CAO: Lecture 24
Memory Reference, Register Reference

Topics Covered

- Common bus system
- Computer instructions
- Memory reference instruction
- Register reference instruction
- Input-output instruction

Detailed data path of a typical register based CPU

- The **internal organization of a digital system** is defined by the **sequence of micro operations** it performs on data stored in its registers.

- **Digital computer** is capable of executing various micro operations & can be instructed as to **what sequence of operations it must perform.**

- The **user of a computer can control** the process by means of a program.

- A **program is a set of instructions** that specify the operations, operands, and the sequence by **which processing has to occur**.

- The **data processing task maybe altered** by specifying a new program with **different instructions** or specifying the same instructions with **different data**.

- A **computer instruction is a binary code** that specifies a sequence of micro operations for the computer. **Instructions codes together with data are stored in memory.**

- The **computer reads each instruction from memory** and places it in a control register. The **control then interprets the binary code** of the instruction and **proceeds to execute** it by issuing a **sequence of micro operations** .

- Every **computer has its own instruction set**. The ability to store and execute , the stored program concept, is the most important property of a general purpose computer.

- <u>**An instruction code is a group of bits that instruct the computer to perform a specific operation.**</u> It usually divided into two parts, each having its own particular interpretation.

- The most basic part of an instruction code is its **operation part**. The operation code of an instruction is a group of bits that **define such operations as add, subtract, multiply, shift and complement.**

- As an illustration, consider a **computer with 64 distinct operations**. One of them being an **ADD operation**. When this **operation code is decoded in the control unit**, the computer **issues control signals to read an operand** **from memory and add the operand to a processor register.**

- The relationship between a computer operation and a micro operation. **An operation is a part of an instruction** stored in computer memory. It is a binary code that **tells the computer to perform a specific operation.**

- The control unit receives the instruction from memory and interprets the computer code bits.

- **It then issues a sequence of control signals to initiate micro operations in internal computer registers.**

- *For every operation code, the control issues a sequence of micro operations needed for the hardware implementation of the specified operation.*

- For this reason, an operation code is sometimes called a macro operations because it specifies a set of micro operations.
- The operation part of an instruction code specifies the operation to be performed. This operation must be performed on some data stored in processor

- Instruction Code specifies operation and registers where the operands are to be found.
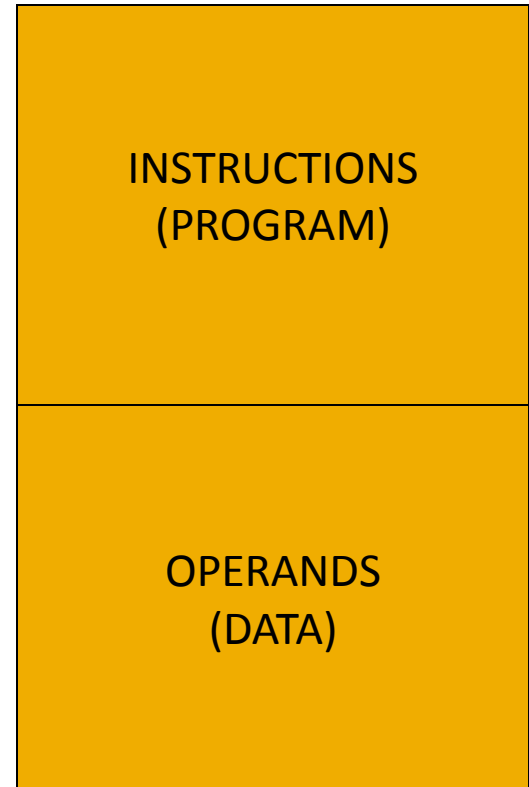
- Instruction Code format with two parts

OPCODE	ADDRESS

- Opcode specifies the operation to be performed
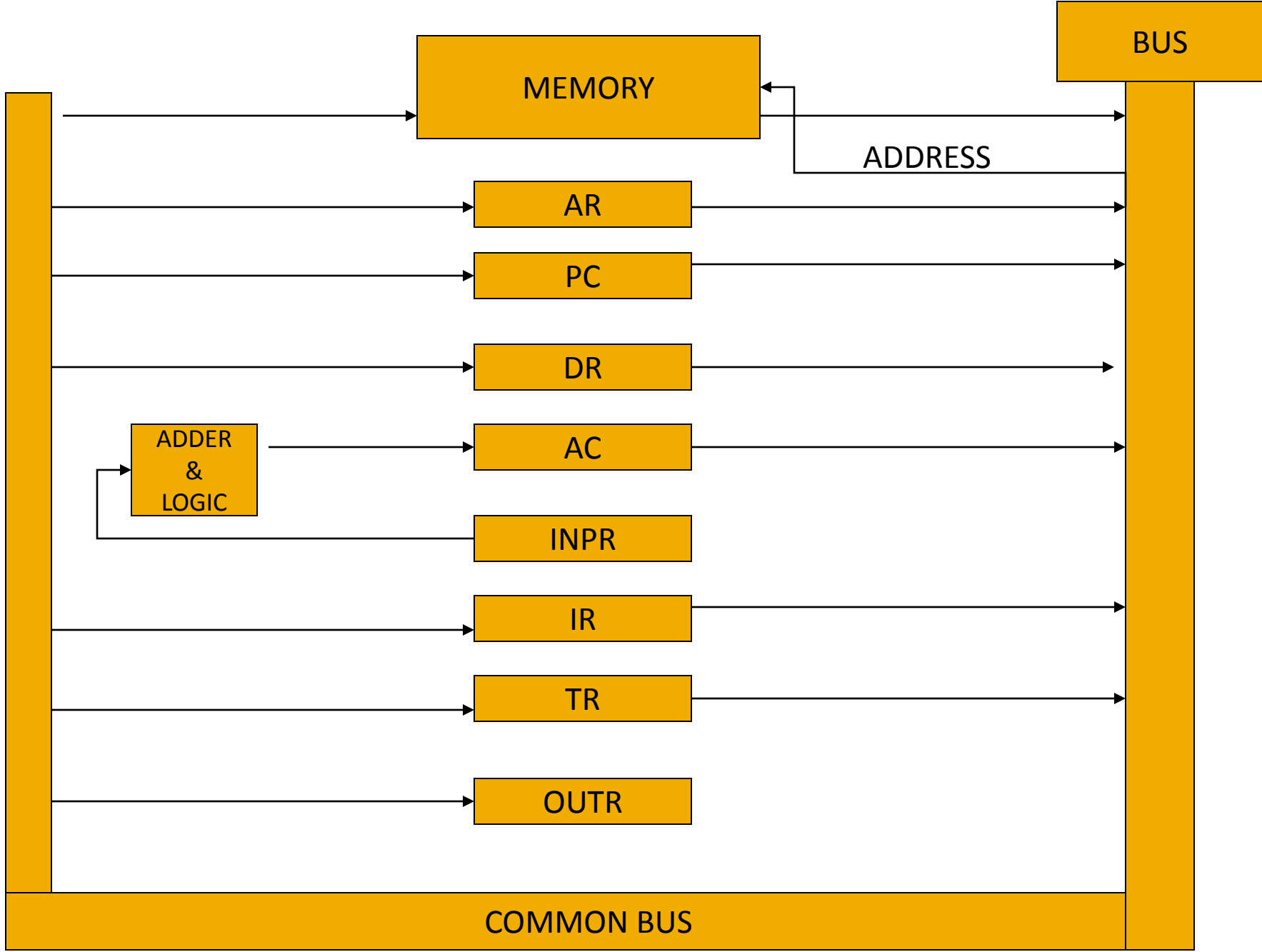- Address tells the control where to find an operand in memory.

Memory

OPCODE	ADDRESS

Instruction Format

INSTRUCTIONS
(PROGRAM)

OPERANDS
(DATA)
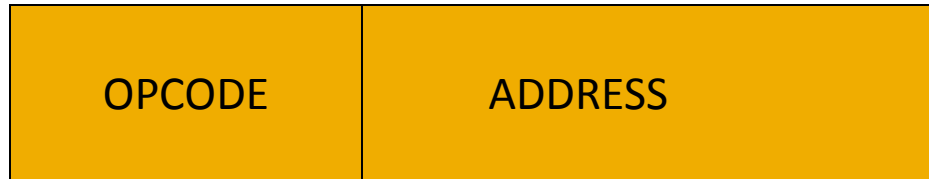
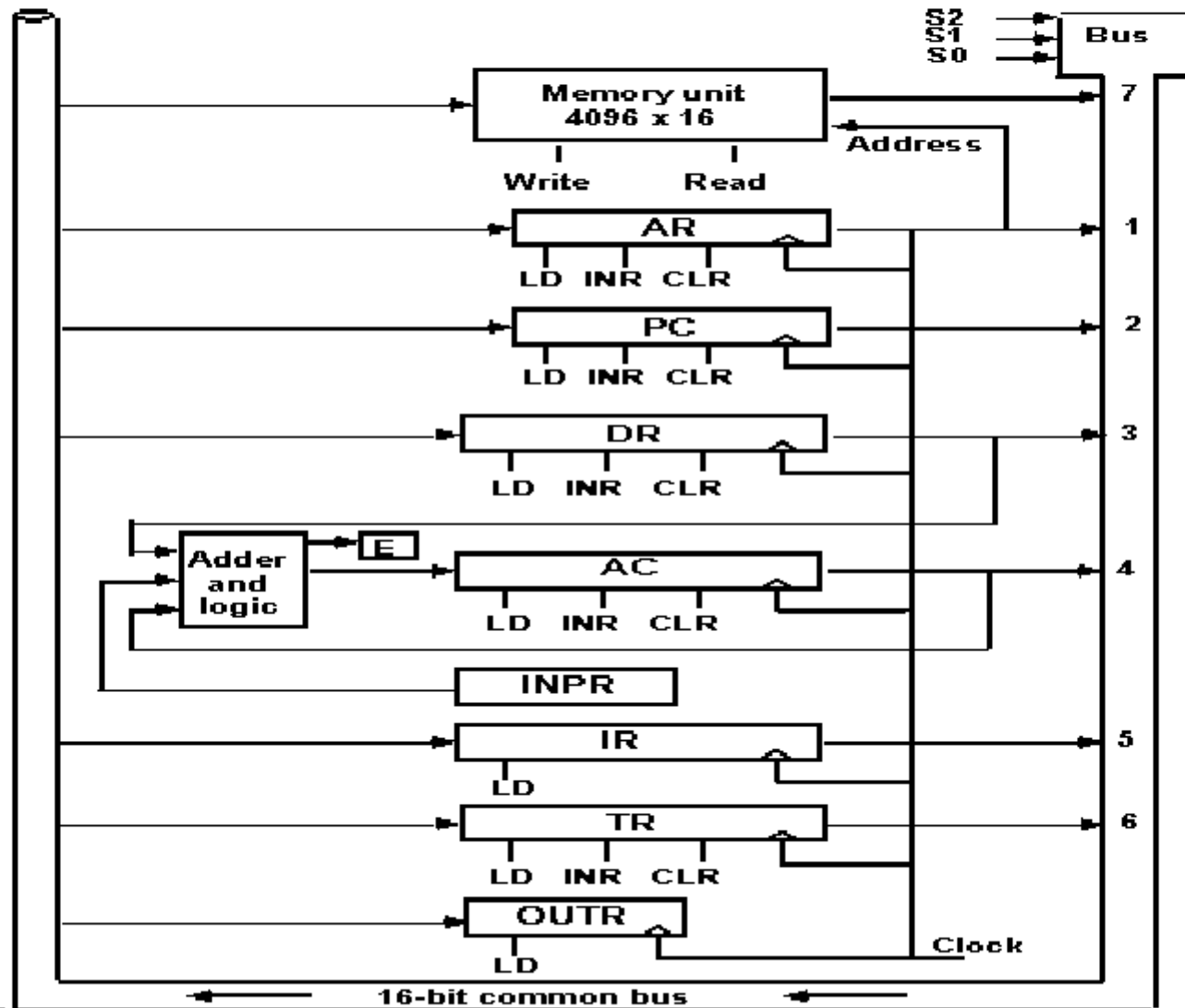STORED PROGRAM ORGANIZATION

PROCESSOR
REGISTERS

COMMON BUS SYSTEM

- BASIC COMPUTER HAS
  - **EIGHT REGISTERS**
  - **MEMORY UNIT**
  - **CONTROL UNIT**

- Path must be provided to transfer information from one register to another and between memory and registers.

- The number of wires will be excessive if connections are made between the o/p of each register and i/p of other registers.

- A more efficient scheme of transferring the information in a system with many registers is to use a common bus
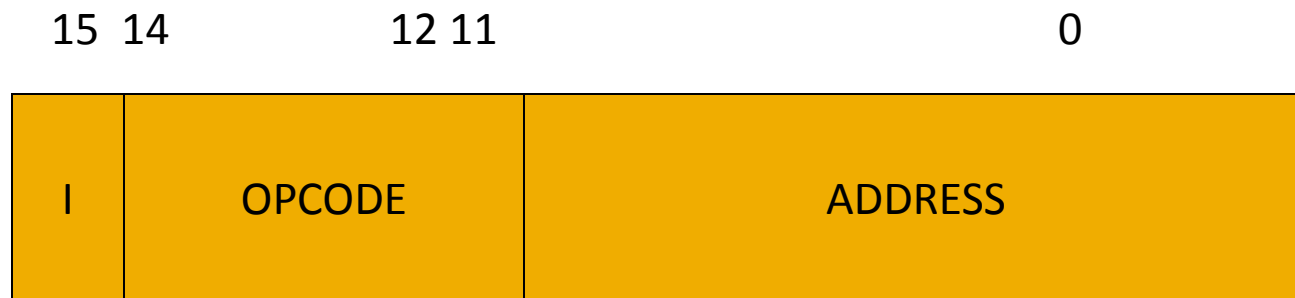
MEMORY

BUS

ADDRESS

AR

PC

DR

ADDER & LOGIC

AC

INPR

IR

TR

OUTR

COMMON BUS

COMMON BUS SYSTEM

S2
S1
S0

Bus

**Memory unit
4096 x 16** ─── 7

Address

Write   Read

AR ─── 1

LD INR CLR

PC ─── 2

LD INR CLR

DR ─── 3

LD   INR   CLR

**Adder
and
logic**   E

AC ─── 4

LD   INR   CLR

INPR

IR ─── 5

LD

TR ─── 6

LD   INR   CLR

OUTR

Clock

LD

← **16-bit common bus** ←

Computer Instructions

- Computer instruction code format has 16 bits

- OPCODE : part of the instruction contains three bits and the meaning of the 13 bits depends upon the operation code encountered.

15 14	12 11	0
I	OPCODE	ADDRESS

I is 0 for direct address and 1 for indirect address

Computer instruction are of three types

- Memory reference instruction
- Register reference instruction
- Input-output instruction
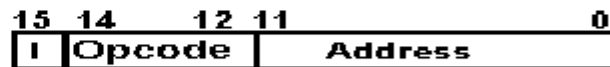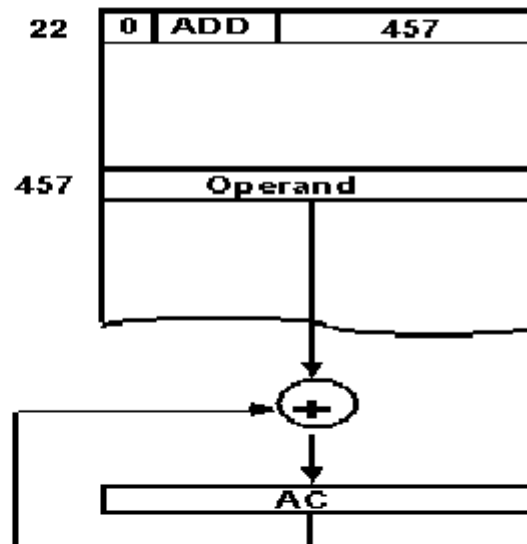
Memory reference instruction

15 14	12 11	0
I	OPCODE	ADDRESS

I is 0 for direct address and 1 for indirect address

INDIRECT ADDRESS

Instruction Format

15	14	12	11	0
I	Opcode		Address	

Direct Address

22	0	ADD	457
457		Operand	

Indirect address

35	1	ADD	300
300		1350	
1350		Operand	

+

AC
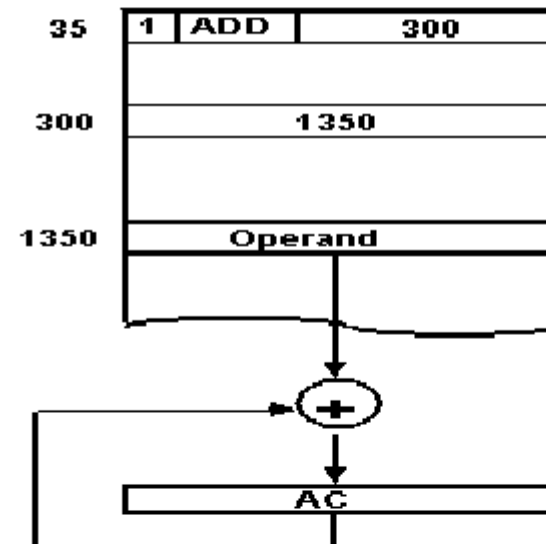
Effective Address(EFA, EA)

The address, that can be directly used without modification to access an operand for a computation-type instruction, or as the target address for a branch-type instruction

Register reference instruction

- They are recognized by the OPCOde 111 and 0 with the left most bit

- The other 12 bits specifies the operation.

- Register reference instruction specifies operation on register.

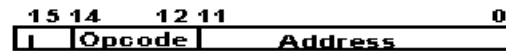- So, does not need any reference to memory

Input-output instruction

- They are recognized by the OPCOde 111 and 1 with the left most bit

- The other 12 bits specifies the operation.

- Input-output instruction does not need any reference to memory

COMPUTER(BC) INSTRUCTIONS

Basic Computer Instruction code format

Memory-Reference Instructions(OP-code = 000 ~ 110)

```
 15 14     12 11                      0
| I | Opcode |        Address          |
```

Register-Reference Instructions(OP-code = 111, I = 0)

```
 15         12 11                     0
| 0 1 1 1 |    Register operation      |
```

Input-Output Instructions (OP-code =111, I = 1)

```
 15         12 11                     0
| 1 1 1 1 |      I/O operation          |
```