CBCS SCHEME

USN | 2 | V | D | L | 8 | E | C | 0 | 6 | 8 |

18EC35

Third Semester B.E. Degree Examination, Dec.2019/Jan.2020
Computer Organization and Architecture

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. With a neat diagram, explain basic operational concept of computer. **(10 Marks)**
b. Explain in brief different types of key parameters that affect the processor performance. **(05 Marks)**
c. Explain the Bus Structures. **(05 Marks)**

OR

2 a. Illustrate Instruction and Instruction sequencing with an example. **(10 Marks)**
b. Define Byte Addressability, Big-endian and Little-endian assignment. **(06 Marks)**
c. Represent 85.125 in IEEE floating point using single precision. **(04 Marks)**

Module-2

3 a. What is an addressing mode? Explain any five types of addressing modes with example. **(10 Marks)**
b. Write a program to add 'n' number using indirect addressing mode. **(06 Marks)**
c. Explain various assembler directives used in assembly language program. **(04 Marks)**

OR

4 a. Explain stack operation with an example **(10 Marks)**
b. Explain subroutine linkage with an example using linkage register. **(06 Marks)**
c. Explain the shift and rotate operations with example. **(04 Marks)**

Module-3

5 a. Showing the possible register configuration in I/O interface, explain program controlled input/output. **(10 Marks)**
b. What is an interrupt? With an example illustrate the concept of interrupt. **(10 Marks)**

OR

6 a. Explain in detail, the situations where a number of devices capable of initiating interrupts are connected to processor. How to resolve the problems? **(10 Marks)**
b. Explain the registers involved in a DMA interface, to illustrate DMA. **(06 Marks)**
c. Explain the concept of Vectored Interrupt. **(04 Marks)**

Module-4

7 a. With figure, explain Internal Organization of 2M×8 dynamic memory chip. **(10 Marks)**
b. Illustrate Internal structure of static memories. **(10 Marks)**

OR

8 a. With a neat diagram, explain virtual memory organization. **(10 Marks)**
b. Briefly explain any four non-volatile memory concepts. **(05 Marks)**
c. Briefly explain secondary storage devices. **(05 Marks)**

Module-5

9 a. Explain the three-bus organization of the processor and its advantages. **(10 Marks)**

b. Discuss the organization of hardwired control unit. **(05 Marks)**

c. Discuss the control sequence for execution of instruction ADD(R_3), R_1 **(05 Marks)**

OR

10 a. With a block diagram, describe the organization of a micro programmed control unit. **(10 Marks)**

b. Describe the sequence of control signals to be generated to fetch an instruction from memory in a single bus organization. **(10 Marks)**

* * * * *

Name of Faculty : Prof. Shree Gowri S S

Name of Institution : KLS's VDIT Haliyal

Department : Electronics and Communication Engg.

Sem : III Semester B.E. Degree

Subject : Computer Organization and Architecture

Subject Code : 18EC35

Note : Scheme is as per the requirement of Subject

Subject Teacher

(Prof. Shree Gowri SS)

14.03.2022

Head of the Department
Dept. of Electronics & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)

Module - 1

1 a: With a neat diagram, explain basic operational concept of computer - 10 Marks

Answer

Block Diagram - 03 M
Explanation - 07 M
10 M

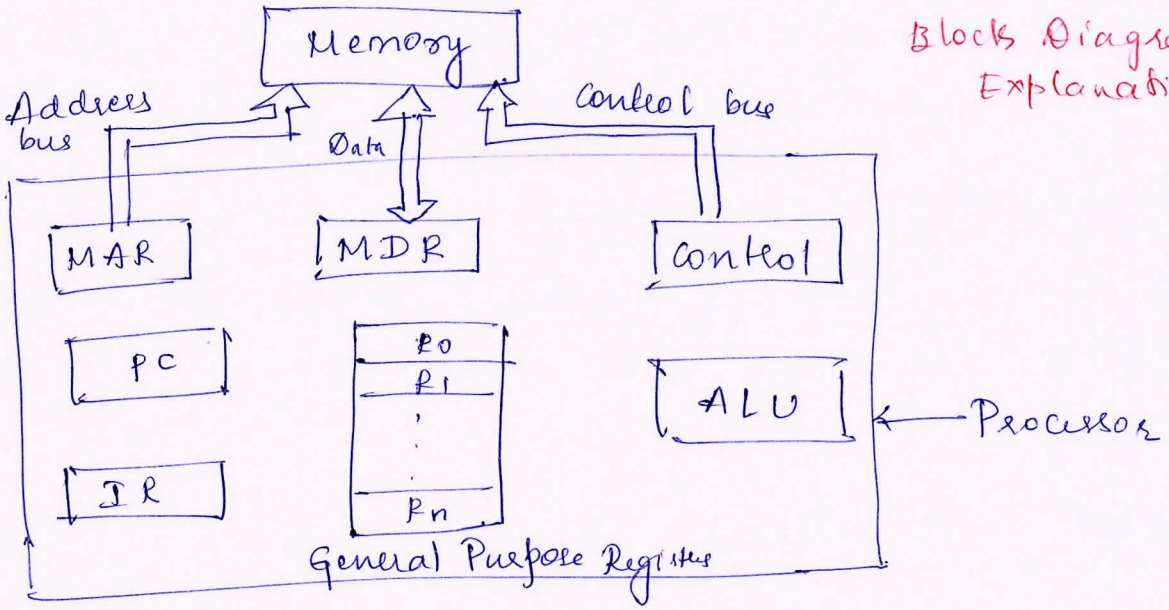


fig: Connection between processor and Memory

Program :- Set of instruction

The program must be stored in a memory and execution takes place by the processor

Processor consists of ALU, control unit, n-number of General purpose register ($R_0 - R_n$), Special purpose register PC (Program Counter), IR (Instruction Register) MDR (Memory Data Register) & MAR (Memory Address Register)

IR (Instruction Register) :- This register holds the instruction that is currently being executed

PC (Program Counter) :- The purpose of PC is to keep tracks of execution of program It contains

address of memory of next instruction to be fetched and executed i.e during execution of current instruction the content of PC is updated or incremented to next instruction memory address.

MAR (Memory Address Register) :- This Register holds the address of Memory location that needs to accessed either for data or instruction.

MDR (Memory Data Register) :- This register holds the data that is either fetched from the memory or ~~stored~~ that needs to be stored into the memory.

General Purpose Register (R_0 - R_n) :- These registers may be used either to hold the data (opeaand) ~~of~~ or intermediate results.

Control unit :- It Generates control signals to perform read and write operation

ALU :- ALU performs arithmetic and logical operations

Basic Operational concepts of a memory :- Program is stored in the memory. Execution of program starts when PC is set to point to the first instruction of program i.e PC holds the address of memory address of first instruction. The content of PC is transferred to MAR and read control signal is sent to memory. After the required time ~~for~~ memory elapses the addressed information is read out of memory and loaded into the MDR

Ans:- further the content of MDR is transferred to the IR register this is called as fetch stage.

Now the instruction is ready to decode and execute. The execution of instruction by ALU needs an operand hence it is necessary to obtain the required operand. The operand may be in memory or it may be available in general purpose registers.

If operand is in memory then address of that memory location is placed in MAR & once the read signal is issued, then operand is fetched from the memory and placed in MDR and finally it is transferred from MDR to ALU. It may access one or more operand in similar way.

Now ALU performs the desired operation and this stage is called as execute stage.

If the result of this operation needs to be stored in memory, then result is sent to MDR. The address of memory location where the data needs to be stored is placed in MAR & write signal is initiated. At the execution stage, the content of PC is incremented so PC points to the address of next instruction.

Hence as soon as execution of current instruction is completed, next instruction fetch can be started.

Explanation 07M

1b Explain in brief different types of key parameters that affect the processor performance. — 05 Marks

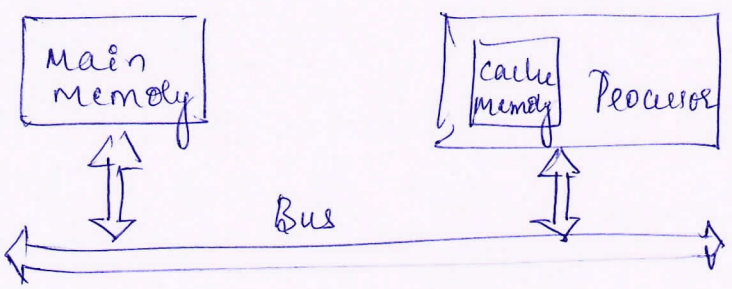
Ans :- Performance is how quickly a processor or computer system can execute the programs. Explanation – 05 M

Performance of a processor is sum of time periods only when the processor is active.

The processor time depends on a hardware involved in the execution of individual instructions

Hardware mainly includes processor and memory

In order to increase the speed of execution then a small memory called cache memory can be included along with the processor in same chip.



The instruction that is fetched from the main memory one after the other is placed in cache memory. This cache memory improves the performance of processor, when the processor make use of same data or same set of instruction repeatedly then rather than fetching from main memory its available in the cache memory itself

1C. Explain Bus Structure — 05 Marks

Ans:- Bus is a group of wires or lines, which is used to interconnect the functional units of computer. Mainly there are three types of bus & they are
1) Address bus 2) Data bus 3) Control bus

Bus Structure provides different methods or ways of connecting the device
The different types are
1) Single bus Structure
2) Multiple bus Structure

Block diagram of Single bus structure 02M
Explanation 03M
05M

Single Bus Structure :- Means only one bus or common bus is used for communication.

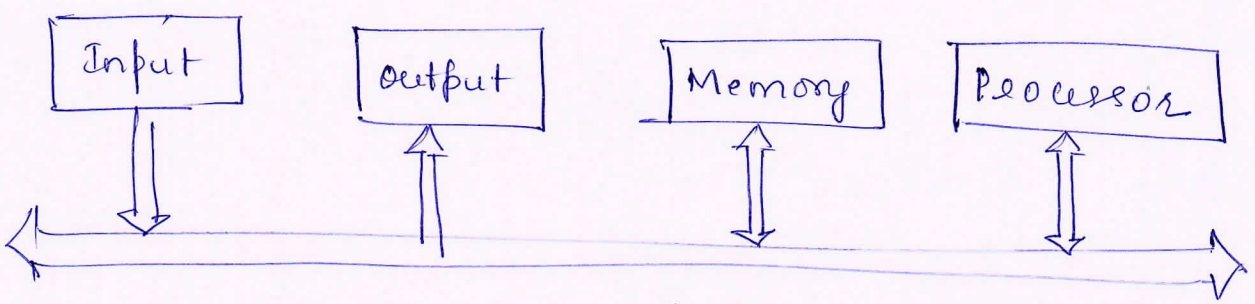


fig: Single bus Structure

Single bus Structure offers low cost and flexibility for attaching peripheral devices. There is only one transfer at a time i.e only two units can actively use the given bus at any given time

Where as Multiple bus Structure have more concurrency in operations allowing two or more transfer to be carried out at the same time. This leads to a better performance at an increased cost.

OR

2a Illustrate instruction and instruction Sequencing with an example — 10 marks

Ans:- Illustration of instruction execution — 05 M

Consider example $C \leftarrow [A] + [B]$ statement need to be executed by the processor

for the above statement program is written using two-address instruction

Program for $C \leftarrow [A] + [B]$

```
MOVE  A , R0
Add   B , R0
MOVE  R0 , C
```

Assume every instruction size is of 32-bits or 4 bytes long and also first instruction of the program is stored ~~from~~ at an address i and subsequent instruction on subsequent memory location

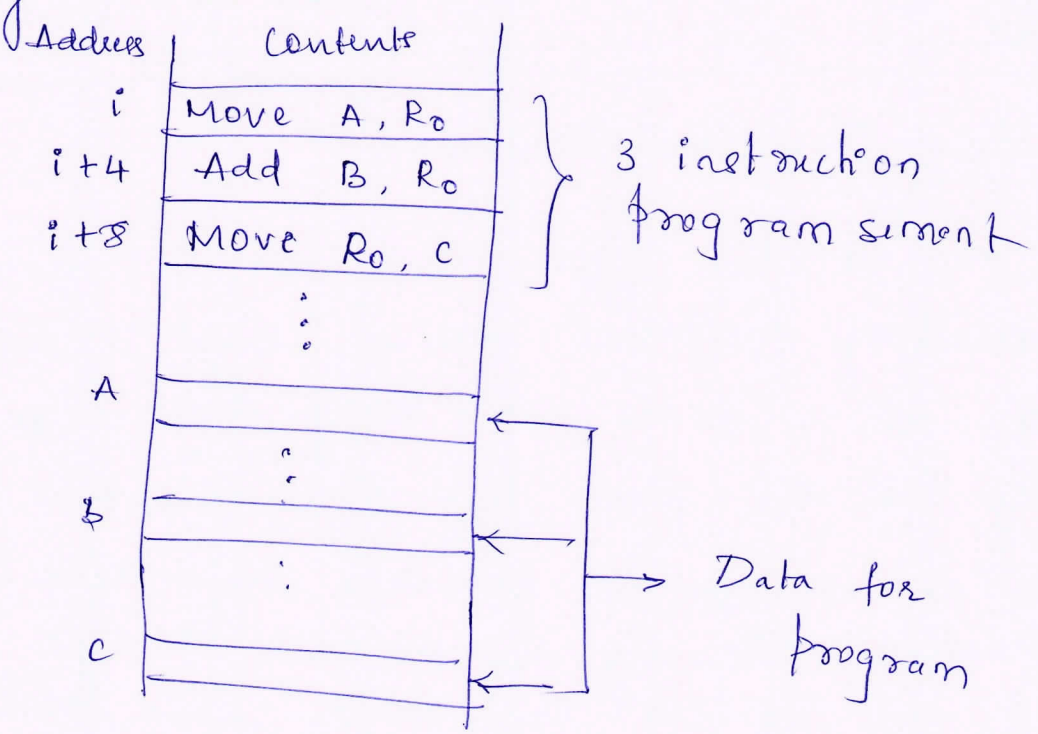


fig: Memory structure for program $C \leftarrow [A] + [B]$

One of the processor register is PC (Program counter) which always holds the address of the next instn that needs to be fetched for execution

To begin execution, the address of the first instruction i is placed into PC. The address that is in PC is used to fetch and execute by the processor control circuit. Only one at a time and in increasing address. This is called as "Straight line Sequencing". — OSM

During execution of each instruction with respect to above example, the PC is incremented by 4 to point to the next instruction.

Thus after execution i+8 address instruction, the PC contains i+12, which is the address of the first instruction of next program.

Executing a given instruction is a two-phase procedure i.e
1) first phase → Instruction fetch
2) Second phase → Execute phase.

1) first phase :- The instruction is fetched from the memory location whose address is in PC. The fetched instruction is placed in Instruction Register.

2) Second phase :- The instruction in IR is examined to determine which operation to be performed. The specified operation is performed by the processor i.e it involves
1) fetching of operands either from memory

or from processor register

2) performing arithmetic and logical operation

3) Storing the result in the destination location.

At some point during execute phase the content of PC is incremented to point to next instruction.

Q6 Define Byte addressability, Big-endian and Little-endian assignment ——— 06 Marks.

Ans:- Byte addressability is assigning address to every byte of memory.

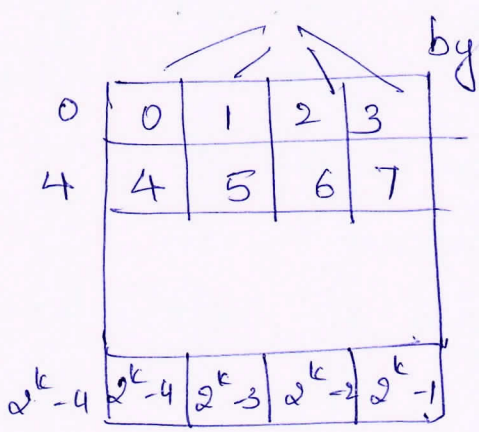
There are two ways of assigning byte address across word address

1) Big-endian assignment ——— 03M

2) Little-endian assignment. ——— 03M

06M

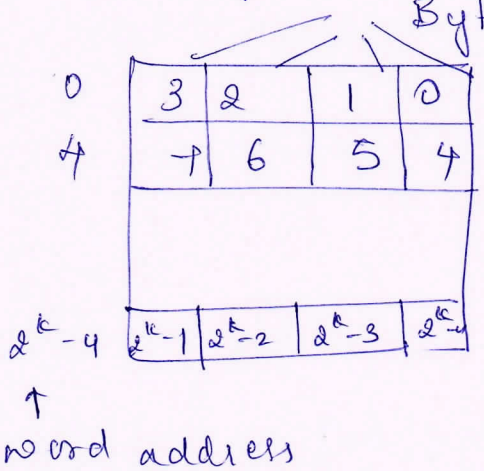
1) Big-endian assignment :- Big-end of the data is MSB and it is stored at lower byte address.



i.e storing of data starts from MSB to LSB.

b) Little - endian assignment

Little - endian means Little - end of data is LSB and it is stored at lower - byte address i.e storing of data starts from LSB



Q) Represent 85.125 in IEEE floating - point using Single - precision — 04 Marks

Ans:- — 04M

85.125

85 = 1010101

0.125 = 001

$85.125 = 1010101.001 \times 2^0$

$= 1010101.001 \times 2^1$

$= 1.01010101 \times 2^6$

1) Sign = 0 → Number is positive

2) Exponent in IEEE field is E' which is equal E+127

$\therefore E' = E + 127$

$= 6 + 127$

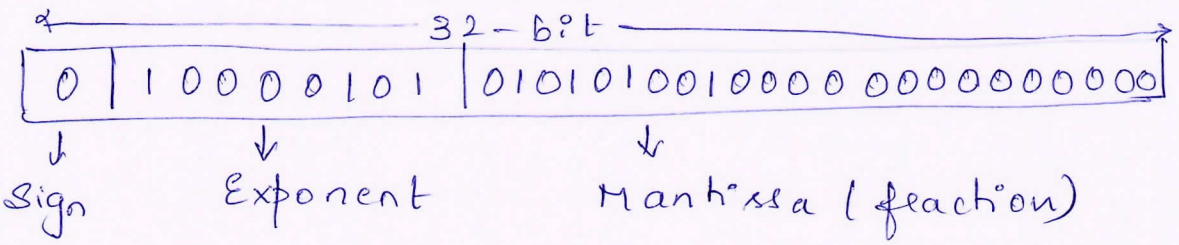
$E' = 133$

$\therefore 133 = 10000101$

Page

3) Mantissa = 01010100 1

in order to make it as 23-bits append 0's at the end



Module - 2

3a What is an addressing mode? Explain any five types of addressing modes with example — 10 M

Ans:- Addressing mode provides different methods or various ways of accessing data — 02M

Five different types of addressing modes are — 08 M

10 M

1) Immediate Mode
2) Register Mode
3) Direct Mode
4) Indirect Mode
5) Index Mode

1) Immediate Mode :- In immediate addressing mode the data is available in the instruction itself. Immediate mode is identified by # sign

Example :- Move #200, R0

After execution of instruction R0 = 200.

2) Register Mode :- In register addressing mode both source and destination are register. In register addressing mode data is available in one of the general purpose register.

Example :- MOVE R_0, R_1

Operand or data is in R_0, the content of R_0 register is copied to R_1 register.

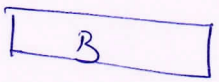
3) Direct Mode :- In direct mode, address of memory of the operand is explicitly specified in the instruction.

Example :- MOVE NUM, R_2

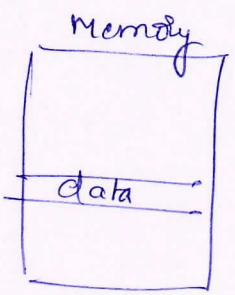
Where NUM is the address of memory which holds the data/operand.

4) Indirect Mode :- In indirect mode, address of memory of the operand is not directly specified rather memory address of the operand is specified in one of the register.

Example :- MOVE (R_0), R_2



With respect to above example R_0 register holds the ad memory address of the memory i.e B



In B memory location, the data is present & that data is transferred to R_2 Register

5) Index Mode :- In index addressing mode EA is calculated by adding an offset value to the content of index registers. After calculating EA, the content of index registers is not changed. Index mode is in the form of X(Ri) where X is an offset, Ri is an index register

$\therefore$ EA = X + [Ri]

Example :- Move 20(R0), R2 R0 | 1000

$\therefore$ EA = X + [Ri]

where Ri = R_0

$\therefore$ EA = 20 + 1000

= 1020 → is the address of the data

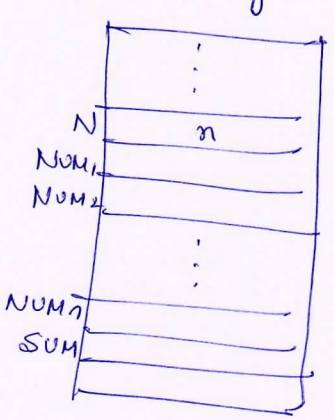
3b. Write a program to add 'n' numbers using indirect addressing mode —— 06 Marks

Program :-

```
        MOVE   N, R1
        MOVE   #NUM1, R2
        Clear  R0
loop:   Add    (R2), R0
        Add    #4, R2
        Decrement R1
        Branch >0 loop
        MOVE   R0, SUM
```

Program — 06 M

Memory



3C Explain assembler directives used in assembly language program — OR Marks

Ans:- Assembler directives :- They are the commands given to assembler i.e it gives direction to the assembler how the part of the program must be treated. They are called as Pseudocodes hence they are never executed by the processor.

Assembler directives are — 1X4 = 04M

1) EQU :- EQU is an equate directive

Ex:- SUM EQU 200

This directives equates or assigns value to the name. i.e 200 value is assigned to SUM. Now in program wherever SUM appears then it is replaced by 200.

2) ORIGIN

ORIGIN is an assembler directive, which tells where the program or data required for program should be stored in a memory i.e from which memory location it must stored.

Example ORIGIN 100

This example says from memory location 100 the below mentioned instruction/data after ORIGIN statement should be saved from an address 100.

3) END

END is an assembler directive. It is the end of the program and it also specify the address from where the processor starts the execution of program Example:- END start

Page 12

OR

Q4a:- Explain stack operation with an example — 10 Marks

Ans:- Stack is an area of memory, it is a data structure that works on the concept of LIFO in order to store and retrieve the information. storing the data into stack is called as Push operation retrieving the data from stack is called as POP operation.

Push Operation :- Push is used to store data onto stack. SP → Stack pointer is a register always holds the address of the top current element of the stack.

In order to store the element onto the stack the SP must be decremented, before storing the data

Example :-

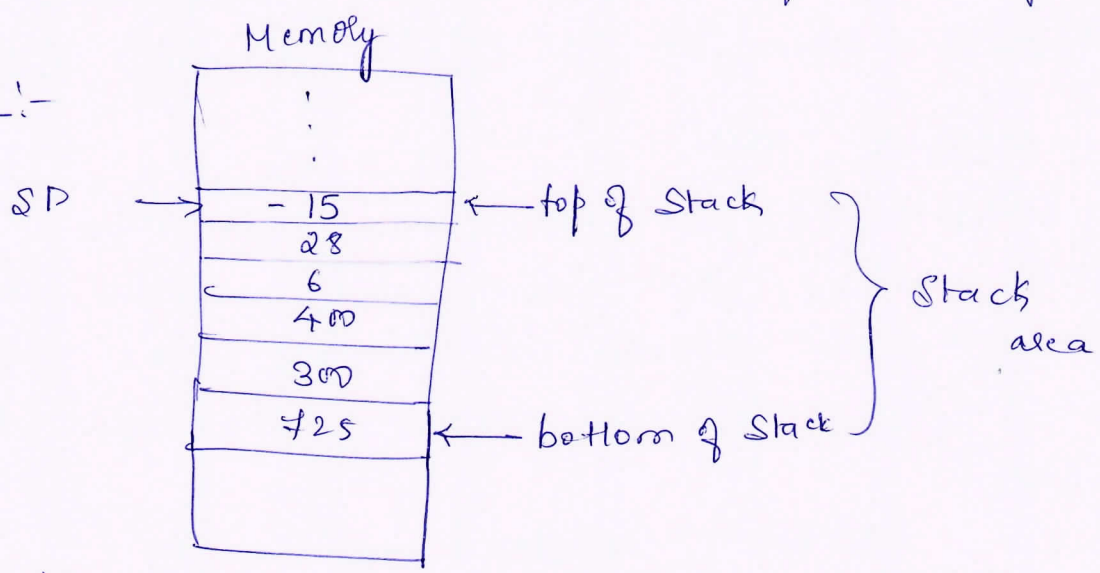


fig : Stack Memory structure

In the above example SP is pointing to the current top element

To perform Push operation the below instruction can be used

Ex:- Subtract # 4, SP

Move # 19, (SP)

(or)

Move # 19, -(SP)

either use Subtract instruction or auto decrement mode before storing data onto stack.

Pop operation:- To retrieve the data from the stack, the data that was pushed onto the stack for the last time is the first one to remove. We know SP points for current top element. After retrieving the data from stack, the SP must be incremented.

Therefore instruction required are

```
MOVE    (SP), NUM
Add     #4, SP
```

Push operation - 05M
Pop operation - 05M
10 M

OR

```
MOVE    (SP)+, NUM
```

either use add instruction or auto ↑ increment mode after retrieving the data from stack so that it can point to next memory location or points to current top element of stack.

4b Explain Subroutine Linkage method with an example using Linkage Register. — 06 Marks

Ans:- Subroutine linkage method is used to save the return address in the special register called as Linkage Register

Explanation 03M
Example 03M
06M

when the subroutine completes its tasks, the return instruction return to the calling program by using the content of link register

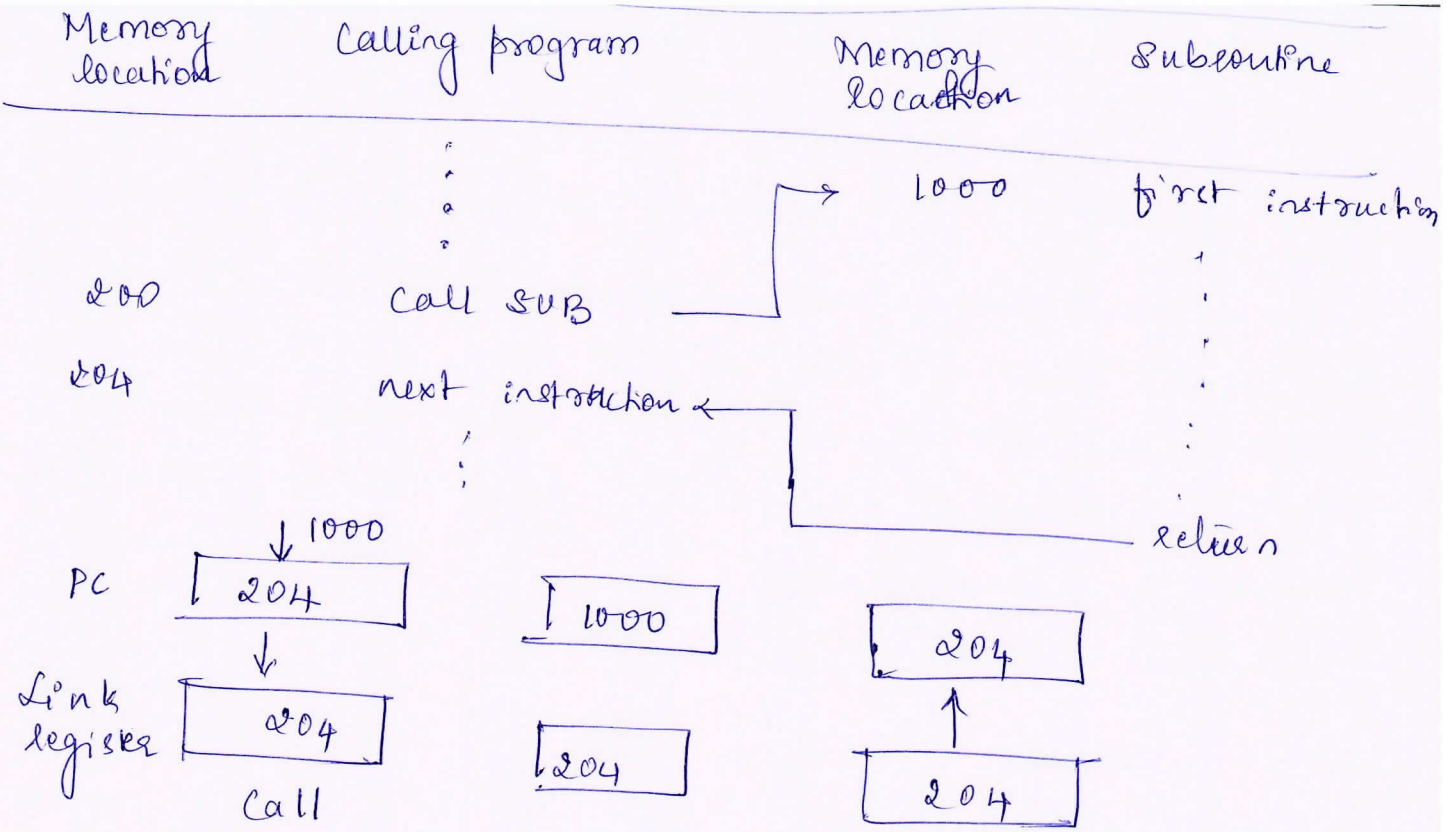


fig: Subroutine linkage using link register

Call instruction performs the following operations

1) Store the content of PC in the link register
2) Branch to target address specified in the call instruction

Return instruction performs the following operations

Branch to address contained in link register

4. Explain Shift and rotate operations with example — 04 marks

Any Shift operations are :-

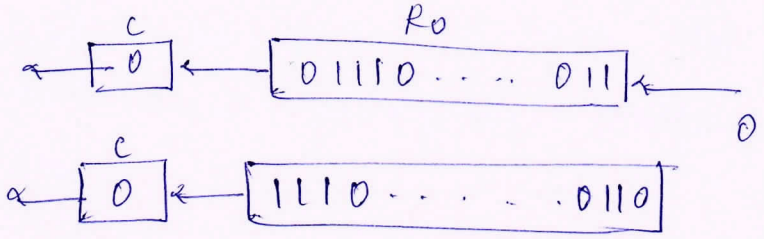
1) Logical Shift < LShift L / LShift R
2) Arithmetic Shift < A Shift L / A Shift R

Shift operation 02M
Rotate operation 02M
04M

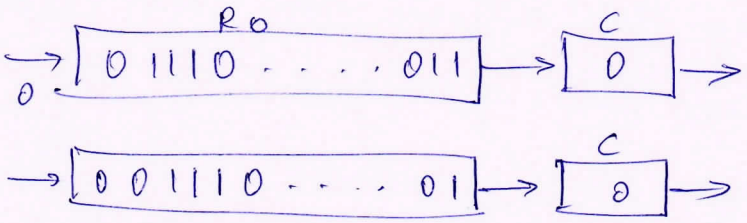
Rotate Operations are

1) Rotate L
2) Rotate R
3) Rotate LC
4) Rotate RC

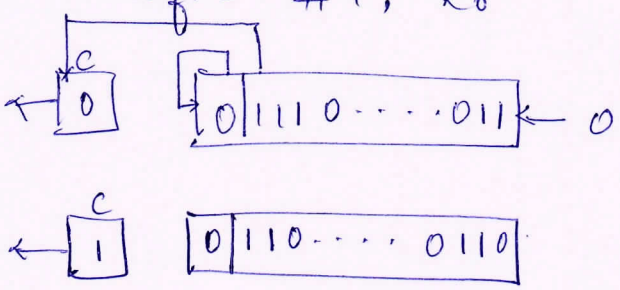
Example :- 1) LShift #1, R0 let R0 = 01110 ... 011



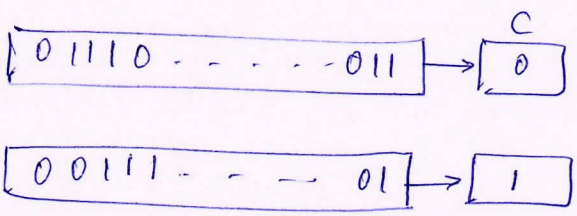
2) LShiftR #1, R0



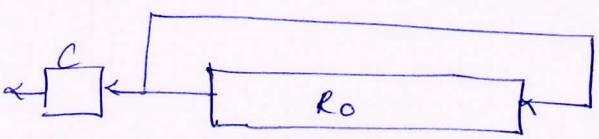
3) AShiftL #1, R0



4) AShiftR #1, R0

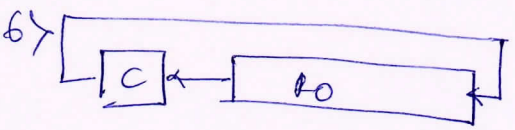


5)



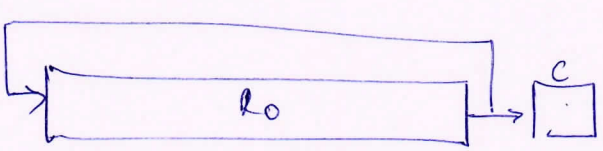
Rotate left

6)



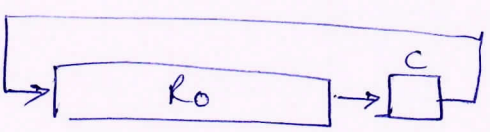
Rotate left with carry

7)



Rotate Right

8)



Rotate Right without carry

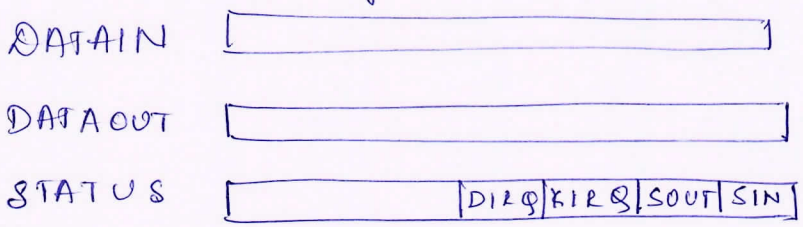
Module 3

5a. Showing the possible register configuration in I/O interface, explain program controlled input/output — 10 Marks

Ans:- In program controlled I/O, the processor repeatedly checks a status flag to achieve the required synchronization between the processor and an input or output device i.e processor polls the device. i.e It checks the flags in Status Register

Consider an example of an I/O operations which involves keyboard and display device in a computer system.

The register and status register associated with Keyboard and display device are
1) DATAIN register 2) DATAOUT register 3) STATUS Register



The Status register contains two control flags SIN & SOUT which provides Status information for the keyboard and display unit respectively. Data from keyboard is stored in DATAIN register until it is sent to processor. The data to be sent to display device is stored in DATAOUT register.

Program to understand Program controlled I/O

* program to read one line from the keyboard, stores it in memory buffer and echoes it back to the display.

Program:-

```
        MOVE    # line, R0
WAIT Read  Testbit  #0, Status
           Branch=0   WAIT read
           MOVE    DATAIN, R1
WAIT write Testbit  #1, Status
           Branch=0   WAIT write
           Move    R1, DATAOUT
           Move    R1, (R0)+
           Compare #$0D, R1
           Branch ≠0  WAIT read
           Move    #$0A, DATAOUT
           Call Process
```

Explanation 05M
Program 05M
10M

Disadvantage of Program controlled I/O :- The processor waste its time in checking the status of the device before the actual data transfer takes place.

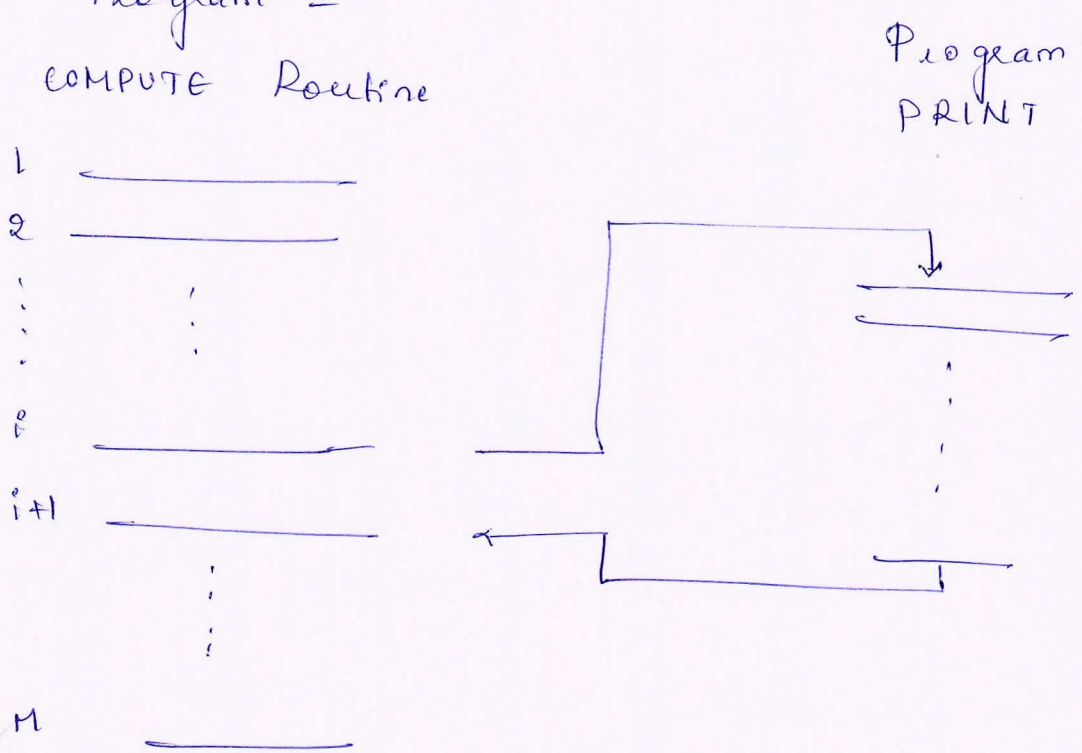
5b what is an interrupt ? with an example Illustrate the concept of interrupt — 10 Marks.

Ans:- In order to achieve synchronization, as soon as I/O device becomes ready to transfer or receive the data, it sends a special signal over the bus to the processor. This mechanism is called as Interrupt. or a signal used to alert the processor

when I/O device becomes ready it an interrupt.

Interrupt request line :- Interrupt request line is one control bus line which carries interrupt signal.

Example : To illustrate the idea of interrupt consider a task that requires some computations and the results to be printed on a printer. Let the program contains two routines, COMPUTE and PRINT routine. COMPUTE routine after computation produces a set of n lines as output, the set of n lines needs to be printed by PRINT routine



Defn - 02M
Example
Explanation - 08M
10M

Fig: Transfer of control through the use of interrupts

Let us assume processor and printer works simultaneously in order to save the processor time Hence consider

1) COMPUTE routine is executed to produce the first n lines of output, Then PRINT routine

is executed to send first line of output to printer, now when the printer is executing the first line of output instead of waiting for a printer till it prints.

2. Execution of COMPUTE routine is resumed back by suspending the execution of PRINT routine.

3. When the printer completes printing the first line of output and it is ready to print the next line

4. When printer becomes ready it sends an interrupt request signal to the processor and at that time processor is executing COMPUTE routine

5. In response to interrupt signal the processor stops executing the COMPUTE routine and transfers the control to the PRINT routine

6. Now processor executes print routine and sends second line in a first set of n-lines and after sending PRINT routine gets suspended control

7. This process repeates until all n lines have been printed and the PRINT routine ends

8. PRINT routine starts whenever next set of n line is available for printing.

OR

6a Explain in detail, the situations where a number of devices capable of initiating interrupts are connected to processor. How to resolve the problems — 10 Marks

Ans:- When multiple I/O devices are connected to common interrupt request line may generate an interrupt at the same time.

Hence there are different methods to handle multiple device connected to common interrupt request line. 01M

1) Polling the IRQ - bit in processor register
2) Vectored interrupt
3) Daisy chain

1) Polling the IRQ - bit in processor register:-

When several I/O devices connected to common interrupt request line generates interrupt at the same time then to identify the devices which raised interrupt requires additional information. 03M

i.e when device ~~gets~~ raise the interrupt it sets one of the bit in status register called IRQ to 1.

When multiple devices raise interrupt simultaneously by setting its IRQ bit in status register to 1, now the processor provides service to requested one by breaking the tie
or interrogate

Processor polls the IRQ - bits of all the devices this method is called as polling.

2) Vectored interrupts:- Instead of wasting processor time, the device which generates interrupt also sends special code to processor 03M

3. Daisy chain

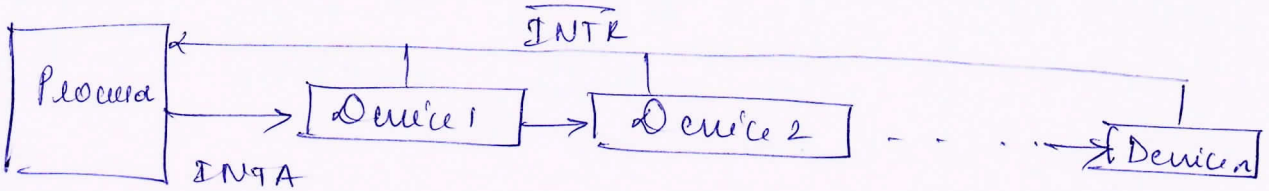


fig: Daisy chain

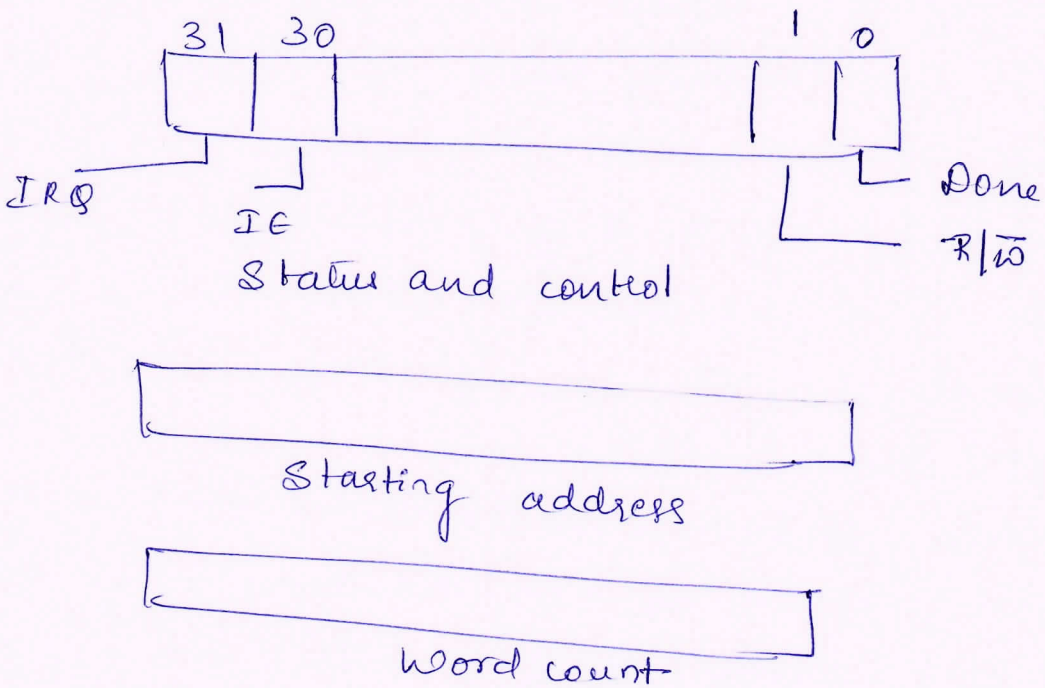
03M

In this method, the device that is electrically closest to the processor gets the service by blocking INTA. If device does not requires any service then INTA signal will be passed to next device.

01
+03
+03
+03
10

6b. Explain the registers involved in DMA interface to illustrate DMA — 06 Marks.

Ans:- Registers that are involved to initiate DMA transfer operations are — 03M



1) R/W - bit - indicates the direction of the transfer. — 03M

R/W = 1 then controller performs read operation

06M

R/$\bar{W}$ = 0 performs write operation

2) Done flag → This bit is set to 1 when the controller completes transferring block of data

3) IE - bit → After DMA transfer is completed DMA controller informs the processor that data transfer is completed by raising an interrupt i.e by setting IE-bit = 1, control circuit causes the controller to raise an interrupt.

4) IRQ bit → Controller sets IRQ = 1 after raising an interrupt.

6C. Explain the concept of vectored Interrupts — 04 Marks

Explanation- 04M

When multiple I/O devices connected to common interrupt request line may generate the interrupt simultaneously thus to identify the requesting devices, the requesting device itself will identify by sending a special code to the processor. This enables the processor to identify individual devices even if they share a single interrupt request line.

This approach is called as <u>vectored - interrupts</u>

The code supplied by the I/O device may represent the starting address of ISR. This code is sent to the processor through data bus. Whe The I/O device needs wait for data bus because when a interrupt has occured the currently running instruction ~~may~~ might be using the data bus. Thus once data bus becomes free, special code is sent

Module - 4

7a. With figure, explain internal organization of 2M*8 dynamic memory chip ——— 10 Marks

Block Diagram 04M
Explanation 06M
10M

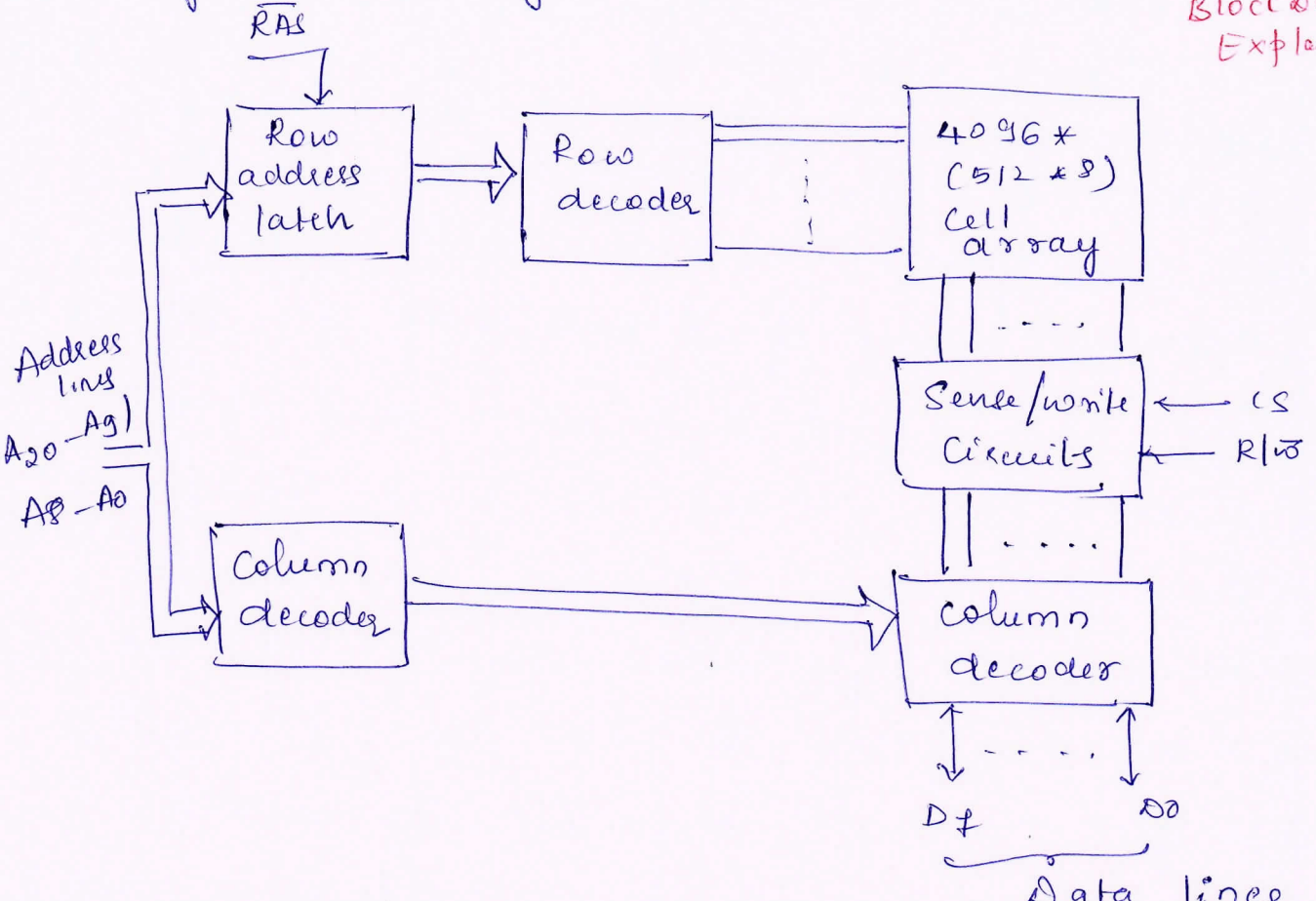


fig: Internal Organization of 2M*8 DRAM chip

The above figure shows the organization of a 16M bit DRAM structured as 2M*8 memory. The cells are organized as 4K*4K array. The 4K cells in each row are divided into 512 groups of 8-bits each i.e the row can store 512 bytes of data. Thus 12 address bits are required to select a row. Another 9 bits are needed to specify a group of 8-bits in the selected row. Thus a 21-bit address is needed to access a byte. The high-order 12-bits and the low-order 9-bits of the address constitute the row and column address of a byte. During a read or write operation, the row address is loaded &

into row address latch in response to a signal on RAS input of the chip. Then a read operation is initiated, in which all cells on the selected row are read and refreshed

Column address is applied to the address pins and loaded into column address latch using CAS signal. The information in this latch is decoded and appropriate group of 8 sense/write are selected

If the R/$\overline{W}$ signal indicates a read operation, the output of the selected circuits is transferred to the data lines $D_7 - D_0$. For a write operation, the information on the $D_7 - D_0$ lines are transferred to the selected circuits. The information will overwrite the contents of the selected cells in the corresponding 8 columns

7b. Illustrate internal structure of Static Memories — 10 Marks

Ans:- Static memory cell is capable of retaining its state as long as power is applied. Block Diagram 04M Explanation 06M 10M

The below figure shows static RAM cell

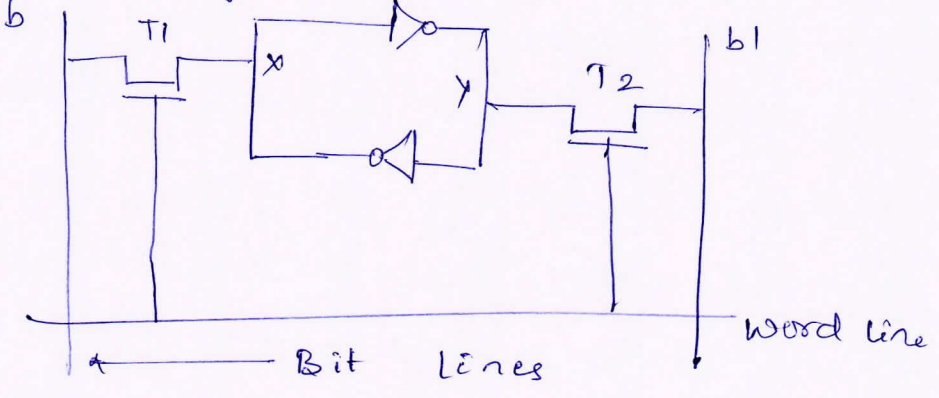


fig: Static RAM cell

In specified SRAM cell, the latch is formed by cross-coupling two inverters. The transistors T_1 and T_2 acts as switches. The latch is connected between two bit lines b and b' by transistors T_1 and T_2

Read operation :- To read the state of SRAM cell, the word line is activated to close switches T_1 and T_2. If the cell is in state 1, the signal on bit line b is high and b' is low. & opposite state also exist.

Write operation :- Appropriate value is put on bit line b and its complement on line b' & then word line is activated. This forces the cell into the corresponding state. The required signals on the bit line are generated by sense/write circuit.

OR

8a. With a neat diagram, explain virtual memory Organization — 10 Marks

Ans:- Physical main memory is not as large as the address space spanned by an address issued by the processor. The size of main memory in a typical computer ranges from a few hundred megabytes to 1G bytes. When a program does not completely fit into a main memory, the parts of it not currently being executed are stored on secondary storage devices such as magnetic disks. Of course, all parts of a program that are eventually executed are first brought into the main memory

when a new segment of a program is to be moved into a full memory, it must replace another segment already in the memory.

Techniques that automatically move program and data blocks into the physical main memory when they are required for execution are called "Virtual-memory techniques". The binary address that the processor issues for either instruction or data are called "Virtual or Logical address"

These addresses are translated into physical addresses by a combination of hardware & software components. On the other hand, if the referenced address is not in main memory, its contents must be brought into suitable location in the memory before they can be used.

Block Diagram 05M
Explanation 07M
10M

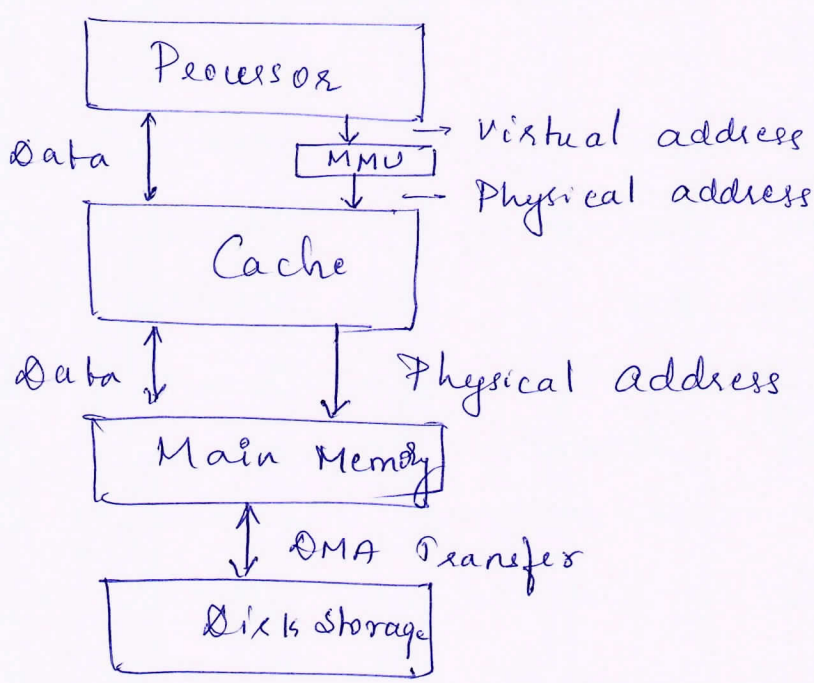


fig: Virtual Memory Organization

The figure shows typical organization that implement virtual memory. A special hardware unit called Memory Managment Unit (MMU), translates virtual address into physical address.

When the desired data are in the main memory, these data are fetched as described. If the data are not in main memory, the MMU causes the operating System to bring the data into memory from the disk. Transfer of data between the disk and main memory is performed using DMA.

8 b. Briefly explain any four non-volatile memory concepts — 05 marks

Ans:- Four non-volatile memory are :- Explanation - 05 M

1) Programmable ROM (PROM)

ROM designs allow the users to load the data. This is PROM. Programmability is achieved by inserting a fuse at point P.

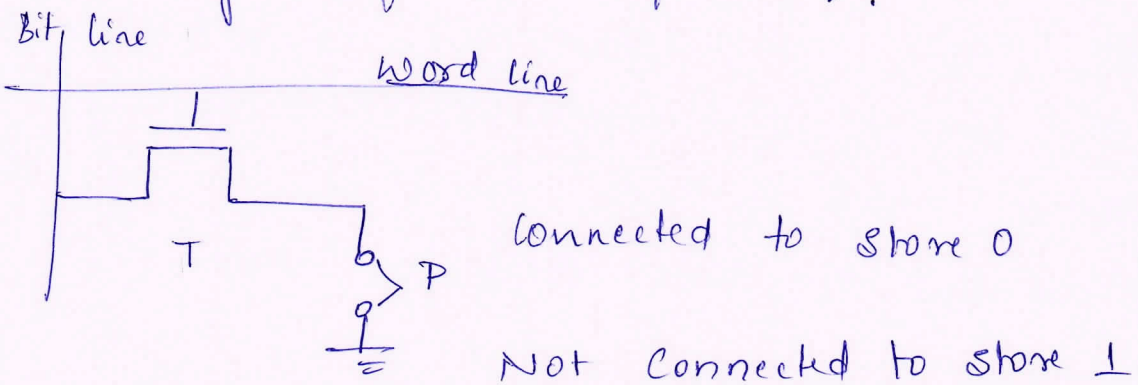


fig: ROM cell

i.e before it is programmed, all the memory cell contains 0's. The user can insert 1's at the required locations by burning out the fuses at these locations using high current pulses

2) Erasable PROM (EPROM):-

A modification to PROM allows the data to be erased and new data to be loaded many number of times. Such an erasable, reprogrammable ROM is called EPROM. They provide greater flexibility during software development stage of a system. The stored information can be erased by dissipating the charges trapped in the transistors of the memory cells. This is done by exposing the chip to UV light for a fixed period.

3) Electrically Erasable PROM (EEPROM) :- EPROM

Another version of EPROM allows programming and erasing electrically. Such chips are known as EEPROM. In EEPROM the contents can be selectively erased by passing current pulse instead of UV light. The disadvantage of EEPROM is that different voltages are needed for erasing, writing and reading the stored data.

4) Flash Memory:- In a flash chip, it is possible to read the contents of a single cell, but writing is done for entire block of cells. Prior to writing, the previous contents of that block are erased.

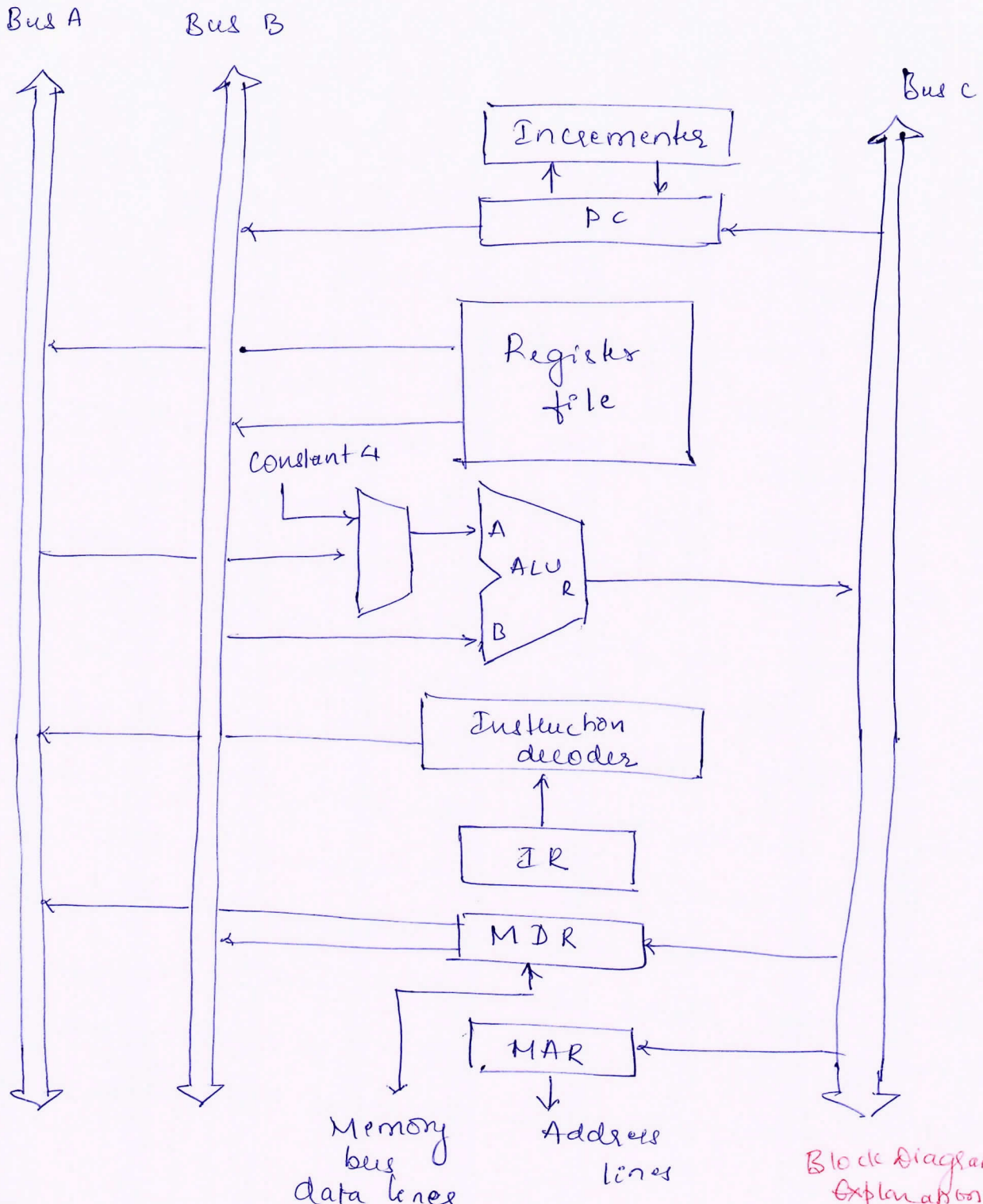
8c. Briefly explain Secondary Storage devices — 05 Mark
Explanation — 05M

Ans:- Magnetic disks System consists of one or more disks mounted on a common spindle. A thin magnetic film is deposited on each disk usually on both sides. The disks are placed in a rotary drive so that magnetized surfaces move in close proximity to read/write heads. The disks rotate at a uniform speed. Each head consists of magnetic yoke and a magnetizing coil. This causes the magnetization of the film in the area immediately underneath the head to switch to a direction parallel to the applied field. The same head can be used for reading the stored information. In this case, changes in the magnetic field in the vicinity of the head caused by the movement of the film relative to yoke induce a voltage in the coil, which now serves as a sense coil. Magnetic Hard disk is one of the example of secondary story device and similarly the other examples are floppy disks & RAID disk.

Module - 5

9a. Explain the three - bus organization of the processor and its advantages — 10 Marks.

Ans:- Multiple bus organization or three bus organization of the processor consists of three internal bus in the processor



Block Diagram 03 M
Explanation 07 M
10 M

fig : Three - bus organization of the data-path

The figure depicts a three-bus structure used to connect the registers and the ALU of the processor. All general purpose registers are combined into the single block called the register file. In VLSI technology the most efficient way to implement a number of registers is in the form of an array of memory

The register file is said to have three ports. There are two outputs, allowing the contents of two different registers to be accessed simultaneously and have their contents placed on bus A and B. The third port allows the data on the bus C to be loaded into a third register during the same clock cycle.

Buses A and B are used to transfer the source operands to the A and B inputs of the ALU, where an arithmetic and logic operation may be performed. The result is transferred to the destination over the bus C.

A second feature is the introduction of the Incrementer unit, which is used to increment the PC by 4. Using the incrementer unit eliminates the need to add 4 to PC using the main ALU. It can be used to increment the other addresses, such as the memory address in Load Multiple & Store Multiple instructions.

Advantage:- As there are multiple paths for data transfers, a significant reduction in the number of clocks cycles needed to execute an instruction is achieved

9b. Discuse the Organization of hardwired control unit — 05 Marks

Ans:- Hardwired control unit is the method used to generate the control signals.

Block Diagram - 02M
Explanation - 03M
05M

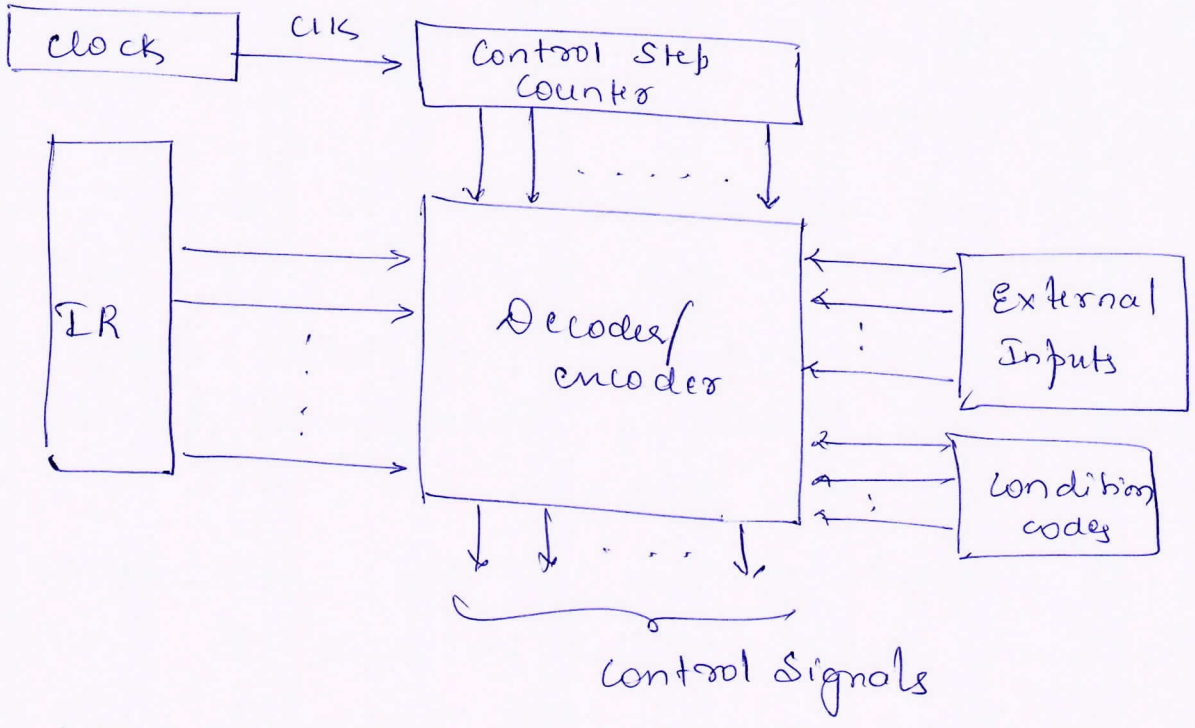


fig: Control Unit Organization.

In hardwired control unit the required control signals are determined by the following information:

1) Contents of the control step counter
2) Contents of instruction register
3) Contents of condition code flags
4) External input signals such as MFC & interrupt requests

The decoder/ encoder block is a combinational circuit that generates the required control outputs, depending on the state of all its inputs.

9c Discuss the control sequence for execution of instruction ADD (R_3), R_1 — 05 Marks

Ans: The control sequence for execution of the instruction ADD (R_3), R_1 — 05M

Steps	Action
1	PCout, MARin, Read, select 4, Add, Zin
2	Zout, PCin, Yin, WMFC
3	MDRout, IRin
4	R3out, MARin, Read
5	R1 out, Yin, WMFC
6	MDRout, Select Y, Add, Zin
7	Zout, R1 in, End.

OR

10a. With a block diagram, describe the organization of a Microprogrammed control unit — 10 Marks.

Ans: Microprogrammed control unit is used to generate control signals i.e control signals are generated by a program. A control word is a word whose individual bits represents the various control signals. Each control steps in the control sequence of an instruction defines a unique combination of 1's and 0's in the CW. The CW corresponds Assume that Select Y is represented by Select = 0 and Select 4 by Select = 1. A sequence of CW's corresponding to control sequence of machine

Page

instruction constitutes the microroutine for that instruction and the individual control words in this microroutine are referred to as Microinstructions.

Diagram – 05M
Explanation – 05M
10M

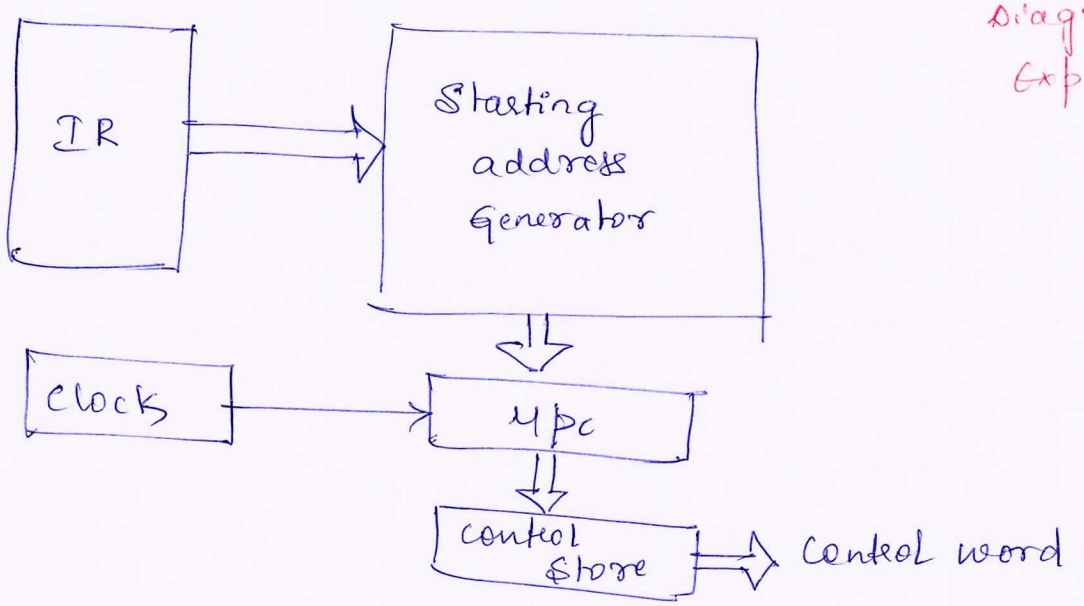


fig: Basic Organization of a Microprogrammed Control

Example: Add (R_3), R_1

Control steps

1. PCout, MARin, Read, Select 4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDRout, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, Select Y, Add, Zin
7. Zout, R1in, End

Micro instruction for the above control sequence

Micro Instruction	PC_{in}	PC_{out}	MAR_{in}	Read	MDR_{out}	IR_{in}	Y_{in}	Select	Add	Z_{in}	$R1_{in}$	R_3	WMFC	End
1	0	1	1	1	0	0	1	1	1	1	0	0	0	0
2														
3														
4														
5														
6														
7														

The microprogrammed control works by loading the instruction into IR register, the output of starting address generator is loaded into μpc. The μpc is automatically incremented by the clock causing successive micro instruction/ control word to read from the control store.

In order to implement conditional or unconditional branch instruction, basic organization of micro-programmed control unit cannot handle so a modified control unit is used for handling conditional branching.

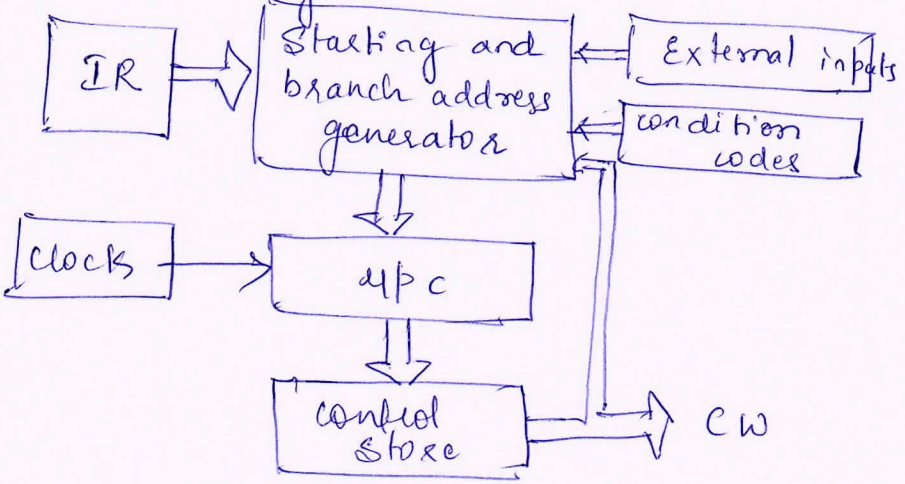


fig: control unit for handling branch instruction

10 b. Describe the sequence of control signals to be generated to fetch an instruction from memory in a single bus organization ——— 10 Marks

Ans:- Consider an example of an instruction

Add (R_3), R_1

control signal for fetching and executing above instruction is ——— 10 M

Step	Action
1	PCout, MARin, Read, Select 4, Add, Zin
2	Zout, PCin, WMFC
3	MDRout, IRin
4	R3 out, MARin, Read
5	R1 out, Yin, WMFC
6	MDRout, Select Y, Add, Zin
7	Zout, R1in, End