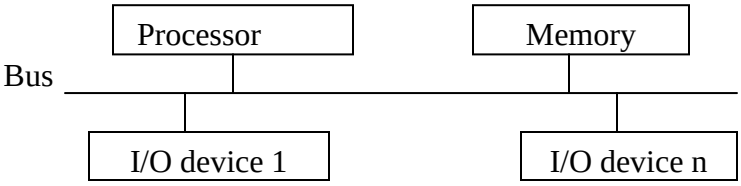
Module: 5	**I/O ORGANIZATION –** Accessing I/O devices – interrupts - interrupt hardware – Direct Memory Access. **THE MEMORY SYSTEM -** Basic concepts – Semiconductor RAMs – Memory system considerations – Semiconductor ROMs –Content Addressable memory – Cache Memory -Mapping Functions.

ACCESSING I/O DEVICES

- A simple arrangement to connect I/O devices to a computer is to use a single busstructure. It consists of three sets of lines to carry
  - Address
  - Data
  - Control Signals.
- When the processor places a particular address on address lines, the devices that recognize this address responds to the command issued on the control lines.
- The processor request either a read or write operation and the requested data are transferred over the data lines.
- When I/O devices & memory share the same address space, the arrangement is called **memory mapped I/O.**

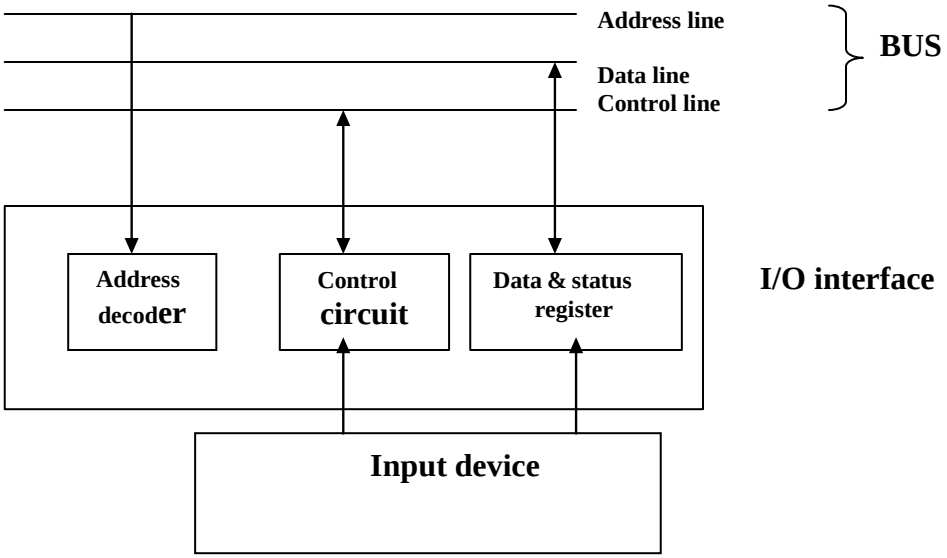
Single Bus Structure



Eg:-

Move DATAIN, R_o - Reads the data from DATAIN then into processor register R_o.
Move R_o, DATAOUT - Send the contents of register Ro to location DATAOUT.
DATAIN - Input buffer associated with keyboard.
DATAOUT - Output data buffer of a display unit / printer.

Fig: I/O Interface for an Input Device



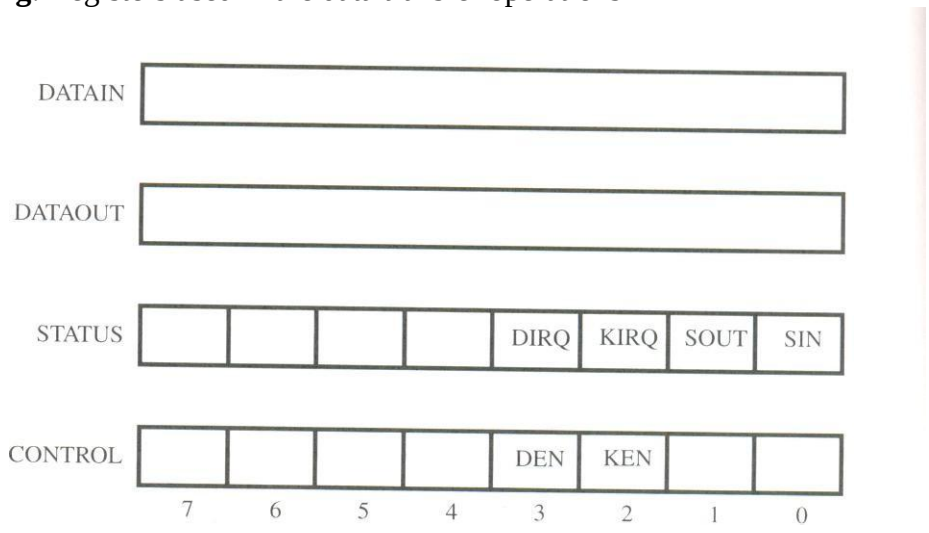
Address Decoder:

- It enables the device to recognize its address when the address appears on addresslines.

Data register - It holds the data being transferred to or from the processor.
Status register - It contains information relevant to the operation of the I/O devices.

- The address decoder, data & status registers and the control circuitry required to co-ordinate I/O transfers constitute the device's I/O interface circuit.
- For an input device, SIN status flag is used SIN = 1, when a character is entered at the keyboard, SIN = 0, once the char is read by processor.
- For an output device, SOUT status flag is used.

Eg: Registers used in the data transfer operations



DIRQ - Interrupt Request for display.
KIRQ - Interrupt Request for keyboard.
KEN - Keyboard enable.
DEN - Display Enable.
SIN, SOUT - Status flags.

The data from the keyboard are made available in the DATAIN register & the data sent to the display are stored in DATAOUT register.

Program:

```
        Move      #LINE,R0          Initialize memory pointer.
WAITK   TestBit   #0,STATUS         Test SIN.
        Branch=0  WAITK             Wait for character to be entered.
        Move      DATAIN,R1         Read character.
WAITD   TestBit   #1,STATUS         Test SOUT.
        Branch=0  WAITD             Wait for display to become ready.
        Move      R1,DATAOUT        Send character to display.
        Move      R1,(R0)+          Store charater and advance pointer.
        Compare   #$0D,R1           Check if Carriage Return.
        Branch≠0  WAITK             If not, get another character.
        Move      #$0A,DATAOUT      Otherwise, send Line Feed.
        Call      PROCESS           Call a subroutine to process the
                                      the input line.
```

Figure 4.4 A program that reads one line from the keyboard, stores it in memory buffer, and echoes it back to the display.

EXPLANATION:

- This program, reads a line of characters from the keyboard & stores it in a memory buffer starting at locations LINE.
- Then it calls the subroutine "PROCESS" to process the input line.
- As each character is read, it is echoed back to the display.
- Register Ro is used as a pointer to memory buffer area. The contents of R0 are updated using Auto – increment mode so that successive characters are stored in successive memory location.
- Each character is checked to see if there is carriage return (CR), char, which has the ASCII code 0D (hex).
- If it is, a line feed character (ASCII character 0A) is sent to move the cursor one line down on the display & subroutine PROCESS is called. Otherwise, the program loops back towait for another character from the keyboard.

PROGRAM CONTROLLED I/O

In the above example, the processor repeatedly checks a status flag to achieve the required synchronization between Processor & I/O device. (ie) the processor polls the device.

There are 2 mechanisms to handle I/o operations. They are,

- **Interrupt** - Synchronization is achieved by having I/O device send special signal over the bus where is ready for data transfer operation.

- **DMA** - It is a technique used for high speed I/O device. Here, the device interface transfer data directly to or from the memory without continuous involvement by the processor.

INTERRUPTS

When a program enters a wait loop, it will repeatedly check the device status. During this period, the processor will not perform any function. There are many situations where other tasks can be performed while waiting for an I/O device to become ready. To allow this to happen, we can arrange for the I/O device to alert the processor when it becomes ready.

It can do so by sending a hardware signal called an **interrupt** to the processor. At least one of the bus control lines called an **interrupt request line** is usually dedicated for this purpose.

Since the processor is no longer required to continuously check the status of external devices, it can use the waiting period to perform other useful functions. Indeed by using interrupts such waiting periods can ideally be eliminated.

The routine executed in response to an interrupt request is called **Interrupt Service Routine.**

Fig: Transfer of control through the use of interrupts

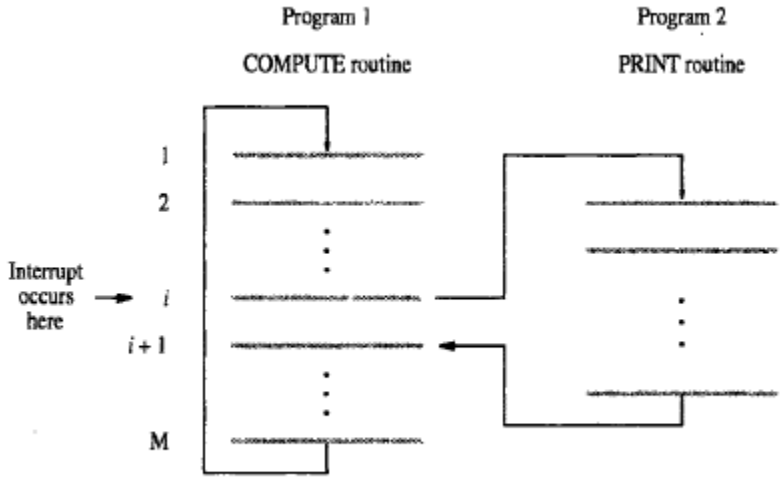


Figure 4.5 Transfer of control through the use of interrupts.

- Assume that an interrupt requests arrives during the execution of instruction i.
- The processor first completes the execution of instruction i. Then it loads the PC (Program Counter) with the address of the first instruction of the ISR.
- After the execution of ISR, the processor has to come back to instruction i + 1.
- Therefore, when an interrupt occurs, the current contents of PC which point to i+1 is put in temporary storage in a known location.
- A return from interrupt instruction at the end of ISR reloads the PC from that temporary storage location, causing the execution to resume at instruction i+1.
- When the processor is handling the interrupts, it must inform the device that itsrequest has been recognized so that it remove its interrupt requests signal.
- This may be accomplished by a special control signal called the **interruptacknowledge signal.**
- The task of saving and restoring the information can be done automatically by theprocessor.

- The processor saves only the contents of **program counter & status register** ie; it saves only the minimal amount of information to maintain the integrity of the program execution.
- Saving registers also increases the delay between the time an interrupt request is received and the start of the execution of the ISR. This delay is called the **Interrupt Latency.**
- Generally, the long interrupt latency in unacceptable.
- The concept of interrupts is used in Operating System and in Control Applications, where processing of certain routines must be accurately timed relative to external events. This application is also called as **real-time processing**.

Interrupt Hardware:

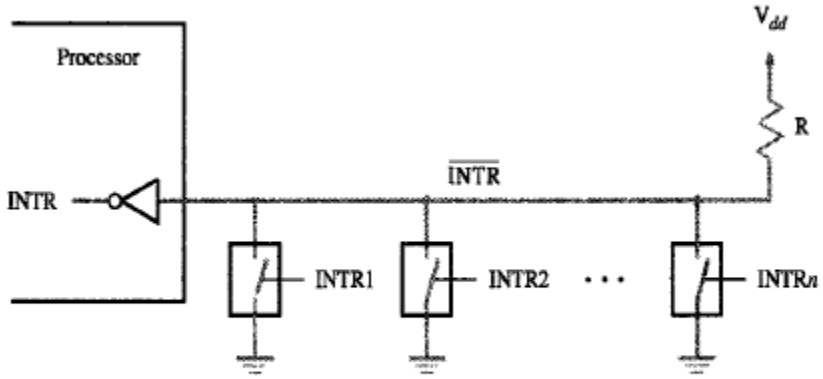


Figure 4.6 An equivalent circuit for an open-drain bus used to implement a common interrupt-request line.

- A single interrupt request line may be used to serve n devices. All devices are connected to the line via switches to ground.
- To **request** an interrupt, a device **closes** its associated switch, the voltage on INTR line drops to 0**(zero).**
- If all the interrupt request signals (INTR1 to INTRn) are **inactive**, all switches are open and the voltage on INTR line is equal to **Vdd**..
- When a device requests an interrupt by closing its switch, the voltage on the line drops to 0, causing INTR request signal received by the processor to go to 1.
- Since closing one or more switches will cause line voltage to drop to 0, the value of INTR is the logical OR of the requests from individual devices. ie;

INTR = INTR1+… +INTRn

$\overline{INTR}$ - It is used to name the INTR signal on common line it is active in the low voltage state.

In figure special gates called,

- **Open collector** (bipolar ckt) or **Open drain** (MOS circuits) is used to drive $\overline{INTR}$ line. The Output of the Open collector (or) Open drain control is equal to a switch to the ground that is open when gates input is in „0" state and closed when the gates input is in „1" state.
- Resistor „R" is called a **pull-up resistor** because it pulls the line voltage upto the high voltage state when the switches are open.

Enabling and Disabling Interrupts:

- The arrival of an interrupt request from an external device causes the processor to suspend the execution of one program & start the execution of another because the interrupt may alter the sequence of events to be executed.
- INTR is active during the execution of **Interrupt Service Routine**.
- There are 3 mechanisms to solve the problem of infinite loop which occurs due to successive interruptions of active INTR signals.
- The following are the typical scenario.

  - The device raises an interrupt request.
  - The processor interrupts the program currently being executed.
  - Interrupts are disabled by changing the control bits is PS (Processor Status register)
  - The device is informed that its request has been recognized & in response, it deactivates the INTR signal.
  - The actions are enabled & execution of the interrupted program is resumed.

Edge-triggered:

The processor has a special interrupt request line for which the interrupt handling circuit responds only to the leading edge of the signal. Such a line said to be edge-triggered.

Handling Multiple Devices:

- When several devices requests interrupt at the same time, it raises some questions. They are.

  - How can the processor recognize the device requesting an interrupt?
  - Given that the different devices are likely to require different ISR, how can the processor obtain the starting address of the appropriate routines in each case?
  - Should a device be allowed to interrupt the processor while another interrupt is being serviced?
  - How should two or more simultaneous interrupt requests be handled?

Polling Scheme:

- If two devices have activated the interrupt request line, the ISR for the selected device (first device) will be completed & then the second request can be serviced.
- The simplest way to identify the interrupting device is to have the ISR polls all the encountered with the IRQ bit set is the device to be serviced
- IRQ (Interrupt Request) -> when a device raises an interrupt requests, the status
- register IRQ is set to 1.

Advantage: It is easy to implement.
Disadvantages: The time spent for interrogating the IRQ bits of all the devices that may not be requesting any service.

Vectored Interrupt:

- Here the device requesting an interrupt may identify itself to the processor by sending a special code over the bus & then the processor start executing the ISR.
- The code supplied by the processor indicates the starting address of the ISR for the device.
- The code length ranges from 4 to 8 bits.
- The location pointed to by the interrupting device is used to store the staring address to ISR.
- The processor reads this address, called the interrupt vector & loads into PC.
- The interrupt vector also includes a new value for the Processor Status Register.
- When the processor is ready to receive the interrupt vector code, it activate the interrupt acknowledge (INTA) line.

Interrupt Nesting:
Multiple Priority Scheme:

- In multiple level priority scheme, we assign a priority level to the processor that can be changed under program control.
- The priority level of the processor is the priority of the program that is currently being executed.
- The processor accepts interrupts only from devices that have priorities higher than its own.
- At the time the execution of an ISR for some device is started, the priority of the processor is raised to that of the device.
- The action disables interrupts from devices at the same level of priority or lower.

Privileged Instruction:

- The processor priority is usually encoded in a few bits of the Processor Status word. It can also be changed by program instruction & then it is write into PS. These instructions are called **privileged instruction.** This can be executed only when the processor is in supervisor mode.
- The processor is in supervisor mode only when executing OS routines.
- It switches to the user mode before beginning to execute application program.

Privileged Exception:

- User program cannot accidently or intentionally change the priority of the processor & disrupts the system operation.

- An attempt to execute a privileged instruction while in user mode, leads to aspecial type of interrupt called the privileged exception.

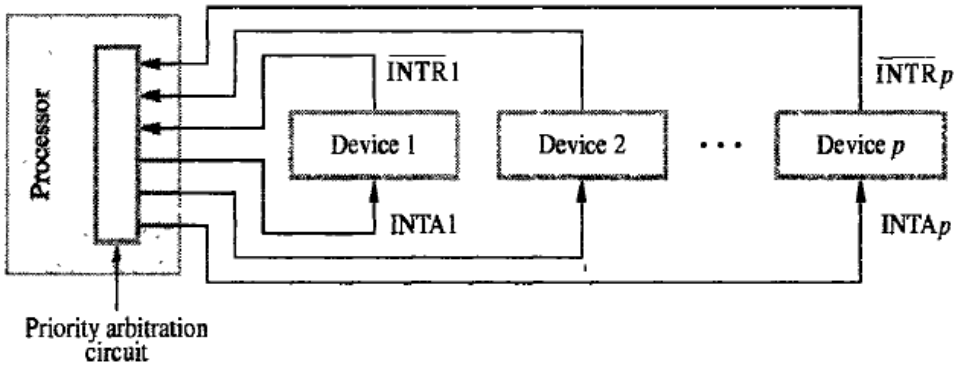
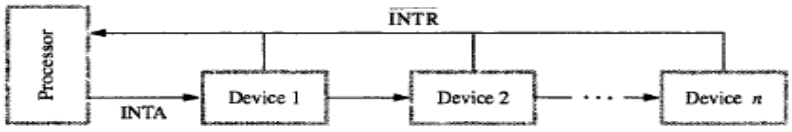


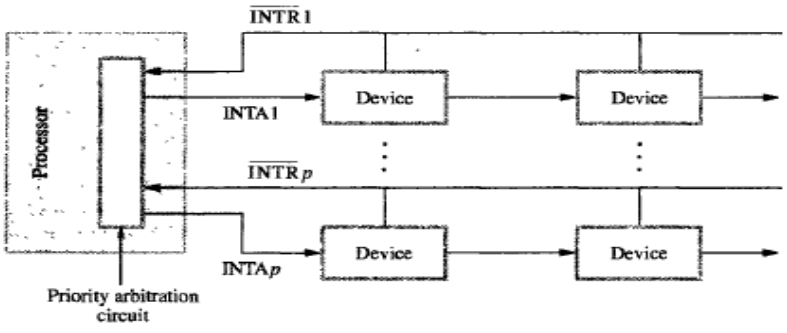
Figure 4.7 Implementation of interrupt priority using individual interrupt-request and acknowledge lines.

- Each of the interrupt request line is assigned a different priority level.
- Interrupt request received over these lines are sent to a priority arbitration circuit in the processor.
- A request is accepted only if it has a higher priority level than that currently assigned to the processor,

Simultaneous Requests:
Daisy Chain:



(a) Daisy chain



(b) Arrangement of priority groups

Figure 4.8 Interrupt priority schemes.

- The interrupt request line INTR is common to all devices. The interrupt acknowledge line INTA is connected in a daisy chain fashion such that INTA signal propagates serially through the devices.
- When several devices raise an interrupt request, the INTR is activated & the processor responds by setting INTA line to 1. this signal is received by device.

- Device1 passes the signal on to device2 only if it does not require any service.
- If devices1 has a pending request for interrupt blocks that INTA signal & proceeds to put its identification code on the data lines.
- Therefore, the device that is electrically closest to the processor has the highest priority.

Merits:

It requires fewer wires than the individual connections.

Arrangement of Priority Groups:

- Here the devices are organized in groups & each group is connected at a different priority level.
- Within a group, devices are connected in a daisy chain.

Controlling Device Requests:

KEN	□	Keyboard Interrupt Enable
DEN	□	Display Interrupt Enable
KIRQ / DIRQ	□	Keyboard / Display unit requesting an interrupt.

- There are two mechanism for controlling interrupt requests.
- At the devices end, an interrupt enable bit in a control register determines whether the device is allowed to generate an interrupt requests.
- At the processor end, either an interrupt enable bit in the PS (Processor Status) or a priority structure determines whether a given interrupt requests will be accepted.

Initiating the Interrupt Process:

- □ Load the starting address of ISR in location INTVEC (vectored interrupt).
- □ Load the address LINE in a memory location PNTR. The ISR will use this location as a pointer to store the input characters in the memory.
- □ Enable the keyboard interrupts by setting bit 2 in register CONTROL to 1.
- □ Enable interrupts in the processor by setting to 1, the IE bit in the processor status register PS.

Exception of ISR:

- □ Read the input characters from the keyboard input data register. This will cause the interface circuits to remove its interrupt requests.
- □ Store the characters in a memory location pointed to by PNTR & increment PNTR.
- □ When the end of line is reached, disable keyboard interrupt & inform program main.
- □ Return from interrupt.

Exceptions:

- An interrupt is an event that causes the execution of one program to be suspended and the execution of another program to begin.
- The Exception is used to refer to any event that causes an interruption.

Kinds of exception:

- ❖ Recovery from errors
- ❖ Debugging
- ❖ Privileged Exception

Recovery From Errors:

- Computers have error-checking code in Main Memory, which allows detection of errors in the stored data.
- If an error occurs, the control hardware detects it informs the processor by raising an interrupt.
- The processor also interrupts the program, if it detects an error or an unusual condition while executing the instance (ie) it suspends the program being executed and starts an execution service routine.
- This routine takes appropriate action to recover from the error.

Debugging:

- System software has a program called debugger, which helps to find errors in a program.
- The debugger uses exceptions to provide two important facilities
- They are
  - ❖ Trace
  - ❖ Breakpoint

Trace Mode:

- When processor is in trace mode, an exception occurs after execution of every instance using the debugging program as the exception service routine.
- The debugging program examine the contents of registers, memory location etc.
- On return from the debugging program the next instance in the program being debugged is executed
- The trace exception is disabled during the execution of the debugging program.

Break point:

- Here the program being debugged is interrupted only at specific points selected by the user.

- An instance called the Trap (or) software interrupt is usually provided for this purpose.
- While debugging the user may interrupt the program execution after instance „I"
- When the program is executed and reaches that point it examine the memory and register contents.

Privileged Exception:

- To protect the OS of a computer from being corrupted by user program certain instance can be executed only when the processor is in supervisor mode. These are called privileged exceptions.
- When the processor is in user mode, it will not execute instance (ie) when theprocessor is in supervisor mode , it will execute instance.

DIRECT MEMORY ACCESS

- It is a technique used for high speed I/O device.
- Here, the device interface transfer data directly to or from the memory without continuous involvement by the processor
- A special control unit may be provided to allow the transfer of large block of data at high speed directly between the external device and main memory, without continuous intervention by the processor. This approach is called **DMA.**
- DMA transfers are performed by a control circuit called the **DMA Controller**.

  To initiate the transfer of a block of words , the processor sends,

  - Starting address
  - Number of words in the block
  - Direction of transfer.
- When a block of data is transferred , the DMA controller increment the memory address for successive words and keep track of number of words and it also informs the processor by raising an interrupt signal.
- While DMA control is taking place, the program requested the transfer cannot continue and the processor can be used to execute another program.
- After DMA transfer is completed, the processor returns to the program that requested the transfer.

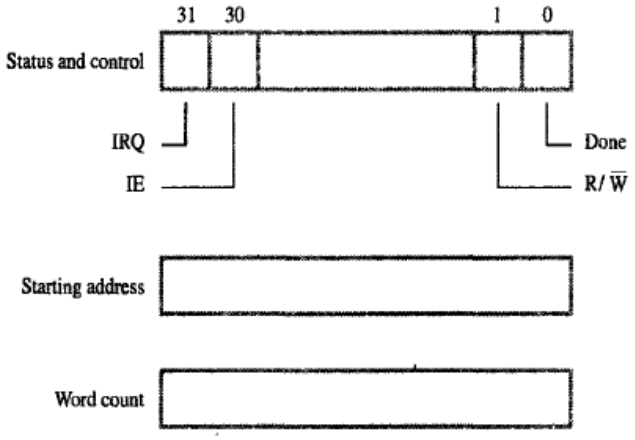


Figure 4.18 Registers in a DMA interface.

R/W - Determines the direction of transfer.
When

R/W =1, DMA controller read data from memory to I/O device.
R/W =0, DMA controller perform write operation.
Done Flag=1, the controller has completed transferring a block of data and is ready to receive another command.
IE=1, it causes the controller to raise an interrupt (interrupt Enabled) after it has completed transferring the block of data.
IRQ=1, it indicates that the controller has requested an interrupt.

Fig: Use of DMA controllers in a computer system

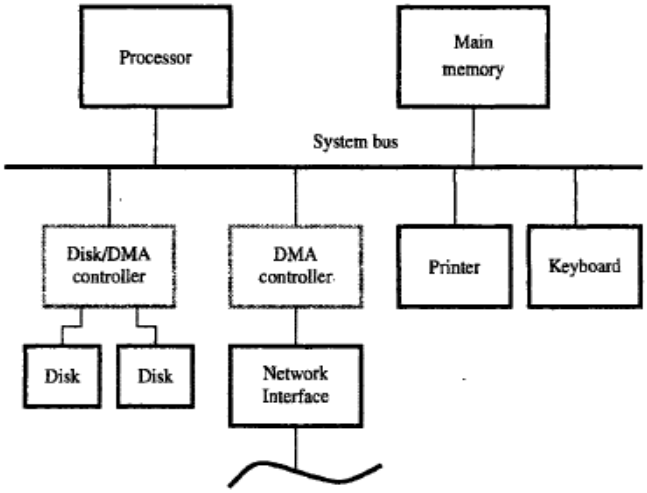


Figure 4.19 Use of DMA controllers in a computer system.

- A DMA controller connects a high speed network to the computer bus. The disk controller two disks, also has DMA capability and it provides two DMA channels.
- To start a DMA transfer of a block of data from main memory to one of the disks, the program write s the address and the word count information into the registers of the corresponding channel of the disk controller.
- When DMA transfer is completed, it will be recorded in status and controlregisters of the DMA channel (ie) **Done bit=IRQ=IE=1**.

Cycle Stealing:

- Requests by DMA devices for using the bus are having higher priority than processor requests.
- Top priority is given to high speed peripherals such as ,
  - Disk
  - High speed Network Interface and Graphics display device.

- Since the processor originates most memory access cycles, the DMA controller can be said to steal the memory cycles from the processor.
- This interviewing technique is called **Cycle stealing.**

Burst Mode:

The DMA controller may be given exclusive access to the main memory to transfer a block of data without interruption. This is known as **Burst/Block Mode**

Bus Master:

The device that is allowed to initiate data transfers on the bus at any given time is called the bus master.

Bus Arbitration:

It is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it.

Types:

There are 2 approaches to bus arbitration. They are,

- Centralized arbitration (A single bus arbiter performs arbitration)
- Distributed arbitration (all devices participate in the selection of next bus master).

Centralized Arbitration:

- Here the processor is the bus master and it may grants bus mastership to one of its DMA controller.
- A DMA controller indicates that it needs to become the bus master by activating the Bus Request line (BR) which is an open drain line.
- The signal on BR is the logical OR of the bus request from all devices connected to it.
- When BR is activated the processor activates the Bus Grant Signal (BGI) and indicated the DMA controller that they may use the bus when it becomes free.
- This signal is connected to all devices using a daisy chain arrangement.
- If DMA requests the bus, it blocks the propagation of Grant Signal to other devices and it indicates to all devices that it is using the bus by activating open collector line, Bus Busy (BBSY).

Fig: A simple arrangement for bus arbitration using a daisy chain

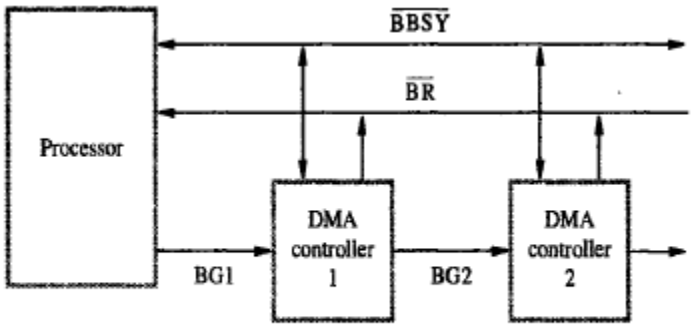


Figure 4.20 A simple arrangement for bus arbitration using a daisy chain.

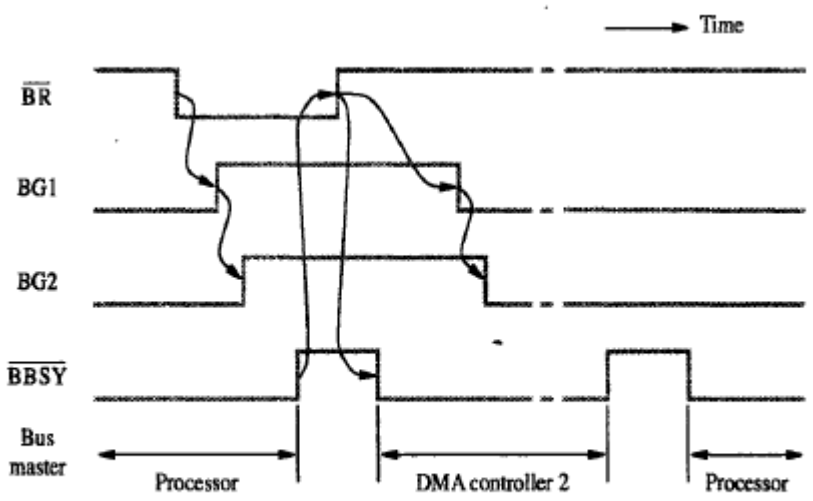


Figure 4.21 Sequence of signals during transfer of bus mastership for the devices in Figure 4.20.

- The timing diagram shows the sequence of events for the devices connected to the processor is shown.
- DMA controller 2 requests and acquires bus mastership and later releases the bus.
- During its tenture as bus master, it may perform one or more data transfer.
- After it releases the bus, the processor resources bus mastership

Distributed Arbitration:

It means that all devices waiting to use the bus have equal responsibility in carrying out the arbitration process without using a central arbiter.

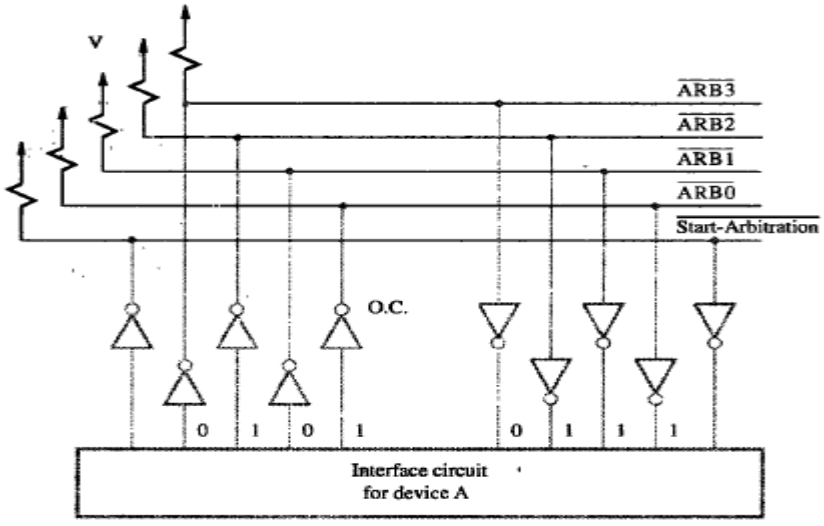


Figure 4.22 A distributed arbitration scheme.

- Each device on the bus is assigned a 4 bit id.
- When one or more devices request the bus, they assert the Start-Arbitration signal & place their 4 bit ID number on four open collector lines, ARB0 to ARB3.

- A winner is selected as a result of the interaction among the signals transmitted over these lines.
- The net outcome is that the code on the four lines represents the request that has the highest ID number.
- The drivers are of open collector type. Hence, if the input to one driver is equal to 1, the input to another driver connected to the same bus line is equal to 0, then bus will be in low-voltage state.

Example

- Assume two devices A & B have their ID 5 (0101), 6(0110). They are requesting the use of bus.
- Device A transmits the pattern 0101 and B transmits 0110. The code seen by both devices is 0111.
- Each devices compares the pattern on the arbitration line to its own ID starting from MSB.
- If it detects a difference at any bit position, it disables the drivers at that bit position and for all lower order bits.
- It does this by placing 0 at the input of these drivers.

- In our example. A detects a difference in line ARB1, hence it disables the drivers on lines ARB1 & ARB0.
- This causes the pattern on the arbitration line to change to 0110 which means that B has won the contention.
- Note that since the code on the priority line is 0111 for a shorter period, device B may be temporarily disable its driver on line ARB0. However, it will enable this driver once it sees a 0 on line ARB1 resulting from the action by device A.

Advantages:

Highly reliable – because operation of the bus is not dependent on any single device.

MEMORY SYSTEM - INTRODUCTION

Programs and data they operate on are resided in the memory of the computer. The execution speed of the program is dependent on how fast the transfer of data and instructions in-between memory and processor. There are three major types of memory available: Cache, Primary and Secondary Memories.

A good memory would be fast, large and inexpensive. Unfortunately, it is impossible to meet all three of these requirements simultaneously. Increased speed and size are achieved at increased cost.

BASIC CONCEPTS:

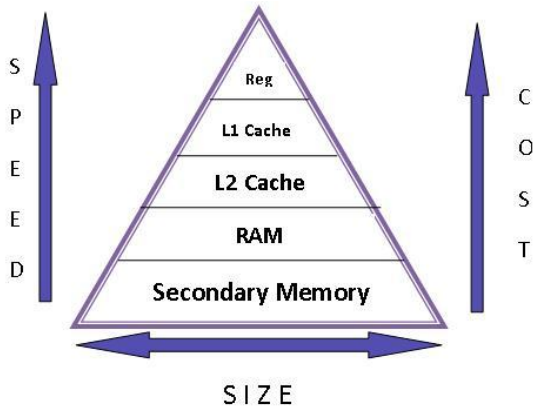
A memory unit is considered as a collection of ***cells***, in which each cell is capable of ***storing a bit*** of information. It stores information in group of bits called byte or word.The maximum size of the memory that can be used in any computer is determined by the addressing scheme.

Address	Memory Locations
16 Bit	2^{16} = 64 K
32 Bit	2^{32} = 4G (Giga)
40 Bit	2^{40} = IT (Tera)

Word length is the number of bits that can be transferred to or from the memory, it can be determined from the width of data bus, if the data bus has a width of n bits, it means word length of that computer system is n bits.

Memory access time: time elapses between the initiation of an operation and the completion of that operation.

Memory cycle time: minimum time delay required between the initiations of two successive memory locations.



Compared to processor, the main memory unit is very slow. So in order to transfer something between memory and processor takes a long time. The processor has to wait a lot. To avoid this speed gap between memory and processor a new memory called cache memory is placed in between main memory and processor.

In the memory hierarchy, speed will decrease and size will increase from top to bottom level.An important design issue is to provide a computer system with as large and fast a memory as possible, within a given cost target.

Random Access Memory (RAM)is a memory system in which any location can be accessed for a Read or Write operation in some fixed amount of time that is independent of the location's address.

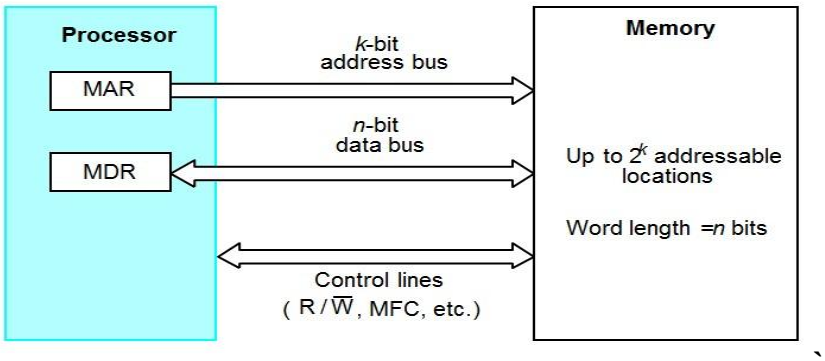
Several techniques to increase the effective size and speed of the memory: Cache memory (to increase the effective speed) &Virtual memory (to increase the effective size

Connection of a memory to a processor

The processor reads data from the memory by loading the address of the location into the MAR

register and setting the R/W line to 1.Upon receiving the MFC signal, the processor loads the data on the data lines into the MDR register.

The processor writes data into the memory location by loading the data into MDR. It indicates that a write operation is involved by setting the R/W line to 0. If MAR is k bits long and MDR is n bits long, then the memory unit may contain up to 2k addressable locations. Memory access can be synchronized by using a clock.



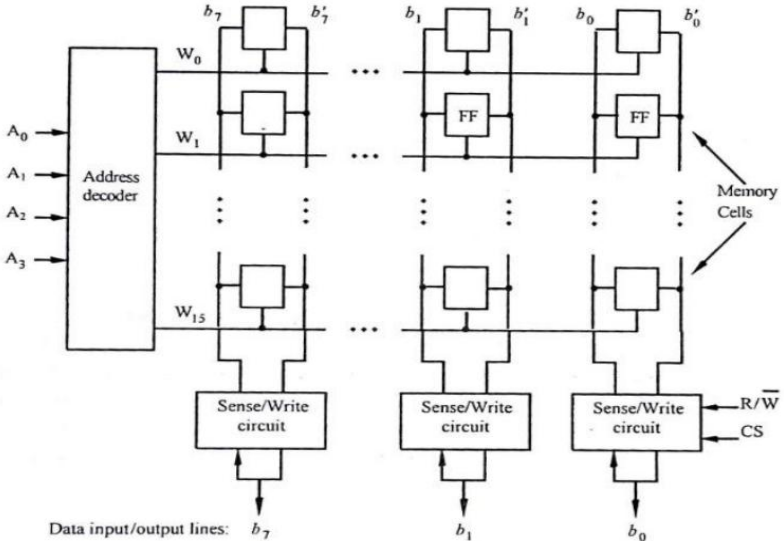
SEMICONDUCTOR RAM MEMORIES

Semiconductor memories are available in a wide range of speeds. Their cycle times range from 100ns to less than 10 ns.

When first introduced in late 1990s, they were much more expensive than the magnetic-core memories they replaced. Because of rapid advances in VLSI (Very Large Scale Integration) technology, the cost of semiconductor memories has dropped dramatically. As a result, they are now used almost exclusively in implementing memories.

Internal Organization of Memory Chips

Memory cells are usually organized in the form of array, in which each cell is capable of storing one bit of information.



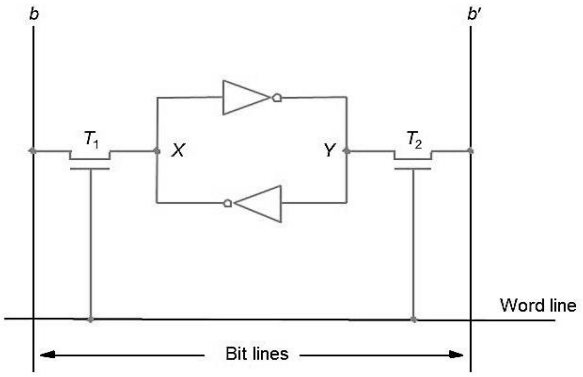
Each row of cells consists of memory word, and all cells of a row are connected to a common line referred to as the word line, which is driven by the address decoder on a chip.The cells in each column are connected to a Sense/Write circuit by two bit lines. Sense/write circuits are connected to the data input/output lines of the memory chip.

During read operation, these circuits sense or read the information stored in the cells selected by a word line and transmit this information to the output data lines. During a write operation, the sense/write circuits receive input information and store it in the cells of the selected word.

Two control lines, R/W and CS, are provided in addition to address and data lines. The Read/Write input specifics the required operation, and the CS input select a given chip in multichip memory system

Static Memories (SRAM)

Static memories are the memories that consist of circuits capable of retaining their state as long as power is applied. Two transistor inverters are cross connected to implement a basic flip-flop. The cell is connected to one word line and two bits lines by transistors T1 and T2. When word line is at ground level, the transistors are turned off and the latch retains its state.



Most of the static RAMs are built using MOS (Metal Oxide Semiconductor) technology, but some are built using bipolar technology. If the cell is in state 1/0, the signal on b is high/low and signal on bit line b' is low/high.

Read operation: In order to read state of SRAM cell, the word line is activated to close switches T1 and T2. Sense/Write circuits at the bottom monitor the state of b and b'.

Write operation: During the write operation, the state of the cell is set by placing the appropriate value on bit line b and its complement on b' and then activating the word line. This forces the cell into the corresponding state. The major advantage of SRAM is very quicly accessed by the processor. The major disadvantage is that SRAM are expensive memory and SRAM are Volatile memory. If the power is interrupted, the cell's content will be lost. Continuous power is needed for the cell to retain its state.

Dynamic Memories (DRAM)

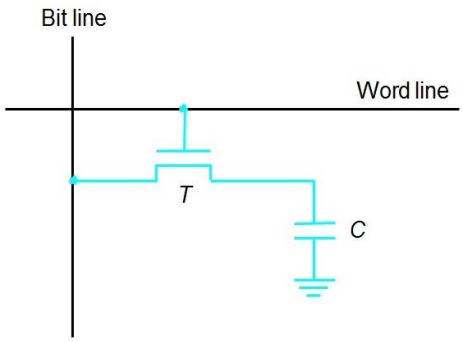
Static RAMs are fast, but the cost is too high because their cells require several transistors.Less expensive RAMs can be implemented if simpler cells are used. Such cells don't retain their state indefinitely; hence they are called dynamic RAMs

Dynamic RAMs (DRAMs) are cheap and area efficient, but they cannot retain their state indefinitely – need to be periodically refreshed. Dynamic memory cell consists of a **capacitor C**, and a

transistor T.

Information is stored is a dynamic memory cell in the form of charge on a capacitor and this charge can be maintained for only tens of milliseconds.

Since the cell is required to store information for a much longer time, its contents must be **periodically refreshed** by restoring the capacitor charge to its full value.

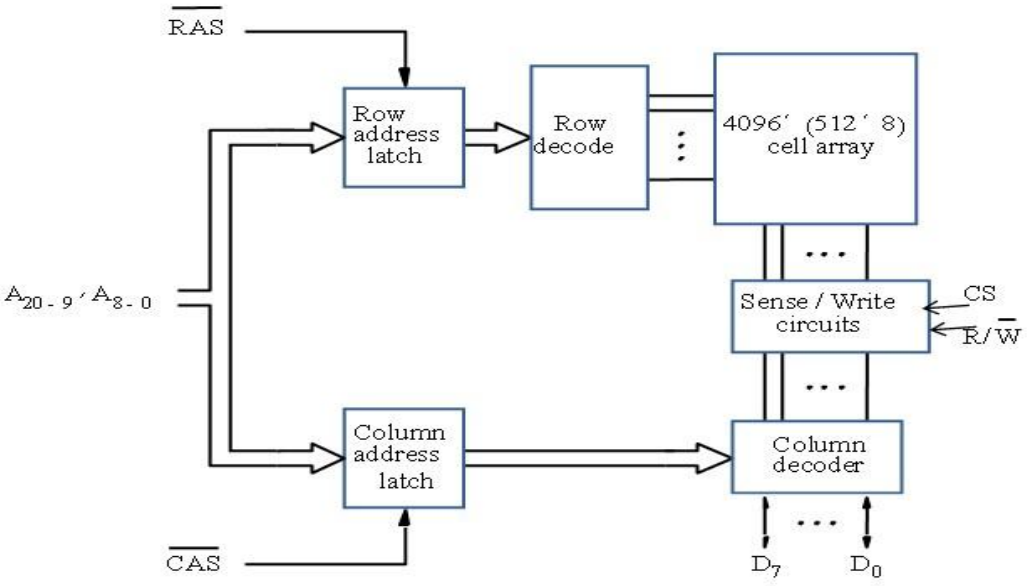


Read Operation: Transistor turned on, Sensor check voltage of capacitor. If voltage is less than Threshold value, Capacitor discharged and it represents logical '0' else if voltage is above Threshold value, Capacitor charged to full voltage and it represents Logical '1'

Write Operation - Transistor is turned on and a voltage is applied/removed to the bit line.

Asynchronous Dynamic RAM:

In Asynchronous dynamic RAM, the timing of the memory device is controlled asynchronously.A specialized memory controller circuit provides the necessary control signals, RAS and CAS, which govern the timing. The processor must take into account the delay in the response of the memory.



In the diagram above, we can see that there are two extra elements with two extra lines attached to them: the Row Address Latch is controlled by the RAS (or Row Address Strobe) pin, and the Column Address Latch is controlled by the CAS (or Column Address Strobe) pin.

Read Operation:

1. The row address is placed on the address pins via the address bus.
2. The RAS pin is activated, which places the row address onto the Row Address Latch.
3. The Row Address Decoder selects the proper row to be sent to the sense amps.

4. The Write Enable (not pictured) is deactivated, so the DRAM knows that it's not being written to.
5. The column address is placed on the address pins via the address bus.
6. The CAS pin is activated, which places the column address on the Column Address Latch.
7. The CAS pin also serves as the Output Enable, so once the CAS signal has stabilized the sense amps, it place the data from the selected row and column on the Data Out pin so that it can travel the data bus back out into the system.
8. RAS and CAS are both deactivated so that the cycle can begin again.

Write Operation:

1. In the write operation, the information on the data lines is transferred to the selected circuits. For this write enable is activated

Fast Page Mode

Suppose if we want to access the consecutive bytes in the selected row. This can be done without having to reselect the row. Add a latch at the output of the sense circuits in each row.
All the latches are loaded when the row is selected.Different column addresses can be applied to select and place different bytes on the data lines. Consecutive sequence of column addresses can be applied under the control signal CAS, without reselecting the row.

This methodology allows a block of data to be transferred at a much faster rate than random accesses.A small collection/group of bytes is usually referred to as a block.This transfer capability is referred to as the fast page mode feature. This mode of operation is useful when there is requirement for fast transfer of data *(Eg: Graphical Terminals)*

Synchronous DRAM's

Operation is directly synchronized with processor clock signal. The outputs of the sense circuits are connected to a latch. During a Read operation, the contents of the cells in a row are loaded onto the latches. During a refresh operation, the contents of the cells are refreshed without changing the contents of the latches.

Data held in the latches correspond to the selected columns are transferred to the output.

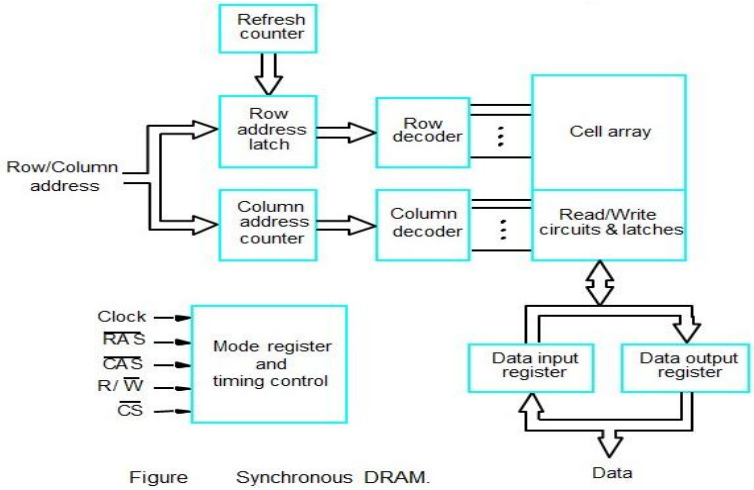


Figure Synchronous DRAM.

For a burst mode of operation, successive columns are selected using column address counter and clock. CAS signal need not be generated externally. A new data is placed duringraising edge of the clock

Memory latency is the time it takes to transfer a word of data to or from memory.

Memory bandwidth is the number of bits or bytes that can be transferred in one second.

Double Data Rate SDRAM

DDR-SDRAM is a faster version of SDRAM. The standard SDRAM perform all actions on the rising edge of the clock signal.DDR SDRAM is also access the cell array in the same way but transfers data on both edges of the clock. So bandwidth is essentially doubled for long burst transfers.

To make it possible to access the data at a high enough rate, the cell array is organized in two banks. Each bank can access separately.Consecutive words of a given block arc stored in different banks. Such interleaving of words allows simultaneous access to two words that are transferred on successive edges of the clock.

Static RAM	Dynamic RAM
More expensive	Less expensive
No refresh	Deleted & refreshed
High power	Less power
Less storage capacity	Higher storage capacity
MOS transistors	Transistor & capacitor
Faster	Slow
More reliable	Less reliable

Structure of Larger Memories

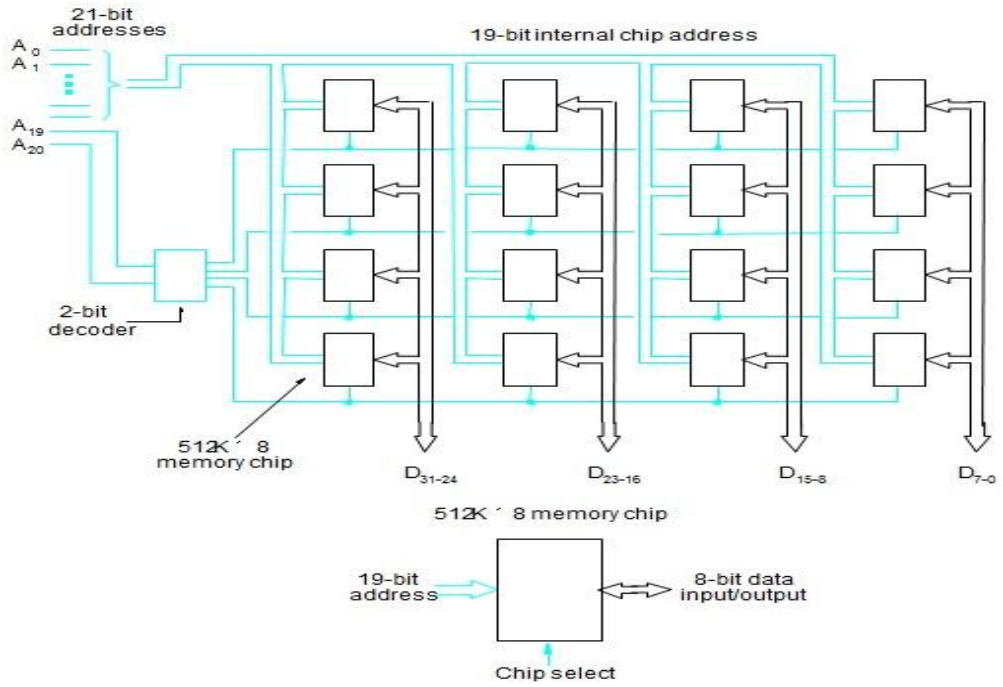
Let discuss about how memory chips may be connected to form a much larger memory.

Static Memory Systems

Implement a memory unit of 2M words of 32 bits each. Use 512x8 static memory chips. Each column consists of 4 chips. Each chip implements one byte position. A chip is selected by setting its chip select control line to 1. Selected chip places its data on the data output line, outputs of other chips are in high impedance state. 21 bits to address a 32-bit word and high order 2 bits are needed to select the row, by activating the four Chip Select signals. 19 bits are used to access specific byte locations inside the selected chip.

Dynamic Memory Systems

Large dynamic memory systems can be implemented using DRAM chips in a similar way to static memory systems. Placing large memory systems directly on the motherboard will occupy a large amount of space. Also, this arrangement is inflexible since the memory system cannot be expanded easily.



Packaging considerations have led to the development of larger memory units known as SIMMs (Single In-line Memory Modules) and DIMMs (Dual In-line Memory Modules). Memory modules are an assembly of memory chips on a small board that plugs vertically onto a single socket on the motherboard in order to occupy less space on the motherboard. And also allows for easy expansion by replacement.

MEMORY SYSTEM CONSIDERATIONS

The choice of a RAM chip for a given application depends on several factors: Cost, speed, power, size etc. SRAMs are faster, more expensive and smaller. In the case of DRAMs are slower, cheaper and larger.

If speed is the primary requirement static RAMS are the most appropriate one. Static RAMs are mostly used in cache memories. If cost is the prioritized factor then we are going for dynamic RAMs. Dynamic RAMs are used for implementing computer main memories.

Refresh overhead:

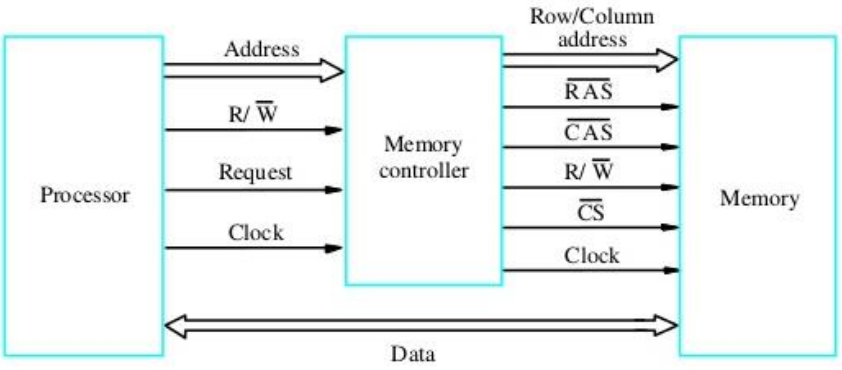
All dynamic memories have to be refreshed. In DRAM, the period for refreshing all rows is 16ms whereas 64ms in SDRAM.

Eg: Suppose a SDRAM whose cells are in 8K (8192) rows; 4 clock cycles are needed to access (read) each rows; then it takes 8192×4=32,768 cycles to refresh all rows; if the clock rate is 133 MHz, then it takes $32,768/(133\times10^{-6})=246\times10^{-6}$ seconds; suppose the typical refreshing period is 64ms, then the refresh overhead is 0.246/64=0.0038<0.4% of the total time available for accessing the memory.

Memory Controller

Dynamic memory chips use multiplexed address inputs so that we can reduce the number of pins. The address is divided into two parts and they are the High order address bits and Low order address bits. The high order selects a row in the cell array and the low order address bits selects a column in the

cell array. The address selection is done under the control of RAS and CAS signal respectively for high order and low order address bits.

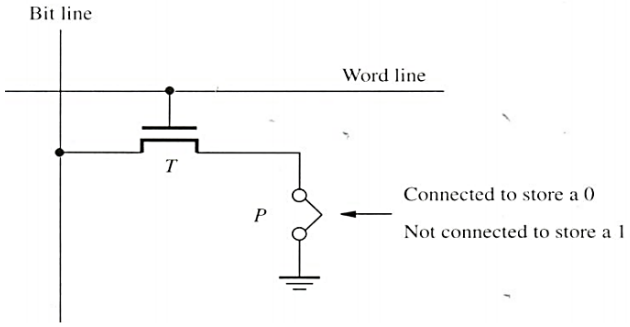


READ ONLY MEMORY

SRAM and SDRAM chips are volatile: Lose the contents when the power is turned off.Many applications need memory devices to retain contents after the power is turned off.

For example, computer is turned on; the operating system must be loaded from the diskinto the memory. For this we need to store instructions which would load the OS from the disk that they will not be lost after the power is turned off. So we need to store the instructions into a non-volatile memory.

Non-volatile memory is read in the same manner as volatile memory. The normal operation involves only reading of data, this type of memory is called Read-Only memory (ROM). The data are written into a ROM when it is manufactured and is permanent memory**.**



At Logic value '0': Transistor(T) is connected to the ground point(P). Transistor switch is closed & voltage on bit line nearly drops to zero. At Logic value '1': Transistor switch is open. The bit line remains at high voltage.

To read the state of the cell, the word line is activated. A Sense circuit at the end of the bit line generates the proper output value.

Types of ROM

Different types of non-volatile memory are

- PROM
- EPROM
- EEPROM
- Flash Memory

Programmable Read-Only Memory (PROM):

PROM allows the data to be loaded by the user. Programmability is achieved by inserting a 'fuse' at point P in a ROM cell. Before it is programmed, the memory contains all 0's. The user can insert 1's at the required location by burning out the fuse at these locations using high-current pulse. This process is irreversible.

The PROMs provides flexibility and faster data access. It is less expensive because they can be programmed directly by the user.

Erasable Reprogrammable Read-Only Memory (EPROM):

EPROM allows the stored data to be erased and new data to be loaded. In an EPROM cell, a connection to ground is always made at 'P' and a special transistor is used, which has the ability to function either as a normal transistor or as a disabled transistor that is always turned 'off'.

During programming, an electrical charge is trapped in an insulated gate region. The charge is retained for more than 10 years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid). This exposure to ultra-violet light dissipates the charge. During normal use, the quartz lid is sealed with a sticker.

EPROM can be erased by exposing it to ultra-violet light for duration of up to 40 minutes. Usually, an EPROM eraser achieves this function.

Merits: It provides flexibility during the development phase of digital system. It is capable of retaining the stored information for a long time.

Demerits: The chip must be physically removed from the circuit for reprogramming and its entire contents are erased by UV light.

Electrically Erasable Programmable Read-Only Memory (EEPROM):

EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of reprogramming is flexible but slow.

Merits:It can be both programmed and erased electrically. It allows the erasing of all cell contents selectively. **Demerits:**It requires different voltage for erasing ,writing and reading the stored data.

Flash memory:

Flash memory is a non-volatile memory chip used for storage and for transferring data between a personal computer (PC) and digital devices. It has the ability to be electronically reprogrammed and erased. It is often found in USB flash drives, MP3 players, digital cameras and solid-state drives.

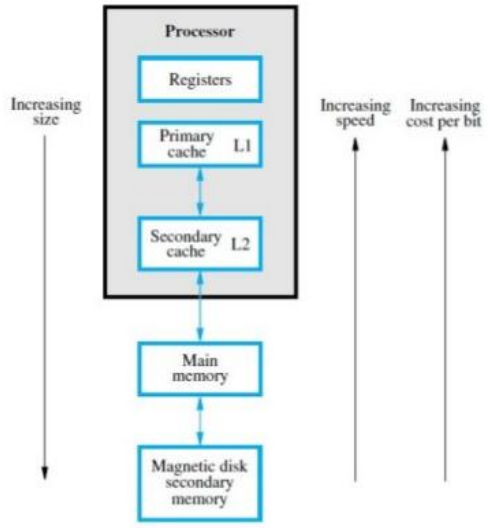
Flash memory is a type of electronically erasable programmable read only memory (EEPROM), but may also be a standalone memory storage device such as a USB drives. EEPROM is a type of data memory device using an electronic device to erase or write digital data. Flash memory is a distinct type of EEPROM, which is programmed and erased in large blocks.

Flash memory incorporates the use of floating-gate transistors to store data. Floating-gate transistors, or floating gate MOSFET (FGMOS), is similar to MOSFET, which is a transistor used for

amplifying or switching electronic signals. Floating-gate transistors are electrically isolated and use a floating node in direct current (DC). Flash memory is similar to the standard MOFSET, except the transistor has two gates instead of one.

SPEED, SIZE AND COST

A big challenge in the design of a computer system is to provide a sufficiently large memory, with a reasonable speed at an affordable cost.



Static RAM: Very fast, but expensive, because a basic SRAM cell has a complex circuit making it impossible to pack a large number of cells onto a single chip.

Dynamic RAM: Simpler basic cell circuit, hence are much less expensive, but significantly slower than SRAMs.

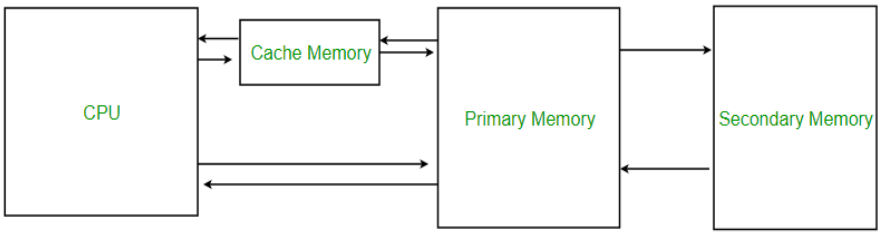
Magnetic disks: Storage provided by DRAMs is higher than SRAMs, but is still less than what is necessary. Secondary storage such as magnetic disks provides a large amount of storage, but is much slower than DRAMs.

CACHE MEMORIES

Processor is much faster than the main memory. As a result, the processor has to spend much of its time waiting while instructions and data are being fetched from the main memory. These create a major obstacle towards achieving good performance. Speed of the main memory cannot be increased beyond a certain point.

Cache Memory is a special very high-speed memory. It is used to speed up and synchronizing with high-speed CPU. Cache memory is costlier than main memory or disk memory but economical than CPU registers. Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.

Cache memory is used to reduce the average time to access data from the Main memory. The cache is a smaller and faster memory which stores copies of the data from frequently used main memory locations. There are various different independent caches in a CPU, which store instructions and data.



Cache memory is based on the property of computer programs known as "locality of reference". Prefetching the data into cache before the processor needs it. It needs to predict processor future access requirement [Locality of Reference].

Locality of Reference

Analysis of programs indicates that many instructions in localized areas of a program are executed repeatedly during some period of time, while the others are accessed relatively less frequently. These instructions may be the ones in a loop, nested loop or few procedures calling each other repeatedly. This is called **"locality of reference".**

Temporal locality of reference:

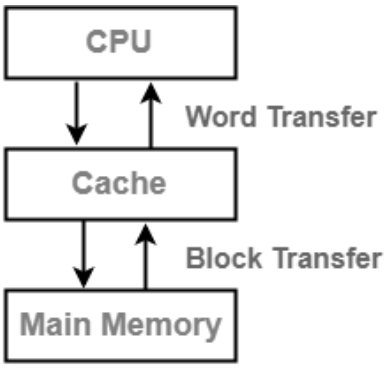
Recently executed instruction is likely to be executed again very soon.

Spatial locality of reference:

Instructions with addresses close to a recently instruction are likely to be executed soon.

Basic Cache Operations

Processor issues a Read request; a block of words is transferred from the main memory to the cache, one word at a time. Subsequent references to the data in this block of words are found in the cache.



At any given time, only some blocks in the main memory are held in the cache, which blocks in the main memory in the cache is determined by a "**mapping function**".

When the cache is full, and a block of words needs to be transferred from the main memory, some block of words in the cache must be replaced. This is determined by a "**replacement algorithm**".

Cache hit : Existence of a cache is transparent to the processor. The processor issues Read and Write requests in the same manner. If the data is in the cache, it is called a Read or Write hit.

Read hit: The data is obtained from the cache.

Write hit: Cache has a replica of the contents of the main memory. Contents of the cache and the main memory may be updated simultaneously. This is the **write-through protocol**. Update the contents of the cache, and mark it as updated by setting a bit known as the **dirty bit or modified bit.** The contents of the main memory are updated when this block is replaced. This is **write-back or copy-back protocol.**

Cache miss: If the data is not present in the cache, then a Read miss or Write miss occurs.

Read miss: Block of words containing this requested word is transferred from the memory. After the block is transferred, the desired word is forwarded to the processor. The desired word may also be forwarded to the processor as soon as it is transferred without waiting for the entire block to be transferred. This is called **load-through or early restart.**

Write-miss: Write-through protocol is used, and then the contents of the main memory are updated directly. If write-back protocol is used, the block containing the addressed word is first brought into the cache. The desired word is overwritten with new information.

MAPPING FUNCTIONS

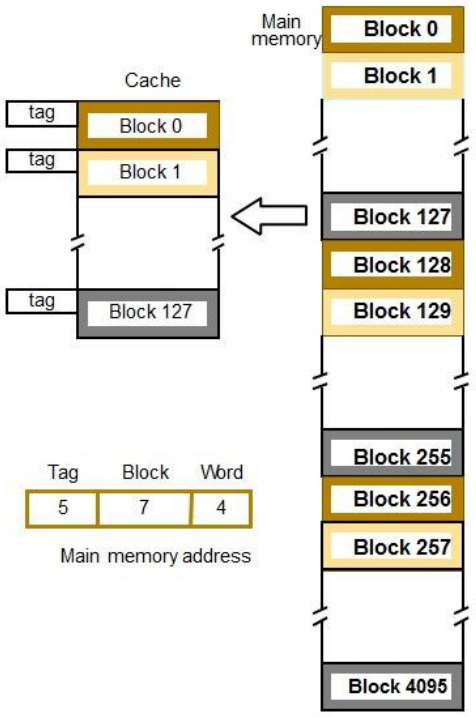
The mapping functions are used to map a particular block of main memory to a particular block of cache. This mapping function is used to transfer the block from main memory to cache memory. Mapping functions determine how memory blocks are placed in the cache.

Three mapping functions:

- Direct mapping.
- Associative mapping.
- Set-associative mapping.

Direct Mapping

A particular block of main memory can be brought to a particular block of cache memory. So, it is not flexible. The simplest way of associating main memory blocks with cache block is the direct mapping technique.



In this technique, block k of main memory maps into block **k modulo m** of the cache **(k % m)**, where m is the total number of blocks in cache. In this example, the value of m is 128.

In direct mapping technique, one particular block of main memory can be transferred to a particular block of cache which is derived by modulo function.

Example:
Block j of the main memory maps to (j modulo 128) of the cache.
Block 0, 128, 256 of main memory is maps to block 0 of cache memory.
Block1, 129, 257 of main memory maps to block 1 of cache memory & so on.

More than one memory block is mapped onto the same position in the cache. This may lead to contention for cache blocks even if the cache is not full. Resolve the contention by allowing new block to replace the old block, leading to a trivial replacement algorithm.

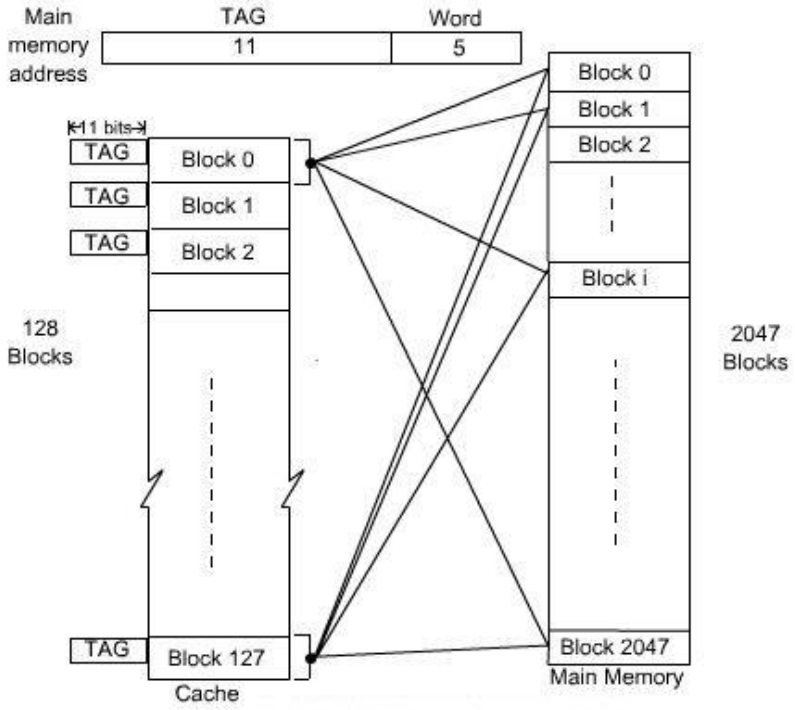
Memory address is divided into three fields: Low order 4 bits determine one of the 16 ***words*** in a block. When a new block is brought into the cache, the next 7 bits determine whichcache ***block*** this new block is placed in. High order 5 bits determine which of the possible32 blocks is currently presentin the cache. These are ***tag*** bits.

This mapping methodology is simple to implement but not very flexible.

Associative mapping

In the associative mapping technique, a main memory block can potentially reside in any cache block position. In this case, the main memory address is divided into two groups, a low-order bit identifies the location of a word within a block and a high-order bit identifies the block.

In the example here, 11 bits are required to identify a main memory block when it is resident in the cache , high-order 11 bits are used as TAG bits and low-order 5 bits are used to identify a word within a block. The TAG bits of an address received from the CPU must be compared to the TAG bits of each block of the cache to see if the desired block is present.



In the associative mapping, any block of main memory can go to any block of cache, so it has got the complete flexibility and we have to use proper replacement policy to replace a block from cache if the currently accessed block of main memory is not present in cache.

It might not be practical to use this complete flexibility of associative mapping technique due to searching overhead, because the TAG field of main memory address has to be compared with the TAG field of the entire cache block.

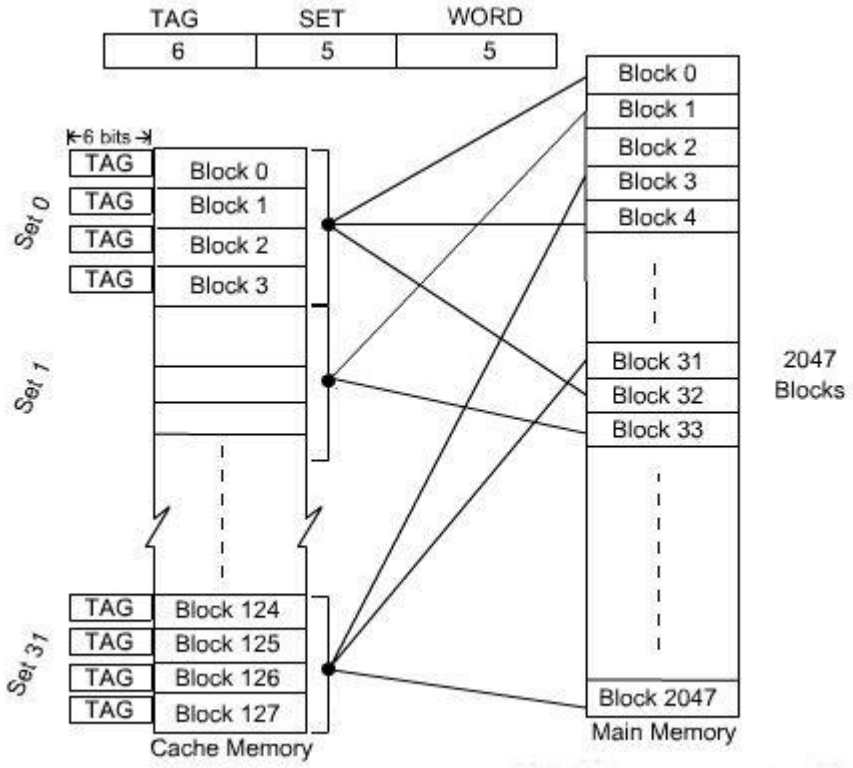
In this example, there are 128 blocks in cache and the size of TAG is 11 bits.

Set-Associative mapping

This mapping technique is intermediate to the previous two techniques. Blocks of the cache are grouped into sets, and the mapping allows a block of main memory to reside in any block of a specific set. Therefore, the flexibility of associative mapping is reduced from full freedom to a set of specific blocks.

This also reduces the searching overhead, because the search is restricted to number of sets, instead of number of blocks. Also the contention problem of the direct mapping is eased by having a few choices for block replacement.

Consider the same cache memory and main memory organization of the previous example. Organize the cache with 4 blocks in each set. The TAG field of associative mapping technique is divided into two groups, one is termed as SET bit and the second one is termed as TAG bit. Each set contains 4 blocks, total number of set is 32. The main memory address is grouped into three parts: low-order 5 bits are used to identifies a word within a block. Since there are total 32 sets present, next 5 bits are used to identify the set. High-order 6 bits are used as TAG bits.



Replacement Algorithms

When the cache is full, there is a need for replacement algorithm for replacing the cache block with a new block. For achieving the high-speed such types of the algorithm is implemented in hardware.

In the cache memory, there are three types of replacement algorithm are used that are:

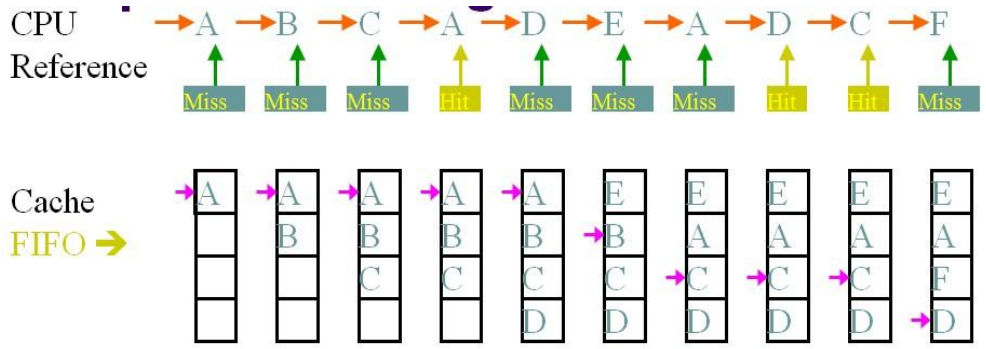
- Random replacement policy.
- First in first Out (FIFO) replacement policy
- Least recently used (LRU) replacement policy.

Random replacement policy

This is a very simple algorithm which used to choose the block to be overwritten at random. In this algorithm replace any cache line by using random selection. It is an algorithm which is simple and has been found to be very effective in practice.

First in first out (FIFO)

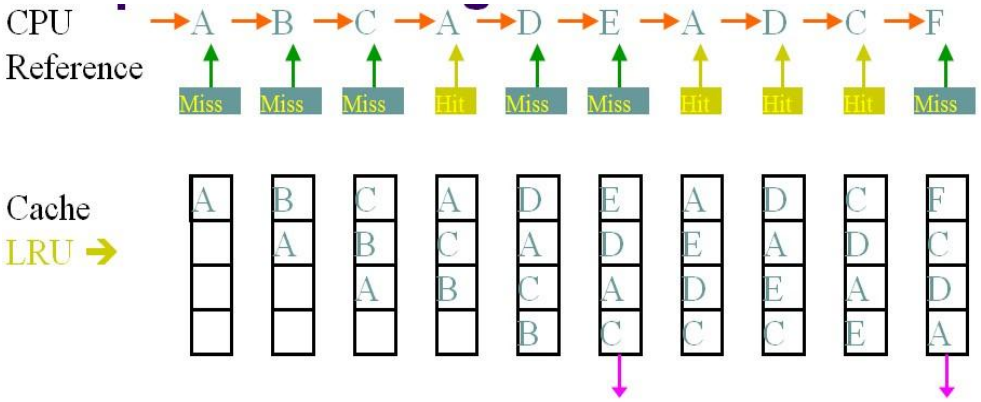
In this algorithm replace the cache block which is having the longest time stamp. While using this technique there is no need of updating when a hit occurs but when there is a miss occur then the block is put into an empty block and the counter values are incremented by one.



Hit Ratio = 3 / 10 = 0.3

Least recently used (LRU)

In the LRU, replace the cache block which is having the less reference with the longest time stamp. In this case also when a hit occurs when the counter value will be set to 0 but when the miss occurs there will be arising of two possibilities in which one possibility is that counter value is set as 0 and in another possibility, the counter value can be incremented as 1.



Hit Ratio = 4 / 10 = 0.4

Hit Rate: Number of hits / Number of all attempted access
Miss Rate: Number of miss / Number of all attempted access
Miss Penalty: The extra time needed to bring the desired information to cache

CONTENT ADDRESSABLE MEMORY (CAM)/ ASSOCIATIVE MEMORY

Many data-processing applications require the search of items in a table stored in memory. An assembler program searches the symbol address table in order to extract the symbol's binary equivalent. An account number may be searched in a file to determine the holder's name and account status.

The established way to search a table is to store all items where they can be addressed in sequence. The search procedure is a strategy for choosing a sequence of addresses, reading the content of memory at each address, and comparing the information read with the item being searched until a match occurs. The number of accesses to memory depends on the location of the item and the efficiency of the search algorithm.

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit accessed by content is called **an associative memory or Content Addressable Memory (CAM).**

This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location. When a word is written in an associative memory, no address is given. The memory is capable of finding an empty unused location to store the word. When a word is to be read from an associative memory, the content of the word, or part of the word, is specified.

The memory locaters all words which match the specified content and marks them for reading. Because of its organization, the associative memory isuniquely suited to do parallel searches by data association.An associative memory is more expensive then a random access memorybecause each cell must have storage capability as well as logic circuits for matching its content with an external argument.For this reason, associative memories are used in applications where the search time is very critical and must be very short.

HARDWARE ORGANIZATION

The block diagram of an associative memory consists of a memory array and logic from words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word.

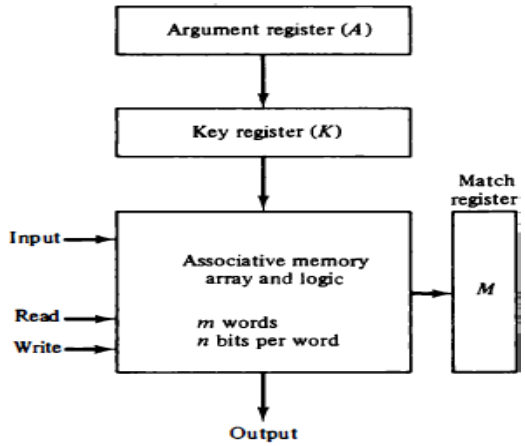


Fig: **Block Diagram of Associative Memory**

The match register M has m bits, one for each memory word. Each word in memory is compared in parallel with the content of the argument register.

The words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate the fact that their corresponding words have been matched. Reading is accomplished by a sequential access to memory for those words whose corresponding bits in the match register have been set.

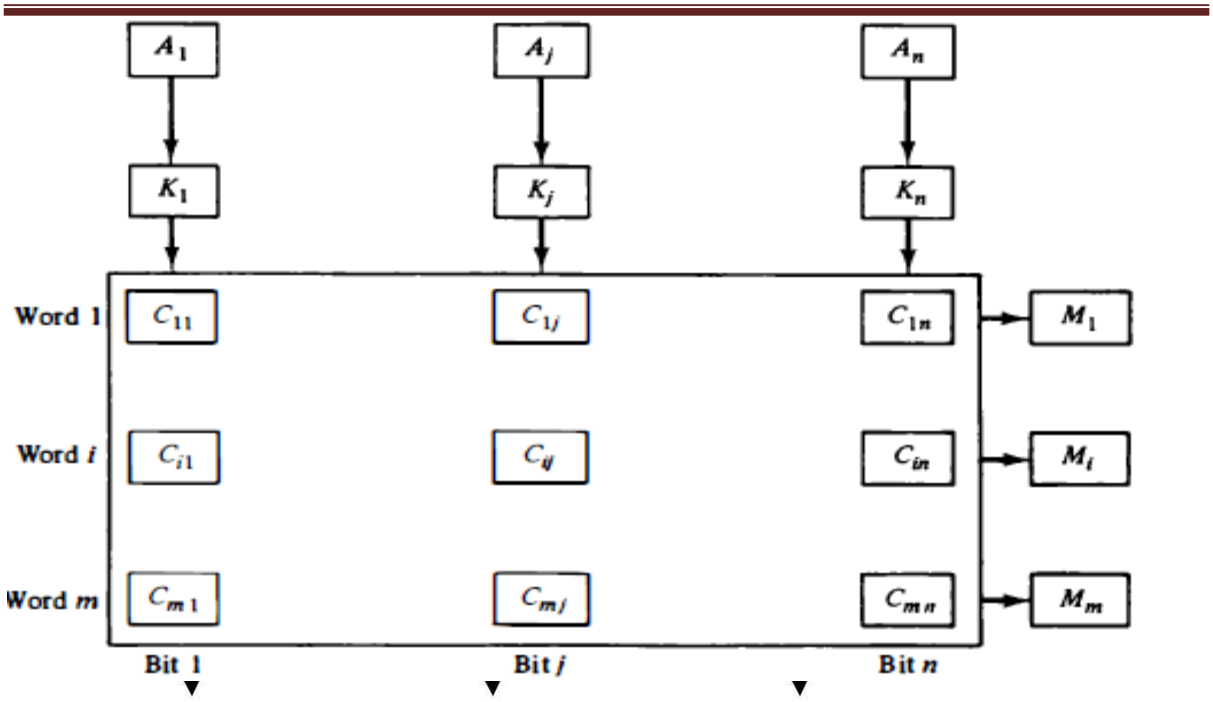
The key register provides a mask for choosing a particular field or key in the argument word. The entire argument is compared with each memory word if the key register contains all 1's. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus the key provides a mask or identifying piece of information which specifies how the reference to memory is made.

To illustrate with a numerical example, suppose that the argument register A and the key register K have the bit configuration shown below. Only the three leftmost bits of A are compared with memory words because K has 1's in these positions.

A	101 111100	
K	111 000000	
Word 1	100 111100	no match
Word 2	101 000001	match

Word 2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.

The relation between the memory array and external registers in an associative memory is shown in below figure.



The cells in the array are marked by the letter C with two subscripts. The first subscript gives the word number and the second specifies the bit position in the word. Thus cell Cij is the cell for bit j in word i. A bit A j in the argument register is compared with all the bits in column j of the array provided that K j =1. This is done for all columns j = 1, 2,…,n. If a match occurs between all the unmasked bits of the argument and the bits in word i, the corresponding bit Mi in the match register is set to 1.

If one or more unmasked bits of the argument and the word do not match, Mi is cleared to 0.
Flop storage element Fij and the circuits for reading, writing, and matching the cell. The input bit is transferred into the storage cell during a write operation. The bit stored is read out during a read operation. The match logic compares the content of the storage cell with the corresponding unmasked bit of the argument and provides an output for the decision logic that sets the bit in Mi.

READ OPERATION

The matched words are read in sequence by applying a read signal to each word line whose corresponding Mi bit is a 1. In most applications, the associative memory stores a table with no two identical items under a given key. In this case, only one word may match the unmasked argument field. By connecting output Mi directly to the read line in the same word position (instead of the M register), the content of the matched word will be presented automatically at the output lines and no special read command signal is needed. Furthermore, if we exclude words having a zero content, an all-zero output will indicate that no match occurred and that the searched item is not available inmemory.

WRITE OPERATION

If the entire memory is loaded with new information at once prior to a search operation then the writing can be done by addressing each location in sequence. This will make the device a random-access memory for writing and a content addressable memory for reading. The advantage here is that the address for input can be decoded as in a random-access memory. Thus instead of having m address lines, one for each word in memory, the number of addresslines can be reduced by the decoder to d lines, where $\mathbf{m = 2^d}$.

If unwanted words have to be deleted and new words inserted one at a time, there is a need for a special register to distinguish between active and inactive words. This register, sometimes called a tag register, would have as many bits as there are words in the memory. For every active word stored in memory, the corresponding bit in **the tag register** is set to 1. A word is deleted from memory by clearing its tag bit to 0. Words are stored in memory by scanning the tag register until the first 0 bit is encountered. This gives the first available inactive word and a position for writing a new word. After the newword is stored in memory it is made active by setting its tag bit to 1. An unwanted word when deleted from memory can be cleared to all 0's if this valueis used to specify an empty location.