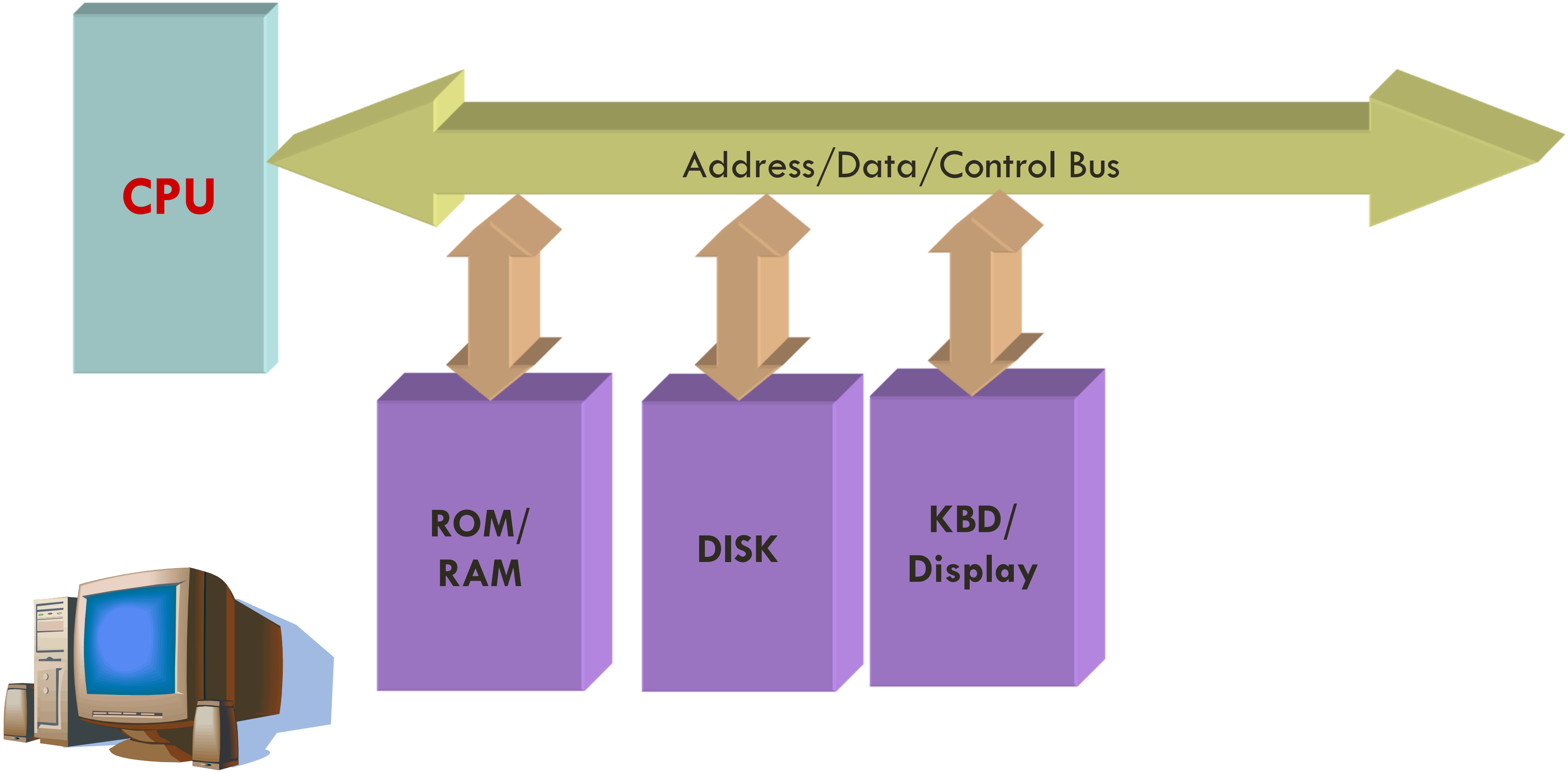
INTRODUCTION TO MICROPROCESSORS

BLOCK DIAGRAM OF A COMPUTER SYSTEM



CPU

Address/Data/Control Bus

ROM/ RAM

DISK

KBD/ Display

MICROPROCESSOR

CPU on a Single VLSI Chip

WHAT HAPPENS WHEN YOU TURN ON YOUR COMPUTER ?

BIOS – Basic Input Output System

- Resident in ROM

Orchestrates loading the computer's operating system from the hard disk drive into RAM

OS Loads Program from Disk (Secondary Storage) to RAM(Primary Storage)

Program - Set of Instructions – Executed by µp

WHAT IS INSTRUCTION ?

Tells the µp what action to perform

- Arithmetic, Logic Operation
- Read Data from Input Device
- Write to memory
- Reset
- Stop

INSTRUCTION

ADD A, B, C

Assembly Language

A,B,C - Registers

0000 001 000 001 010

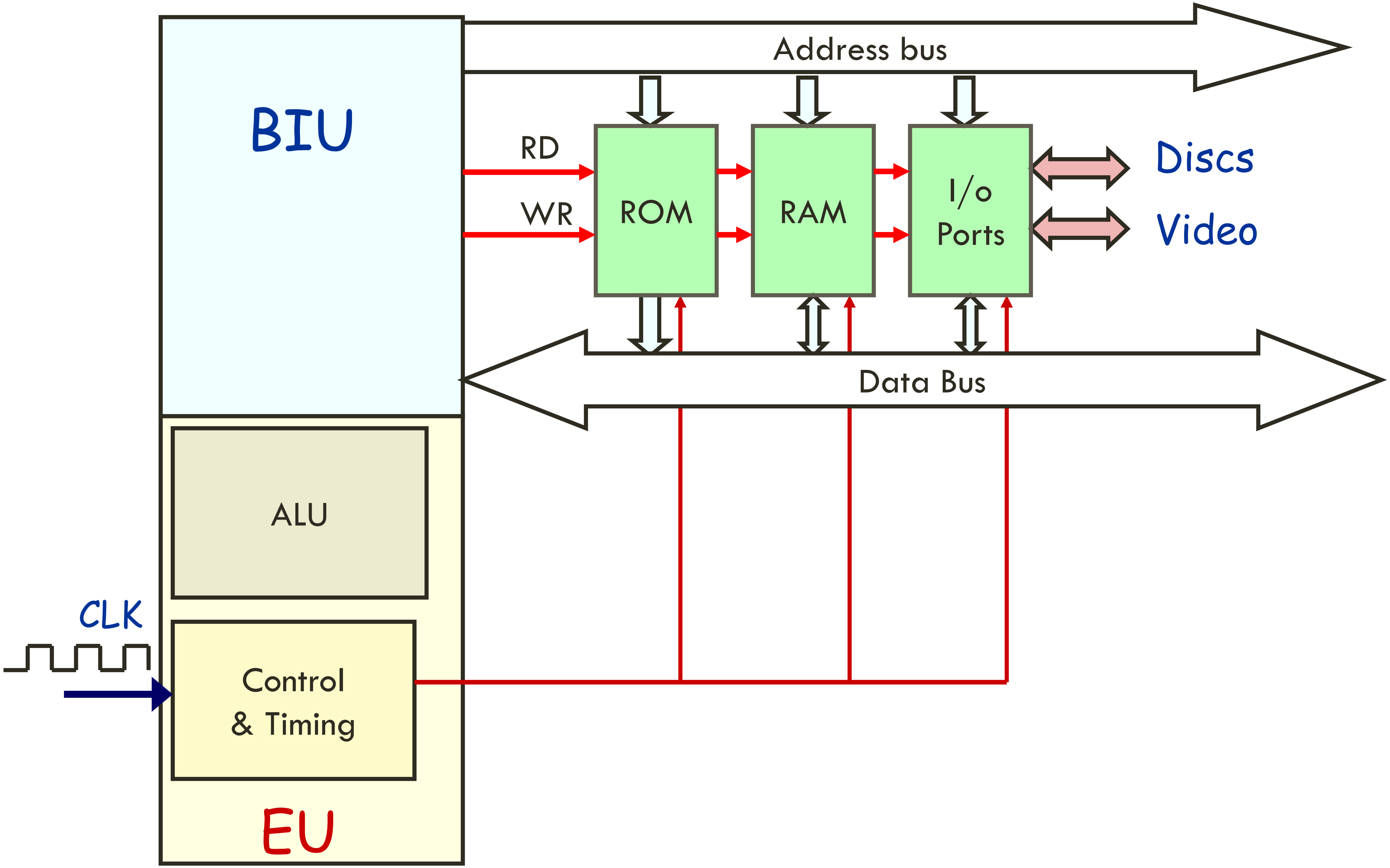
HOW DOES A MICROPROCESSOR HANDLE AN INSTRUCTION?

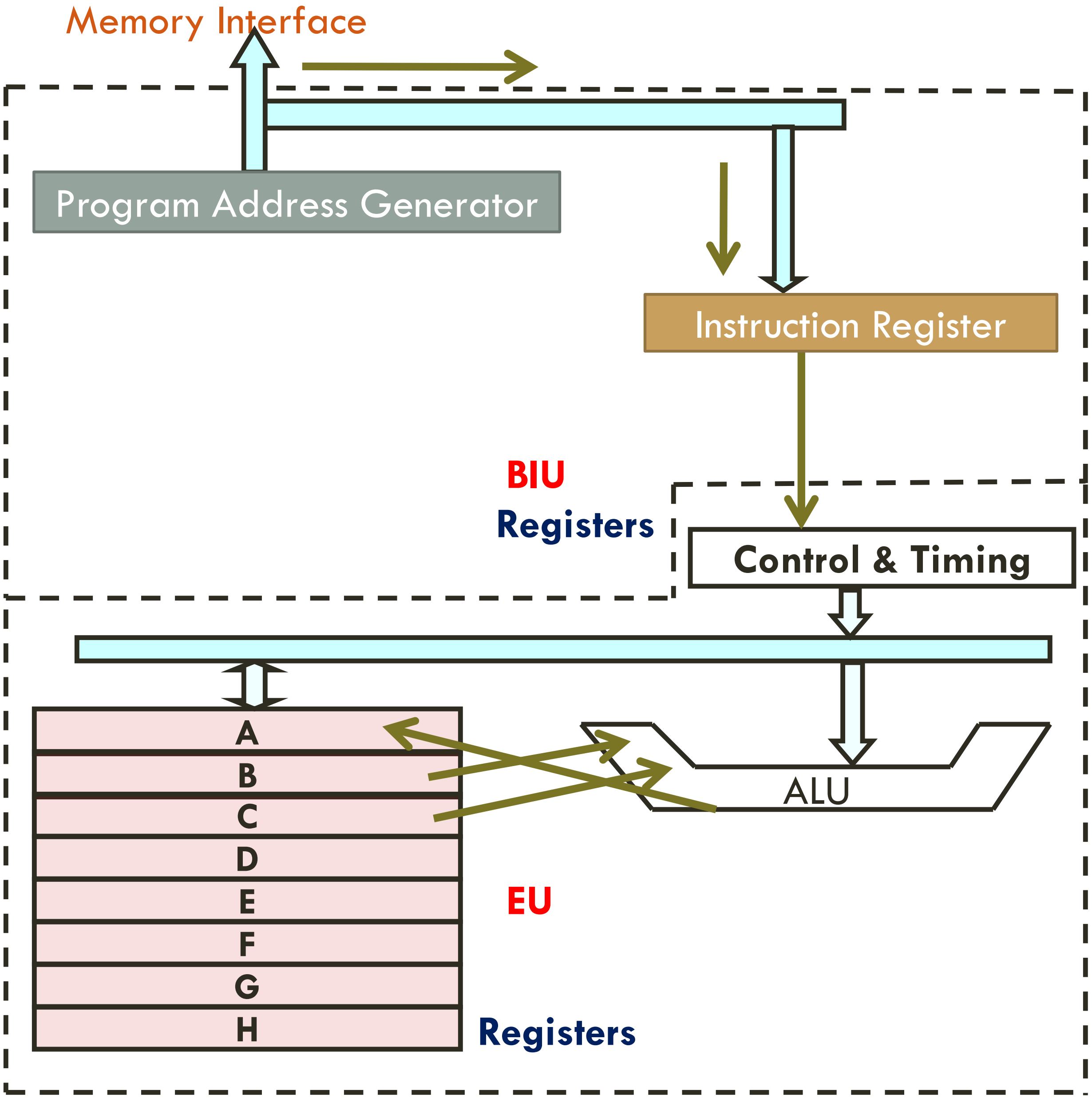
Fetch Cycle

- The fetch cycle takes the instruction required from memory, stores it in the instruction register

Execute Cycle

- The actual actions which occur during the execute cycle of an instruction


Address bus
BIU
RD
WR
ROM
RAM
I/o
Ports
Discs
Video
Data Bus
ALU
CLK
Control
& Timing
EU



Block Diagram of a Microprocessor

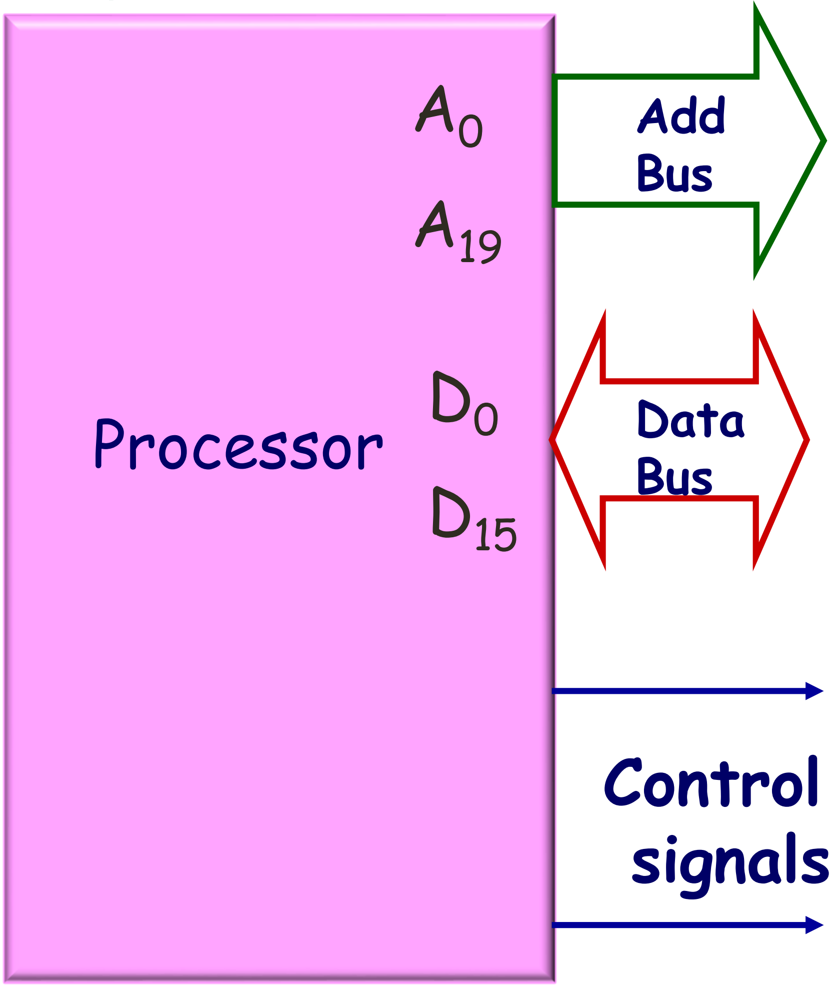
SIZE OF A MICROPROCESSOR

Size of Data Bus

Size of Registers

Size of ALU

PROCESSOR BUSES



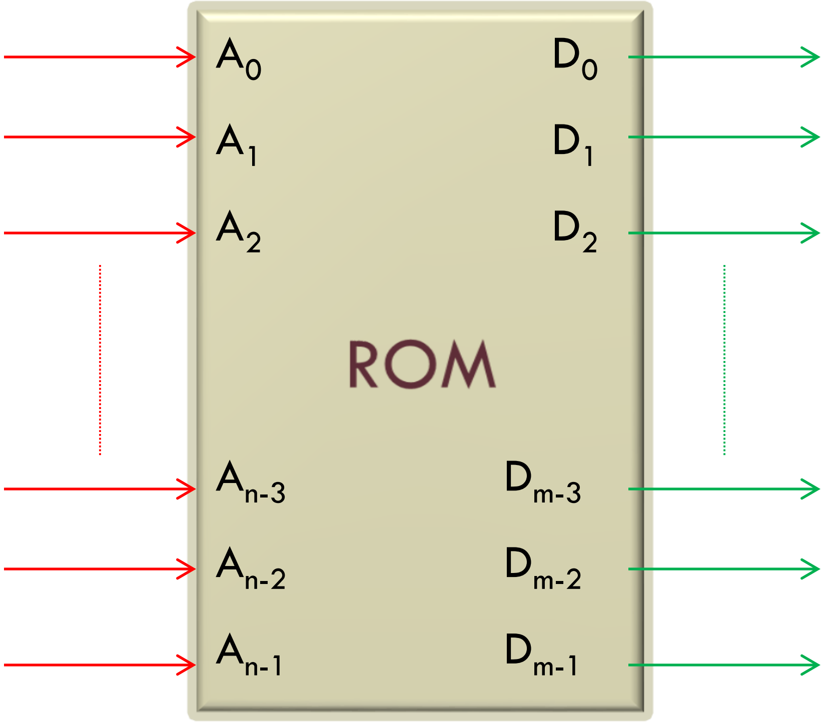
MEMORY

ROM

- Non-Volatile
- Read Only

RAM

- Volatile
- Random Access Memory



$2^n \quad x \; m$

ADDRESS BUS

No of Address lines

- 20 lines – A_{19} – A_0
- 1 MB

MEMORY DATA SIZE

Bit Organised

Nibble Organised

Byte Organised

PROCESSORS

ISA

- Execution model
- Processor registers
- Address and Data formats

Microarchitecture

- Interconnections - various micro architectural elements of machine
- ALU
- Data Path
- Control Path

Physical Realization

INSTRUCTION

ADD   A, B

Assembly Language

A,B - Registers

0000 0001 0000 0001

INSTRUCTION

ADD   M1, B

Assembly Language

B – Register

M1 – Memory Location

0000 0001 1111 0001 1000 0000 1000 1010

INSTRUCTION

ADD   M1, M2

Assembly Language

M1,M2 – Memory Location

0000 0001 1111 1111 1000 0000 1000 1010

1110 1111 0010 0010

INSTRUCTION

ADD   A, B, M1

Assembly Language

A,B – Registers

M1 – Memory Location

0000 001 000 001 111 0000 0010 1000 1000

INSTRUCTION

ADD   A, M1,M2

Assembly Language

A – Register

M1, M2 – Memory Locations

0000 001 000 111 111 0000 0010 1000 1000

0000 1000 1000 1001

INSTRUCTION

ADD   M3, M1,M2

Assembly Language

M1, M2,M3 – Memory Locations

0000 001 111 111 111 0000 0010 1000 1000

0000 1000 1000 1001 0010 1000 1111 1010

WHAT IS THE EFFECT ?

If Instructions can be present anywhere

- Size of Instruction Varies
- Complicates Instruction Decoder

ISA

- CISC
  - Operands for Arithmetic/Logic operation can be in Register/ Memory
- RISC
  - Operands for Arithmetic/Logic operation only in Registers
  - Register – Register Architecture

RISC Vs CISC

Goal: Multiply data in mem A with B- put it back in A

CISC:

MUL A,B

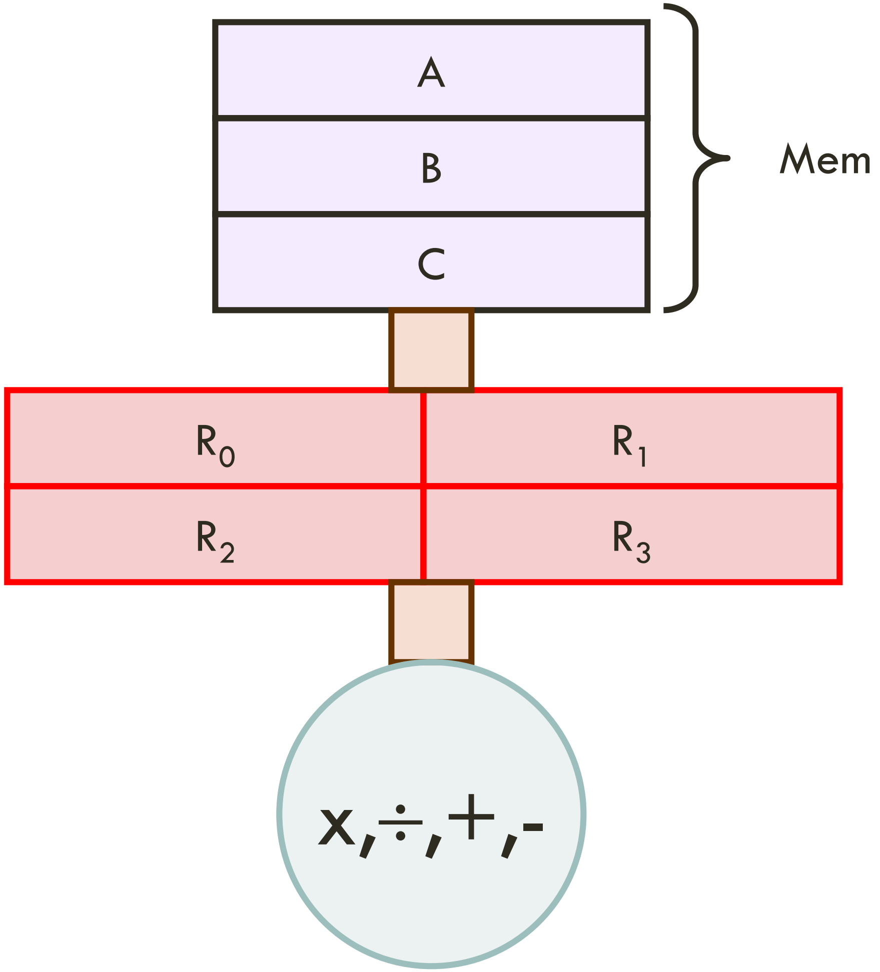
RISC:

LDA R_0,A

LDA R_1,B

MUL R_0,R_1

STR A,R_0



$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Time}}{\text{cycle}} \times \underset{\substack{\uparrow \\ \textbf{RISC}}}{\frac{\text{Cycles}}{\text{Inst}}} \times \underset{\substack{\uparrow \\ \textbf{CISC}}}{\frac{\text{Instructions}}{\text{Program}}}$$

CPU- SPEEDUP

1 Instruction Per Cycle (1 IPC)

BASIC PARALLEL TECHNIQUES

Pipelining

Replication

INSTRUCTION PIPELINES

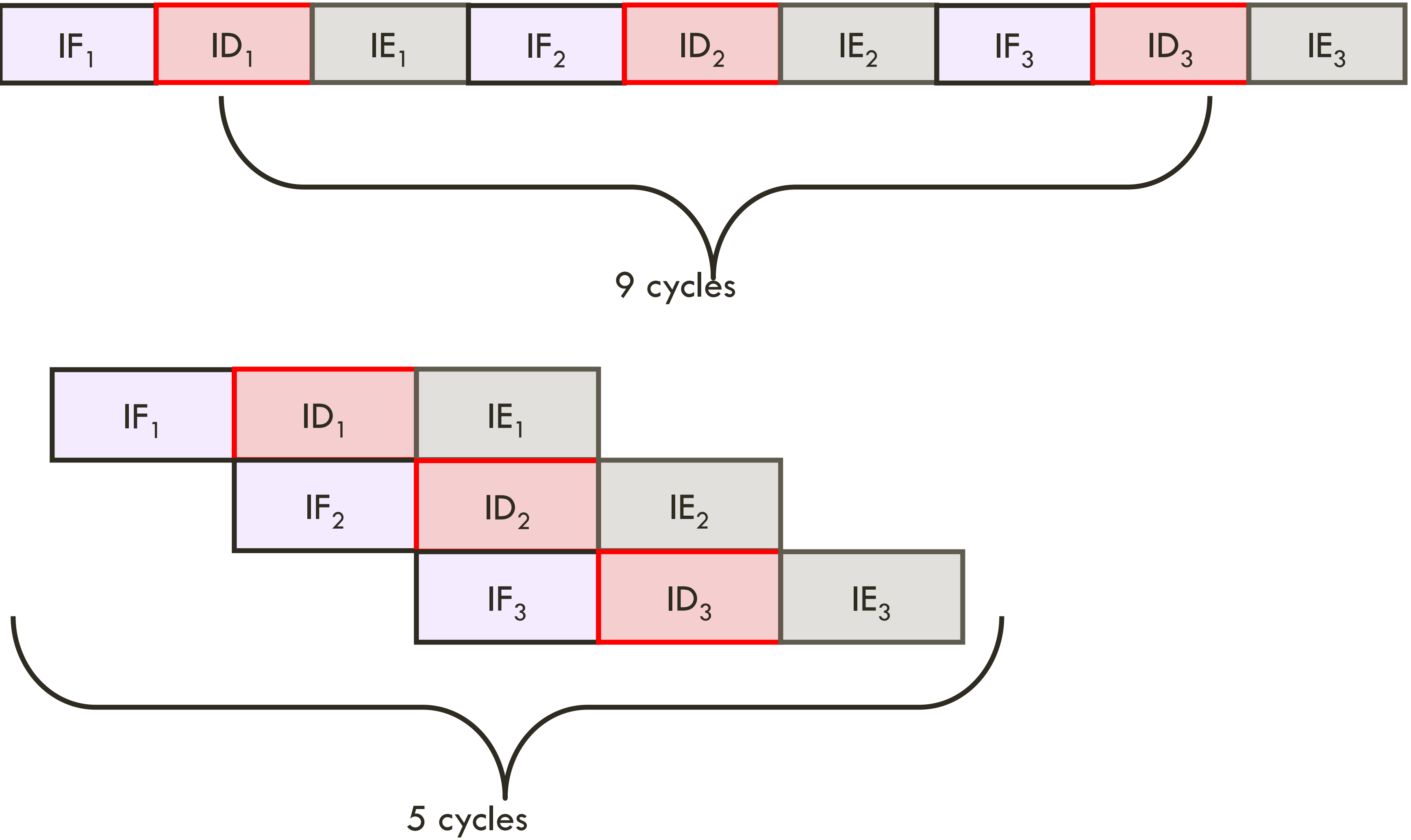
Instruction:

Fetch

Decode

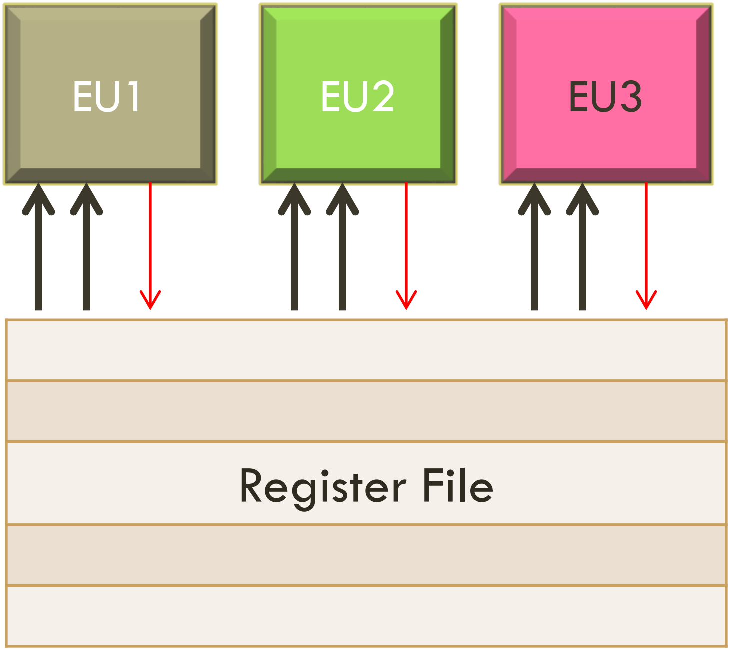
Execute

1. ADD R2,R1,R3
2. SBR R2,R3,R2
3. STR R2,b



Pipeline Hazards

VLIW & SUPERSCALAR ARCHITECTURE



FLYNN'S TAXNOMY

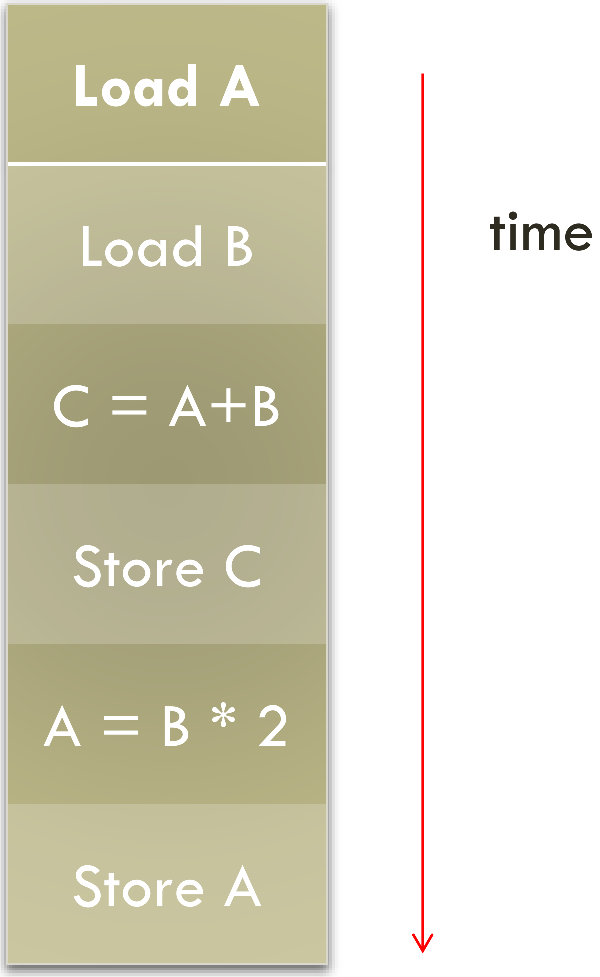
SISD

SIMD

MISD

MIMD

SISD



Load A

Load B

C = A+B

Store C

A = B * 2

Store A

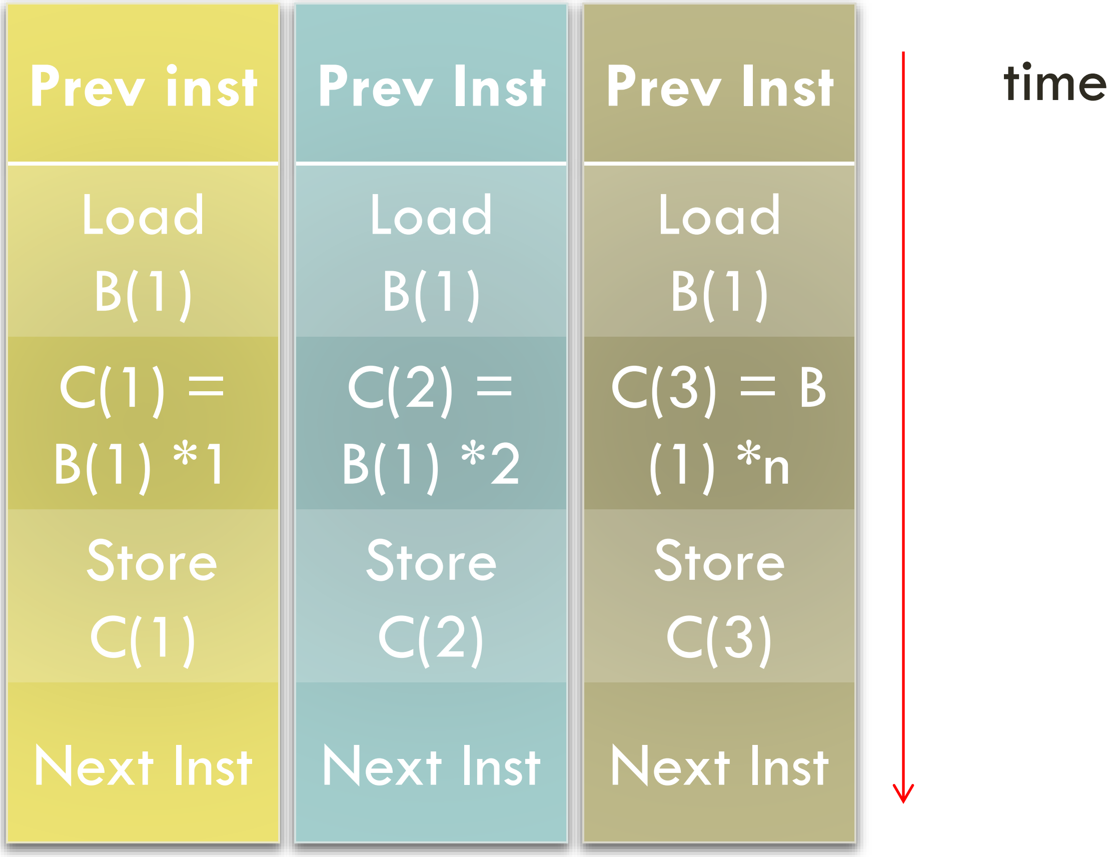
time

SIMD

Load A(0)	**Load A(1)**	**Load A(2)**
Load B(0)	Load B(1)	Load B(2)
C(0) = A(0)+B(0)	C(1) = A(1)+B(1)	C(2) = A(2)+B(2)
Store C(0)	Store C(1)	Store C(2)
A (0)= B (0)* 2	A (1)= B (1)* 2	A (2)= B (2)* 2
Store A(0)	Store A(1)	Store A(2)

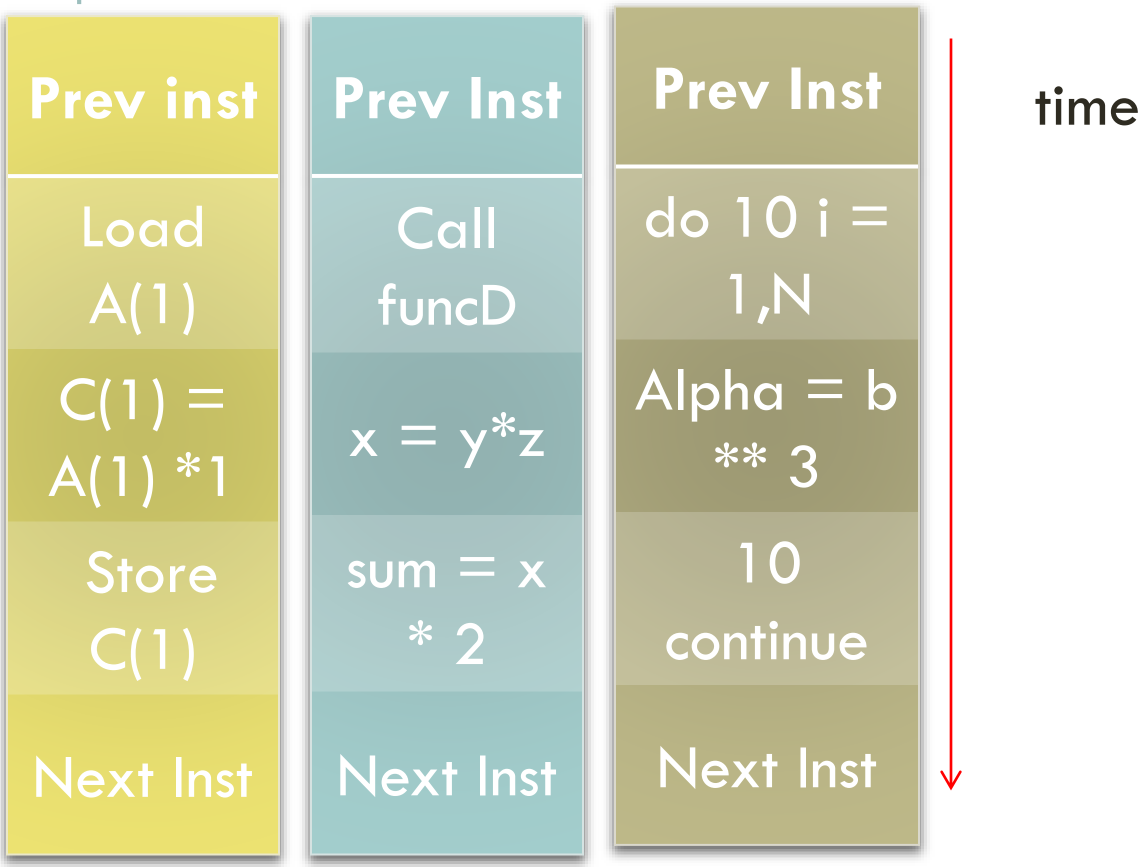
time

MISD



Prev inst	Prev Inst	Prev Inst
Load B(1)	Load B(1)	Load B(1)
C(1) = B(1) *1	C(2) = B(1) *2	C(3) = B (1) *n
Store C(1)	Store C(2)	Store C(3)
Next Inst	Next Inst	Next Inst

MIMD



Prev inst	Prev Inst	Prev Inst
Load A(1)	Call funcD	do 10 i = 1,N
C(1) = A(1) *1	x = y*z	Alpha = b ** 3
Store C(1)	sum = x * 2	10 continue
Next Inst	Next Inst	Next Inst

Name	Date	Transistors	Clock speed	Data width
8080	1974	6K	2MHz	8
8086	1978	29K	5MHz	16
80286	1982	134K	12 MHz	16
80386	1985	275K	16-33 MHz	32
80486	1989	1.2 M	20 -100 MHz	32
Pentium	1993	3.1M	60-200 MHz	32 /64
Pentium II	1997	7.5 M	233-450 MHz	32/ 64
Pentium III	1999	9.5M	450 -933 MHz	32 /64
Pentium 4	2000	42 M	1.5 GHz	32/ 64

The Evolution of Microprocessors