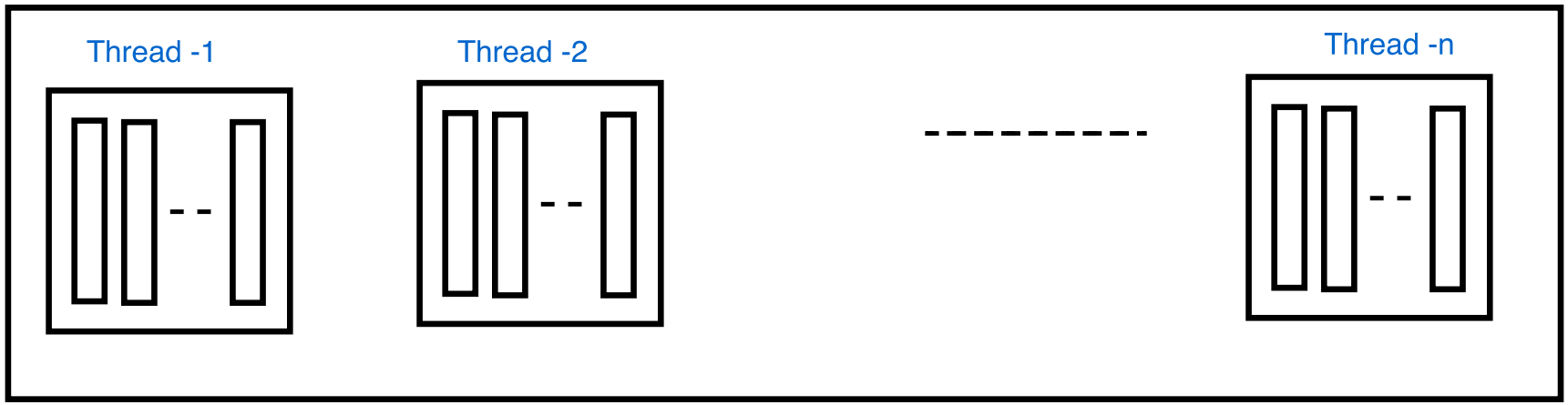
Super Scalar Processor
(Scalar refers to an Instruction)



Super Scalar Processor with 'K' Pipelines

- If a super scalar processor with 'K' scalar pipelines (SSP_K) then performance would be : K (Instructions per Cycle) = K IPC.

Simultaneous Multi-threaded Architecture



Simultaneous Multi-threaded Architecture (SMA)

- Super Scalar Processors exploit instruction level parallelism.
- SMAs/SMTs exploit task level parallelism.

Sample C Code

```
#include<stdio.h>
#define PI 3.1415
int main()
{
int a=10000;
int b=a*PI;
printf("%d",b);
return 0;
}
```

```
$ gcc -E B1.c > B1.i
$ gcc -S B1.i > B1.s
$ as B1.s -o B1.o
$ gcc -o B1.exe B1.o
$ ./B1.exe
```