Computer Architecture ECS-355

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

(Week 4) Lecture 7-8

Learning objectives:

- Instruction codes: To understand how instructions and data share memory for execution and how an address points to another location for data access.
- Computer Registers: To understand how multiple registers share a single bus system for efficient data transfer within the CPU.
- Computer Instructions: To understand the meaning of computer instruction with respect to the processor and execution over the processor and what are the basic types of instructions.
- Instruction cycle: To understand the instruction execution cycle that how a processor executes the whole program instruction by instruction.
- Timing of processor w.r.t to instruction execution over the processor.
- To understand the how a processor determine a specific type of instruction that intends to be processed.
- Finally, how a specific instruction is determined by the processor and what are the steps a processor takes to execute a specific instruction.

(Note: It is important to note that we are considering "Basic computer architecture" processor which we were discussing in the class and it's the continuation of the same concepts).

Resources: Beside these lecture handouts, this lesson will draw from the following

Text Book: Computer System Architecture by Morris Mano (3rd Edition) and
Reference book:Computer Architecture, by William Stallings (4th Edition).

Computer Architecture ECS-355

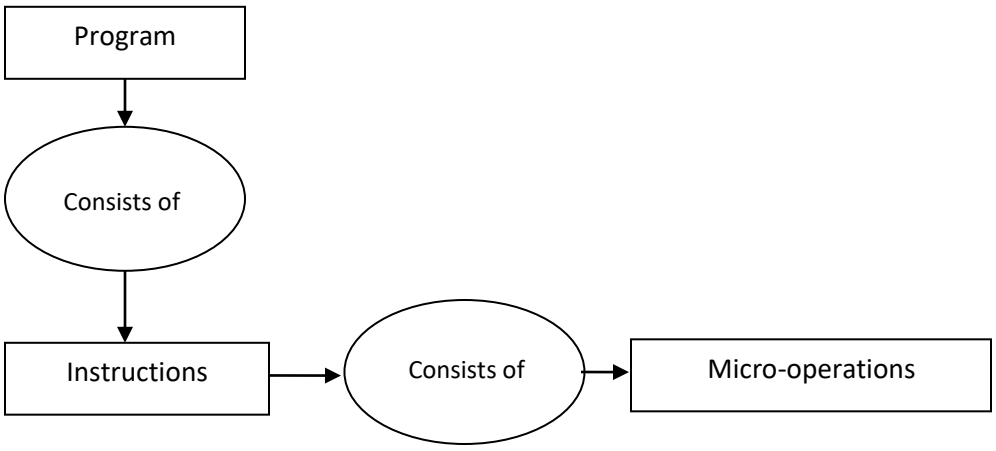
Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

Lecture:

COMPUTER ORGANIZATION

Since we have sufficient knowledge of digital computer components like ALU, Shift unit, memory unit, flip-flops, registers, common bus etc, we are able to construct a basic digital computer. Now we move our attention towards more practical sort of things that is now we need to put all these components together to get what we want? In this manner we first see those particular codes which define what a computer should do? Whenever we are intended somebody to do something we have to instruct him. So similarly, we shall instruct computer what to do? To accomplish this we firstly see what a computer instruction is, latterly we shall see its format and organization.

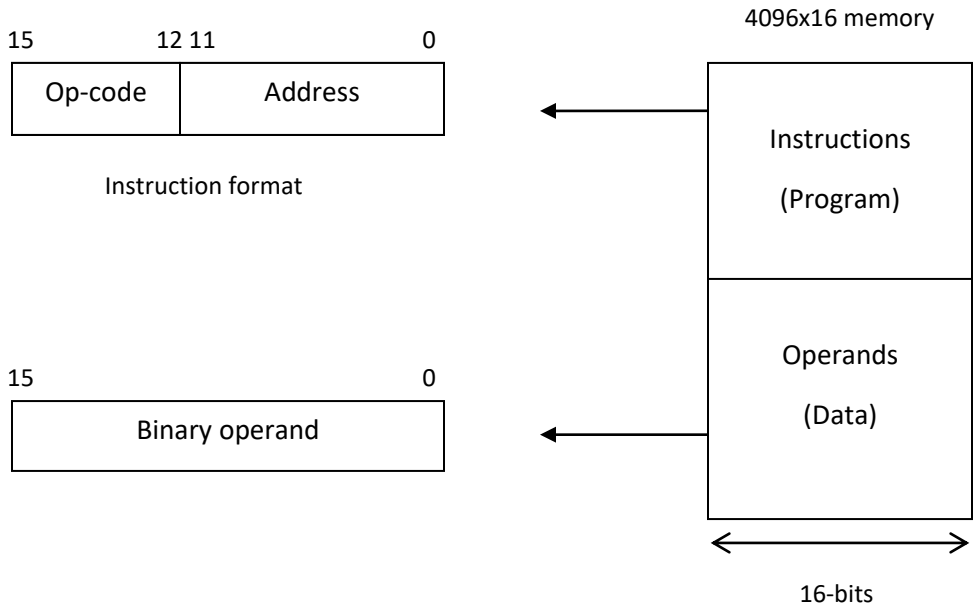
INSTRUCTION CODES AND ORGANIZATION



As above picture mentioned, a computer *program* is combinations of *instructions*, while each instruction is further combination of *micro-operations*. A program is written in a format which is *machine understandable* called *language*. So language is software which let computer to

understand what we want computer to perform? An *instruction code* is a group of bits that instructs the computer to perform specific operation. It is usually divided into parts, each having its own particular interpretation.

This most basic part of an instruction code is *operation code (op-code)*, which specifies the type of instruction; hence computer instructions can be categorized according to op-code.



Stored program organization

Initially a program (set of instruction) is loaded into the memory, where each word of the memory is addressed by some specific sequence. Let we have a memory of size 4096x16, where 16 is size of memory word, and there are $4096=2^{12}$ such words each distinguished by 12 bit long, unique code called *address*. Since a program consisted of instructions so these instructions are loaded inside the memory words specified by some specific address. Also the operands (data) for the program are loaded into the memory.

An instruction, in our discussion consisted of 16 bit long sequence of bits, which is normally fetched in ***IR*** **(Instruction Register)**. Following is given the description. As figure describes an instruction consists of 12 bit long address field and a four bit long op-code. Here address specifies one of address out of 4096 cells of memory, where corresponding operand resides.

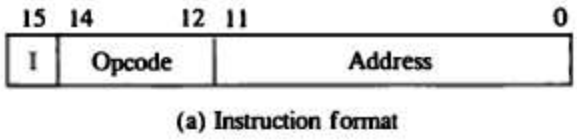
Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

Similarly, the other operand resides inside the ***accumulator*** **AC** register of CPU. And op-code describes type of instruction to be executed.

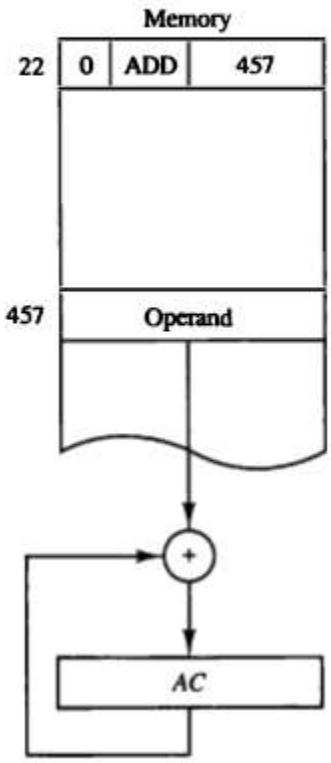
Addressing Modes

It is sometimes convenient to use the address bits of an instruction code not as an address but as the actual operand. When the second part of an instruction code specifies an operand, the instruction is said to have an **immediate operand**. When the second part specifies the address of an operand, the instruction is said to have a **direct address**. This is in contrast to a third possibility called **indirect address**, where the bits in the second part of the instruction designate an address of a memory word in which the address of the operand is found. **One bit** of the instruction code can be used to distinguish between a direct and an indirect address.

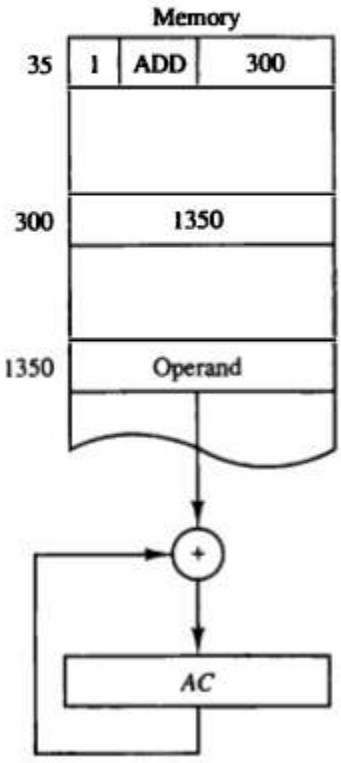
Basic Computer Organization (Instruction format, direct and indirect addressing)



(a) Instruction format



(b) Direct address



(c) Indirect address

Computer Architecture ECS-355

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

Computer Registers

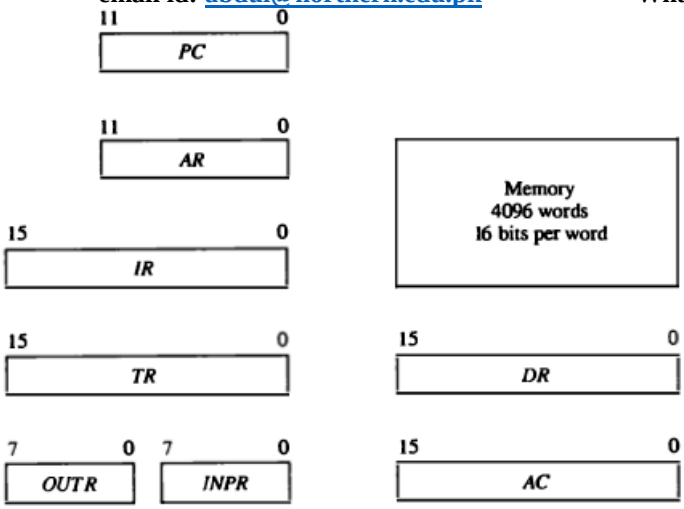
Computer instructions are normally stored in consecutive memory locations and are executed sequentially one at a time. The control reads an instruction from a specific address in memory and executes it. It then continues by reading the next instruction in sequence and executes it, and so on. This type of instruction sequencing needs a counter to calculate the address of the next instruction after execution of the current instruction is completed. It is also necessary to provide a register in the control unit for storing the instruction code after it is read from memory. The computer needs processor registers for manipulating data and a register for holding a memory address. These requirements dictate the register configuration shown in Fig. 5-3. The registers are also listed in Table 5-1 together with a brief description of their function and the number of bits that they contain.

The memory unit has a capacity of 4096 words and each word contains 16 bits. Twelve bits of an instruction word are needed to specify the address of an operand. This leaves three bits for the operation part of the instruction and a bit to specify a direct or indirect address. The data register (DR) holds the operand read from memory. The accumulator (AC) register is a general purpose processing register. The instruction read from memory is placed in the instruction register (IR). The temporary register (TR) is used for holding temporary data during the processing.

Register symbol	Number of bits	Register name	Function
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

Computer Architecture ECS-355

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

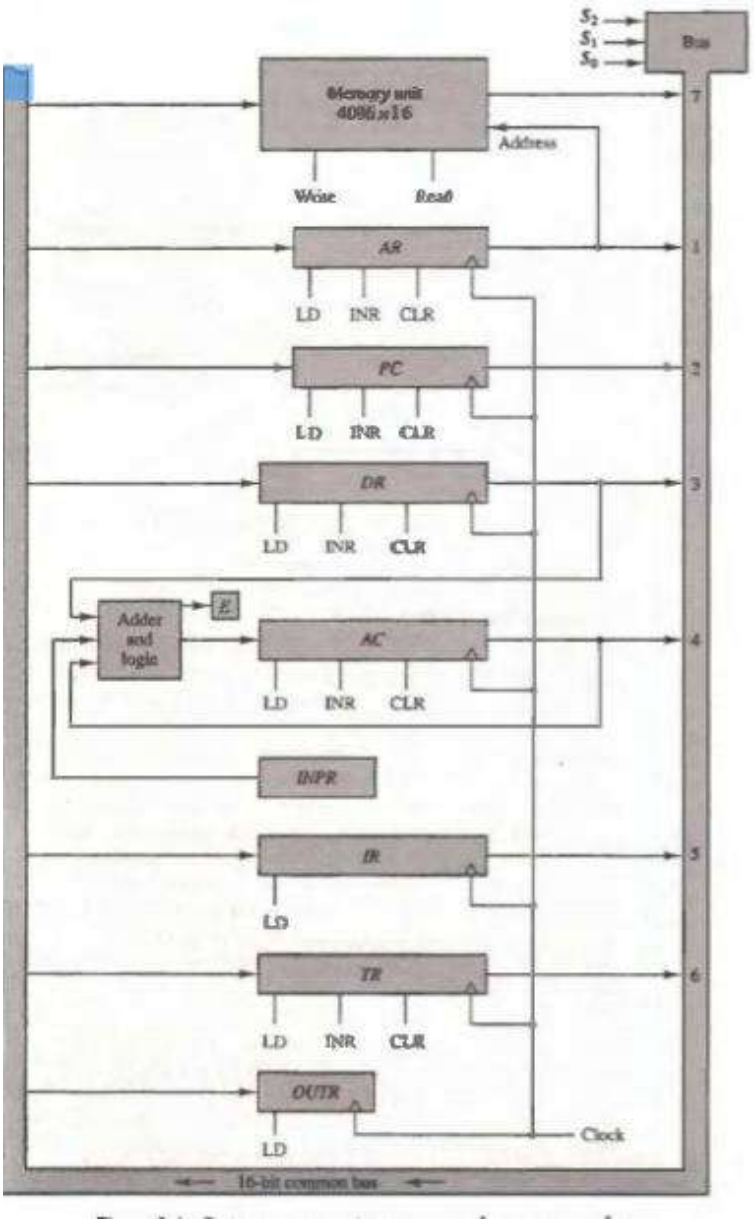


Common Bus System

The basic computer has eight registers, a memory unit, and a control unit. Paths 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 outputs of each register and the inputs of the other registers. A more efficient scheme for transferring information in a system with many registers is to use a common bus. We have shown in figure below how to construct a bus system using multiplexers or three-state buffer gates.

Computer Architecture ECS-355

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**



Examples discussion

A computer uses a memory unit with 256k words of 32 bits each. A binary instruction code is stored in one word of memory. The instruction has four parts: an indirect bit, an operation code, a register code part to specify one of 64 registers, and an address part

1. How many bits are there in the operation code, the register code part, and the address part?
2. Draw the instruction word format and indicate the number of bits in each part?
3. How many bits are there in the data and address inputs of the memory?

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

Solution:

256 K = $2^8 \times 2^{10} = 2^{18}$

64 = 2^6

(a) Address: 18 bits

Register code: 6 bits

Indirect bit: 1 bit

25 32 – 25 = 7 bits for opcode.

(b) Instruction format

1	7	6	18

(c) Data: 32 bits

Address: 18 bits

COMPUTER INSTRCUTIONS

Computer instructions are those particular codes that put a computer on work. A computer program consists of a number of instructions, while instructions consist of micro-operations.

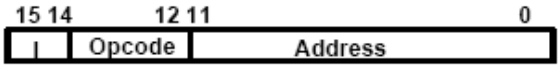
The contents of IR are interpreted in such a way that each chunk of that register reveals some specific code of information. Especially the 4 most significant bits are designated as operation code. Where 3 bits reveal the operation specified and last *IR (15)* tells about the nature of instruction.

Computer Architecture ECS-355

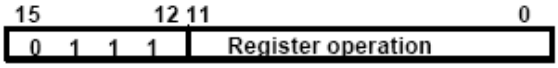
Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

Basic Computer Instruction code format

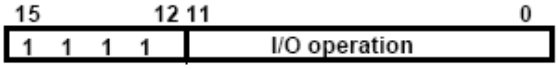
Memory-Reference Instructions (OP-code = 000 ~ 110)



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



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



Opcode bits (11-14) are between (000-110) then they are called *memory reference instructions*, where for 15th bit if zero then *direct address* otherwise *indirect address*. Similarly, if opcode =111 then this is a *non-memory reference* instruction, further if 15th bit is 0 then *register reference* otherwise *input/output* instruction.

Here are given the total number of instruction of the basic computer with their codes and function associated.

Symbol	Hex Code I = 0	Hex Code I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instr. if AC is positive
SNA	7008		Skip next instr. if AC is negative
SZA	7004		Skip next instr. if AC is zero
SZE	7002		Skip next instr. if E is zero
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

Computer Architecture ECS-355

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

A computer should have a set of instructions so that the user can construct machine language programs to evaluate any function that is known to be computable.

Instruction Types

- Functional Instructions
  - Arithmetic, logic, and shift instructions
  - ADD, CMA, INC, CIR, CIL, AND, CLA
- Transfer Instructions
  - Data transfers between the main memory and the processor registers
  - LDA, STA
- Control Instructions
  - Program sequencing and control
  - BUN, BSA, ISZ
- Input/Output Instructions
  - Input and output
  - INP, OUT

TIMING AND CONTROL

Another important feature associated with the instructions and their execution is timing and control. An instruction is executed in a sequential way, as they are loaded into the main memory.

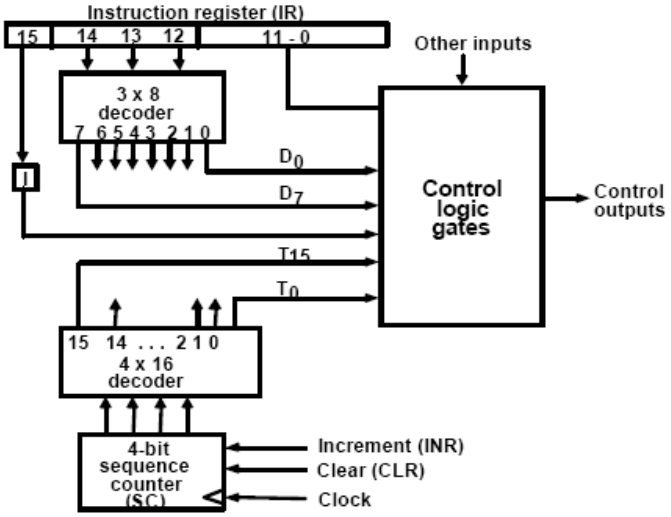
Further an instruction needed certain steps from fetching to execution. Also each instruction is associated with some register and memory operations, we know that a single common bus is available to communicate among them. It means we need entire bus for a single operation; and if there are a lot of operations executing in a sequential way then there must be some timing and control, in order to prevent from collisions.

The circuits that produce these timing or control signals are associated with instruction so that to provided sufficient controls.

Computer Architecture ECS-355

Mr. Abdul Rehman **email id: abdul@northern.edu.pk** **Whatsapp# 0308-7792217**

Control unit of basic computer



Control unit implementation

Hardwired Implementation

Microprogrammed Implementation

A control unit is capable of doing all these tasks as shown in figure above. Here 4-bit sequence counter is given which can generate some value of hexadecimal which is further decoded in one of the 16 discrete values. These values are connected with control logic gates. The operations associated are increment, clearing the contents and clock for synchronization purposes.

Similarly, from above circuit opcode bits are decoded to any of 8th bit also mode bit is input into the control logic unit, which generates control signals.

- Generated by 4-bit sequence counter and 4x16 decoder
- The SC can be incremented or cleared.

- Example: $T_0, T_1, T_2, T_3, T_4, T_0, T_1, \ldots$
  Assume: At time T_4, SC is cleared to 0 if decoder output D3 is active.

D_3T_4: SC ← 0