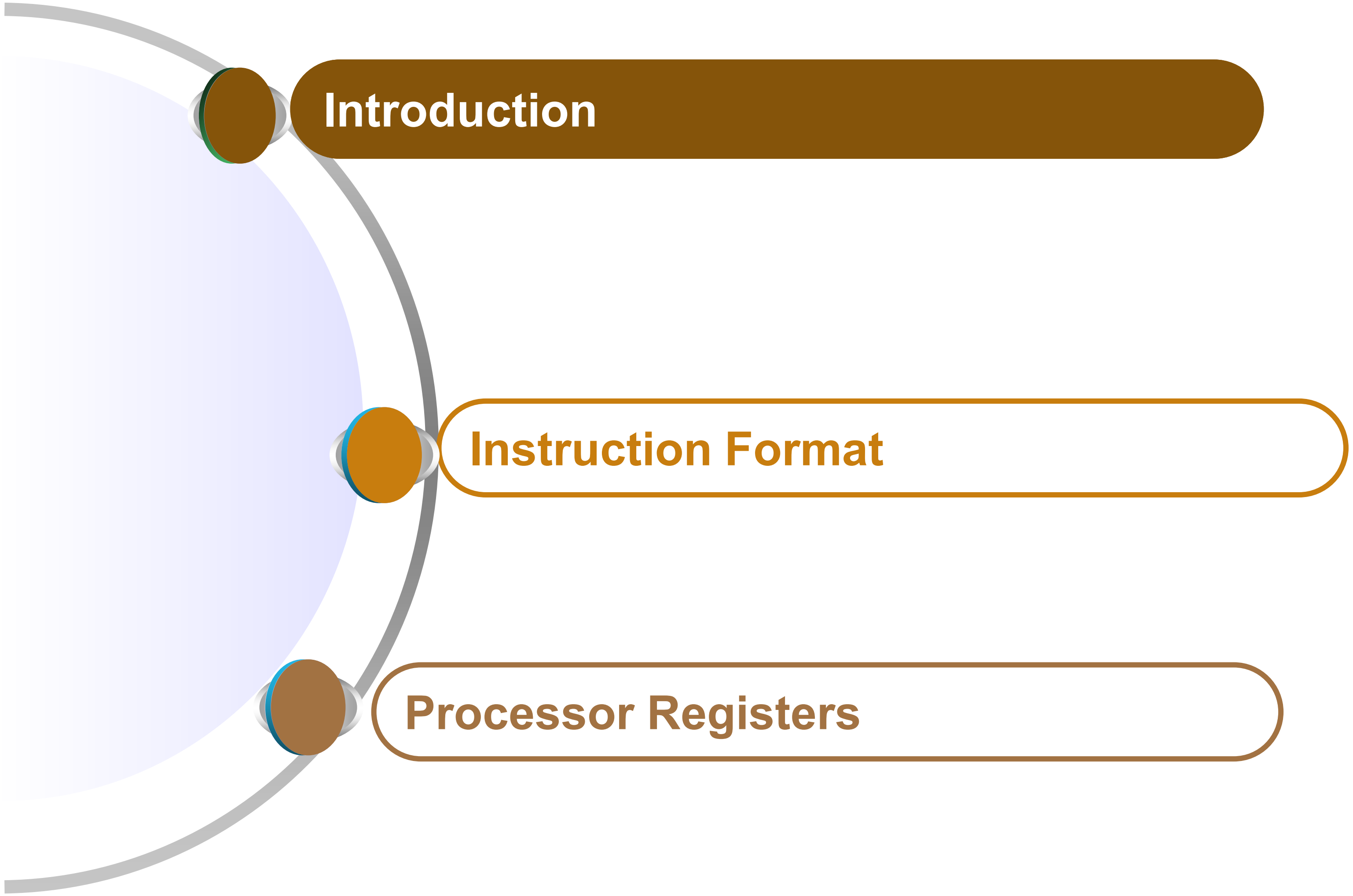
Computer Architecture

Lecture 1

What You Will Learn

- How programs are translated into the machine language
  - And how the hardware executes them
- The hardware/software interface
- What determines program performance
  - And how it can be improved
- How hardware designers improve performance

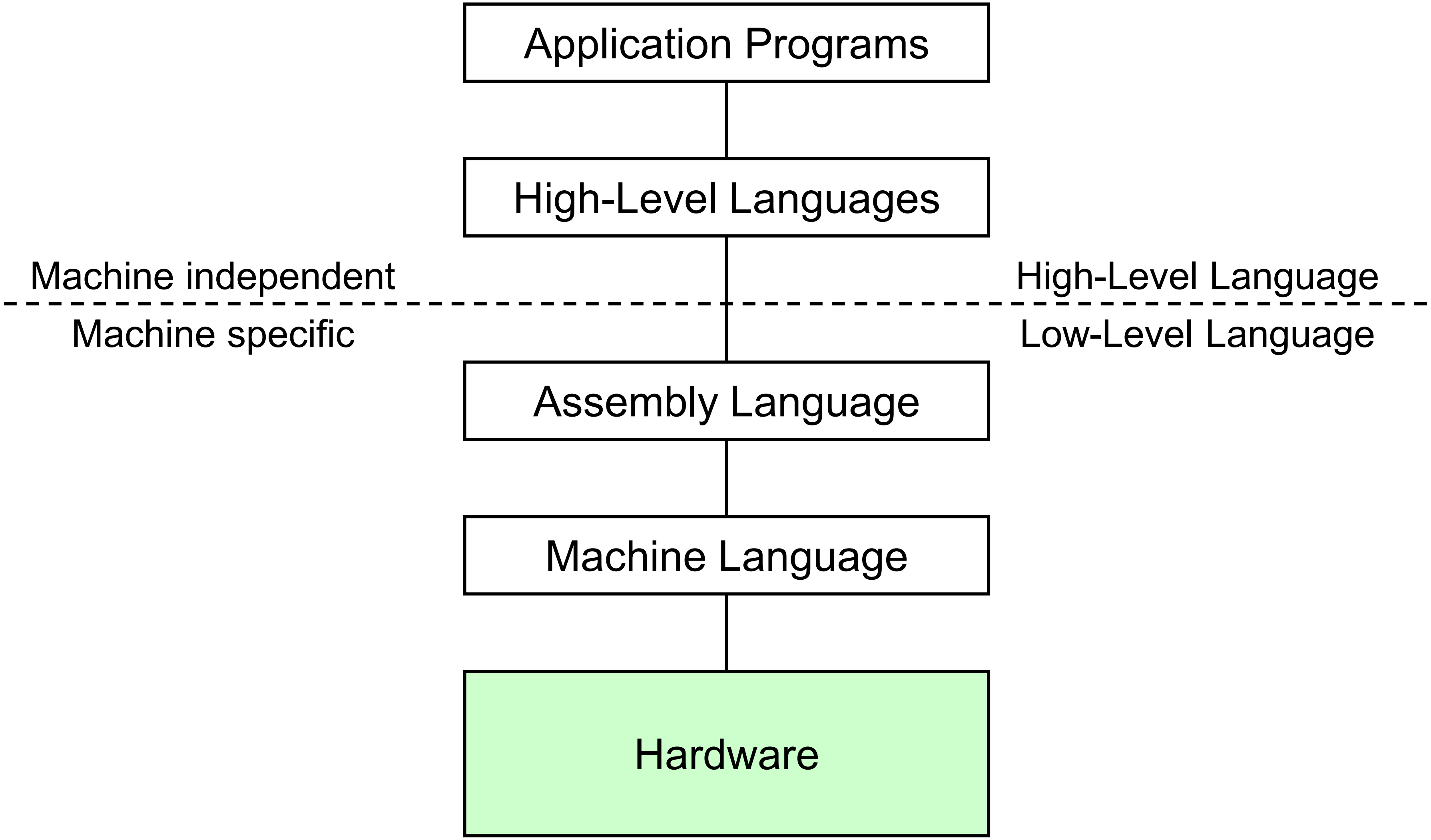


Introduction

Instruction Format

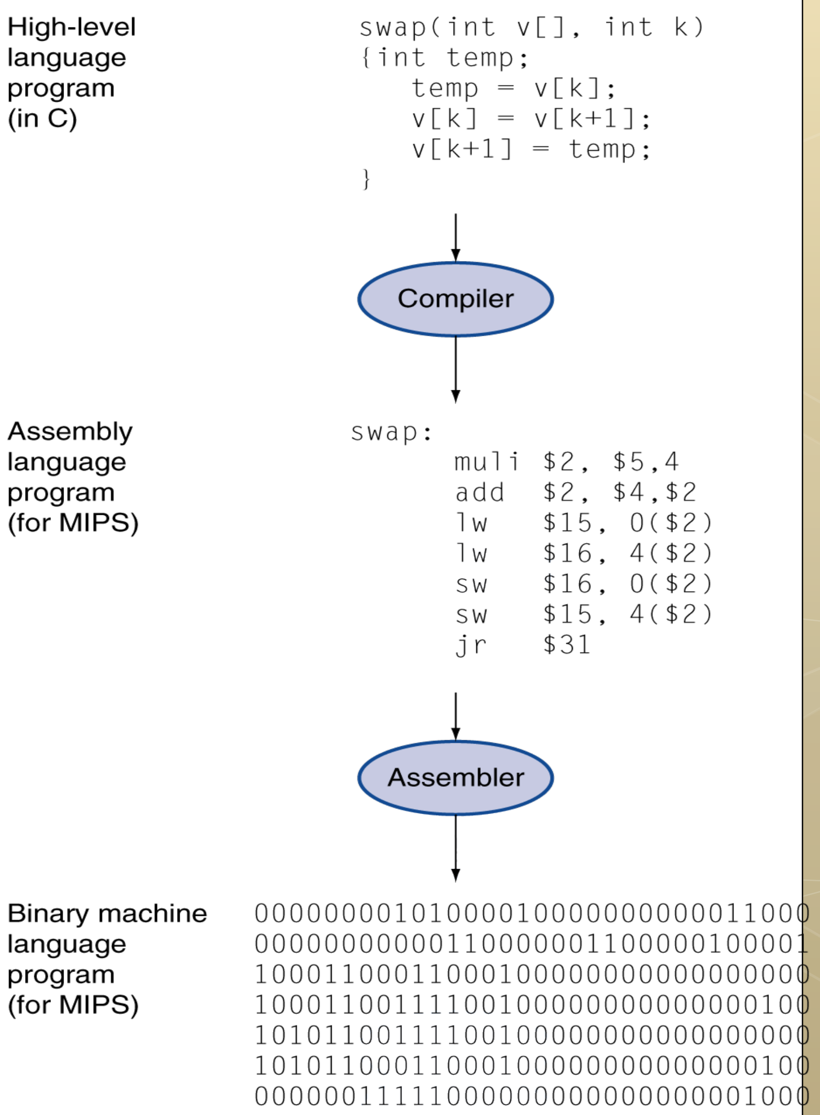
Processor Registers

A Hierarchy of Languages



Assembly and Machine Language

- Machine language
  - Native to a processor: directly executed by hardware.
  - Instructions have only binary code: 1s and 0s.
- Assembly language
  - Readability of instructions is better than machine language
  - Every instruction an assembly translated to one instruction in machine language.
- Assemblers translate assembly to machine code
- Compilers translate high-level programs to machine code or to assembly code



High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Assembly language program (for MIPS)

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```

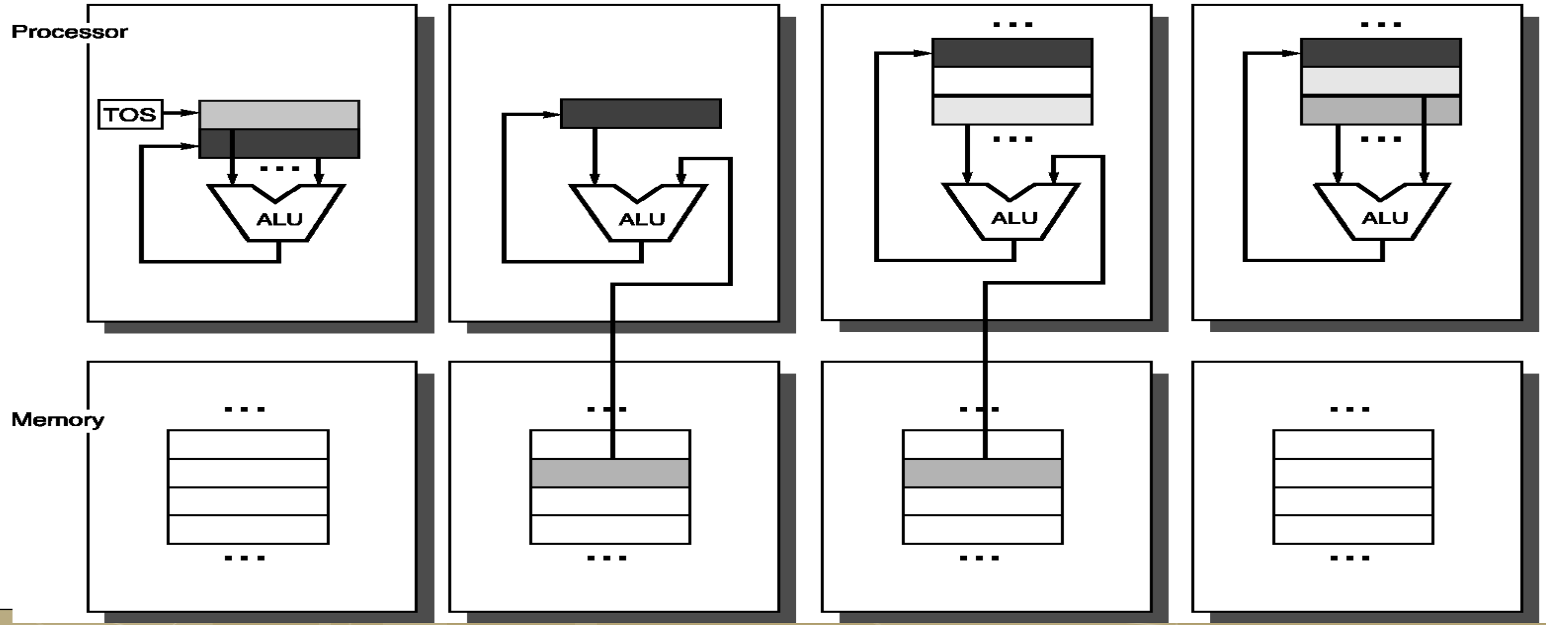
Binary machine language program (for MIPS)

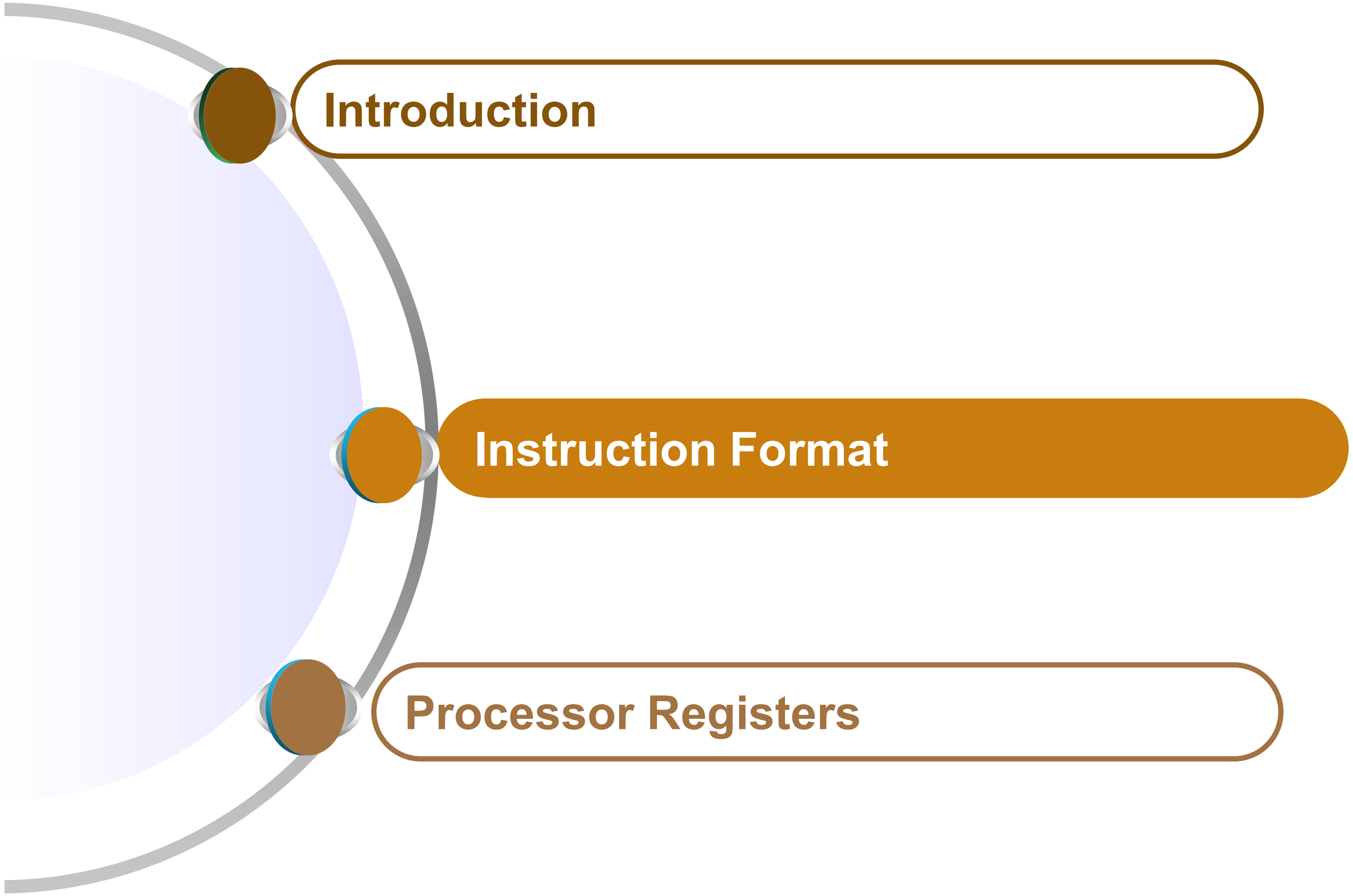
```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Architecture Classes

Assembly for **C:=A+B**:

Stack	Accumulator	Register (register-memory)	Register (load-store)
`Push A`	`Load A`	`Load R1,A`	`Load R1,A`
`Push B`	`Add B`	`Add R1,B`	`Load R2,B`
`Add`	`Store C`	`Store C,R1`	`Add R3,R1,R2`
`Pop C`			`Store C,R3`



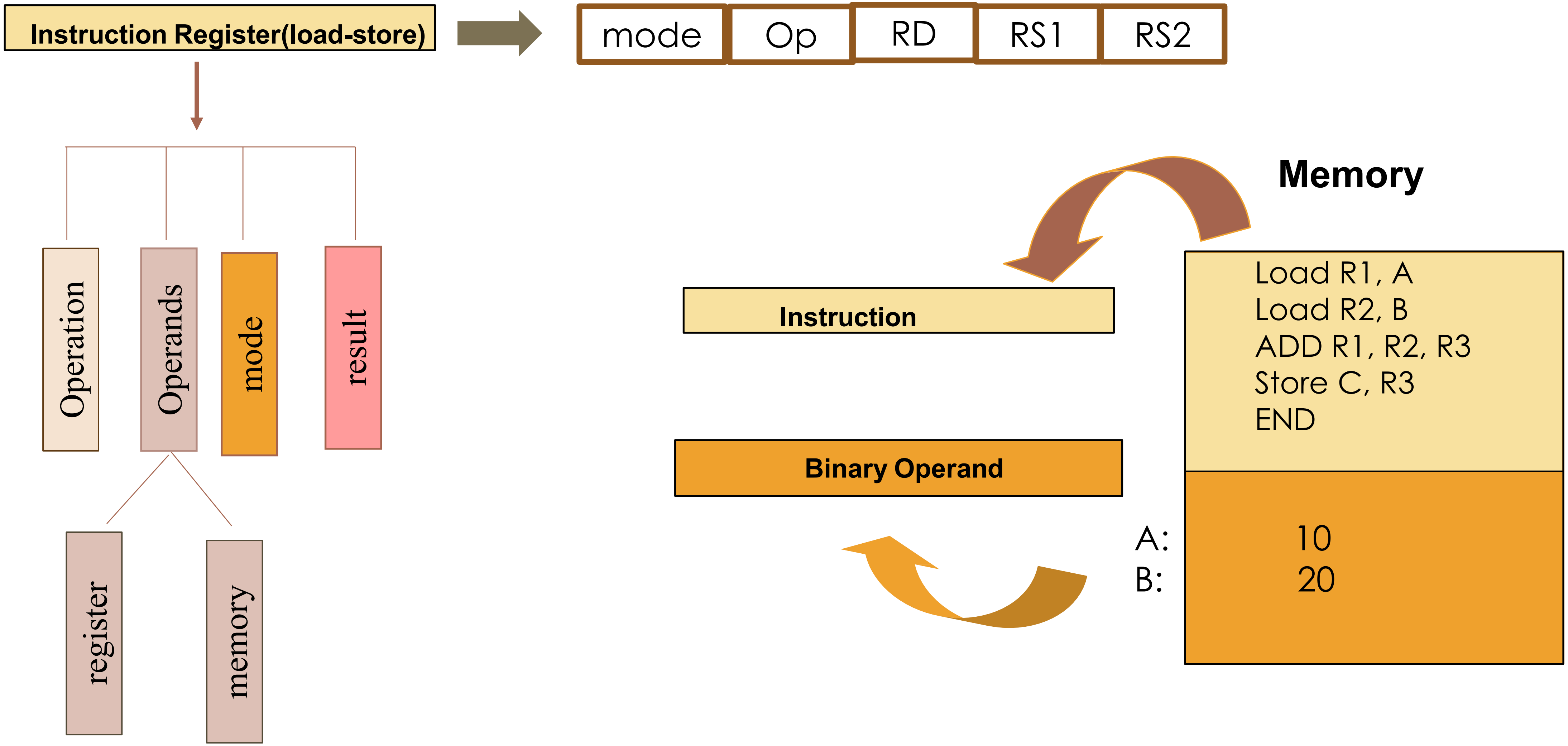


Introduction

Instruction Format

Processor Registers

Stored Program Organization



Instruction Register(load-store)

mode	Op	RD	RS1	RS2

Memory

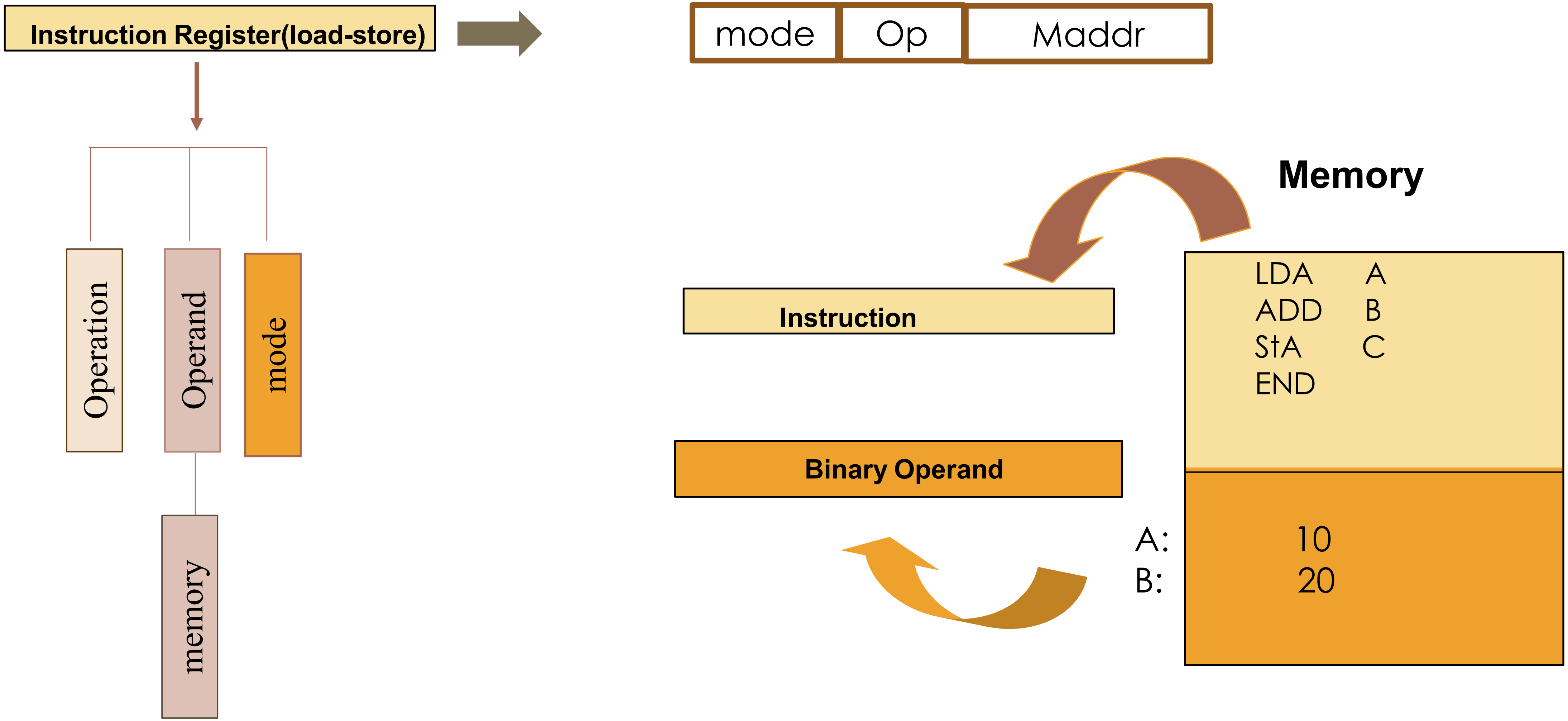
Instruction

Binary Operand

```
Load R1, A
Load R2, B
ADD R1, R2, R3
Store C, R3
END
```

A:	10
B:	20

Stored Program Organization



Instruction Register(load-store)

mode | Op | Maddr

Operation

Operand

mode

memory

Memory

Instruction

Binary Operand

LDA A
ADD B
StA C
END

A: 10
B: 20

Basic Computer

- **Program**
  - **A program is a set of instructions that specify the operations, operands, and the sequence by which processing has to occur.**
- **(Machine) Instruction**
  - **A group of bits that tell the computer to *perform a specific operation* (a sequence of micro-operation).**
- **Instruction codes together with data are stored in memory.**
- **The CPU reads the next instruction from memory and placed in an *Instruction Register* (IR).**
- **The control unit is responsible of translating the instruction into the sequence of microoperations necessary to implement it.**

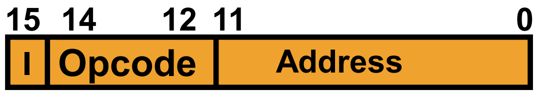
Instruction Formats

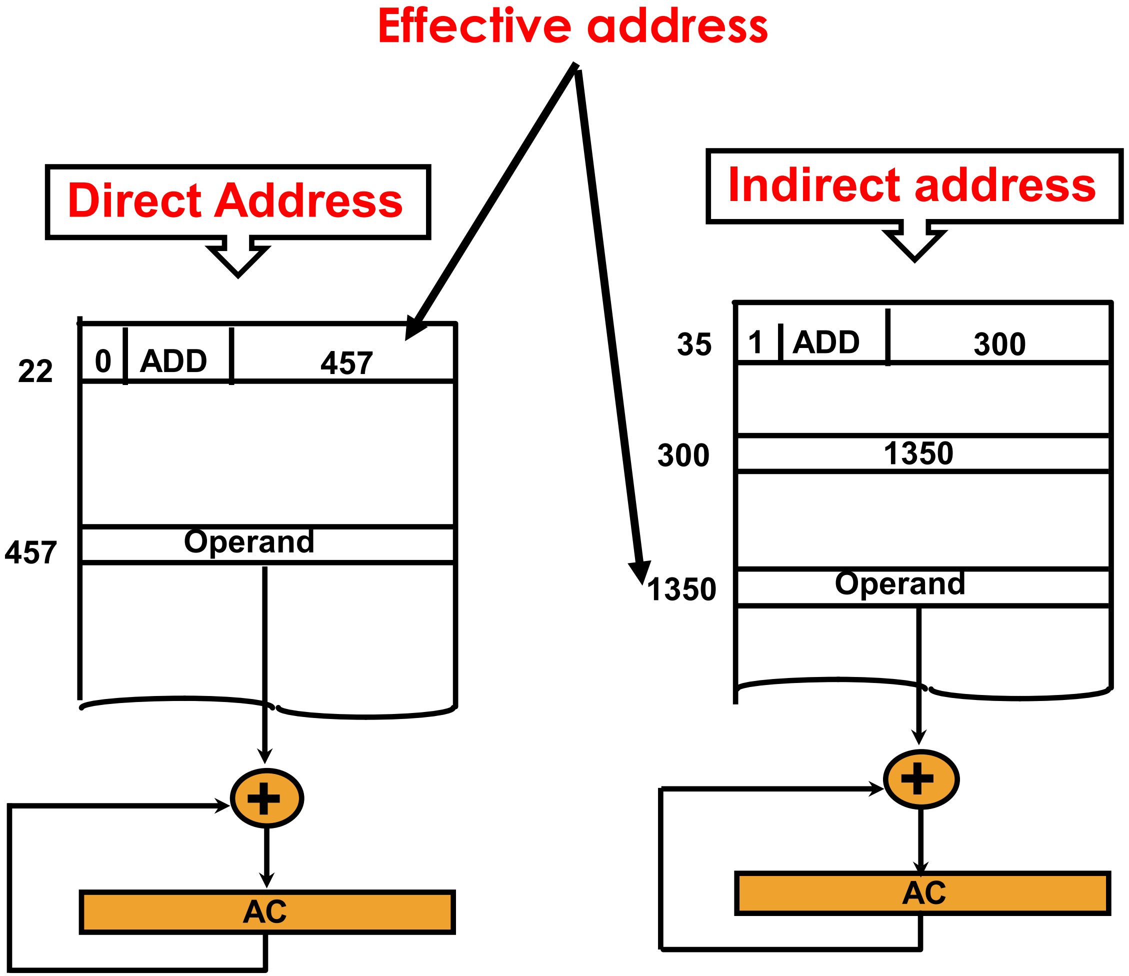
- The most common fields in instruction formats:
- **Operation field**
  - The operation code field of an instruction is a group of bits that define various processor operations, such as add, subtract, complement, and shift.
- **Address field**
  - Operations specified by computer instructions are executed on some data stored in memory or processor registers.
  - Operands residing in memory are specified by their memory address.
  - Operands residing in processor registers are specified with a register address.
- **Addressing Modes**
  - Specifies a rule for interpreting or modifying the address field of the instruction (before the operand is actually referenced)

Addressing Modes

- **Direct Address Mode**
  - ➢ Instruction specifies the memory address which can be used directly to access the memory
- **Indirect Addressing Mode**
  - ➢ The address field of an instruction specifies the address of a memory location that contains the address of the operand

Instruction Format



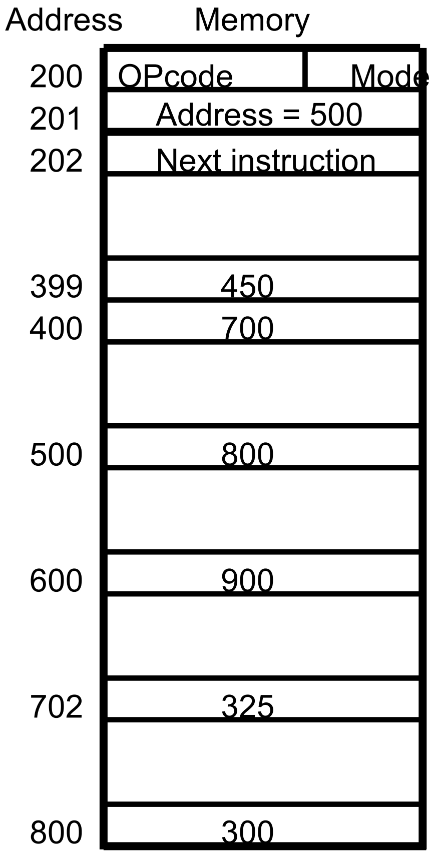


ADDRESSING MODES EXAMPLE

What is the Ac content and EA after executing the following instruction with two addressing modes?

Addressing Mode	Effective Address	Content of AC
Direct address	500	/* AC ← 200 + M [500] */ 1000
Indirect address	800	/* AC ← 200 + M[M[500]] */ 500

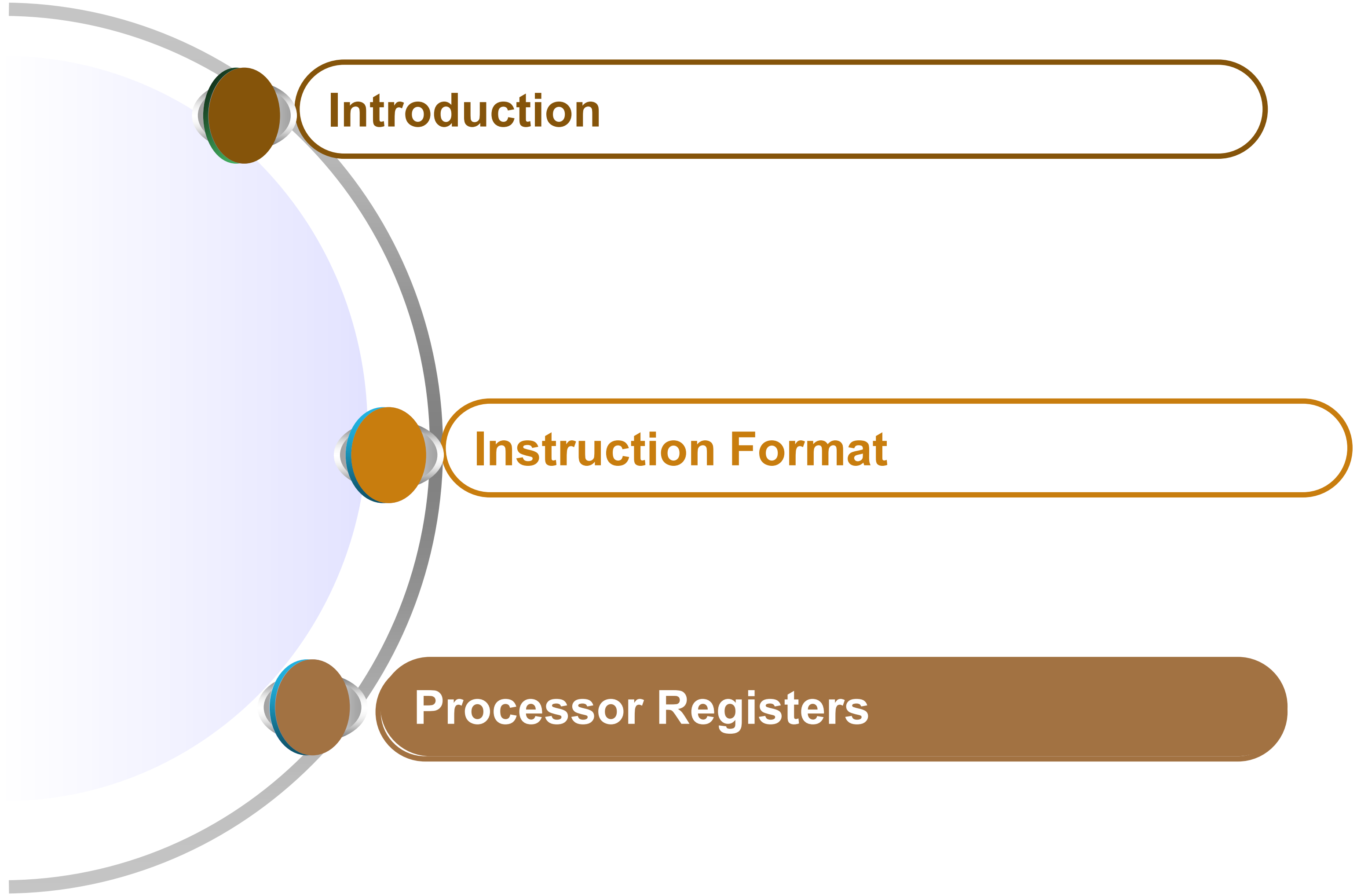
ADD operation means adding AC value with the content of memory then saves the result in AC



Address	Memory
200	OPcode | Mode
201	Address = 500
202	Next instruction
399	450
400	700
500	800
600	900
702	325
800	300

PC = 202

AC = 200

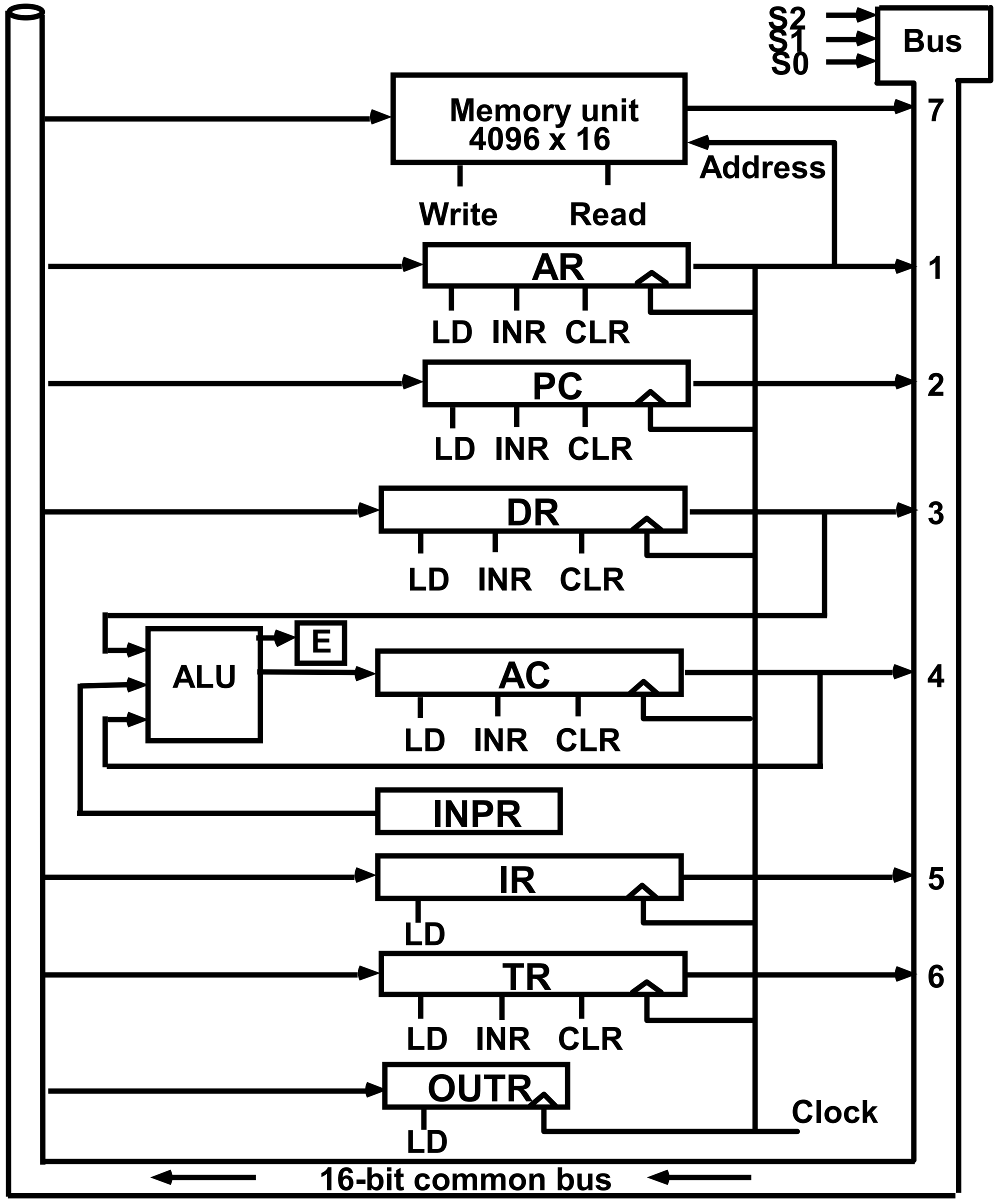


Introduction

Instruction Format

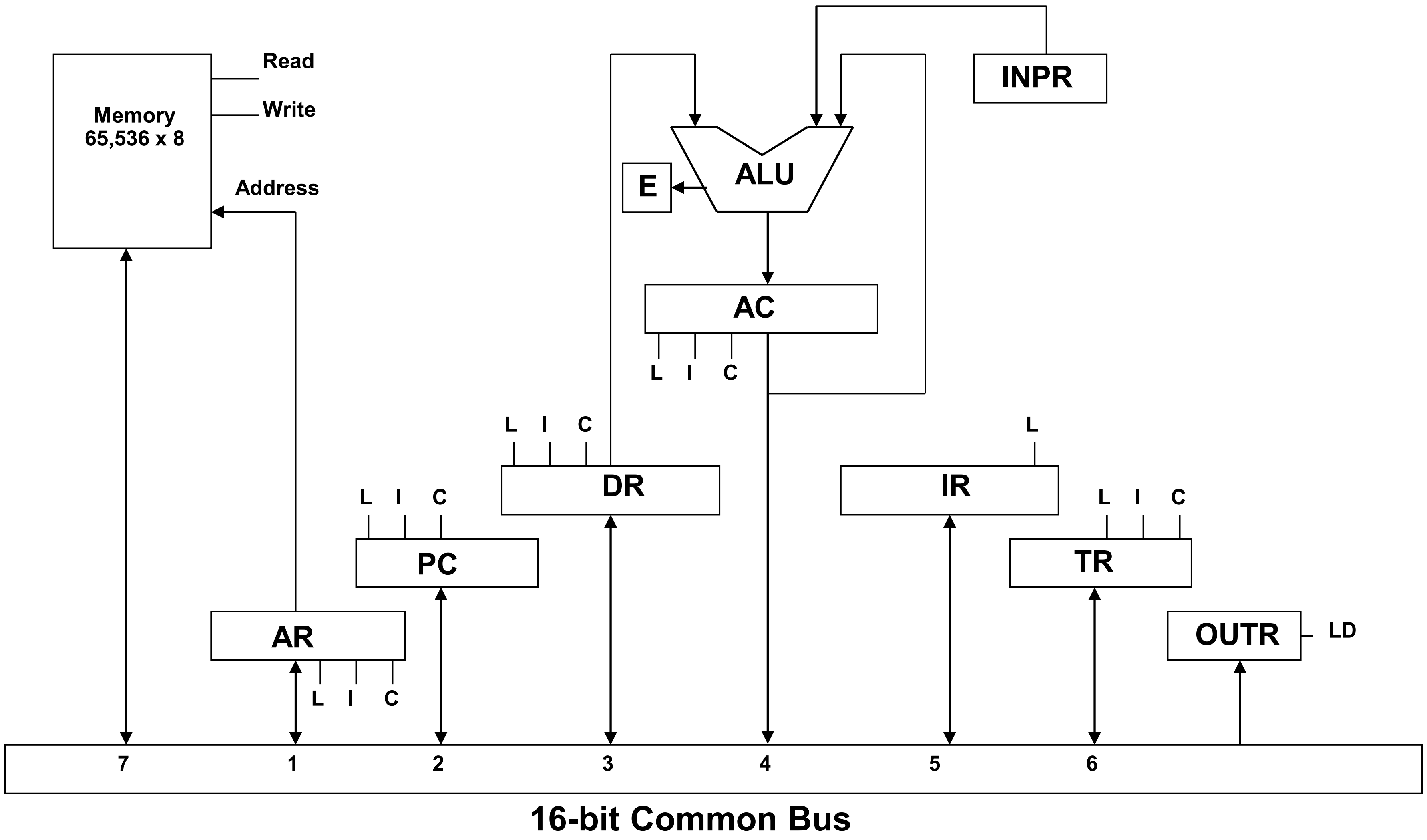
Processor Registers

Common Bus System



- **The connection between register is done through bus which saves the circuitry.**

COMMON BUS SYSTEM



PROCESSOR REGISTERS

- A processor has several registers each is used for different task such as holding instructions, addresses.
- **The *Program Counter* (PC)** stores the memory address of the next instruction to be executed.
  - the PC is 12 bits long Since the memory has 4096 words.
- **The *Address Register* (AR)** is used to keep track of what locations in memory it is addressing: a direct or indirect addressing.
  - The AR is a 12 bit register in the Basic Computer
- **The *Data Register* (DR)** holds the operand read from memory . The processor then uses this value as data for its operation.
- ***The Accumulator (AC)*** is a single *general-purpose register* in basic computer.

PROCESSOR REGISTERS

- Sometimes a processor needs a scratch register to store intermediate results or other temporary data. This register is known as **the *Temporary Register* (TR).**

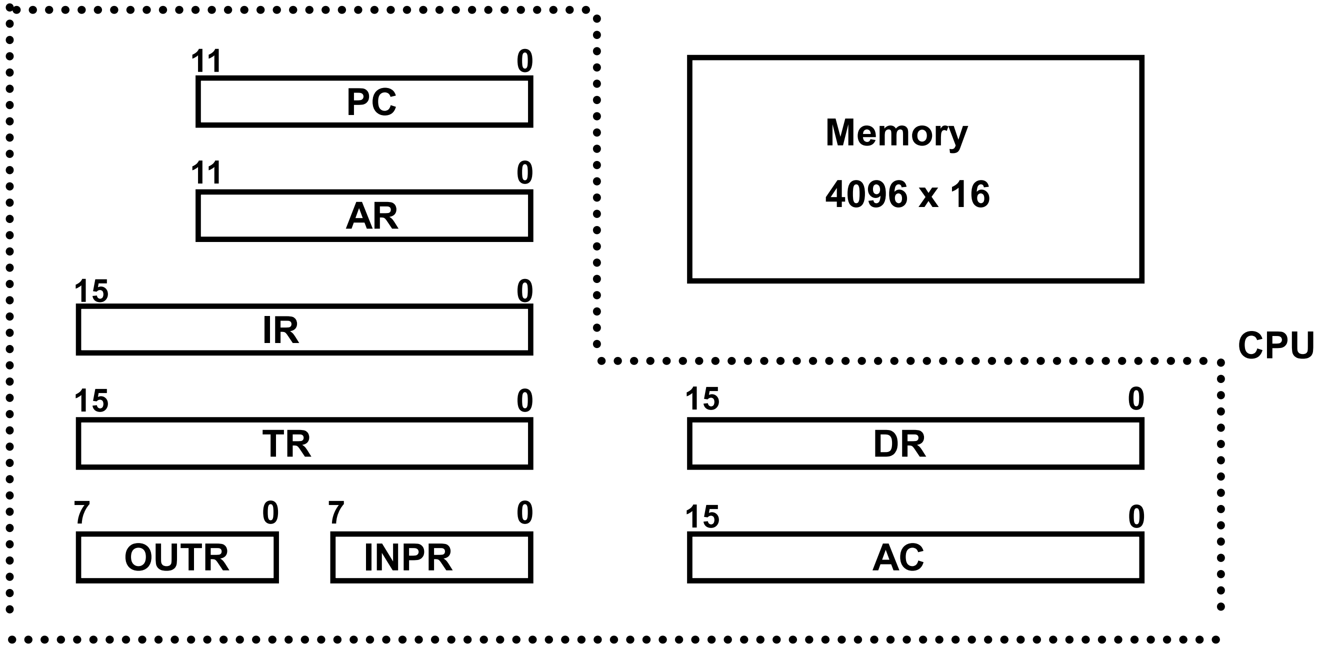
- The Basic Computer uses a very simple model of input/output (I/O) operations
  - Input devices are considered to send 8 bits of character data to the processor
  - The processor can send 8 bits of character data to output devices

- **The *Input Register* (INPR)** holds an 8-bit character received from an input device

- **The *Output Register* (OUTR)** holds an 8-bit character to be send to an output device

PROCESSOR REGISTERS

Registers in the Basic Computer



List of Registers

DR	16	Data Register	Holds memory operand
AR	12	Address Register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Holds instruction code
PC	12	Program Counter	Holds address of instruction
TR	16	Temporary Register	Holds temporary data
INPR	8	Input Register	Holds input character
OUTR	8	Output Register	Holds output character

Thank you