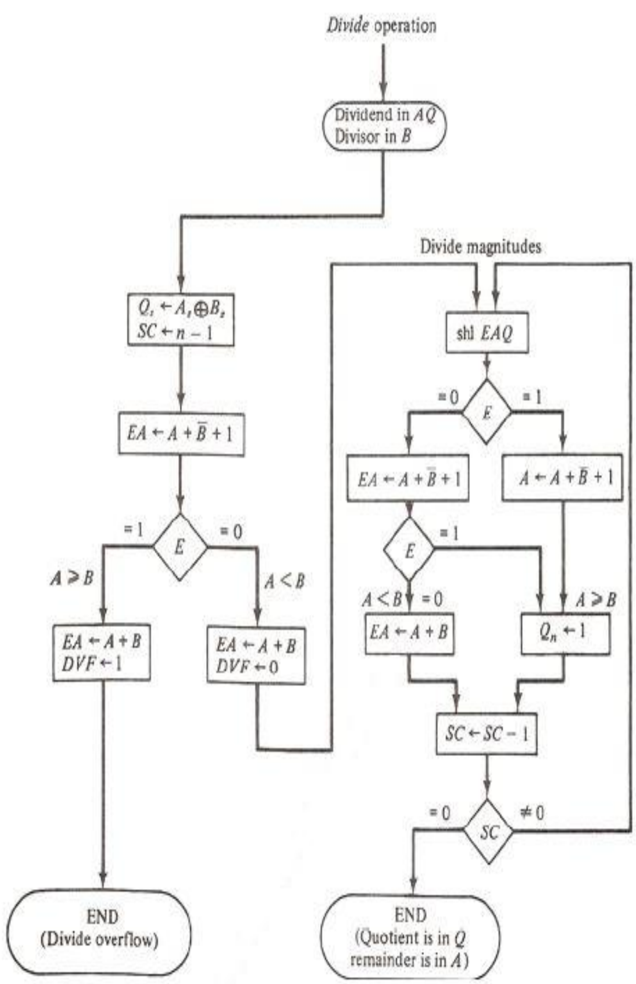
2068

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

1. Explain the restoring division algorithm with example.

Flowchart for restoring division algorithm is shown below:



- B: Divisor, AQ: Dividend
- If A>=B (oh yes, magnitudes are compared subtracting one from another and testing E flip-flop), DVF is set and operation is terminated prematurely. If A<B, no overflow and dividend is restored by adding B to A (since B was subtracted previously to compare magnitudes).
- Division starts by left shifting AQ (dividend) with high order bit shifted to E. Then E=1, EA>B so B is subtracted from EA and Q_n is set to 1. If E=0, result of subtraction is stored in EA, again E is tested. E=1 signifies A>=B, thus Q_n is set to 1 and E=0 denotes A<B, so original number is **restored** by adding B to A and we leave 0 in Q_n.
- Process is repeated again with register A holding partial remainder. After n-1 times Q contains magnitude of Quotient and A contains remainder. Quotient sign in Q_s and remainder sign in A_s.

Fig: flow chart for divide operation

Numerical Example: Binary division with digital hardware

Divisor $B = 10001$, $\overline{B} + 1 = 01111$

	E	A	Q	SC
Dividend:		01110	00000	5
shl EAQ	0	11100	00000	
add $\overline{B} + 1$		01111		
$E = 1$	1	01011		
Set $Q_n = 1$	1	01011	00001	4
shl EAQ	0	10110	00010	
Add $\overline{B} + 1$		01111		
$E = 1$	1	00101		
Set $Q_n = 1$	1	00101	00011	3
shl EAQ	0	01010	00110	
Add $\overline{B} + 1$		01111		
$E = 0$; leave $Q_n = 0$	0	11001	00110	
Add B		10001		
Restore remainder	1	01010		2
shl EAQ	0	10100	01100	
Add $\overline{B} + 1$		01111		
$E = 1$	1	00011		
Set $Q_n = 1$	1	00011	01101	1
shl EAQ	0	00110	11010	
Add $\overline{B} + 1$		01111		
$E = 0$; leave $Q_n = 0$	0	10101	11010	
Add B		10001		
Restore remainder	1	00110	11010	0
Neglect E				
Remainder in A:		00110		
Quotient in Q:			11010	

2. What do you mean by I/O interface? Explain the I/O bus and Interface module.

Input-output interface provides a method for transferring information between internal storage and external I/O devices. It resolves the differences between the computer and peripheral devices. The major differences are:

- Peripherals are electromechanical and electromagnetic devices and manner of operation is different from that of CPU which is electronic component.
- Data transfer rate of peripherals is slower than that of CPU. So some synchronization mechanism may be needed.

- Data codes and formats in peripherals are different from each other and each must be controlled so as not to disturb other.

To resolve these differences, computer system usually include special hardware unit between CPU and peripherals to supervise and synchronize I/O transfers, which are called interface units since they interface processor bus and peripherals.

I/O Bus and Interface Modules

Peripherals connected to a computer need special communication link to interface with CPU. This special link is called I/O bus. Figure below clears the idea:

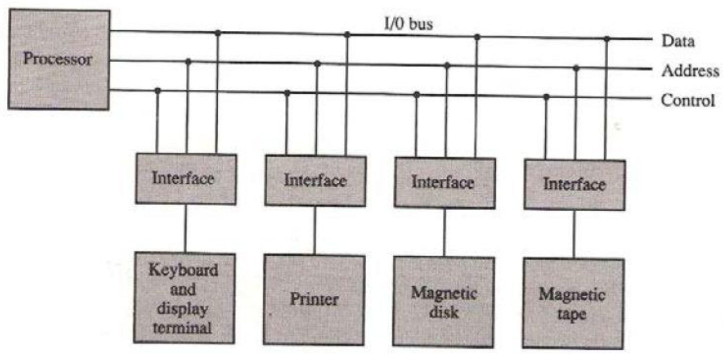


Fig: Connection of I/O bus to I/O devices

- I/O bus from the processor is attached to all peripheral interfaces.
- I/O bus consists of **Data lines**, **address** and **control lines**.
- To communicate with a particular device, the processor places a device address on the address lines. Each peripheral has an **interface module** associated with its interface.

Functions of an interface are as below:

- Decodes the device address (device code)
- Decodes the I/O commands (operation or function code) in control lines.
- Provides signals for the peripheral controller.
- Synchronizes the data flow.
- Supervises the transfer rate between peripheral and CPU or memory.

3. What do you mean by memory organization? Explain the memory management hardware with example.

Memory unit is an essential component in any general purpose computer since it is needed to store programs and data. The memory unit that communicates directly with the CPU is called the main memory and devices that provide backup storage are called auxiliary memory. Auxiliary memory devices such as magnetic disk and tapes are used to store system programs, large data files and other backup information. Only programs and data currently needed by the processor reside in main memory. All other information is stored in main memory and transferred to main memory when needed.

A memory management system is a collection of hardware and software procedures for managing various programs (effect of multiprogramming support) residing in memory. Basic components of memory management unit (MMU) are:

- A facility for dynamic storage relocation that maps logical memory references into physical memory addresses.
- A provision for sharing common programs by multiple users.
- Protection of information against unauthorized access.

The dynamic storage relocation hardware is a mapping process similar to paging system.

Segment: It is more convenient to divide programs and data into logical parts called segments despite of fixed-size pages. A segment is a set of logically related instructions or data elements. Segments may be generated by the programmer or by OS. Examples are: a subroutine, an array of data, a table of symbols or user's program.

Logical address: The address generated by the segmented program is called a logical address. This is similar to virtual address except that logical address space is associated with variable-length segments rather than fixed-length pages.

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

4. Explain the error detection code with example.

Binary information transmitted through some form of communication medium is subject to external noise that could change bits from 1 to 0 and vice versa. An error detection code is a binary code that detects digital errors during transmission. The detected errors cannot be corrected but their presence is indicated. The most common error detection code used is the parity bit. A parity bit(s) is an extra bit that is added with original message to detect error in the message during data transmission.

Even Parity: One bit is attached to the information so that the total number of 1 bits is an even number.

Message	Parity
1011001	0
1010010	1

Odd Parity: One bit is attached to the information so that the total number of 1 bits is an odd number.

Message	Parity
1011001	1
1010010	0

5. Differentiate between logic micro operations and shift micro operations.

Logic micro operations are bit-wise operations, i.e., they work on the individual bits of data. Useful for bit manipulations on binary data and for making logical decisions based on the bit value. There are, in principle, 16 different logic functions that can be defined over two binary input variables. However, most systems only implement four of these

AND ($\wedge$), OR ($\vee$), XOR ($\oplus$), Complement/NOT

The others can be created from combination of these four functions.

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.

6. Explain the I/O instruction with example.

Input/output Instructions

- Input and output
- Examples: INP, OUT

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

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

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

7. What do you mean by memory mapping? Explain.

Memory mapping is a process whereby some item of digital hardware is connected to a processor's address bus and data bus in such a way that it can be accessed (for reading and/or writing) exactly as if it were a memory cell.

This is used as an alternative to connecting it to I/O port, especially in embedded system. For example, an analog-to-digital converter could be memory mapped to a certain address. When that address is written to, the conversion is started; when the address is read from, the data is transferred to the processor.

Sometimes only partial address decoding is used, meaning that the device efficiently occupies a much larger block of memory space than is strictly necessary. This can be a problem if the memory space is small (e.g. with a i6-bit address bus which can only access 65,536 different locations.

The memory bus is only bus in the system. CPU can manipulate I/O data residing in interface registers with same instructions that are used to access memory words. Memory space is not only ordinary system memory. It can refer to all the address that the programmer may specify. These addresses correspond to all possible valid addresses that the CPU may place on its memory bus address line.

8. What do you mean by control memory? Explain the micro instructions and micro operations format.

A computer that employs a micro programmed control unit will have two separate memories: main memory and a control memory. The user's program in main memory consists of machine instructions and data whereas control memory holds a fixed micro program that cannot be altered by the user. Each machine instruction initiates a series of microinstructions in control memory.

Microinstruction Format
We know the computer instruction format for different set of instruction in main memory. Similarly, microinstruction in control memory has 20-bit format divided into 4 functional parts as shown below:

3	3	3	2	2	7
F1	F2	F3	CD	BR	AD

F1, F2, F3: Micro operation fields
CD: Condition for branching
BR: Branch field
AD: Address field

9. What do you mean by addressing modes? Differentiate between indexed addressing mode and base register addressing mode.

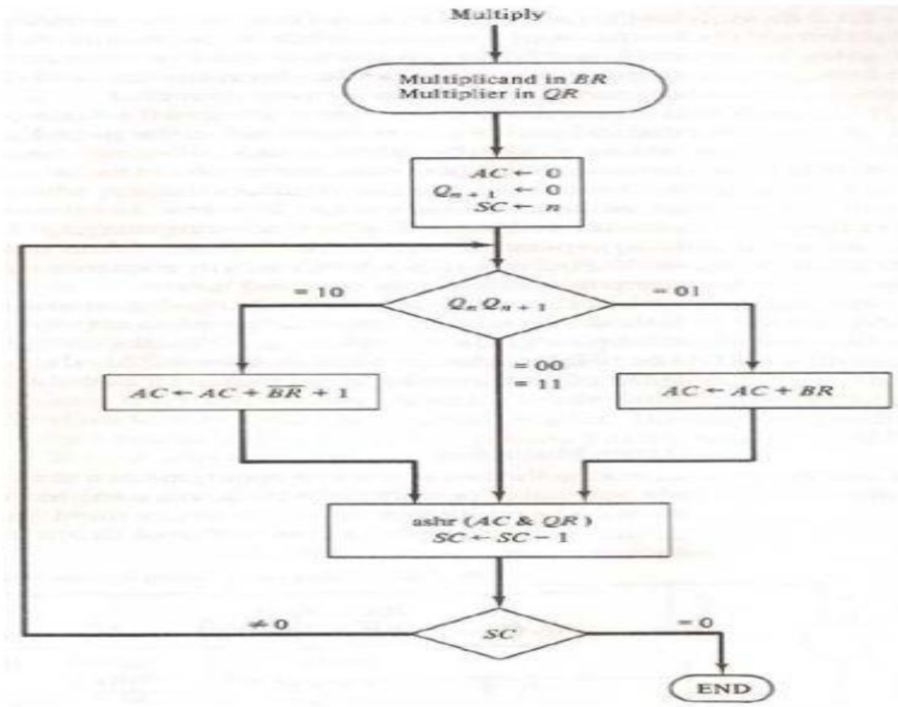
Addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced. There are different types of addressing modes. They are:

- Implied mode
- Immediate mode
- Register mode
- Register indirect mode
- Auto increment or auto decrement mode
- Direct addressing mode
- Indirect addressing mode
- Relative addressing mode
    - PC relative addressing mode
    - Indexed addressing mode
    - Base register addressing mode

Indexed Addressing Mode
EA = IX + IR (address) {IX is index register}
Base Register Addressing Mode
EA = BAR + IR (address)

10. Explain the Booth algorithm. Multiply 3 × 5 using booth algorithm.

Booth algorithm gives a procedure for multiplying binary integers in signed 2's complement notation. Algorithm for booth multiplication is given below:



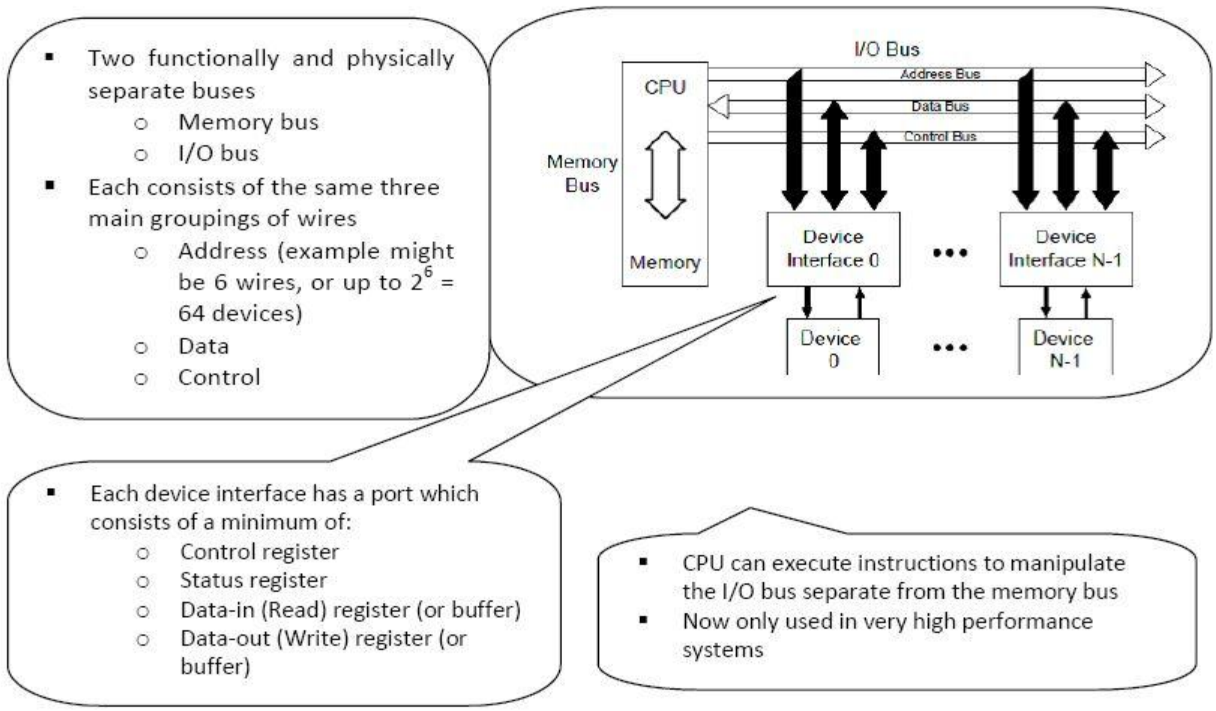
We have, $3 \rightarrow 11$ and $5 \rightarrow 101$

So, $BR = 011$ and $QR = 101$

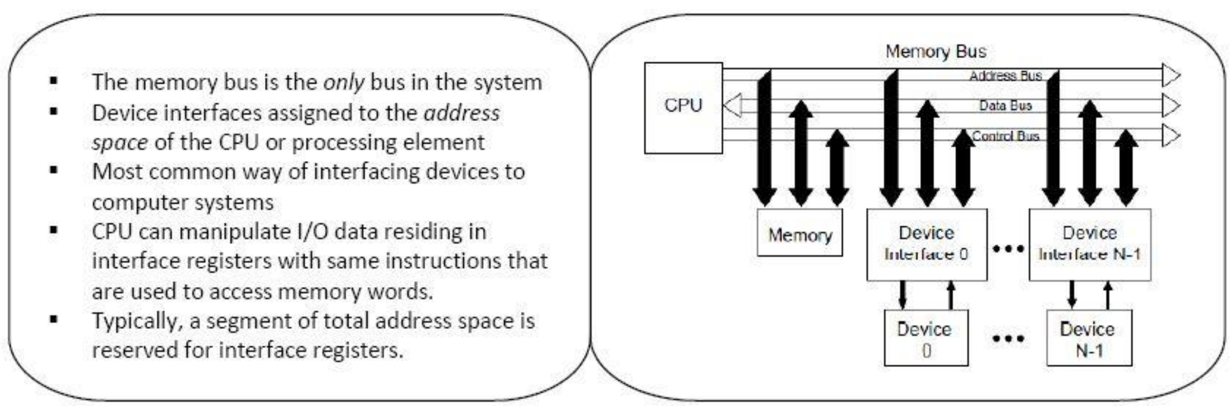
Q_n	Q_{n+1}	$BR = 011$ $\overline{BR} + 1 = 101$	AC	QR	Q_{n+1}	SC
		Initials	000	101	0	011
1	0	Sub BR	+101 =101			
		ashr	010	010	1	10
0	1	Add BR	+011 =101			
		ashr	010	101	0	01
1	0	Sub BR	+101 =111	010	0	00

11. Differentiate between isolate and memory mapped I/O.

Isolated I/O Configuration



Memory-Mapped I/O configuration



12. Explain the I/O processor with block diagram.

IOP is a processor with DMA capability that communicates with I/O devices. In this configuration, the computer system can be divided into a memory unit, and a number of processors comprised of CPU and one or more IOP's. IOP is similar to CPU except that it is designed to handle the details of I/O processing. Unlike DMA controller, IOP can fetch and execute its own instructions. IOP instructions are designed specifically to facilitate I/O transfers. Instructions that are read from memory by an IOP are called commands to differ them from instructions read by CPU. The command words constitute the program for the IOP. The CPU informs the IOP where to find commands in memory when it is time to execute the I/O program.

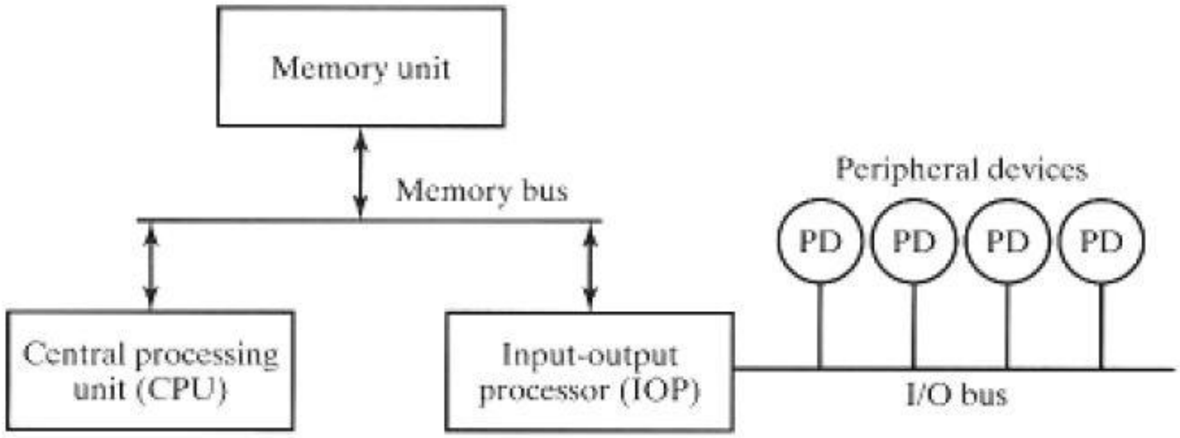


Fig: Block diagram of computer with I/O processor

The memory occupies a central position and can communicate with each processor by means of DMA. CPU is usually assigned the task of initiating the I/O program, from then on; IOP operates independent of the CPU and continues to transfer data from external devices and memory.

13. Explain data transfer instruction with example.

Data transfer instructions causes transfer of data from one location to another without modifying the binary information content. The most common transfers are:

- between memory and processor registers
- between processor registers and I/O
- between processor register themselves

Table below lists 8 data transfer instructions used in many computers.

Name	Mnemonic
Load	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP

Load: denotes transfer from memory to registers (usually AC)
Store: denotes transfer from a processor registers into memory
Move: denotes transfer between registers, between memory words or memory & registers.
Exchange: swaps information between two registers or register and a memory word.
Input & Output: transfer data among registers and I/O terminals.
Push & Pop: transfer data among registers and memory stack.

14. Differentiate between RISC and CISC processor.

The major differences between RISC and CISC processor is based upon their characteristics.

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

15. Write short notes on the following:

a. Interrupt cycle

b. DMA

Interrupt cycle: This is a hardware implementation of a branch and save return address operation.

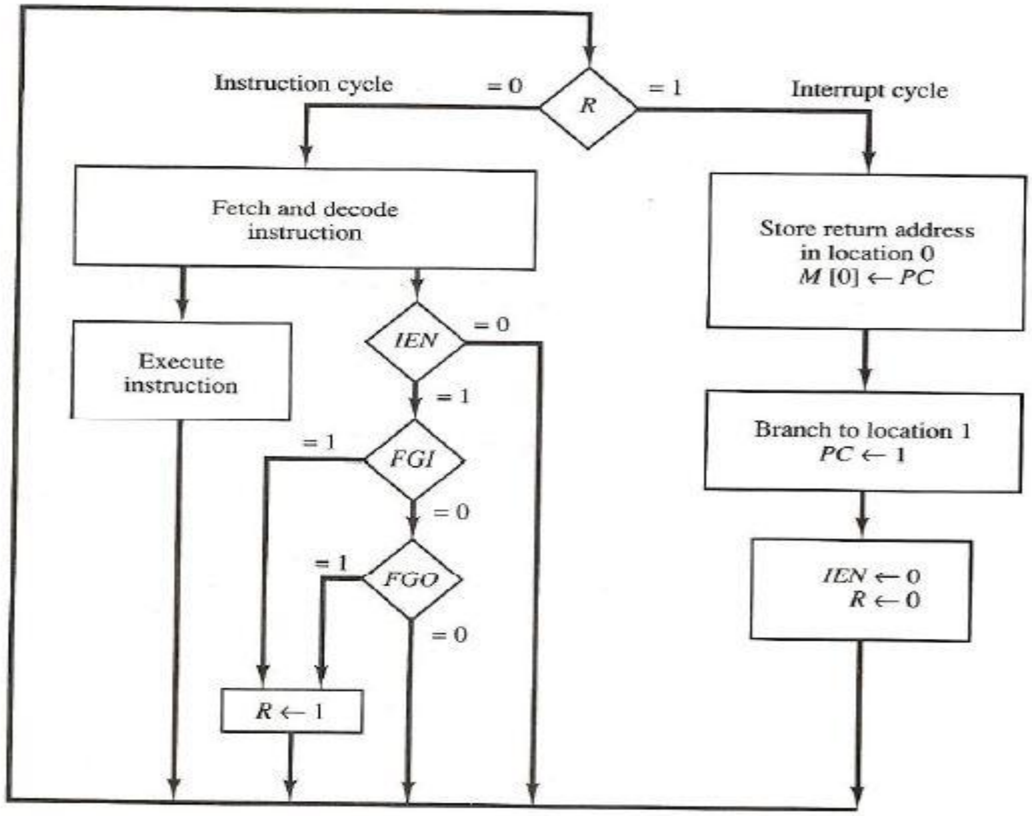


Fig: flowchart of interrupt cycle

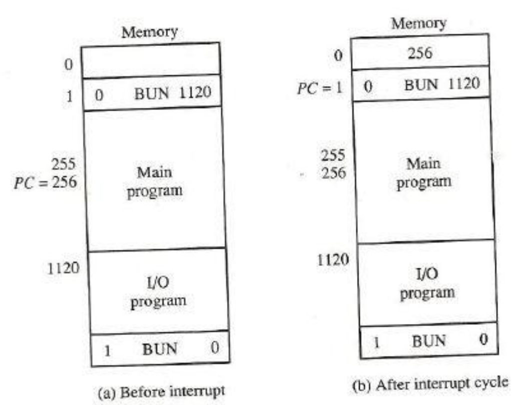


Fig: Demonstration of interrupt cycle

- At the beginning of the instruction cycle, the instruction that is read from memory is in address 1.
- At memory address 1, the programmer must store a branch instruction that sends the control to an interrupt service routine
- The instruction that returns the control to the original program is "indirect BUN 0"

DMA: 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. Interrupt driven and programmed I/O require active CPU intervention (all data must pass through CPU). Transfer rate is limited by processor's ability to service the device and hence CPU is tied up managing I/O transfer.

Removing CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. Extensively used method to capture buses is through special control signals: Bus Request (BR) and Bus Grant (BG). When DMA takes control of bus system, the transfer with the memory can be made in following two ways: Burst transfer and Cycle stealing.