2067

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

1. Explain the Micro program sequence with example.

Basic components of a micro programmed control unit are control memory and the circuits that select the next address. This address selection part is called a micro program sequencer. The purpose of micro program sequencer is to load CAR so that microinstruction may be read and executed. Commercial sequencers include within the unit an internal resister stack to store addresses during micro program looping and subroutine calls.
Internal structure of a typical micro program sequencer is shown below in the diagram. It consists of input logic circuit having following truth table.

BR Field		Input			MUX 1		Load SBR
		I_1	I_0	T	S_1	S_0	L
0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	1	1
1	0	1	0	×	1	0	0
1	1	1	1	×	1	1	0

Fig: Input Logic Truth for Micro program Sequencer

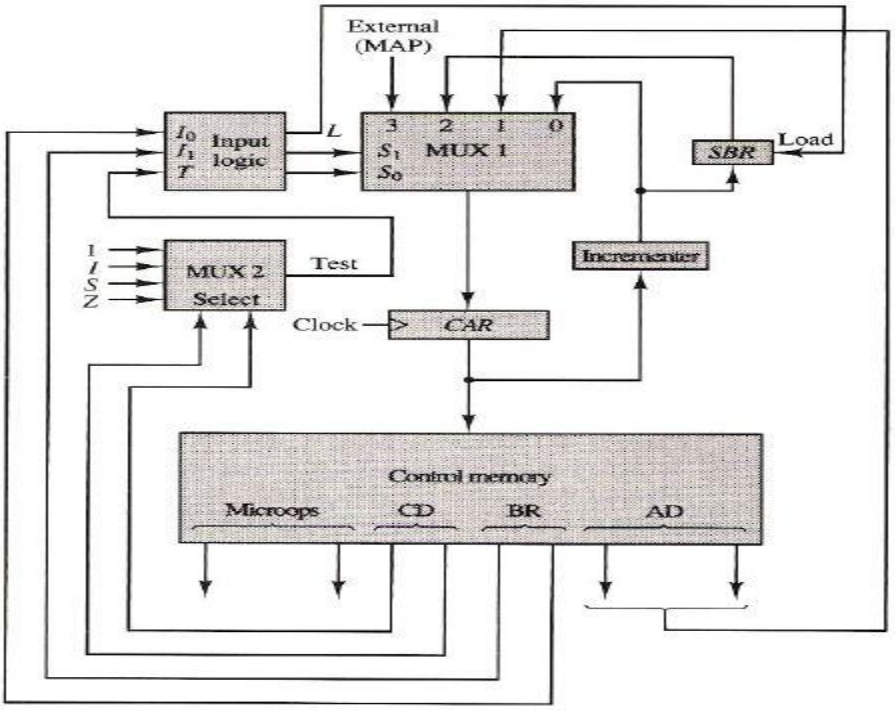


Fig: Microprogram sequencer for a control memory

Above figure represents the micro program sequencer for a control memory in which:

- MUX1 selects an address from one of four sources of and routes it into CAR.
- MUX2 tests the value of selected status bit and result is applied to input logic circuit.
- Output of CAR provides address for the control memory.
- Input logic circuit has 3 inputs I0, I1 and T and 3 outputs S0, S1 and L. variables S0 and S1 select one of the source addresses for CAR. L enables load input of SBR.
- E.g. when S1S0=10, MUX input number 2 is selected and establishes a transfer path from SBR to CAR.

2. Explain with example of Data manipulation instructions.

Data manipulation instructions provide computational capabilities for the computer. These are divided into 3 parts:

- Arithmetic instructions
- Logical and bit manipulation instructions
- Shift instructions

These instructions are similar to the micro operations. But actually, each instruction when executed must go through the fetch phase to read its binary code value from memory. The operands must also be brought into registers according to the rules of different addressing mode. And the last step of executing instruction is implemented by means of micro operations.

Arithmetic instructions

Typical arithmetic instructions are listed below:

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Negate (2's complement)	NEG

- Increment (decrement) instr. adds 1 to (subtracts 1 from) the register or memory word value.
- Add, subtract, multiply and divide instructions may operate on different data types (fixed-point or floating-point, binary or decimal).

Logical and bit manipulation instructions:

Logical instructions perform binary operations on strings of bits stored in registers and are useful for manipulating individual or group of bits representing binary coded information. Logical instructions each bit of the operand separately and treat it as a Boolean variable.

Name	Mnemonic
Clear	CLR
Complement	COM
AND	AND
OR	OR
Exclusive-OR	XOR
Clear carry	CLRC
Set carry	SETC
Complement carry	COMC
Enable interrupt	EI
Disable interrupt	DI

- Clear instr. causes specified operand to be replaced by 0's.
- Complement instr. produces the 1's complement.
- AND, OR and XOR instructions produce the corresponding logical operations on individual bits of the operands.

Shift instructions

Instructions to shift the content of an operand are quite useful and are often provided in several variations (bit shifted at the end of word determine the variation of shift). Shift instructions may specify 3 different shifts:

- Logical shifts
- Arithmetic shifts
- Rotate-type operations

Name	Mnemonic
Logical shift right	SHR
Logical shift left	SHL
Arithmetic shift right	SHRA
Arithmetic shift left	SHLA
Rotate right	ROR
Rotate left	ROL
Rotate right through carry	RORC
Rotate left through carry	ROLC

- Table lists 4 types of shift instructions.
- Logical shift inserts 0 at the end position
- Arithmetic shift left inserts 0 at the end (identical to logical left shift) and arithmetic shift right leave the sign bit unchanged (should preserve the sign).
- Rotate instructions produce a circular shift.
- Rotate left through carry instruction transfers carry bit to right and so is for rotate shift right.

3. Explain the non-restoring Division algorithm, flow chart hardware implementation with example.

In contrast to restoring method, when A-B is negative, B is not added to restore A but instead, negative difference is shifted left and then B is added.

- In flowchart for restoring method, when A<B, we restore A by operation A-B+B. Next time in a loop, this number is shifted left (multiplied by 2) and B subtracted again, which gives: 2(A-B+B)-B=2A-B
- In non-restoring method, we leave A-B as it is. Next time around the loop, the number is shifted left and B is added: 2(A-B)+B=2A-B

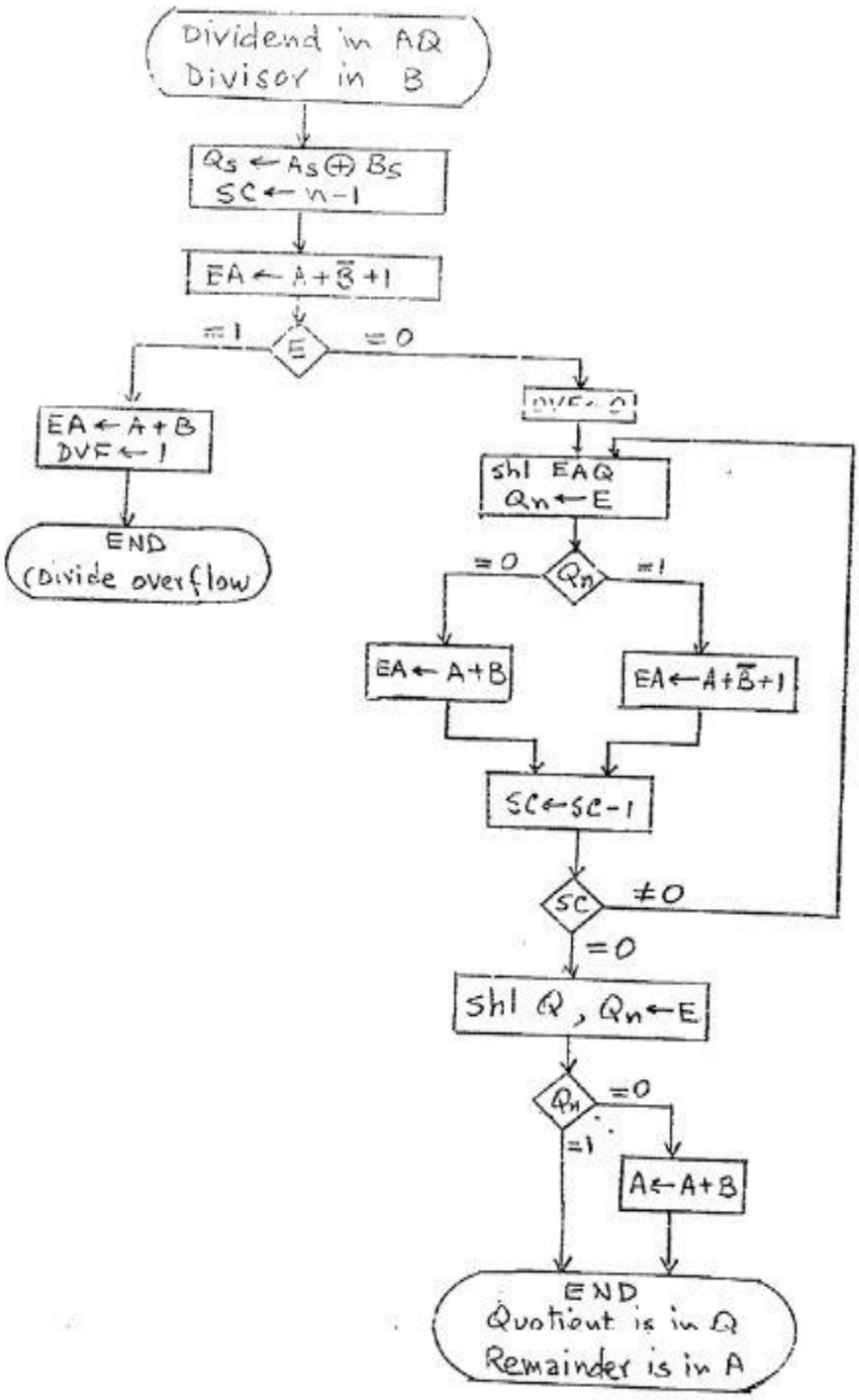


Fig. Hardware implementation flowchart of non-restoring division algorithm

Example for non-restoring division algorithm:

P	A	Operation
00000	1110	Divide 14 = 1110 by 3 = 11. B register always contains 0011
00001	110	step 1(i-b): shift
+00011		step 1(ii-b): subtract b (add two's complement)
11110	1100	step 1(iii): P is negative, so set quotient bit to 0
11101	100	step 2(i-a): shift
+00011		step 2(ii-a): add b
00000	1001	step 2(iii): P is +ve, so set quotient bit to 1
00001	001	step 3(i-b): shift
+11101		step 3(ii-b): subtract b
11110	0010	step 3(iii): P is -ve, so set quotient bit to 0
11100	010	step 4(i-a): shift
+00011		step 4(ii-a): add b
11111	0100	step 4(iii): P is -ve, set quotient bit to 0
+00011		Remainder is negative, so do final restore step
00010		

The quotient is 0100 and the remainder is 00010

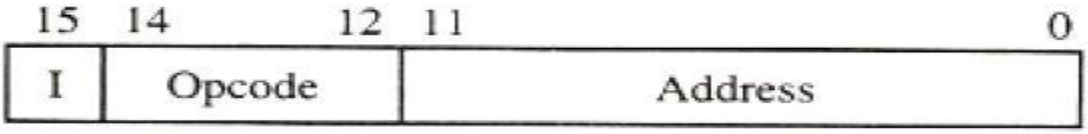
Short questions $(10 \times 6 = 60)$

4. What do you mean by instruction format? Explain.

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

5. Differentiate between Hardwired and Micro program control unit.

Hardwired Control:

- CU is made up of sequential and combinational circuits to generate the control signals.
- If logic is changed we need to change the whole circuitry.
- Expensive
- Fast

Micro program control unit:

- A control memory on the processor contains micro programs that activate the necessary control signals
- If logic is changed we only need to change the micro program
- Cheap
- Slow

6. What do you mean by logic 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.
Logic micro operations can be used to manipulate individual bits or a portion of a word in a register. Consider the data in a register A. Bits of register B will be used to modify the contents of A.

Selective-set	$A \leftarrow A + B$
Selective-complement	$A \leftarrow A \oplus B$
Selective-clear	$A \leftarrow A \cdot B'$
Mask (Delete)	$A \leftarrow A \cdot B$
Clear	$A \leftarrow A \oplus B$
Insert	$A \leftarrow (A \cdot B) + C$
Compare	$A \leftarrow A \oplus B$

7. Differentiate between direct and indirect addressing modes.

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 bits
- Slow to acquire an operand because of an additional memory access
- EA= M[IR (address)]

8. Explain with example of Data transfer instructions.

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.

9. What are the major differences between RISC and CISC architecture?

The major differences between RISC and CISC architecture 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

10. Explain the subtraction algorithm with signed 2's compliment.

In signed 2's complement representation, the leftmost bit represents sign (0-positive and 1-negative). If sign bit is 1, entire number is represented in 2's complement form. Subtraction consists of first taking 2's complement of the subtrahend and then adding it to minuend. When two numbers of n-digits each are added and the sum occupies n+1 bits, overflow occurs when output of the gate is 1.

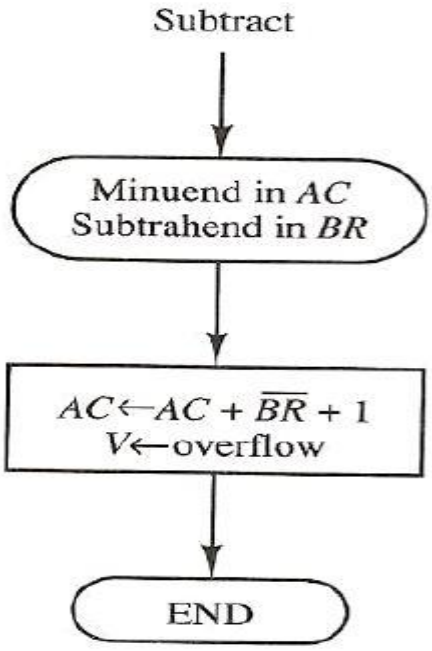
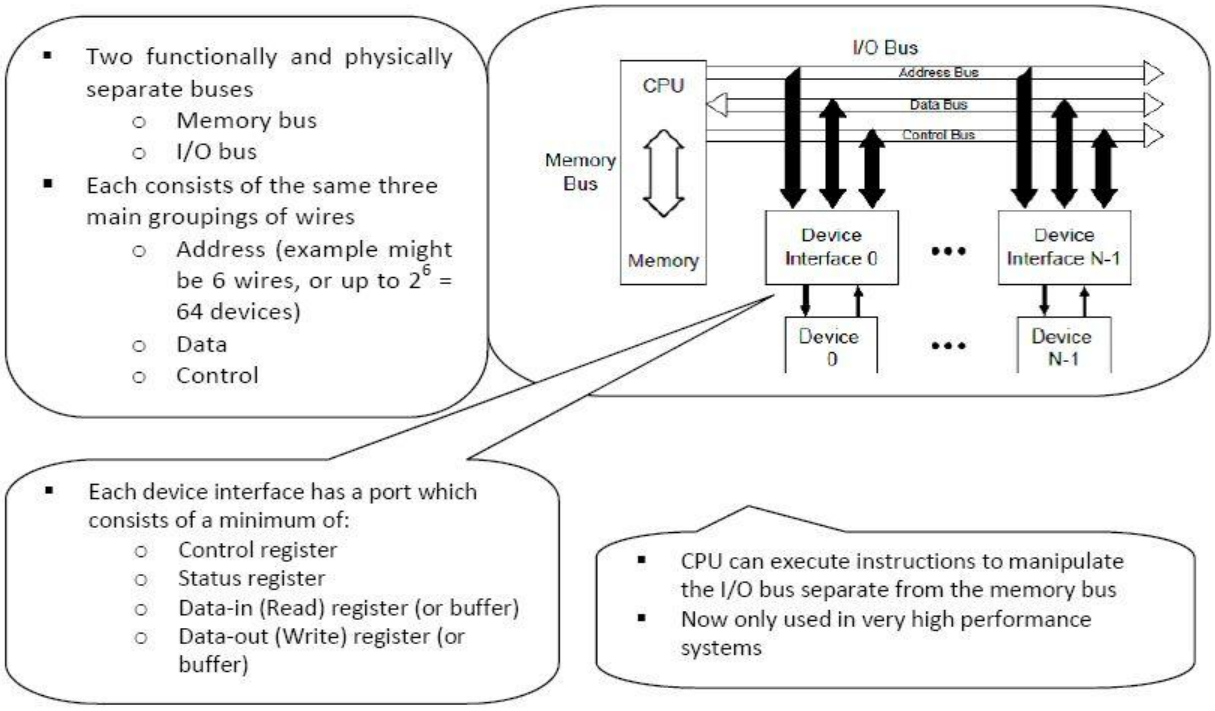


Fig: algorithm for subtraction of numbers in signed 2's complement representation

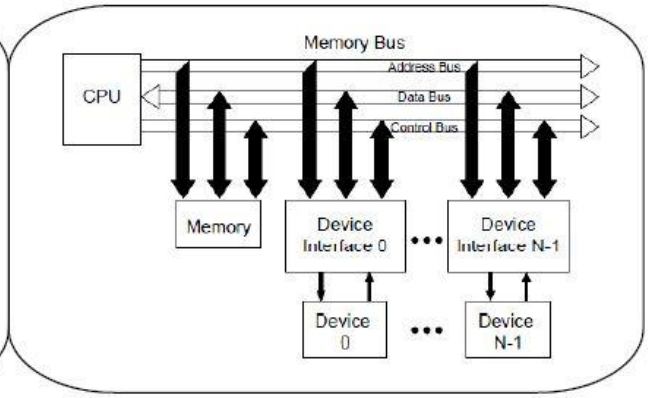
11. Differentiate between isolated I/O and Memory Mapped I/O.

Isolated I/O Configuration

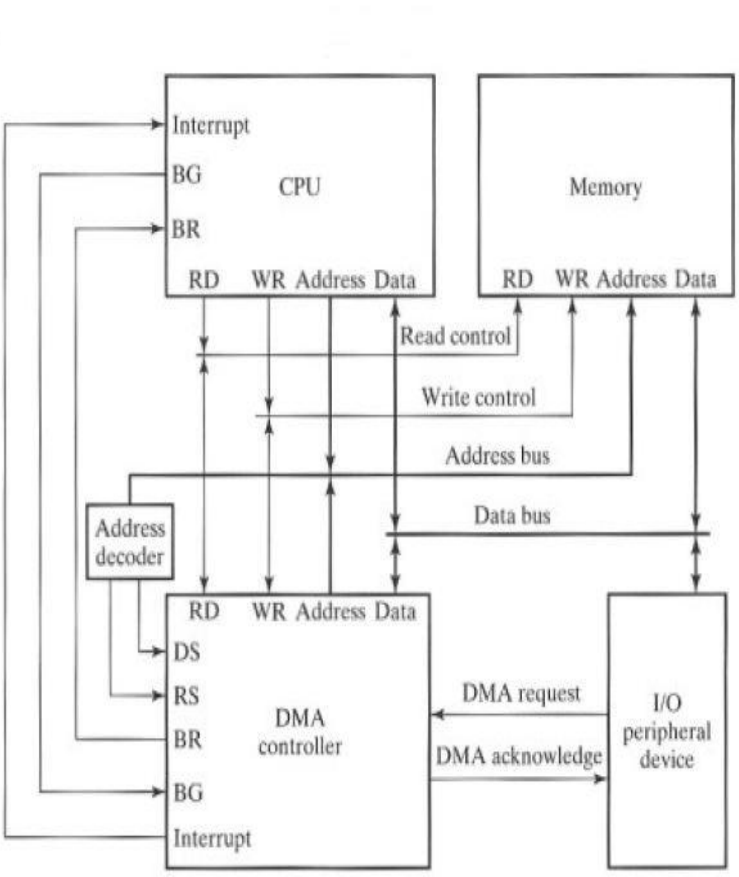


Memory-Mapped I/O configuration

- The memory bus is the *only* bus in the system
- Device interfaces assigned to the *address space* of the CPU or processing element
- Most common way of interfacing devices to computer systems
- CPU can manipulate I/O data residing in interface registers with same instructions that are used to access memory words.
- Typically, a segment of total address space is reserved for interface registers.



12. What is DMA transfer? Explain.



- CPU communicates with the DMA through address and data buses.
- DMA has its own address which activates RS (Register select) and DS (DMA select) lines.
- When a peripheral device sends a DMA request, the DMA controller activates the BR line, informing CPU to leave buses. The CPU responds with its BG line.
- DMA then puts current value of its address register into the address bus, initiates RD or WR signal, and sends a DMA acknowledge to the peripheral devices.
- When BG=0, RD & WR allow CPU to communicate with internal DMA registers and when BG=1, DMA communicates with RAM through RD & WR lines.

Fig: DMA transfer in a computer system

13. What is the role of input-output processor (IOP) in computer system? Explain.

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.

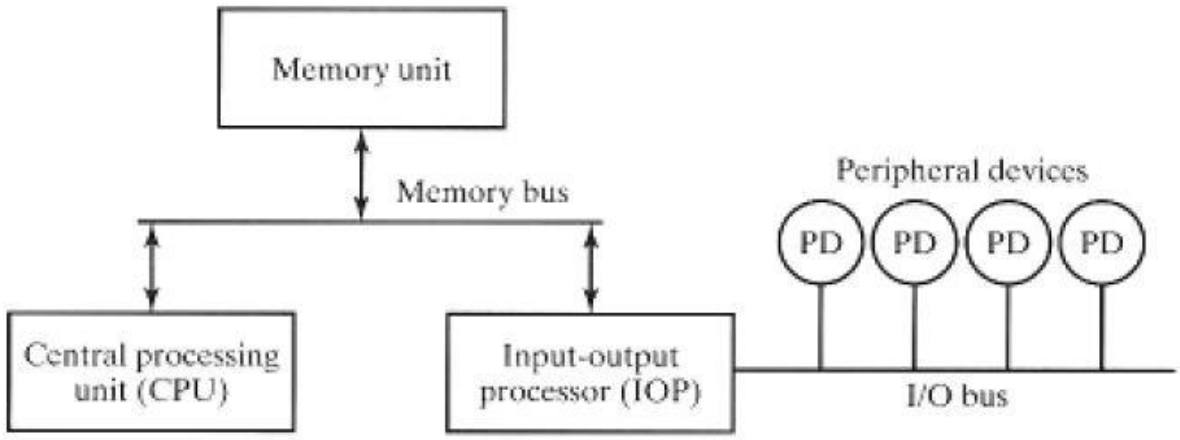


Fig: Block diagram of computer with I/O processor

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.

14. What is memory management hardware? Explain.

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.

15. Write short notes on the following:

a. Sequential memory hierarchy

b. Random memory hierarchy

Sequential memory hierarchy: In computing, SAM is a class of data storage devices that read their data in sequence. This is in contrast to random access memory (RAM) where data can be accessed in any order. Sequential access devices are usually a form of magnetic memory. Magnetic sequential access memory is typically used for secondary storage in general purpose computers due to their higher density at lower cost compared to RAM, as well as resistance to wear and non-volatility. Examples of SAM devices still in use include hard disks, CD-ROMs and magnetic tapes. Historically, drum memory has also been used.

Random memory hierarchy: RAM is a form of computer data storage. Today, it takes the form of integrated circuits that allow stored data to be accessed in any order with a worst case performance of constant time. Strictly speaking, modern types of DRAM are therefore not random access, as data is read in bursts, although the name DRAM/RAM has stuck. However, many types of SRAM, ROM and NOR flash are still random access even in a strict sense. RAM is often associated with volatile types of memory, where its stored information is lost if the power is removed. The first RAM modules to come into the market were created in 1951 and were sold until the late 1960s and 1970s.