2066

Long questions $(2 \times 10 = 20)$

1. Explain the different types of addressing modes and compare each other.

Addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced.

Types of addressing modes:

Implied mode:

Address of the operands is specified implicitly in the definition of the instruction.

- No need to specify address in the instruction
- Example from Basic Computer CLA, CME, INP

  ADD X;

  PUSH Y;

Immediate mode:

Instead of specifying the address of the operand, operand itself is specified in the instruction.

- No need to specify address in the instruction
- However, operand itself needs to be specified
- Sometimes, require more bits than the address
- Fast to acquire an operand

Register mode:

Address specified in the instruction is the address of a register.

- Designated operand need to be in a register
- Shorter address than the memory address
- A k-bit address field can specify one of 2^k registers
- Faster to acquire an operand than the memory addressing

Register indirect mode:

Instruction specifies a register which contains the memory address of the operand.

- Saving instruction bits since register address is shorter than the memory address
- Slower to acquire an operand than both the register addressing or memory addressing
- EA (effective address) = content of R

Auto increment or Auto decrement mode:

It is similar to register indirect mode except that the register is incremented or decremented after (or before) its value is used to access memory. When address stored in the register refers to a table of data in memory, it is necessary to increment or decrement the register after every access to the table.

Direct addressing mode:

Instruction specifies the memory address which can be used directly to access the memory

- Faster than the other memory addressing modes
- Too many bits are needed to specify the address for a large physical memory space
- EA = IR (address)

Indirect addressing mode:

- The address field of an instruction specifies the address of a memory location that contains the address of the operand
- When the abbreviated address is used large physical memory can be addressed with a relatively small number of bit
- Slow to acquire an operand because of an additional memory access
- EA = M[IR (address)]

Relative addressing mode:

The Address field of an instruction specifies the part of the address which can be used along with a designated register (e.g. PC) to calculate the address of the operand.

- Address field of the instruction is short
- Large physical memory can be accessed with a small number of address bits

3 different relative addressing modes:

- PC Relative addressing mode: EA=PC+IR(address)
- Indexed addressing mode: EA=IX+IR(address) {IX is index register}
- Base register addressing mode: EA=BAR+IR(address)

2. What are the major differentiating between I/O bus and Interface modules? What are the advantage and disadvantages of each?

Peripherals connected to a computer need special communication link to interface with CPU. This special link is called I/O bus. I/O bus consists of Data Lines, address and Control lines. To communicate with a particular device, processor place a device address on the address line

whereas control line have function code provided by the processor called I/O command. And data line contains the data that are accepted by the processor. Each peripheral has an interface module associated with its interface which decodes the device address and provides the signal for the peripheral controller.

Given figure gives clear idea about I/O bus and interface module:

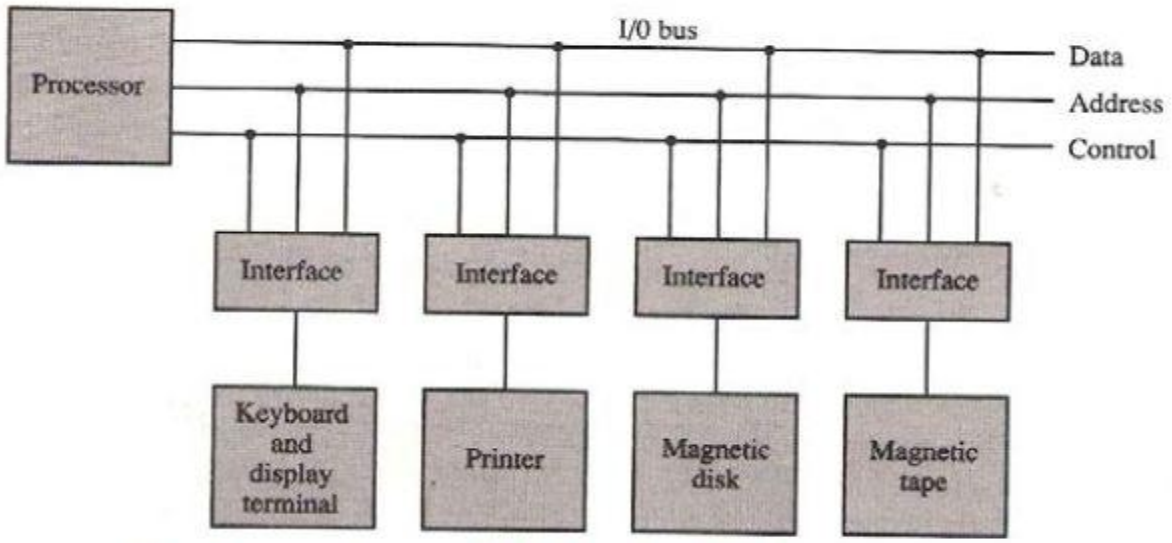


Fig: Connection of I/O bus to I/O devices

The advantages of I/O bus are:

- Provides special communication link to interface with CPU.
- I/O bus has different lines: address, data and control lines which have their own functions.

The advantages of interface module are as follows:

- Interface module decodes device address and decodes I/O commands in control lines.
- Interface module provides signal for the peripheral controller.
- Interface module synchronizes the data flow.
- It supervises the transfer rate between peripheral and memory or CPU.

3. What are the three possible modes to transfer the data to and from peripherals? Explain.

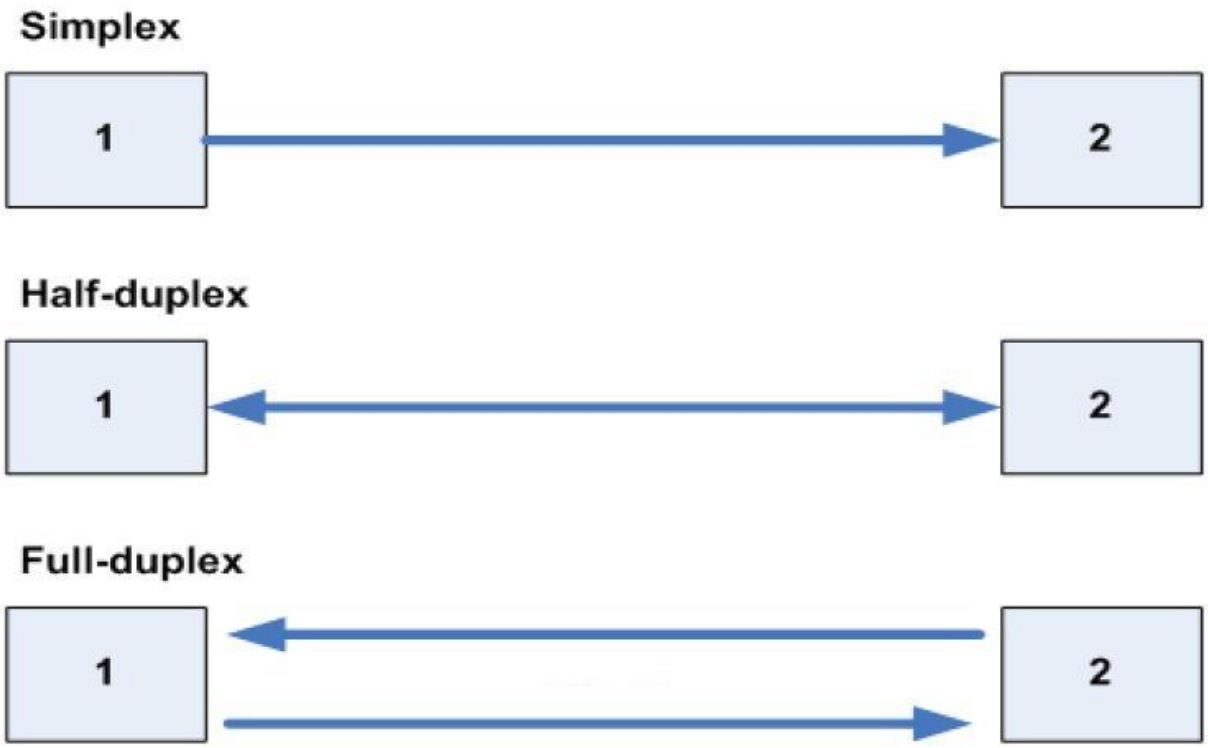
Data can be transmitted in between two points in 3 different modes:

- Simplex:
    - Carries information in one direction only.

- Seldom used.
- Example: PC to printer, radio and TV broadcasting.

➢ Half-duplex:
- Capable of transmitting in both directions but only in one direction at a time.
- Turnaround time: time to switch a half-duplex line from one direction to other.
- Example: walkie-talkie; style two-way radio

➢ Full-duplex:
- Can send and receive data in both directions simultaneously.
- Example: Telephone, Mobile Phone, etc.



Short questions $(10 \times 6 = 60)$

4. Differentiate between parity checker and parity generator.

Parity generator and checker networks are logic circuits constructed with exclusive-OR functions. Consider a 3-bit message to be transmitted with an odd parity bit. At the sending end, the odd parity is generated by a parity generator circuit. The output of the parity checker would be 1 when an error occurs i.e. no. of 1's in the four inputs is even.
$P = \overline{x \oplus y \oplus z}$

Message (xyz)	Parity bit (odd)
000	1
001	0
010	0
011	1
100	0
101	1
110	1
111	0

Considers original message as well as parity bit
$e = \overline{x \oplus y \oplus z}$
e = 1 => No. of 1's in pxyz is even => Error in data
e = 0 => No. of 1's in pxyz is odd => Data is error free

Circuit diagram for parity generator and parity checker is given below:

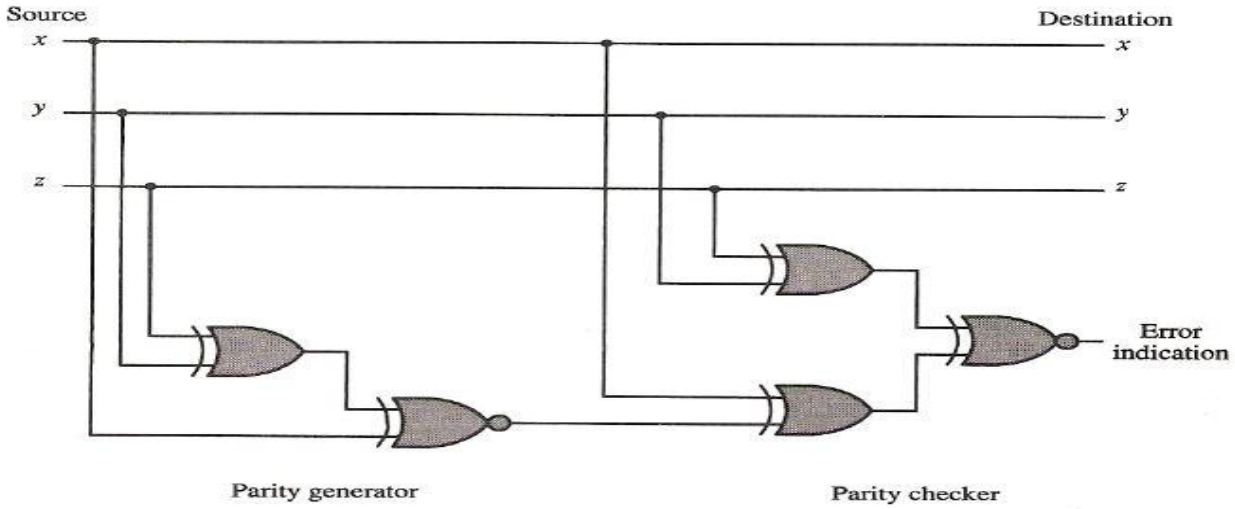
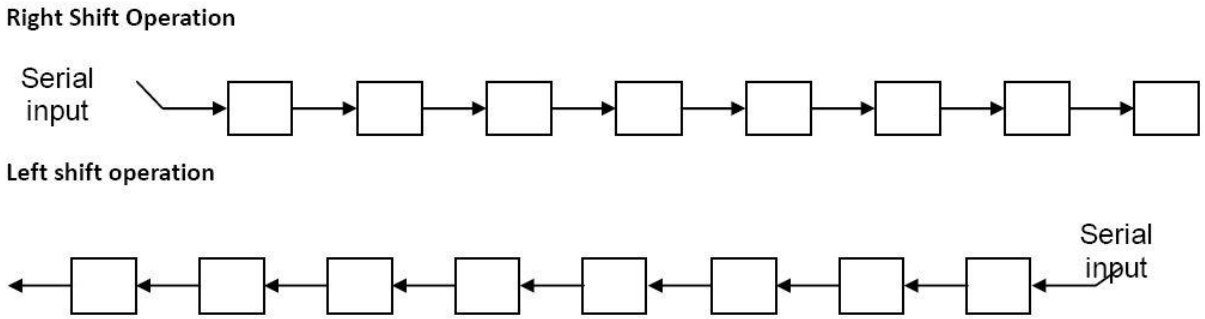


Fig: Error detection with odd parity bit.

5. What do you mean by shift micro-operations? Explain.

Shift micro-operations are used for serial transfer of data. They are also used in conjunction with arithmetic, logic and other data processing operations. The contents of a resister can be shifted left or right. There are three types of shifts: logical, circular and arithmetic shifts.



A logical shift is one that transfers 0 through the serial input. In RTL, the following notations are used:

- shl (for a logical shift left)
- shr (for a logical shift right)

Circular shift circulates the bits of the resister around the two ends without the loss of information. In RTL, the following notations are used:

- cil (for a circular shift left)
- cir (for a circular shift right)

An arithmetic shift is meant for signed binary numbers. An arithmetic left shift multiplies a signed number by 2 and an arithmetic right shift divides a signed number by 2. Arithmetic shifts must leave the sign bit unchanged because the sign of the number remains the same when it is multiplied or divided by 2. The left most bit in a resister holds a sign bit and remaining hold the number. Negative numbers are in 2's complement form. In RTL, the following notations are used:

- ashl (for an arithmetic shift left)
- ashr (for an arithmetic shift right)

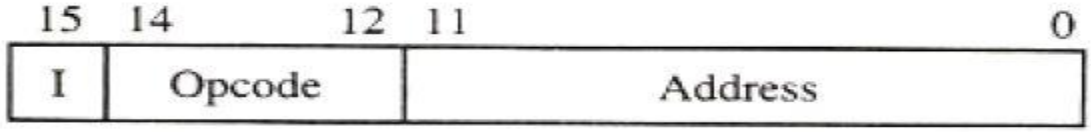
6. Explain the computer instruction with example.

A computer instruction is often divided into two parts:

- An *opcode* (Operation Code) that specifies the operation for that instruction

- An *address* that specifies the registers and/or locations in memory to use for that operation

In the Basic Computer, since the memory contains 4096 (= 212) words, we needs 12 bit to specify the memory address that is used by this instruction. In the Basic Computer, bit 15 of the instruction specifies the addressing mode (0: direct addressing, 1: indirect addressing). Since the memory words, and hence the instructions, are 16 bits long, that leaves 3 bits for the instruction's opcode.



(a) Instruction format

7. Mention the type of interrupt and explain it.

There are 3 major types of interrupts that cause break in the normal execution of a program. They can be classified as external, internal and software interrupts:

External Interrupts:

External interrupts come from Input output devices, from timing devices, from a circuit monitoring the power supply, or from any other external source.

Internal Interrupts:

Internal interrupts are also called traps. Internal interrupts arise from illegal or erroneous use of an instruction or data. The difference between internal and external interrupts is:

- Internal interrupts are initiated by some exceptional condition caused by the program itself rather than by an external event.
- Internal interrupts are synchronous with the program while external interrupts are asynchronous.
- If the program is rerun, the internal interrupts will occur in the same place each time. External interrupts depend on external conditions that are independent of the program being executed at the time.

Software Interrupts:

Software interrupts is a special call instructions that behave like an interrupt rather than a subroutine call. It can be used by the programmer to initiate an interrupt procedure at any desired point in the program. The most common use of software interrupt is associated with a supervisor call instruction. For example, a program written by a user must run in the user mode. When an input or output transfer is required, the supervisor mode is requested by means of a supervisor call instruction. This instruction causes a software interrupt that stored the old CPU state and brings in a new PSW that belongs to the supervisor mode. The calling program must pass information to the operating system in order to specify the particular task requested.

8. What do you mean by field decoding? Explain.

The 9-bits of the micro-operation field are divided into 3 subfields of 3 bits each. The control memory output of each subfield must be decoded to provide distinct micro-operations. The outputs of the decoders are connected to the appropriate inputs in the processor unit.

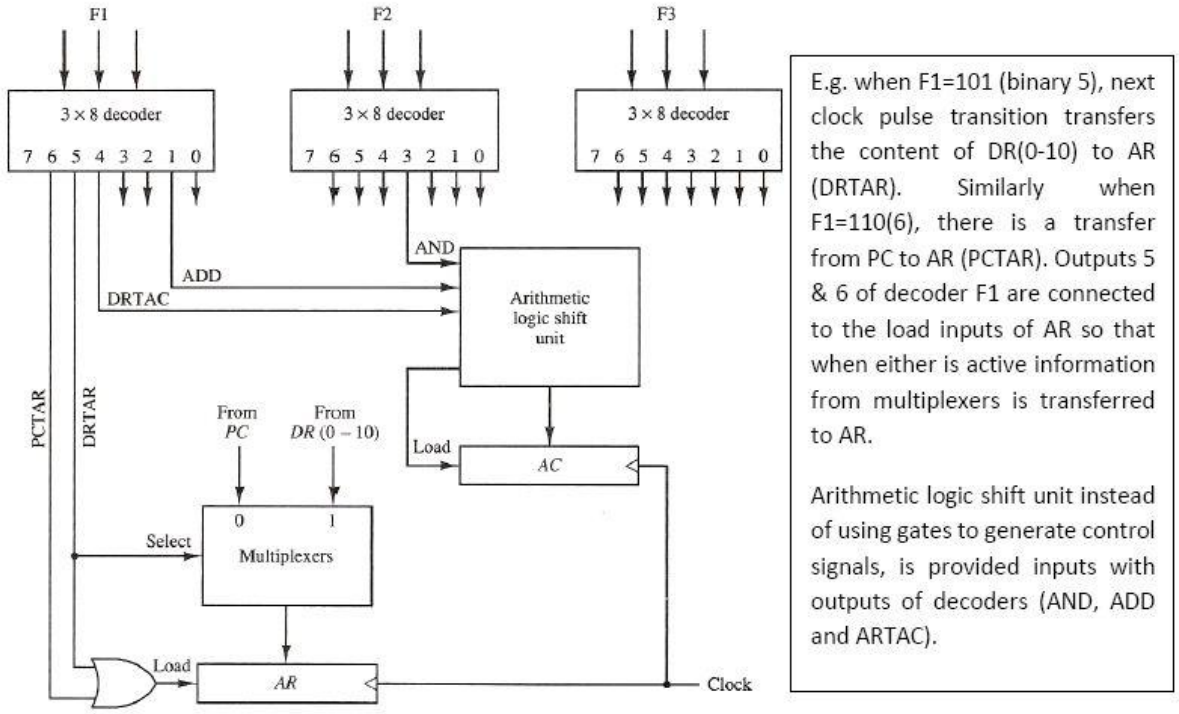


Fig: Decoding of microoperation fields

9. Write down the following equation in three address, two address and one address instruction.

$$Y=AB + (C\times D) + E\,(F/G)$$

In 3-address instruction:

MUL R1, A, B

MUL R2, C, D

ADD R3, R1, R2

DIV R1, F, G

MUL R2, E, R1

ADD Y, R2, R3

In 2-address instruction:

MOV R1, A

MUL R1, B

MOV R2, C

MUL R2, D

ADD R1, R2

MOV R2, F

DIV R2, G

MUL R2, E

ADD R1, R2

MOV Y, R1

In 1-address instruction:

LOAD A

MUL B

STORE T

LOAD C

MUL D

ADD T

STORE T

LOAD F

DIV G

MUL E

ADD T

STORE Y

10. Explain the characteristics of RISC and CISC.

Characteristics of RISC:

- Relatively few instructions and addressing modes
- Memory access limited to load and store instructions
- All operations done with in CPU registers (relatively large no of registers)
- Fixed-length, easily decoded instruction format
- Single cycle instruction execution
- Hardwired rather than micro programmed control
- Use of overlapped-register windows to speed procedure call and return
- Efficient instruction pipeline

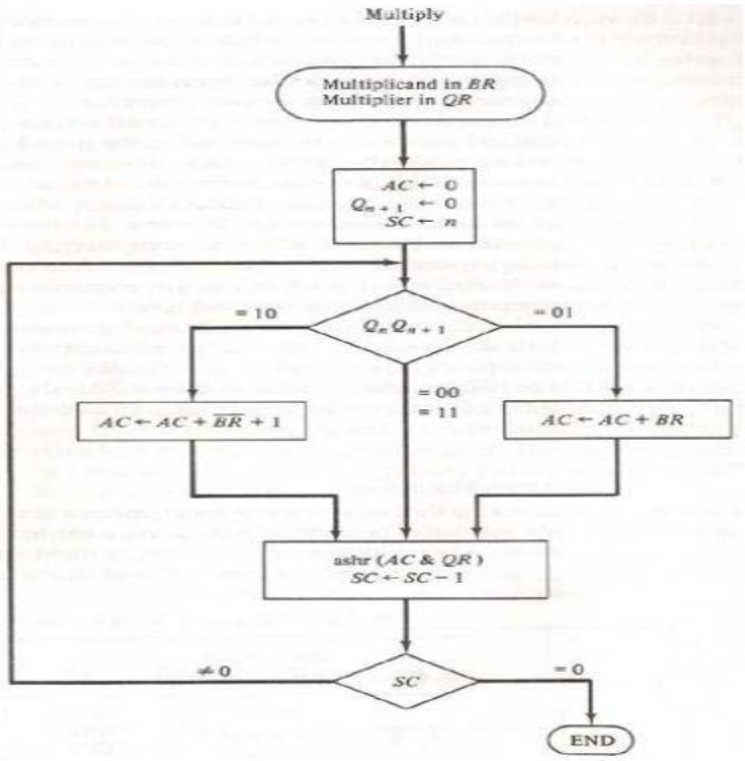
Characteristics of CISC:

- A large no of instructions – typically from 100 to 250 instructions
- A large variety of addressing modes – typically from 5 to 20
- Variable-length instruction formats
- Instructions that manipulate operands in memory

11. Explain the booth algorithm with example.

Booth algorithm gives a procedure for multiplying binary integers in signed 2's complement notation.

Algorithm for booth multiplication is given below:



An example for Booth algorithm is given here below:

Numerical Example: **Booth algorithm**

BR = 10111 (**Multiplicand**)

QR = 10011 (**Multiplier**)

Q_n Q_{n+1}	BR = 10111 $\overline{BR}$ + 1 = 01001	AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	Subtract BR	01001			
		01001			
	ashr	00100	11001	1	100
1 1	ashr	00010	01100	1	011
0 1	Add BR	10111			
		11001			
	ashr	11100	10110	0	010
0 0	ashr	11110	01011	0	001
1 0	Subtract BR	01001			
		00111			
	ashr	00011	10101	1	000

12. What is the main function of DMA? Mention the three points DMA configurations.

DMA is a sophisticated I/O technique in which a DMA controller replaces the CPU and takes care of the access of both, the I/O device and the memory, for fast data transfers. Using DMA you get the fastest data transfer rates possible.

Burst mode: An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system bus by the CPU, it transfers all bytes of data in the data block before releasing control of the system buses back to the CPU. This mode is useful for loading program or data files into memory, but renders the CPU inactive for relatively long periods of time. The mode is also called Block Transfer Mode.

Cycle stealing mode: The cycle stealing mode is used in systems in which the CPU should not be disabled for the length of time needed for burst transfer modes. In the cycle stealing mode, the DMA controller obtains access to the system bus the same way as in burst mode, using BR (Bus Request) and BG (Bus Grant) signals, which are the two signals controlling the interface between the CPU and the DMA controller. However, in cycle stealing mode, after one byte of data transfer, the control of the system bus is disserted to the CPU via BG. It is then continually requested again via BR, transferring one byte of data per request, until the entire block of data has been transferred. Cycle stealing mode is useful for controllers that monitor data in real time.

Transparent mode: The transparent mode takes the most time to transfer a block of data, yet it is also the most efficient mode in terms of overall system performance. The DMA controller only transfers data when the CPU is performing operations that do not use the system buses. It is the primary advantage of the transparent mode that the CPU never stops executing its programs and the DMA transfer is free in terms of time. The disadvantage of the transparent mode is that the hardware needs to determine when the CPU is not using the system buses, which can be complex and relatively expensive.

13. What are the different types of I/O commands? Explain.

The I/O commands allow you to own, use, read from, write to, and close devices. To direct I/O operations to a device, first issue the following commands:

- Issue an OPEN command to establish ownership, unless the device is your principal device.
- Issue a USE command to make the device the current device.
- Subsequent READ and WRITE commands read from and write to that device.
- A CLOSE command releases ownership of the device so that other processes can use the device.

OPEN Command

- OPEN device{:{(parameters)}{:{timeout}{:"mnespace"}}}

USE Command

- USE device:(args):"mnespace"

READ Command

- READ variable:timeout

WRITE Command

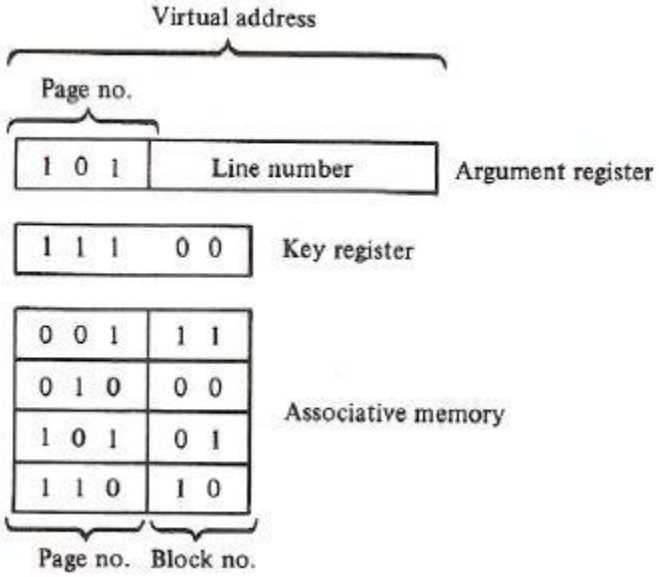
- WRITE variable

CLOSE Command

- CLOSE device[:params]

14. Differentiate between associative page table and replacement.

Memory page table is used in mapping address space to memory space. In case of random access page table it is inefficient with respect to storage utilization. Since to reduce such condition associative page table is efficient which reduce the size of memory and each location is fully utilized. Here each word in memory contains a page number with its corresponding block number.



➔ The page field in each **associative memory table** word is compared with page number bits in an argument register (which contains page number in the virtual address), if match occurs, the word is read form memory and its corresponding block number is extracted.

When a program starts execution, one or more pages are transferred into main memory and the page table is set to indicate its position. The program is executed from the main memory until it attempts to reference a page that is still in auxiliary memory. This condition is called page fault.

When page fault occurs in a virtual memory system, it signifies that the page referenced by the program is not in main memory. A new page is then transferred from auxiliary memory to main memory. If main memory is full, it would be necessary to remove a page from a memory block to make a room from a new page. The policy for choosing pages to remove is determined from the replacement algorithm that is used. Main goal of page replacement algorithm is try to remove the page least likely to be referenced by in the immediate future.

There are numerous page replacement algorithms, two of which are:

- First In First Out (FIFO): replace a page that has been in memory longest time.
- Least Recently Used (LRU): assumes that least recently used page is the better candidate for removal than the least recently loaded page.

15. Write short notes on the following:

a. Memory space

b. Address space

Memory space: An address in main memory is called a location or physical address. The set of such locations is called the memory space. The memory space consists of actual main memory locations directly addressable for processing. Generally, the memory space is smaller than the address space.

Address space: An address used by programmer is a virtual address and the set of such addresses is the address space. Thus the address space is the set of addresses generated by the programs as they reference instructions and data. Generally, the address space is larger than the memory space.