St. Peter's Engineering College (Autonomous) Dullapally (P), Medchal, Hyderabad – 500100. QUESTION BANK						Dept.	:	
						Academic Year 2024-25		
Subject Code	:	AS22-05PC06	Subject	:	COMPUTER ORGANIZATION AND ARCHITECTURE			
Class/Section	:	B. Tech.	Year	:	II	Semester	:	1

BLOOMS LEVEL					
Remember	L1	Understand	L2	Apply	L3
Analyze	L4	Evaluate	L5	Create	L6

*****

Q. No	Question (s)	Marks	BL	CO
	UNIT – I			
1	**a.** What is computer architecture?	1M		
	b. Define Computer Organization.	1M		
	c. What is ALU?	1M		
	d. Define Register.	1M		
	e. Name the list of microoperations	1M		
2	**a.** Differentiate Computer organization and architecture.	3M		
	b. Give any 3 applications of logic microoperations	3M		
	c. Differentiate hardwired and micro-programmed control organizations.	3M		
	d. What is an arithmetic micro-operation?	3M		
	e. Draw Arithmetic Logic Shift Unit.	3M		
3	**a.** Draw the block diagram of digital computer and explain its components.	5M		
	b. Explain logical shift microoperations.	5M		
	c. Explain in detail about the Memory Transfer.	5M		
	d. Explain the hardware implementation of 4-bit shift operation.	5M		
	e. Write the list of logic microoperations	5M		
4	**a.** Explain in detail the Fetch-Phase of instruction cycle	10M		
	b. Explain in detail about arithmetic micro-operations.	10M		
	c. With the help of block diagram, explain the input-output instructions	10M		

ANSWERS:

1a. Define Computer Architecture.
Ans: Computer Architecture is a functional description of requirements and design implementation for the various parts of a computer. From programmers point of view, computer architecture is viewed as series of instructions, addressing modes, and registers

1b. Define Computer Organization?
Ans: Computer Organization deals with a structural relationship. Computer Organization consists of physical units like circuit designs, peripherals, and adders. From the programmer point of view, computer organization is viewed as implementation of the architecture

1c. What is ALU?
Ans: Arithmetic and Logic unit is a digital component basically used to perform arithmetic, logical and shift operations. ALU is part of central processing unit (CPU)

1d. Define Register
Ans: A *REGISTER* is a digital component used to store digital information in the form of 0's and 1's. The storage capacity of the register depends on number of flip-flops used to design it. For example, a 8-bit register has 8 flip-flops and hence can store 8-bits of information.

1e. Name the list of microoperations
Ans: The list of microoperations is:

i. Register Transfer microoperations
ii. Arithmetic microoperations
iii. Logic microoperations
iv. Shift microoperations

2a. **Differentiate Computer organization and architecture.**
Ans:

Computer architecture	Computer organizations
Computer Architecture is a functional description of requirements and design implementation for the various parts of a computer	Computer Organization deals with a structural relationship
Computer Architecture describes what the computer does	The Computer Organization describes how it does it
Computer Architecture comprises logical functions such as instruction sets, registers, data types, and addressing modes	Computer Organization consists of physical units like circuit designs, peripherals, and adders
As a programmer, computer architecture is viewed as series of instructions, addressing modes, and registers	The implementation of the architecture is viewed as computer organization

2b. Give any 3 applications of logic microoperations

Ans: **selective-set**: the bits of registerA are set to 1where there are corresponding 1's in registerB. It doesn't affect the bits of registerA that has 0's in registerB.

RegisterA	1 0 1 1	1 1 0 0	1 0 1 0
RegisterB	0 1 1 0	1 0 1 0	1 1 0 0
RegA after selective set	**1 1 1 1**	**1 1 1 0**	**1 1 1 0**

From analysis, we understand that the bits in A after selective set is obtained from the OR microoperations of bits in B with the corresponding previous values of A

selective-complement: it complements the bits of A where there are corresponding 1's in B. It doesn't affect the bit positions that have 0 in B.

RegisterA	1 0 1 1	1 1 0 0	1 0 1 0
RegisterB	0 1 1 0	1 0 1 0	1 1 0 0
RegA after selective complement	**1 1 0 1**	**0 1 1 0**	**0 1 1 0**

From analysis, we understand that an XOR operation is performed on A and B.

selective-clear: it clears the bits of A to 0 where there are corresponding 1's of B

RegisterA	1 0 1 1	1 1 0 0	1 0 1 0
RegisterB	0 1 1 0	1 0 1 0	1 1 0 0
RegA after selective clear	**1 0 0 1**	**0 1 0 0**	**0 0 1 0**

From analysis, we understand that logic microoperation is $\mathbf{A} \wedge \mathbf{B}'$

2c. Differentiate hardwired and micro-programmed control organizations

Ans:

Hardwired Control	Microprogrammed control
The control logic is implemented using the logic gates, multiplexers, decoders and other digital circuits.	Control information is stored in the control memory
Design is optimized for faster operation	The control memory is programmed to initiate the microoperations
For any modifications we need to make changes to wires connected between different components and may have to add/remove the components.	For any modifications, we just need to reprogram
Because of regular modifications, the design is complex	The design here is simple

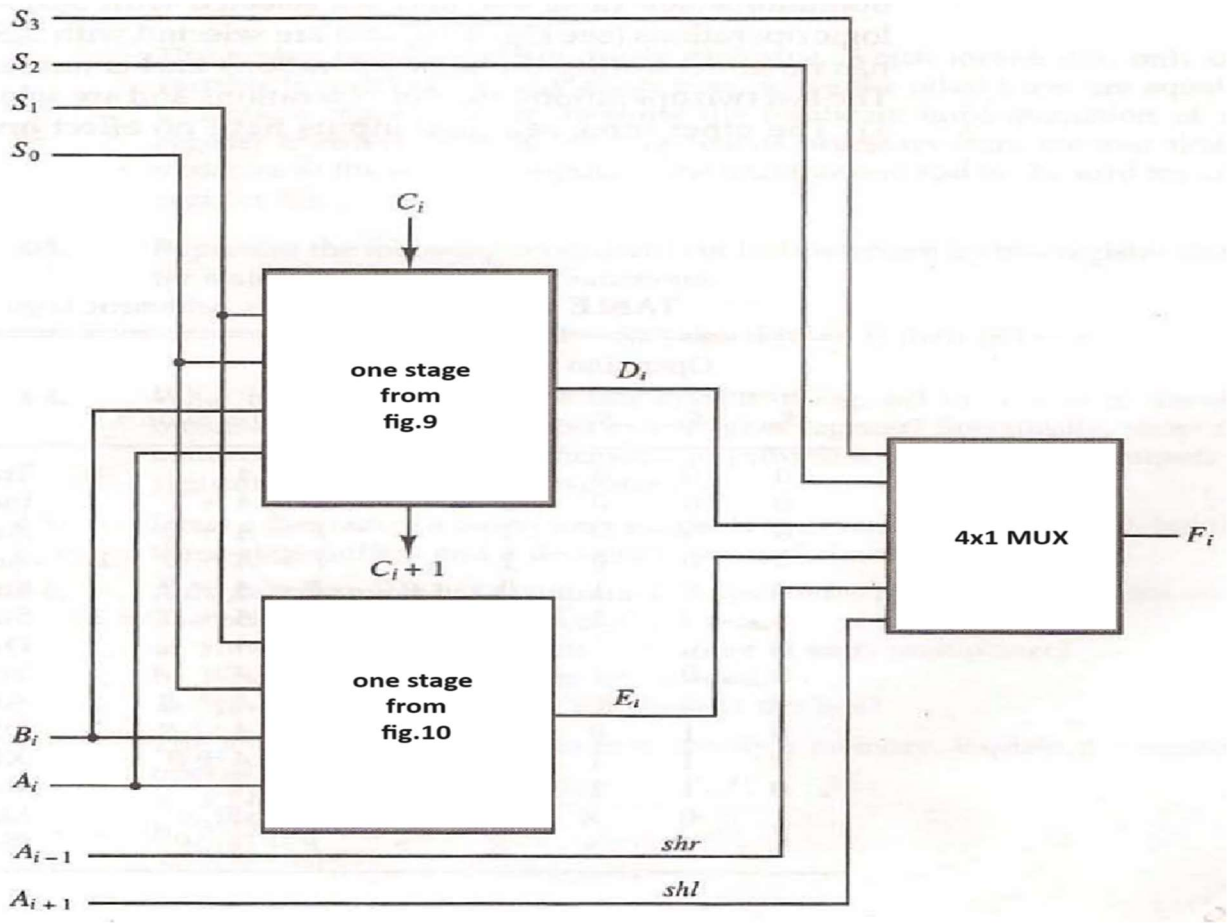
2d. What are arithmetic micro-operation?

Ans:

Basic arithmetic operations are addition, subtraction, increment, decrement. Other arithmetic operations like multiplication and division are performed using addition with shift and subtraction with shift respectively.

2e. Draw Arithmetic Logic Shift Unit.

Ans:



3a. Draw the block diagram of digital computer and explain its components
Ans:

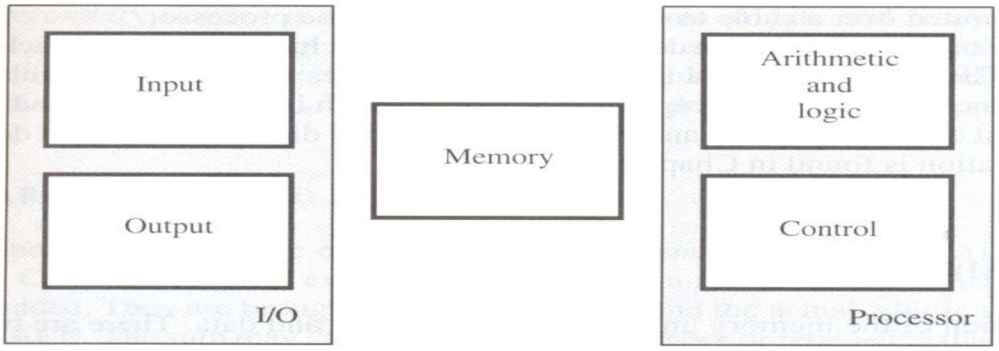


Figure 1.1 Basic functional units of a computer.

This diagram represents the main components of a computer system and their connections. The CPU coordinates operations, memory stores data and instructions, input devices accept user inputs, output devices present information, and various connectors and buses facilitate

Components of a Computer System:

1. **Central Processing Unit (CPU):** The CPU is the brain of the computer responsible for executing instructions. It consists of:
   - **Control Unit (CU):** Directs and coordinates operations within the CPU.
   - **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical operations on data.
   - **Registers:** Temporarily stores data, instructions, and addresses being processed.
2. **Memory:**
   - **RAM (Random Access Memory):** Provides volatile storage for data and instructions that the CPU can quickly access.
   - **ROM (Read-Only Memory):** Holds firmware and instructions that remain intact even when the computer is powered off.
3. **Input Devices:**
   - **Keyboard, Mouse, Touchpad:** Allow users to input data and commands into the computer system.
   - **Microphone, Webcam:** Capture audio and video inputs.
4. **Output Devices:**
   - **Monitor, Display:** Present visual information to the user.
   - **Printer, Speakers:** Produce hard copies or audio outputs.
5. **Storage Devices:**
   - **Hard Disk Drive (HDD), Solid State Drive (SSD):** Provide non-volatile storage for data, programs, and the operating system.

- **External Storage (USB Drives, External HDDs):** Enable portable data storage.

6. **Motherboard:**
   - **Connects all components:** Provides pathways for communication between CPU, memory, storage, and peripheral devices.
   - **Includes ports and slots:** USB ports, expansion slots, audio jacks, etc.
7. **Expansion Cards:**
   - **Graphics Card (GPU):** Handles graphical computations and output.
   - **Network Interface Card (NIC):** Facilitates communication over a network.
8. **Power Supply Unit (PSU):**
   - **Supplies power:** Converts AC power to DC power for the computer components.
9. **Bus System:**
   - **Data Bus:** Transfers data between CPU, memory, and peripherals.
   - **Address Bus:** Carries addresses of data locations in memory.
   - **Control Bus:** Manages communication and control signals between components.

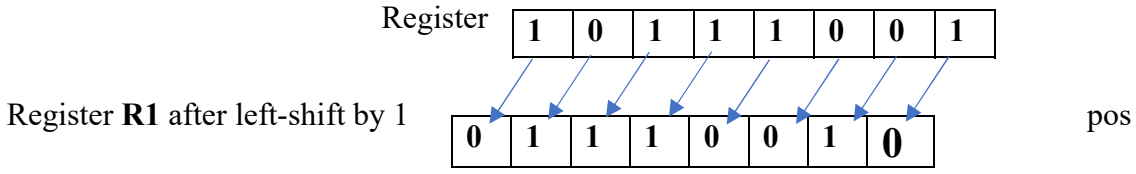
3b. Explain logical shift microoperations

Ans:

a. Logical Shift: Logical shift is one which transfers 0 through the serial data-in. *shl* and *shr* represents logical shift left and logical shift right respectively. The content of RegisterA is shifted either to the left or to right as per the instruction. Note: Source and the destination are the same register.

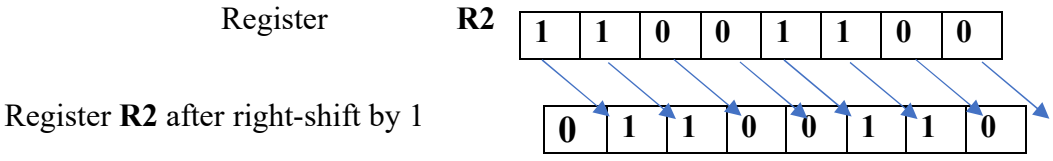
Shown below is an example.

R1 ← shl R1 shifts the content of Register R1 to the left by one position and a 0 is placed in the leftmost bit



pos

R2 ← shr R2 shifts the content of Register R1 to the left by one position and a 0 is placed in the leftmost bit



3c. Explain in detail about the Memory Transfer

Ans:

Transfer of data from memory to outside world is called as READ operation. And, the transfer of data into memory is called as WRITE operation. Memory word is symbolized with **M.** A particular word from the memory is identified by the memory address. It is customary to specify the memory address M during memory transfer operation in square brackets.

Consider a memory unit receives address from register, called address register (AR) and the data is transferred to another data register (DR), this constitute a read operation which is represented as

Read: DR← M[AR]

This causes a transfer of information into DR from the memory word M selected by address AR.

The write operation transfers the data from data register DR to the memory word M specified by the address AR, this write operations is represented as

Read: M[AR]← R1

This causes transfer of information from R1 register into memory word M specified by address AR

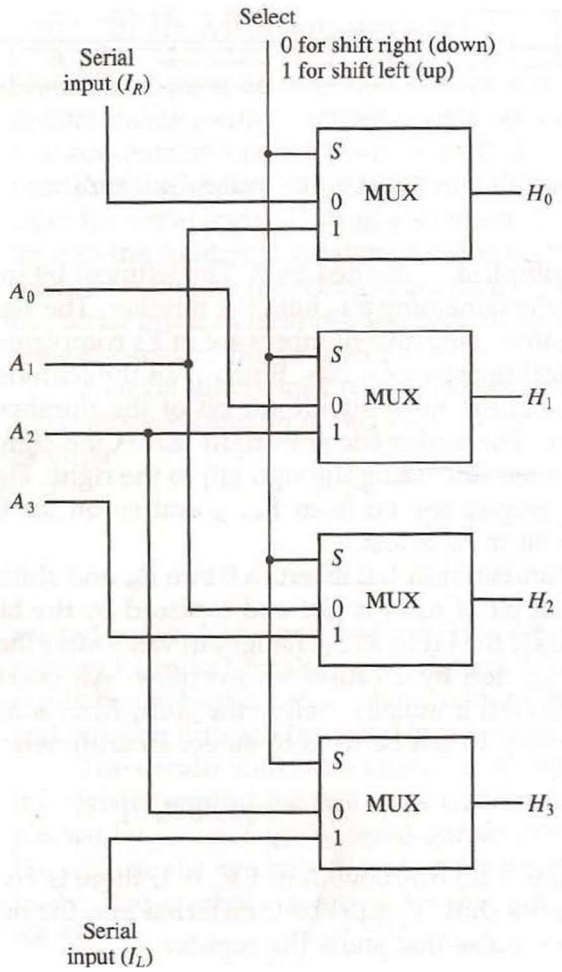
3d. Explain the hardware implementation of 4-bit shift operation.

Ans:

A shifter can be constructed using multiplexers. since we shift ether to the left or to the right, we need 2x1 mux only. Considering 4-bit registerA, we need four 2x1 mux as shown below. When the Select signal is 0, the circuit performs a shift right and when 1 it performs left shift operation. The arrangement of the inputs to multiplexer will depend on the bit postions taken after the shift operations.

During right shift, Bit -A0 moves out and during left shift it moves to A1 position
During right shift Bit-A1 moves into A0 position and during left shift it moves to A2 position
During right shift Bit-A2 moves into A1 position and during left shift it moves to A3 position
During right shift Bit-A3 moves into A2 position and during left shift it moves out

Based on the above analysis the inputs A0,A1,A2,A3 are connected accordingly to the multiplexers.



Function table

Select	Output			
S	H_0	H_1	H_2	H_3
0	I_R	A_0	A_1	A_2
1	A_1	A_2	A_3	I_L

3e. Write the list of logic microoperations

Ans:

There are 16 possible logic microoperations.

Boolean	Microoperation	Description/Name
$F_0 = 0$	F ← 0	Clear
$F_1 = xy$	F ← A ∧ B	AND
$F_2 = xy'$	F ← A ∧ B'	
$F_3 = x$	F ← A	Transfer A
$F_4 = x'y$	F ← A' ∧ B	
$F_5 = y$	F ← B	Transfer B
$F_6 = x \oplus y$	F ← A ⊕ B	
$F_7 = x + y$	F ← A ∨ B	OR
$F_8 = (x + y)'$	F ← (A ∨ B)'	NOR
$F_9 =$ (x xor y)'	F ← (A xor B)'	Exclusive NOR
$F_{10} = y'$	F ← B'	Complement B
$F_{11} = x + y'$	F ← A ∨ B'	
$F_{12} = x'$	F ← A'	Complement A

F_{13} = x' + y	F ← A' ∨ B	
F_{14} = (xy)'	F ← (A ∧ B)'	NAND
F_{15} = 1	F ← 0	All 1's

4a. Explain in detail the Fetch-Phase of instruction cycle

Ans:

Initially ***PC is loaded with address of first instruction in the program*** *and* ***SC is cleared to 0,*** when SC =0, the timing signal T_0 from 4x16 decoder is connected to control logic circuit. For each clock pulse, SC is incremented by 1 and the timing signals T_1, T_2, T_3, …… are made available to control logic gate. The following statements explains the fetch and decode phases

T_0: AR← PC

T_1: IR← [AR], PC=PC+1

T_2: D_0, D_1, …. D_7 ← Decode (12-14), AR←IR (0-11), I ← IR (15)

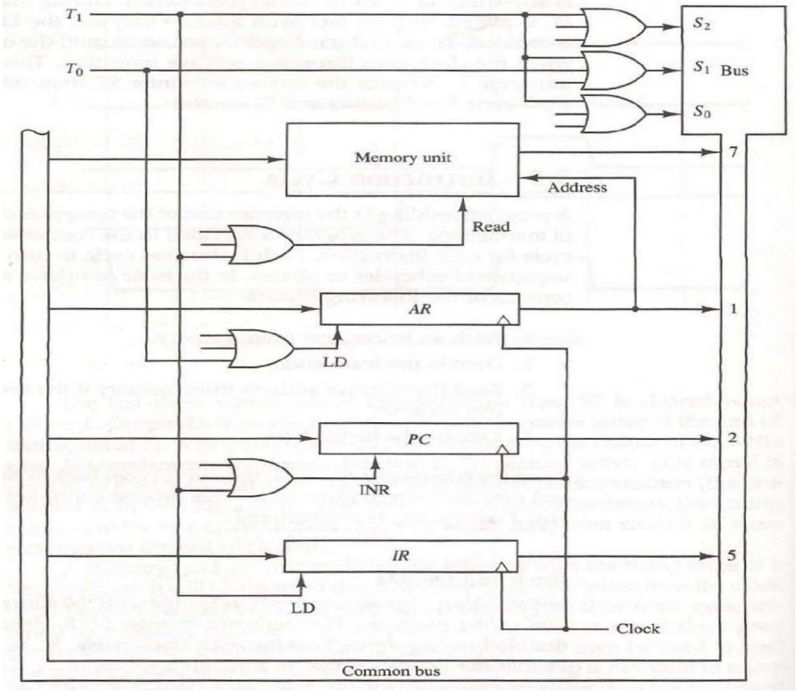


Fig 4: Flow of Fetch Instruction

The fig. 4 show how a fetch phase is processed.

The operation of fetch cycle starts with loading of the address of first instruction onto PC and setting the SC to 0. With SC=0 and rising edge of clock pulse the decoded timing signal is T_0. Subsequent clock cycle will generate the timing signals T_+,T_2,T_3, ….. , T_{15} .

When T0=1:

- ***$S_2S_1S_0$= 010***, PC is selected
- With T0=1, LD pin of AR is 1 which makes the information present on common bus to be loaded onto AR.
- Thus, **AR← PC**

We now need to place the instruction code from the memory into IR register.

When T1=1:

- ***$S_2S_1S_0$= 111***, memory is selected

- Read pin of memory =1, this indicates that a read operation is to be done from memory
- The data present in memory location indicated by AR is placed on the common bus
- At the same time LD pin of PC is 1, hence **PC←PC+1**
- Also, *LD=1* for IR register, hence the data present of common bus is moved to IR.
- Thus, **IR←[AR]** and **PC←PC+1**

When T2=1:

The instruction code is decoded as follows

- Bits 12-14 is decoded as D0, D1, …. D7: **D0, D1, …. D7 ← Decode IR(12-14)**
- Bit 15 refers the type of memory addressing mode: **I ← IR (15)**
- Bit 0-11 refers the address bits: **AR←IR (0-11)**,

At the end of T1, the first instruction is placed in AR. At the end of T2, decoding of instructions code is completed. The subsequent timing signals will then search for the operands and execute the instructions.

After the execution of every instruction SC should be cleared to zero **SC←0.**

4b. Explain in detail about arithmetic micro-operations.

Ans:

Basic arithmetic operations are addition, subtraction, increment, decrement. Other arithmetic operations like multiplication and division are performed using addition with shift and subtraction with shift respectively.

add microoperations:

consider the add microoperation shown below

$$R3 \longleftarrow R2 + R1$$

It performs the addition operation on the content in register R2 with the content in register R1 and the sum result is transferred into Register R3. To implement this with hardware we require and adder circuit and three registers.

subtract microoperations:

consider the subtract microoperation shown below

$$R3 \longleftarrow R2 - R1$$

It subtracts the content in register R1 from the content in register R1 and the difference result is transferred into Register R3. To implement this with hardware we require a subtractor circuit and three registers.

But, subtractor circuit utilizes more hardware when compared with adder. To minimize the circuitry, we will implement subtraction operation using addition operation.

In digital circuits the above operation is implemented as shown.

$$R3 \longleftarrow R2 + \overline{R1} + 1$$

$\overline{R1}$ is 1's complement of R1

$\overline{R1} + 1$ is the 2's complement of R1

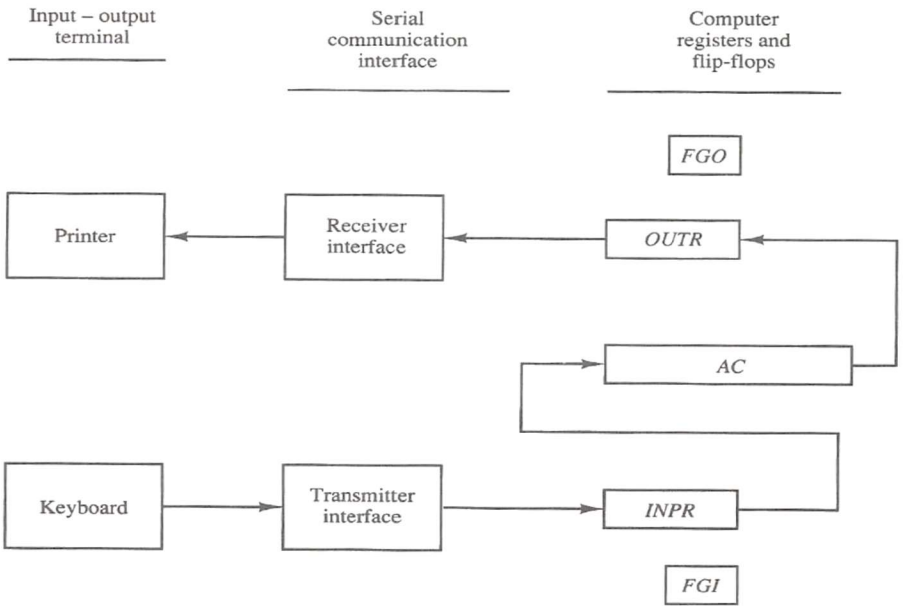
Increment and Decrement microoperations:
Increment and Decrement microoperations are just +1 and -1 operations to the register and are implemented with up-down counters.

Multiplication and Division also forms part of arithmetic operations, but are implemented using adder and subtractor circuits. Multiplication is implemented with a sequence of add and shift operations. Division is implemented with a sequence of subtract and shift operations.

4c. With the help of block diagram, explain the input-output instructions

Instructions and data stored in the memory has to come from some input device and the computational results has to be sent to the output device.

The fig below shows the block diagram of flow of input and output data. It uses keyboard as input device and printer as output device. Each quantity of information is 8-bit in length. INPR is the input register used to stored the input data and OUTR is the output register used to store the output data. Alphanumeric data is transferred serially from keyboard to INPR, but transmitted parallelly from INTR to AC. Similarly, data is loaded parallelly into OUTR and transmitted serially from OUTR to printer.



When we strike the keyboard, the b-bit alphanumeric data is transmitted serially and placed in 8-bit INPR. Before this data is loaded to AC if another keyboard strike occurs, the older data is replaced with new data. To avoid this, a control flipflop FGI (input flag) is used at the input end. Similarly, a flipflop FGO is used at the output end.

Initially FGI is 0, as long as FGI=0 the data can be sent through keyboard. This data is transmitted serially and placed on the INPR register. At this moment, the FGI should be made 1 to avoid any new input data from keyboard. When FGI=1, the data in the INPR is loaded to AC parallelly. Once the data is loaded to INPR, FGI is cleared to zero.

At the output end the computer checks for FGO. If FGO = 1, the data from AC is loaded to OUTR parallelly. At this time, FGO is reset to 0 so that the data on OUTR can be transmitted serially to printer.