COMPUTER ORGANIZATION

Note: These file has notes for Module 2 and 3. It include portions only for your test. Questions are also included in notes. Some figures are not included so refer to textbook.

Module 2 Input Output Organization

BUSES

• Bus
→ is used to inter-connect main-memory, processor & I/O devices

→ includes lines needed to support interrupts & arbitration

• Primary function: To provide a communication-path for transfer of data.
• ***Bus protocol*** is set of rules that govern the behaviour of various devices connected to the buses.
• Bus-protocol specifies parameters such as:
→ asserting control-signals

→ timing of placing information on bus

→ rate of data-transfer

• A typical bus consists of 3 sets of lines: 1) Address, 2) Data and 3) Control lines.
• Control-signals specify whether a read or a write operation is to be performed.
• R/W line specifies
→ read operation when R/W=1

→ write operation when R/W=0

• In data-transfer operation, one device plays the role of a bus-master which initiates data transfers by issuing Read or Write commands on bus (Hence it may be called an initiator).

• Device addressed by master is referred to as a slave (or target).
• Timing of data transfers over a bus is classified into 2 types:
1) Synchronous and 2) Asynchronous

SYNCHRONOUS BUS

• All devices derive timing-information from a common clock-line.
• Equally spaced pulses on this line define equal time intervals.
• Each of these intervals constitutes a bus-cycle during which one data transfer can take place.

A sequence of events during a read operation:

• At time t0, the master (processor)

→ places the device-address on address-lines &

→ Sends an appropriate command on control-lines (Figure 4.23).

• Information travels over bus at a speed determined by its physical & electrical characteristics.

• Clock pulse width (t1-t0) must be longer than the maximum propagation-delay between 2 devices connected to bus.
• Information on bus is unreliable during the period t0 to t1 because signals are changing state.

• Slave places requested input-data on data-lines at time t_1.
• At end of clock cycle(at time t2), master strobes(captures) data on data-lines into its input-buffer

• For data to be loaded correctly into any storage device (such as a register built with flip-flops), data must be available at input of that device for a period greater than setup-time of device.

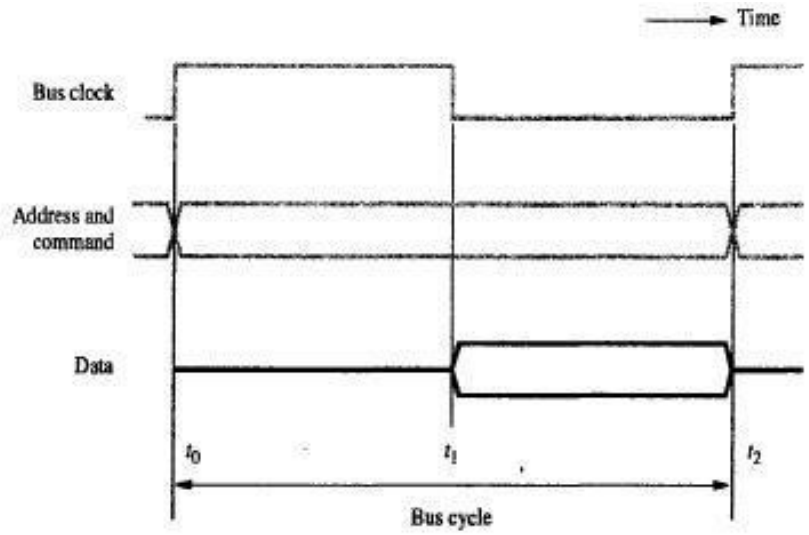
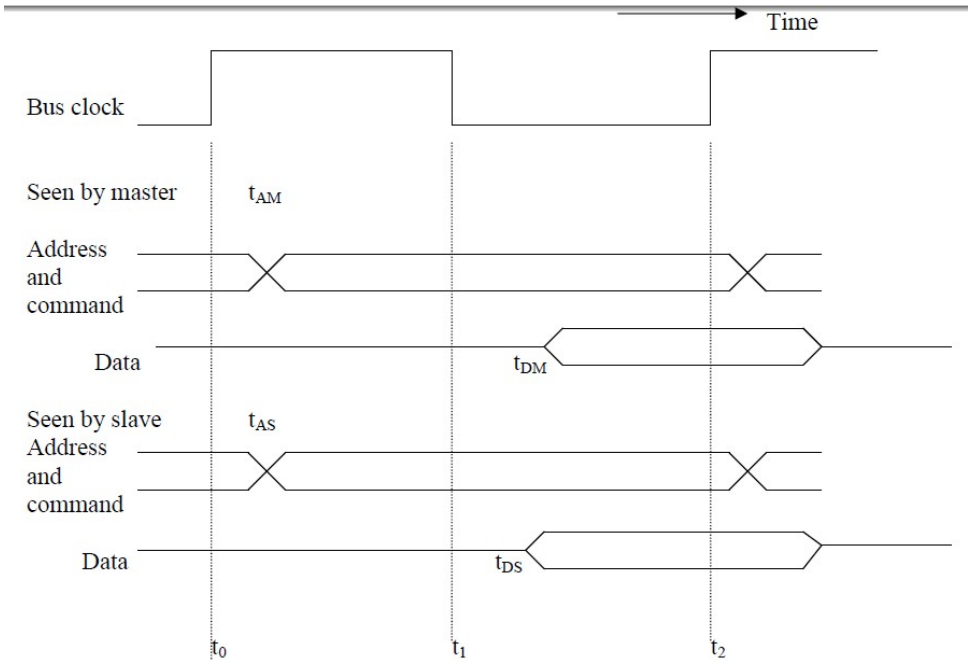


Figure 4.23 Timing of an input transfer on a synchronous bus.



Detailed timing diagram for input transfer of figure 4.23

The above figure gives a realistic picture of what happens in practice. Because signals take time to travel from one device to another, a given signal transition is seen by different devices at different times.

ASYNCHRONOUS BUS

- This method uses handshake-signals between master and slave for coordinating data transfers.
- There are 2 control-lines:
  1) Master-ready(MR) to indicate that master is ready for a transaction
  2) Slave-ready(SR) to indicate that slave is ready to respond

The read operation proceeds as follows:

 At t0, master places address- & command-information on bus. All devices on bus begin to decode this information.

 At t1, master sets MR-signal to 1 to inform all devices that the address- & command-information is ready.

 At t2, selected slave performs required input-operation & sets SR signal to 1 (Figure 4.26).

 At t3, SR signal arrives at master indicating that the input-data are available on bus skew.

 At t4, master removes address- & command-information from bus.

 At t5, when the device-interface receives the 1-to-0 transition of MR signal, it removes data and SR signal from the bus. This completes the input transfer

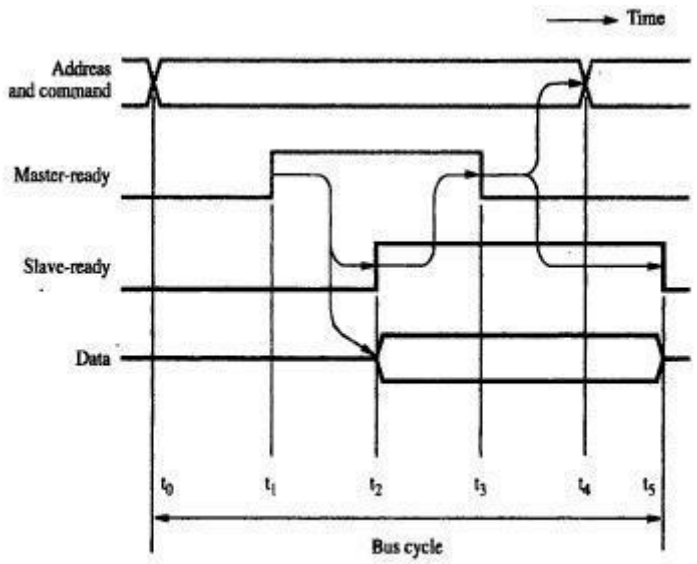


Figure 4.26 Handshake control of data transfer during an input operation.

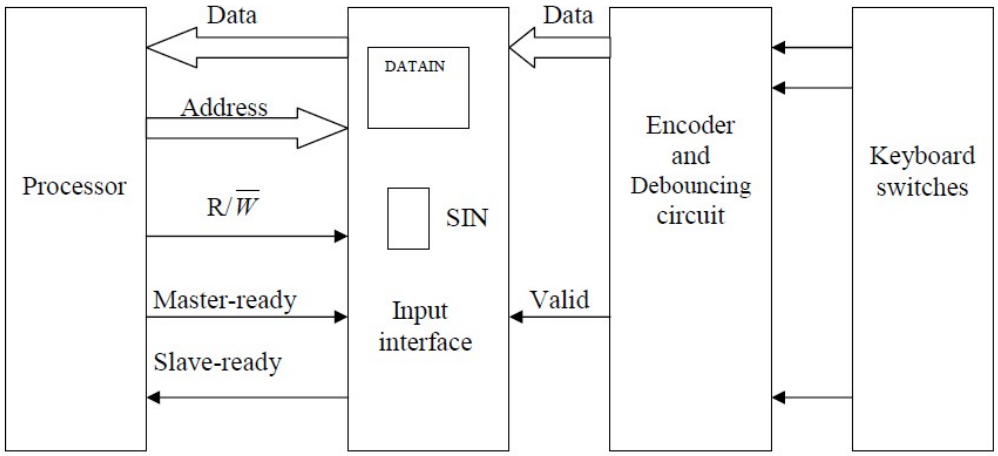
INTERFACE CIRCUITS

- I/O interface consists of the circuitry required to connect an I/O device to a computer bus.
- Side of the interface which connects to the computer has bus signals for:
  - Address,
  - Data
  - Control
- Side of the interface which connects to the I/O device has:

    - Data path and associated controls to transfer data between the interface and the I/O device.
    - This side is called as a "port".
- Ports can be classified into two:
    - Parallel port,
    - Serial port.
- Parallel port transfers data in the form of a number of bits, normally 8 or 16 to or from the device.
- Serial port transfers and receives data one bit at a time.
- Processor communicates with the bus in the same way, whether it is a parallel port or a serial port.
    - Conversion from the parallel to serial and vice versa takes place inside the interface circuit.

Q. Draw the hardware components needed for connecting a keyboard to a processor and explain the input interface circuit in detail.

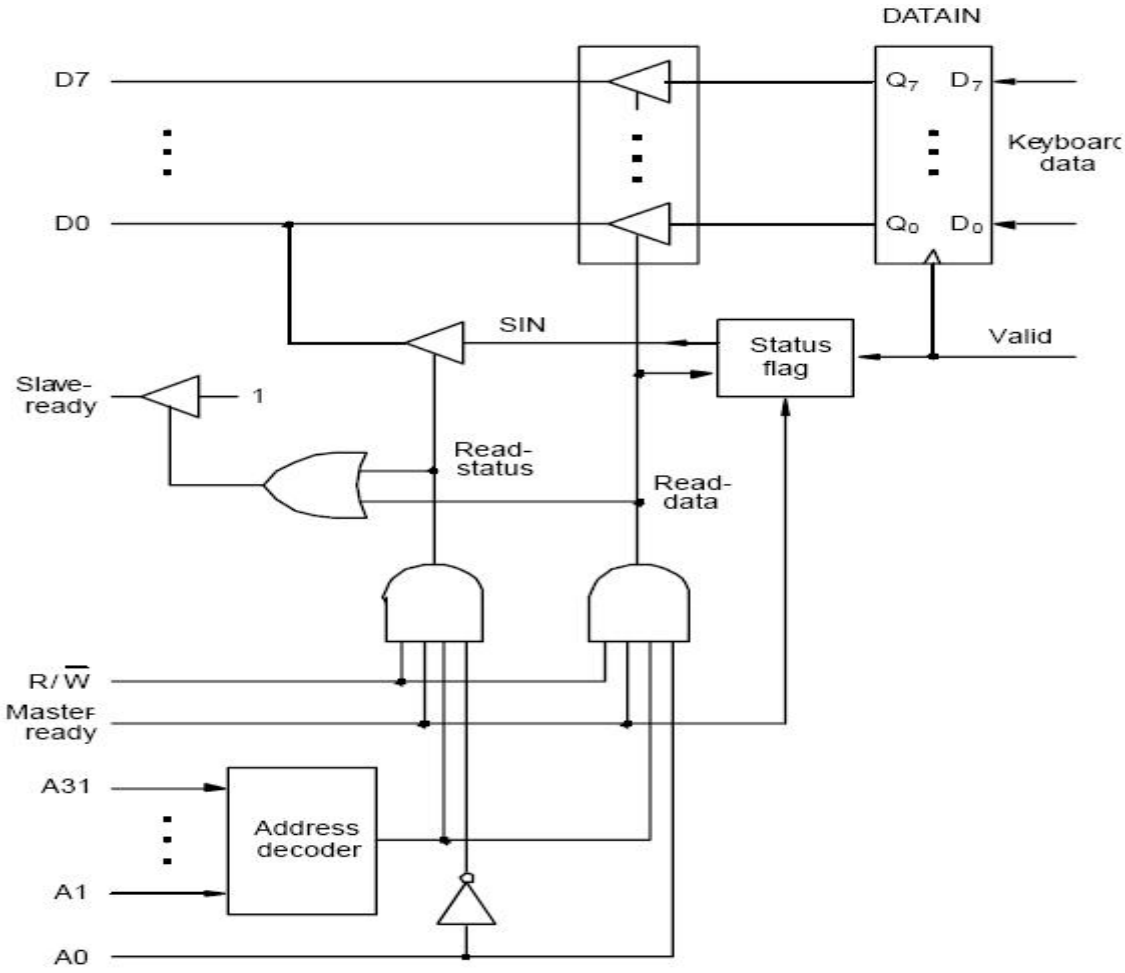
Parallel Port



- A circuits for an 8-bit input port and an 8-bit output port is considered. Then we combine the two circuits to show how the interface for a general-purpose 8-bit parallel port can be designed.
  We assume that the interface circuit is connected to a 32-bit processor that uses memory-mapped I/O and the asynchronous bus protocol.
  Hardware components needed for connecting a keyboard to a processor.
- A typical keyboard consists of mechanical switches that are normally open. When a key is pressed, the switches close and establish a path for an electrical signal. This signal is detected by an encoder circuit that generates the ASCII code for the corresponding character.
- A difficulty with such push-button switches is that contact bounce when a key is pressed. Although bouncing may last only or two milliseconds. This is long enough for a computer to observe a single pressing of a key as several distinct electrical events. The effect of bouncing must be eliminated.
- One control signal called Valid, which indicated that a key is being pressed.
- This information is sent to the interface circuit, which contains data register, DATAIN, and a status flag, SIN.

- When a key is pressed, the Valid signal changes from 0 to 1, causing the ASCII code to be loaded into DATAIN and SIN to be set to 1.
- The status flag SIN is cleared to 0 when the processor reads the contents of the DATAIN register. The interface circuit is connected to an asynchronous bus on which transfers are controlled using the handshake signals Master-ready and Slave-ready.

INPUT INTERFACE CIRCUIT



- Output lines of DATAIN are connected to the data lines of the bus by means of 3 state drivers
- Drivers are turned on when the processor issues a read signal and the address selects this register.
- SIN signal is generated using a status flag circuit.
- It is connected to line D_0 of the processor bus using a three-state driver.
- Address decoder selects the input interface based on bits A_1 through A_{31}.
- Bit A0 determines whether the status or data register is to be read, when Master-ready is active.
- In response, the processor activates the Slave-ready signal, when either the Read-status or Read-data is equal to 1, which depends on line A_0.

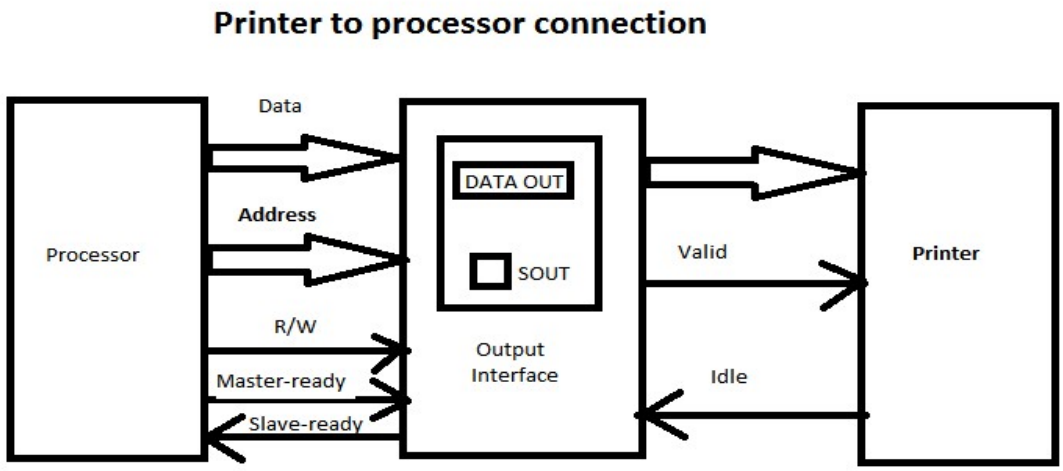
- For Status flag circuit refer to text book
- An edge triggered D flip flop is set to 1 by rising edge on a valid signal line.
- This event neither changes the state of NOR latch that sets SIN to 1.
- The state of latch must not change while SIN is being read by processor.

- Circuit ensures that SIN can be set only when Mater Ready is 0.
- Both flip flop and latch are reset to 0 when Read data is set to 1 to read the DATAIN register.

End of Answer

Q. With a block diagram explain how the printer is interfaced to processor

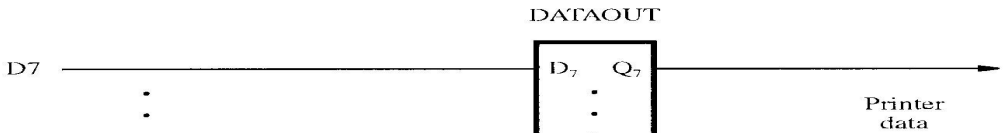
PRINTER TO PROCESSOR CONNECTION



- Printer is connected to a processor using a parallel port.
- Processor is 32 bits, uses memory-mapped I/O and asynchronous bus protocol.
- **On the processor side:**

  Data lines.
  Address lines
  Control or R/W line.
  Master-ready signal and
  Slave-ready signal.
- **On the printer side:**
- Printer operates under the control of handshake signals Mater Ready and Idle .

- Idle signal line which the printer asserts when it is ready to accept a character. This causes the SOUT flag to be set to 1.
- Processor places a new character into a DATAOUT register.
- Valid signal, asserted by the interface circuit when it places a new character on the data lines.
- In response printer starts printing new character and negates Idle signal, Which in turn causes interface circuit to deactivate valid signal.
- **OUTPUT INTERFACE CIRCUIT**



- Include common explanation
- Data lines of the processor bus are connected to the DATAOUT register of the interface.
- The status flag SOUT is connected to the data line D1 using a three-state driver.
- The three-state driver is turned on, when the control Read-status line is 1.
- Address decoder selects the output interface using address lines A1 through A31.
- Address line A0 determines whether the data is to be loaded into the DATAOUT register or status flag is to be read.
- In response, the processor activates the Slave-ready signal, when either the Read-status or Load-data is equal to 1, which depends on line A_0.
- If the Load-data line is 1, then the Valid line is set to 1.
- If the Idle line is 1, then the status flag SOUT is set to 1.

Answer ends here

COMBINED I/O INTERFACE CIRCUITS

- Address bits A2 through A31, that is 30 bits are used to select the overall interface.
- Address bits A1 through A0, that is, 2 bits select one of the three registers, namely, DATAIN, DATAOUT, and the status register.
- Status register contains the flags SIN and SOUT in bits 0 and 1.
- Data lines PA0 through PA7 connect the input device to the DATAIN register.
- DATAOUT register connects the data lines on the processor bus to lines PB0 through PB7 which connect to the output device.
- Separate input and output data lines for connection to an I/O device. Refer fig no. 4.33

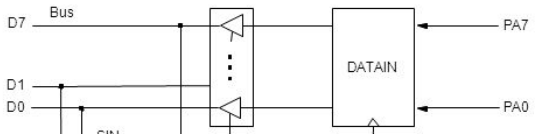
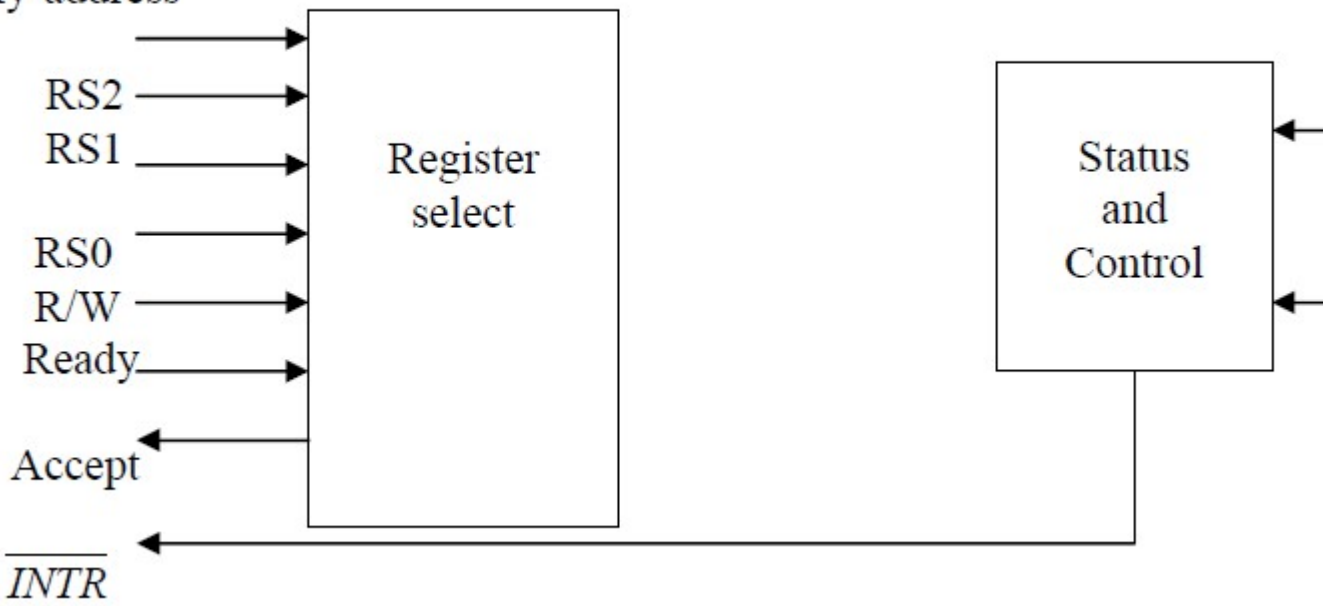


Fig 16 A general 8-bit interface

My-address



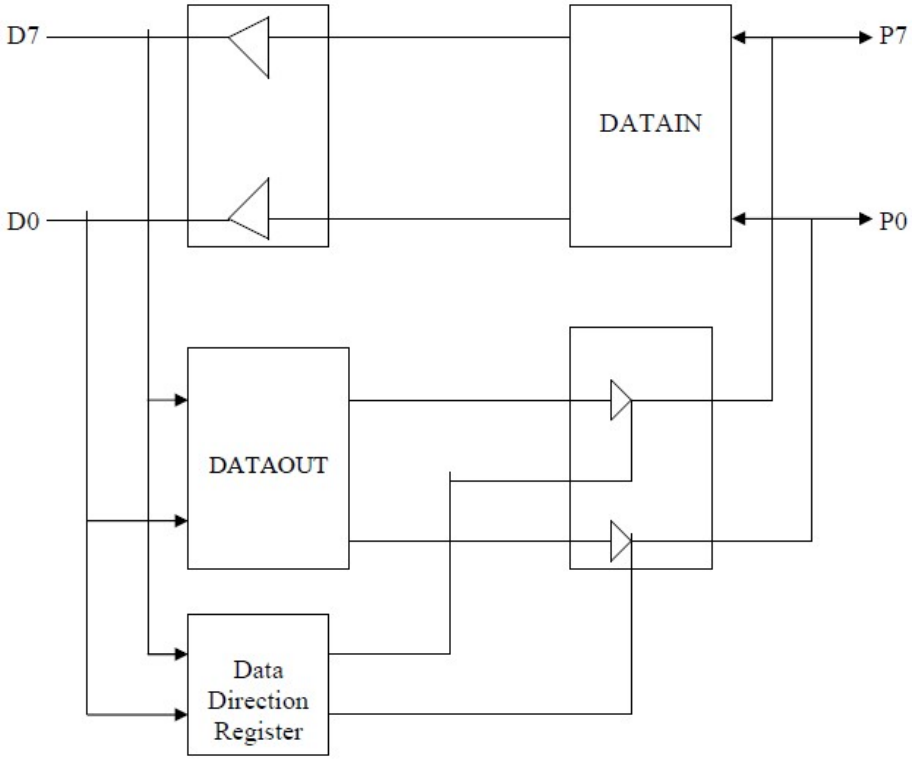
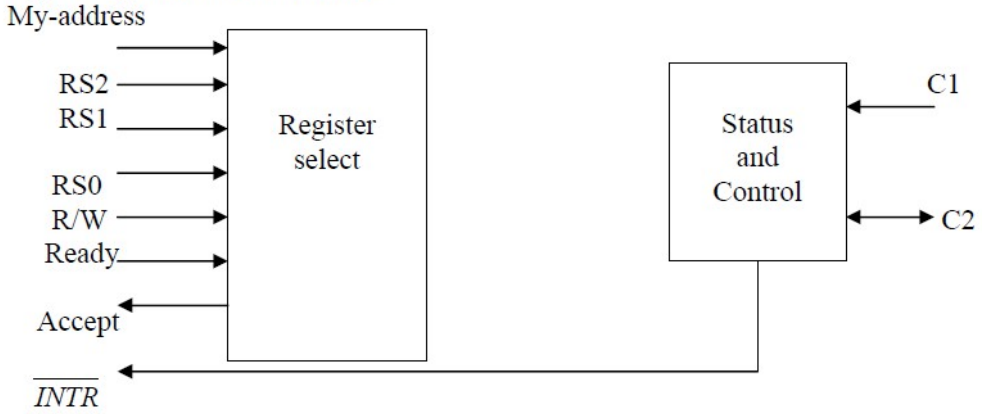


Fig 16 A general 8-bit interface



- Data line P0 through P7 can be used for either input or output purposes.
- The DATAOUT register is connected to these lines via three-state driver that are controlled by a data direction register DDR.
- The processor can write any 8 bit pattern into DDR .
- For a given bit if DDR value is 1 the corresponding data line acts as an output line, else as input line.
- C1 and C2 provide control interaction between interface and I/O device.
- C2 for different modes of signals.
- Ready and Accept lines are handshake control lines .
- The three register select lines allow 8 registers in interface such as DATAIN,DATAOUT, SIN,SOUT etc.
- INTR is for placing interrupt in IRQ line.

Q. Explain the serial port and serial interface.

SERIAL PORT

Refer to textbook for diagram

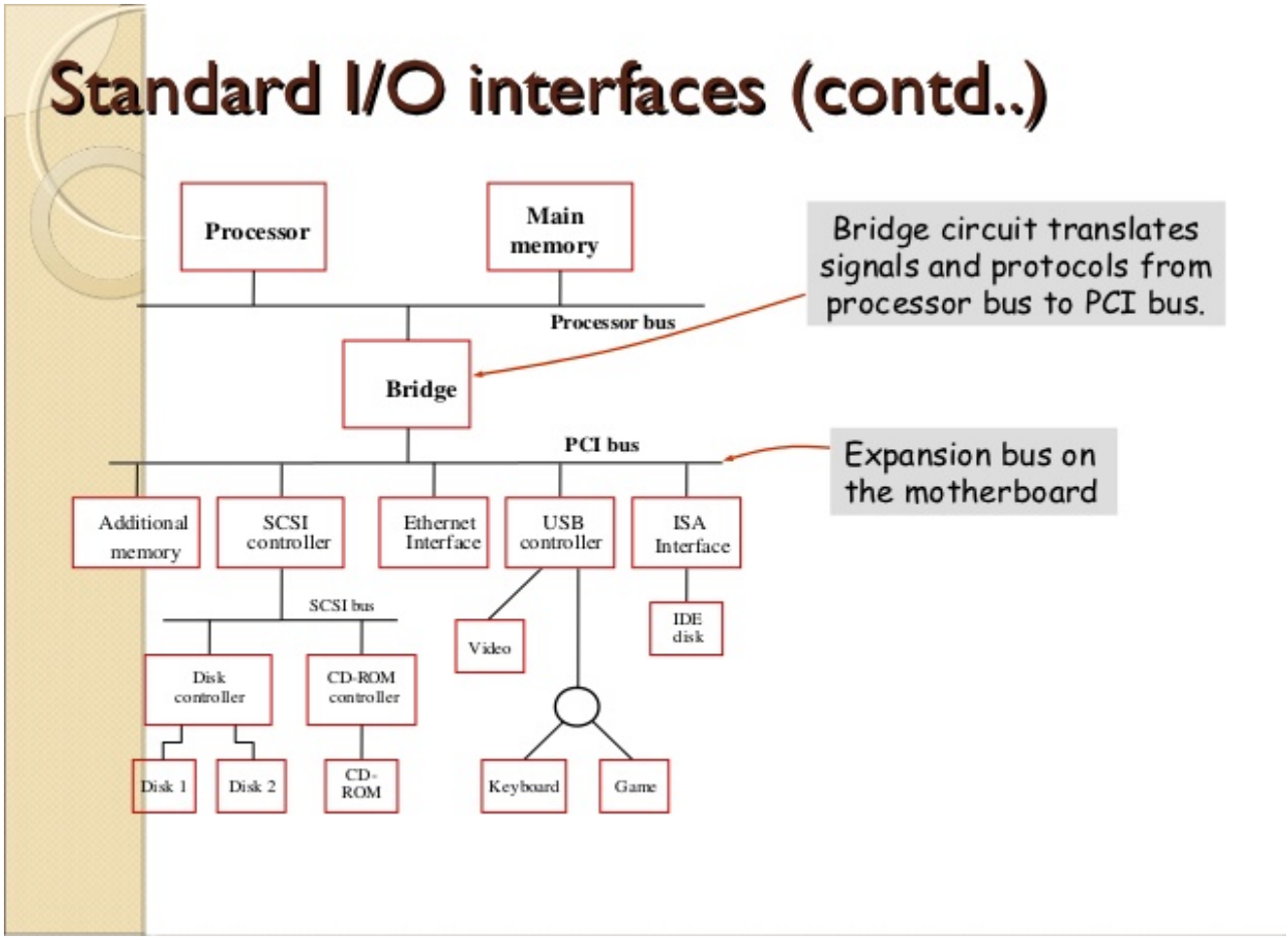
- Serial port is used to connect the processor to I/O devices that require transmission of data one bit at a time.
- Serial port communicates in a bit-serial fashion on the device side and bit parallel fashion on the bus side.
    - Transformation between the parallel and serial formats is achieved with shift registers that have parallel access capability.
- Input shift register accepts input one bit at a time from the I/O device. Refer fig no.4.37
- Once all the 8 bits are received, the contents of the input shift register are loaded in parallel into DATAIN register.
- Output data in the DATAOUT register are loaded into the output shift register.
- Bits are shifted out of the output shift register and sent out to the I/O device one bit at a time.
- As soon as data from the input shift registers are loaded into DATAIN, it can start accepting another 8 bits of data.
- Input shift register and DATAIN registers are both used at input so that the input shift register can start receiving another set of 8 bits from the input device after loading the contents to DATAIN, before the processor reads the contents of DATAIN. This is called as double-buffering.

- Serial interfaces require fewer wires, and hence serial transmission is convenient for connecting devices that are physically distant from the computer.
- Speed of transmission of the data over a serial interface is known as the “bit rate”.
    - Bit rate depends on the nature of the devices connected.
- In order to accommodate devices with a range of speeds, a serial interface must be able to use a range of clock speeds.
- Several standard serial interfaces have been developed:
    - Universal Asynchronous Receiver Transmitter (UART) for low-speed serial devices.
    - RS-232-C for connection to communication links.

Answer ends here

STANDARD I/O INTERFACES

- I/O device is connected to a computer using an interface circuit.
- Do we have to design a different interface for every combination of an I/O device and a computer?
- A practical approach is to develop standard interfaces and protocols.
- A personal computer has:
    - A motherboard which houses the processor chip, main memory and some I/O interfaces.
    - A few connectors into which additional interfaces can be plugged.
- Processor bus is defined by the signals on the processor chip.
    - Devices which require high-speed connection to the processor are connected directly to this bus.
- Because of electrical reasons only a few devices can be connected directly to the processor bus.
- Motherboard usually provides another bus that can support more devices.
    - Processor bus and the other bus (called as expansion bus) are interconnected by a circuit called “bridge”.
    - Devices connected to the expansion bus experience a small delay in data transfers.
- Design of a processor bus is closely tied to the architecture of the processor.
    - No uniform standard can be defined.
- Expansion bus however can have uniform standard defined.
- A number of standards have been developed for the expansion bus.
    - Some have evolved by default.
    - For example, IBM’s Industry Standard Architecture.
- Three widely used bus standards:
    - PCI (Peripheral Component Interconnect)
    - SCSI (Small Computer System Interface)
    - USB (Universal Serial Bus)



Q. Explain PCI Bus

PCI BUS

- *Peripheral Component Interconnect*
- Introduced in 1992
- Low-cost bus
- Processor independent
- Plug-and-play capability
- In today's computers, most memory transfers involve a burst of data rather than just one word. The PCI is designed primarily to support this mode of operation.
- The bus supports three independent address spaces: memory, I/O, and configuration.
- we assumed that the master maintains the address information on the bus until data transfer is completed. But, the address is needed only long enough for the slave to be selected. Thus, the address is needed on the bus for one clock cycle only, freeing the address lines to be used for sending data in subsequent clock cycles. The result is a significant cost reduction.
- A master is called an initiator in PCI terminology. The addressed device that responds to read and write commands is called a target.

Refer figure 4.39 in text book

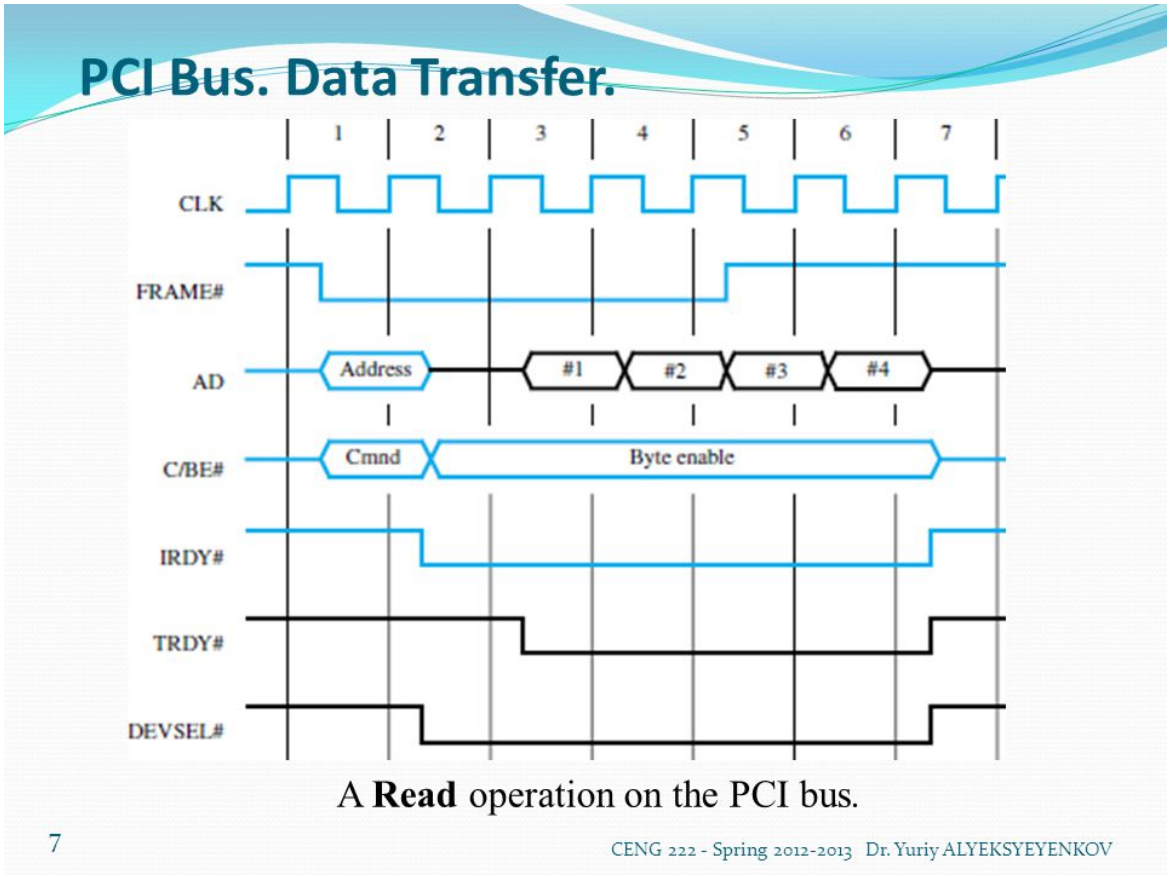
- **Device Configuration**
- When an I/O device is connected to a computer, several actions are needed to configure both the device and the software that communicates with it.
- PCI incorporates in each I/O device interface a small configuration ROM memory that stores information about that device.

- The configuration ROMs of all devices are accessible in the configuration address space. The PCI initialization software reads these ROMs and determines whether the device is a printer, a keyboard, an Ethernet interface, or a disk controller. It can further learn about various device options and characteristics.
- Devices are assigned addresses during the initialization process.
- This means that during the bus configuration operation, devices cannot be accessed based on their address, as they have not yet been assigned one.
- Hence, the configuration address space uses a different mechanism. Each device has an input signal called Initialization Device Select, IDSEL#
- Electrical characteristics:
    - PCI bus has been defined for operation with either a 5 or 3.3 V power supply

For PCI Bus Signals and Timing diagram refer to Table 4.3 and Fig 4.40 in text book
Answer ends here

Q. Explain how read operation happens in PCI Bus

- In clock cycle 1 FRAME# is asserted by processor to indicate beginning of transaction.
- At the same time address on AD lines and Command read on C/BE# line are sent.
- clock cycle 2 processor removes address and disconnects drivers from AD line.
- The selected target enables its drivers on AD lines and fetches the requested data and place it on data bus during clock cycle 3.
- It asserts DEVSEL# and maintains state until end of transaction.
- The initiator sets one or more C/BE# lines to indicate which byte line are enabled for data transfer.
- During clock cycle 3 initiator asserts IRDY# to indicate ready to receive data .
- Target asserts TRDY# and sends a word of data.
- The initiator loads data in buffer at end of clock cycle.
- Target sends three more words in cycle 4 to 6.
- The initiator negates FRAME# during CC5
- After sending word 4 in clock cycle 6 target disconnects its drivers and negates DEVSEL#.



SCSI BUS

- The acronym SCSI stands for Small Computer System Interface.
- It refers to a standard bus defined by the American National Standards Institute (ANSI)
- The SCSI bus standard has undergone many revisions, and its data transfer capability has increased very rapidly, almost doubling every two years.
- SCSI-2 and SCSI-3 have been defined, and each has several options.
- Because of various options SCSI connector may have 50, 68 or 80 pins.
- The SCSI bus is connected to the processor bus through a SCSI controller. This controller uses DMA to transfer data packets from the main memory to the device, or vice versa.
- A packet may contain a block of data, commands from the processor to the device, or status information about the device.
- A controller connected to a SCSI bus is one of two types – an initiator or a target.
- An initiator has the ability to select a particular target and to send commands specifying the operations to be performed. The disk controller operates as a target. It carries out the commands it receives from the initiator.
- The initiator establishes a logical connection with the intended target.
- Once this connection has been established, it can be suspended and restored as needed to transfer commands and bursts of data.
- While a particular connection is suspended, other device can use the bus to transfer information.
- This ability to overlap data transfer requests is one of the key features of the SCSI bus that leads to its high performance.
- Data transfers on the SCSI bus are always controlled by the target controller.

- To send a command to a target, an initiator requests control of the bus and, after winning arbitration, selects the controller it wants to communicate with and hands control of the bus over to it.
- Then the controller starts a data transfer operation to receive a command from the initiator.
- Assume that processor needs to read block of data from a disk drive and that data are stored in disk sectors that are not contiguous.
- The processor sends a command to the SCSI controller, which causes the following sequence of events to take place:
    - The SCSI controller, acting as an initiator, contends for control of the bus.
    - When the initiator wins the arbitration process, it selects the target controller and hands over control of the bus to it.
    - The target starts an output operation (from initiator to target); in response to this, the initiator sends a command specifying the required read operation.
- The target, realizing that it first needs to perform a disk seek operation, sends a message to the initiator indicating that it will temporarily suspend the connection between them. Then it releases the bus.
- The target controller sends a command to the disk drive to move the read head to the first sector involved in the requested read operation. Then, it reads the data stored in that sector and stores them in a data buffer. When it is ready to begin transferring data to the initiator, the target requests control of the bus. After it wins arbitration, it reselects the initiator controller, thus restoring the suspended connection.
- The target transfers the contents of the data buffer to the initiator and then suspends the connection again
- The target controller sends a command to the disk drive to perform another seek operation. Then, it transfers the contents of the second disk sector to the initiator as before. At the end of this transfers, the logical connection between the two controllers is terminated.
- As the initiator controller receives the data, it stores them into the main memory using the DMA approach.
- The SCSI controller sends as interrupt to the processor to inform it that the requested operation has been completed.

Q. List the sequence of events that takes place when processor sends a command to the SCSI controller

1. The SCSI controller, acting as an initiator, contends for control of the bus.
2. When the initiator wins the arbitration process, it selects the target controller and hands over control of the bus to it.
3. The target starts an output operation (from initiator to target); in response to this, the initiator sends a command specifying the required read operation.
4. The target, realizing that it first needs to perform a disk seek operation, sends a message to the initiator indicating that it will temporarily suspend the connection between them. Then it releases the bus.
5. The target controller sends a command to the disk drive to move the read head to the first sector involved in the requested read operation. Then, it reads the data stored in that sector and stores them in a data buffer. When it is ready to begin transferring data to the initiator, the target requests control of the bus. After it wins arbitration, it reselects the initiator controller, thus restoring the suspended connection.
6. The target transfers the contents of the data buffer to the initiator and then suspends the connection again. Data are transferred either 8 or 16 bits in parallel, depending on the width of the bus.
7. The target controller sends a command to the disk drive to perform another seek operation. Then, it transfers the contents of the second disk sector to the initiator, as before. At the end of this transfer, the logical connection between the two controllers is terminated.
8. As the initiator controller receives the data, it stores them into the main memory using the DMA approach.
9. The SCSI controller sends an interrupt to the processor to inform it that the requested operation has been completed.

Q. List SCSI Bus signals with their functionalities.

Category	Name	Function
Data	– DB(0) to – DB(7)	Data lines: Carry one bbyteof information during the information transfer phase and identify device during arbitration, selection and reselection phases
	– DB(P)	Parity bit for the data bus
Phase	– BSY	Busy: Asserted when the bus is not free
	– SEL	Selection: Asserted during selection and reselection
Information type	– C/D	Control/Data: Asserted during transfer of controll information (command, status or message)
	– MSG	Message: indicates that the information being transferred is a message

Category	Name	Function
Handshake	– REQ	Request: Asserted by a target to request a data transfer cycle
	– ACK	Acknowledge: Asserted by the initiator when it has completed a data transfer oporation
Direction of transfer	– I/O	Input/Output: Asserted to indicate an input oporation (relativee to the initiator)
Other	– ATN	Aten ion: Asserted by an initiator when it wishes to send a message to a target
	– RST	Reset: Causes all device controlls to disconnect from the bus and assume their start-up state

Q. Briefly describe the phases involved in SCSI Bus operation

Refer to textbook fig 4.42 for phase's diagram

MAIN PHASES INVOLVED

- **Arbitration**
  - A controller requests the bus by asserting BSY and by asserting it's associated data line
  - When BSY becomes active, all controllers that are requesting bus examine data lines
  - Each controller on bus is assigned a fixed priority with controller 7 having highest priority.
  - SCSI uses distributed arbitration scheme.
  - The controller using highest numbered line realizes that it has won the arbitration process. All other controllers disconnect from bus and wait for BSY to become inactive again.
  - In diagram controller 6 is initiator wishes to establish connection with controller 5.
  - Controller 6 proceeds with selection phase in which it identifies target.
- **Selection**
  - Controller 6 that won arbitration selects target by asserting SEL and data line of target. After that initiator releases BSY line.
  - This informs initiator that the connection it is requesting has been established.
  - 6 Removes Address information from data lines
  - Selection process is completed and Target controller 5 responds by asserting BSY line
  - Target controller 5 will have control on the bus from then.
- **Information Transfer**
  - Handshaking signals are used between initiator and target.
  - Now target controller takes the role of us master.
  - –REQ and –ACK signals replace Master ready and Slave ready signals.
  - The Target asserts –I/O during an input operation and –C/D to indicate that the information transferred is command or a status or data.

- At the end target releases BSY line

- **Reselection**
- When logical connection is suspended and the target is ready to restore it the target must first gain control of the bus.
- It starts an arbitration cycle and after winning it selects the initiator controller.
- But the roles of target and initiator are reversed, the initiator is now asserting -BSY

USB

- Universal Serial Bus (USB) is an industry standard developed through a collaborative effort of several computer and communication companies, including Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, Nortel Networks, and Philips.
- **Speed**
  - Low-speed(1.5 Mb/s)
  - Full-speed(12 Mb/s)
  - High-speed(480 Mb/s)
- **Objectives of USB**

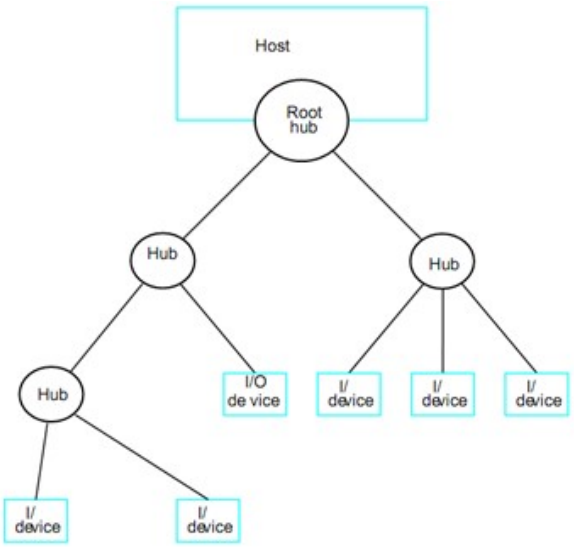
1. **Port Limitation**
   - To add a new port a user must open computer box to gain access to internal expansion bus and install a new interface card.
   - User needs to know how to configure the device and software.
   - But in USB many devices can be added at any time without opening the computer box.
2. **Device Characteristics**

- Speed, volume and timing constraint's vary from one device to another
- Keyboard, Mouse uses asynchronous timing
  - Speed limited to human speed of 100 characters per second.
- Mocrophone Uses isochronous meaning that successive events are separated by equal periods of time.
  - Needs data rate of 1.4 Megabits/s
- Data transfer for images and video need high data rate.

3. **Plug-and-play**
   Connect device at any time while the system is operating. The system should detect the existence of new device automatically identify appropriate device driver software and establish logical connection for communication.

Q. Explain USB Tree Structure.

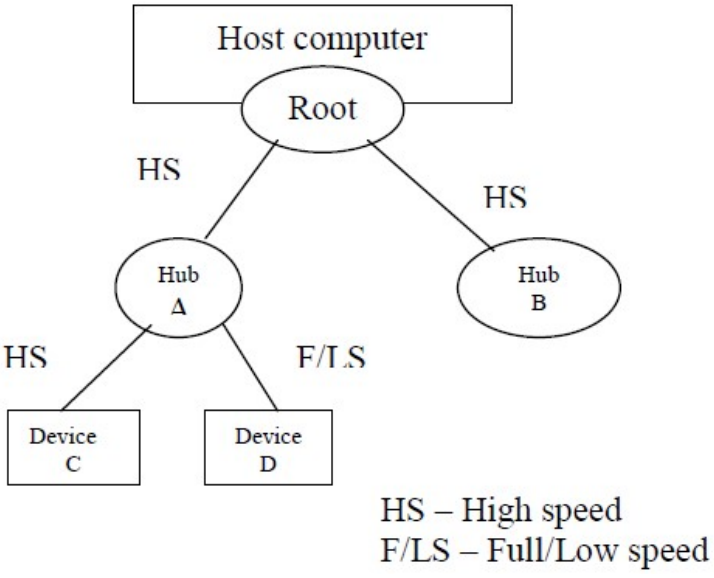
USB TREE STRUCTURE



- To accommodate a large number of devices that can be added or removed at any time, the USB has the tree structure as shown in the figure.
- Each node of the tree has a device called a hub, which acts as an intermediate control point between the host and the I/O devices.
- At the root of the tree, a root hub connects the entire tree to the host computer. The leaves of the tree are the I/O devices being served (for example, keyboard, Internet connection, speaker, or digital TV)
- In normal operation, a hub copies a message that it receives from its upstream connection to all its downstream ports. As a result, a message sent by the host computer is broadcast to all I/O devices, but only the addressed device will respond to that message. However, a message from an I/O device is sent only upstream towards the root of the tree and is not seen by other devices. Hence, the USB enables the host to communicate with the I/O devices, but it does not enable these devices to communicate with each other.
- When a USB is connected to a host computer, its root hub is attached to the processor bus, where it appears as a single device. The host software communicates with individual
    - devices attached to the USB by sending packets of information, which the root hub forwards to the appropriate device in the USB tree.

Q. Explain Split bus operation of USB Bus

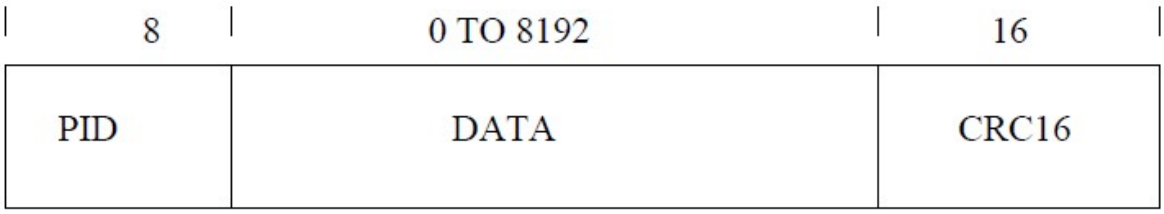
- **Split Bus Operation**
    - USB operates in differents mode such as High speed(HS), Full Speed (FS)and Low Speed(LS).
    - For Example
    - Hub A is connected to root by high speed link. Hub A serves C on HS and D on LS
    - A message to device D is sent on LS from root hub at 1.5 Megabits/s. even a short message takes several microseconds.For duration of this period no other transmission can take place, thus reducing the effectiveness of HS link.
    - To solve this USB protocol requires that a message transmitted on HS link is always transmitted on HS, even if device operates at LS.
    - Hence message intended to device D is sent on HS link from root hub to hubA then forwarded on LS link from hubA to device D.
    - The latter transfer takes longer time during which HS traffic to other nodes is allowed to continue.
    - During these period the bus is said to be split between HS and LS traffic.



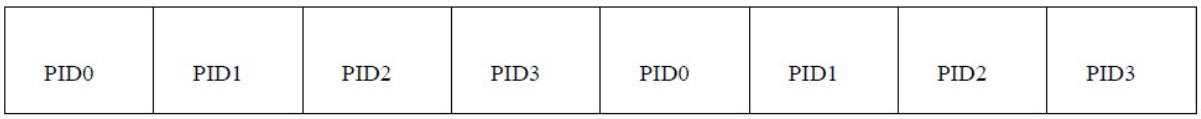
-
- ## Addressing in USB
- Each device on the USB, whether it is a hub or an I/O device, is assigned a 7-bit address. This address is local to the USB tree and is not related in any way to the addresses used on the processor bus.
- A hub may have any number of devices or other hubs connected to it, and addresses are assigned arbitrarily. When a device is first connected to a hub, or when it is powered on, it has the address 0. The hardware of the hub to which this device is connected is capable of detecting that the device has been connected, and it records this fact as part of its own status information. Periodically, the host polls each hub to collect status information and learn about new devices that may have been added or disconnected.
- When the host is informed that a new device has been connected, it uses a sequence of commands to send a reset signal on the corresponding hub port, read information from the device about its capabilities, send configuration information to the device, and assign the
- Device a unique USB address. Once this sequence is completed the device begins normal operation and responds only to the new address.

Q. Explain USB protocols/ Packets

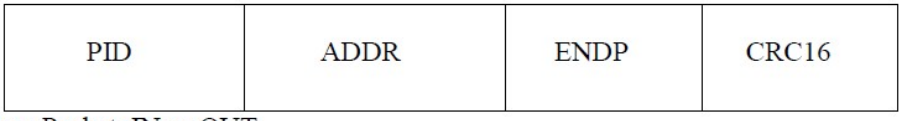
- **USB protocols**
- All information transferred over the USB is organized in packets, where a packet consists of one or more bytes of information. There are many types of packets that perform a variety of control functions.
- The information transferred on the USB can be divided into two broad categories: **control and data.**
    - Control packets perform such tasks as addressing a device to initiate data transfer, acknowledging that data have been received correctly, or indicating an error.
    - Data packets carry information that is delivered to a device.
- A packet consists of one or more fields containing different kinds of information.
- **Token Packet**
    - The first field of any packet is called the packet identifier, PID, which identifies the type of that packet.
    - They are transmitted twice. The first time they are sent with their true values, and the second time with each bit complemented
    - The four PID bits identify one of 16 different packet types such as IN,OUT, ACK, etc.
    - PID is followed by 7 bit address of a device and 4 bit endpoint number within that device(Control/data/status register)
    - 5 bits for error checking using cyclic redundancy check(CRC).
- **Data Packet**
    - Carry input and output data.
    - PID is followed by 8192 bits of data the 16 error checking bits.
    - The three PID pattern are used to identify data packets so that data packets are numbered as 0,1,or 2
    - Data packets do not carry any device address or endpoint information. This information is carried in IN and OUT token packets.



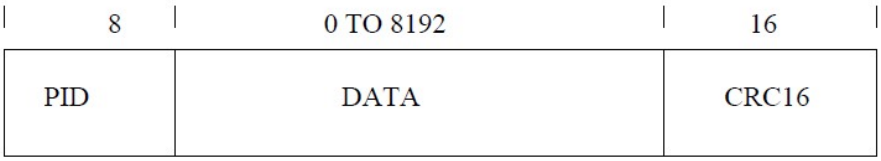
C) Data Packet



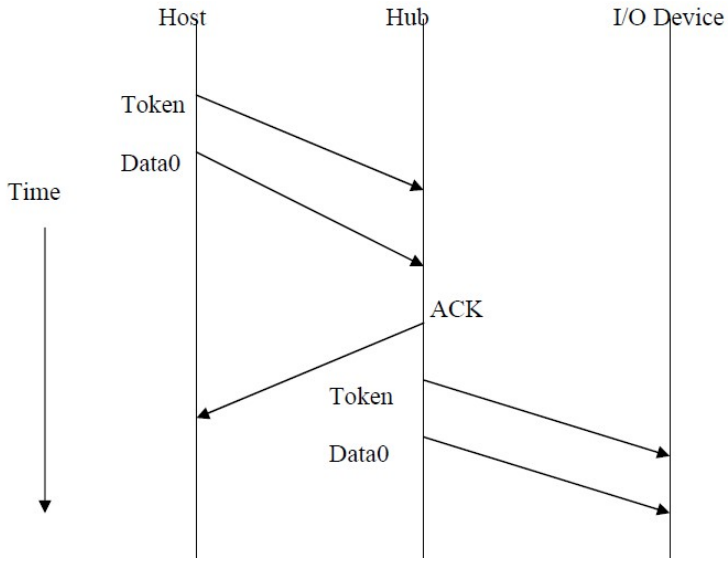
A) packet identifier Field



B) Token Packet ,IN or OUT



C) Data Packet



USB Frames

For diagram refer to fig 4.47

- USB supports isochronous data that require time reference to control the sampling process.
- The transmission over USB is divided into frames of equal length.
- A frame is 1ms long for F/LS data.
- The root hub generates a Start of Frame control packet once every 1ms to marks the beginning of a new frame.

- The arrival of SOF packet at any device constitutes a regular clock signal that the device can use for its own purpose.
- Following SOF host carried input or output transfers.
- All devices will have opportunity for an input or output once every 1ms.
-
- **Electrical Characteristics**

    - The cables used for USB connections consist of four wires.
    - Two are used to carry power, +5V and Ground.
        - Thus, a hub or an I/O device may be powered directly from the bus, or it may have its own external power connection.
    - The other two wires are used to carry data.
    - Different signaling schemes are used for different speeds of transmission.
        - At low speed, 1s and 0s are transmitted by sending a high voltage state (5V) on one or the other o the two signal wires. For high-speed links, differential transmission is used.

MODULE – 3

THE MEMORY SYSTEM

5.1 BASIC CONCEPTS:

The maximum size of the Main Memory (MM) that can be used in any computer is determined by its addressing scheme. For example, a 16-bit computer that generates 16-bit
addresses is capable of addressing upto 2^{16} =64K memory locations. If a machine generates 32-bit addresses, it can access upto 2^{32} = 4G memory locations. This number represents the size of address space of the computer.

If the smallest addressable unit of information is a memory word, the machine is called word-addressable. If individual memory bytes are assigned distinct addresses, the computer is called byte-addressable. Most of the commercial machines are byte-addressable. For example in a byte-addressable 32-bit computer, each memory word contains 4 bytes. A possible word-address assignment would be:

Word Address	Byte Address			
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11
.		…..		
.		…..		
.		…..		

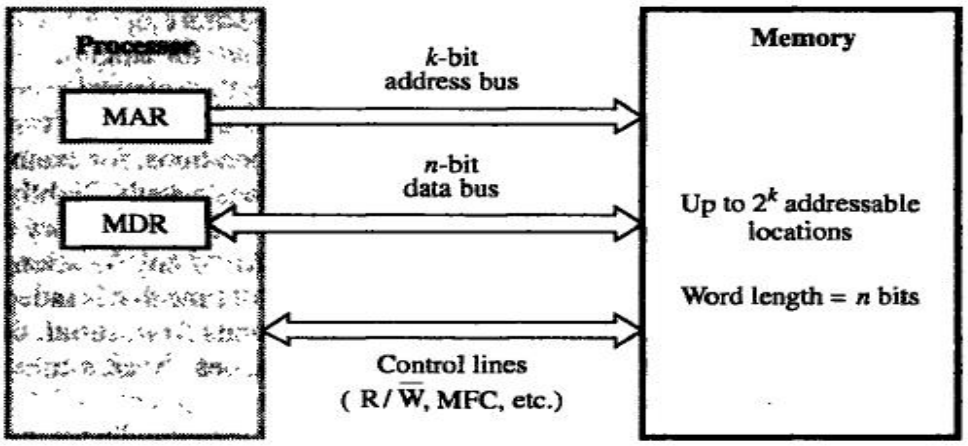
With the above structure a READ or WRITE may involve an entire memory word or it may involve only a byte. In the case of byte read, other bytes can also be read but ignored by the CPU. However, during a write cycle, the control circuitry of the MM must ensure that only the specified byte is altered. In this case, the higher-order 30 bits can specify the word and the lower-order 2 bits can specify the byte within the word.

CPU-Main Memory Connection – A block schematic: -

Data transfer between CPU and MM takes place through the use of two CPU registers, usually called MAR (Memory Address Register) and MDR (Memory Data Register). If MAR is K bits long and MDR is „n" bits long, then the MM unit may contain

upto 2^k addressable locations and each location will be „n" bits wide, while the word length is equal to „n" bits. During a "memory cycle", n bits of data may be transferred between the

· and CPU. This transfer takes place over the processor bus, which has k address lines (address bus), n data lines (data bus) and control lines like Read, Write, Memory Function completed (MFC), Bytes specifiers etc (control bus). For a read operation, the CPU loads the address into MAR, set R/W to 1 and sets other control signals if required. The data from the MM is loaded into MDR and MFC is set to 1. For a write operation, MAR, MDR are suitably loaded by the CPU,R/W is set to 0 and other control signals are set suitably. The MM control circuitry loads the data into appropriate locations and sets MFC to 1. This organization is shown in the following block schematic.



· **Figure 5.1** Connection of the memory to the processor.

Some Basic Concepts

Memory Access Times: -

It is a useful measure of the speed of the memory unit. It is the time that elapses between the initiation of an operation and the completion of that operation (for example, the time between READ and MFC).

Memory Cycle Time :-

It is an important measure of the memory system. It is the minimum time delay required between the initiations of two successive memory operations (for example, the time between two successive READ operations). The cycle time is usually slightly longer than the access time.

RAM: A memory unit is called a Random Access Memory if 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. Main memory units are of this type. This distinguishes them from serial or partly serial access storage devices such as magnetic tapes and disks which are used as the secondary storage device.

Cache Memory:-

The CPU of a computer can usually process instructions and data faster than they can be fetched from compatibly priced main memory unit. Thus the memory cycle time becomes the bottleneck in the system. One way to reduce the memory access time is to use cache memory. This is a small and fast memory that is inserted between the larger, slower main memory and the CPU. This holds the currently active segments of a program and its data. Because of the locality of address references, the CPU can, most of the time, find the relevant information in the cache memory itself (cache hit) and infrequently needs access to the main memory (cache miss) with suitable size of the cache memory, cache hit rates of over 90% are possible leading to a cost-effective increase in the performance of the system.

Memory Interleaving: -

This technique divides the memory system into a number of memory modules and arranges addressing so that successive words in the address space are placed in different modules. When requests for memory access involve consecutive addresses, the access will be to different modules. Since parallel access to these modules is possible, the average rate of fetching words from the Main Memory can be increased.

Virtual Memory: -

In a virtual memory System, the address generated by the CPU is referred to as a virtual or logical address. The corresponding physical address can be different and the required mapping is implemented by a special memory control unit, often called the memory

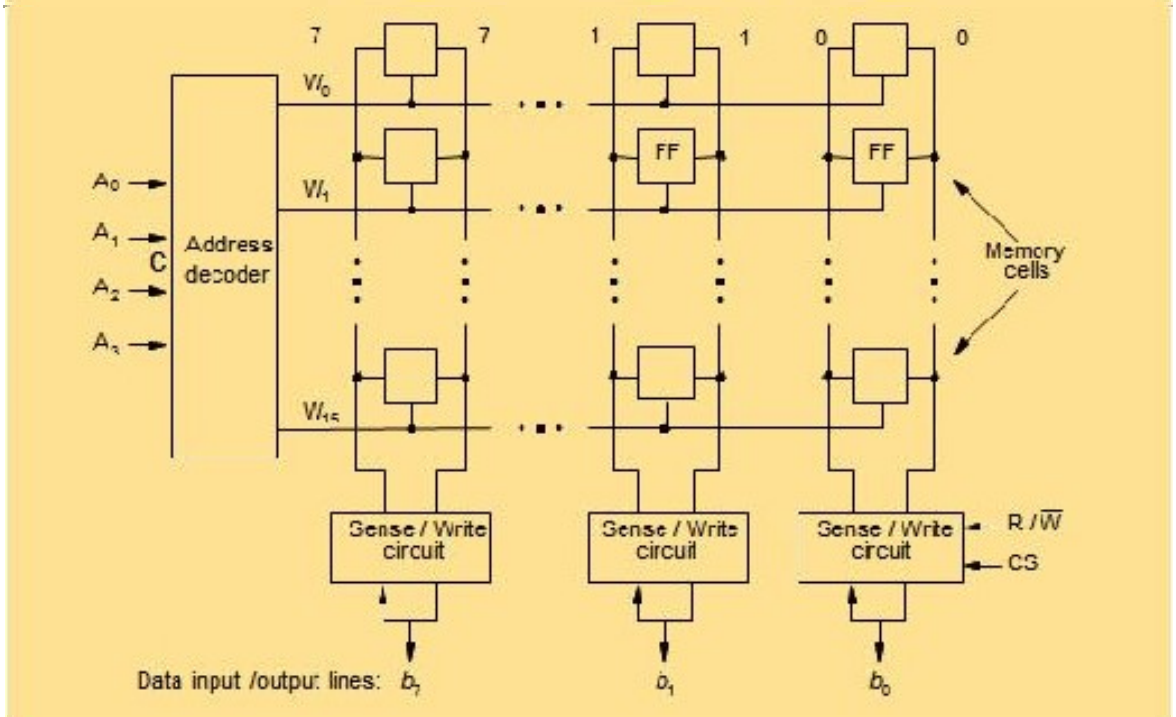
management unit. The mapping function itself may be changed during program execution according to system requirements.

Because of the distinction made between the logical (virtual) address space and the physical address space; while the former can be as large as the addressing capability of the CPU, the actual physical memory can be much smaller. Only the active portion of the virtual address space is mapped onto the physical memory and the rest of the virtual address space is mapped onto the bulk storage device used. If the addressed information is in the Main Memory (MM), it is accessed and execution proceeds. Otherwise, an exception is generated, in response to which the memory management unit transfers a contigious block of words containing the desired word from the bulk storage unit to the MM, displacing some block that is currently inactive. If the memory is managed in such a way that, such transfers are required relatively infrequency (ie the CPU will generally find the required information in the MM), the virtual memory system can provide a reasonably good performance and succeed in creating an illusion of a large memory with a small, in expensive MM.

5.2 SEMICONDUCTOR RAM MEMORIES

5.2.1 Internal Organization of Memory Chips

- Memory cells are usually organized in the form of an **array**, in which each cell is capable of storing on **bit** of information.
- Each row of cells constitutes a 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 the chip.
- The cells in each column are connected to a Sense/Write circuit by two **bit lines**. The Sense/Write circuits are connected to the data I/O lines of the chip. During the 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 the write operation, the Sense/Write circuits receive the input information and store in the cells of the selected word.



The above figure is an example of a very small memory chip consisting of 16 words of 8 bits each. This is referred to as a 16×8 organization. The data input and the data output of each Sense/Write circuit are connected to a single bidirectional data line that can be connected to the data bus of a computer. Two control lines, R/W (Read/ Write) input specifies the required operation, and the CS (Chip Select) input selects a given chip in a multichip memory system.

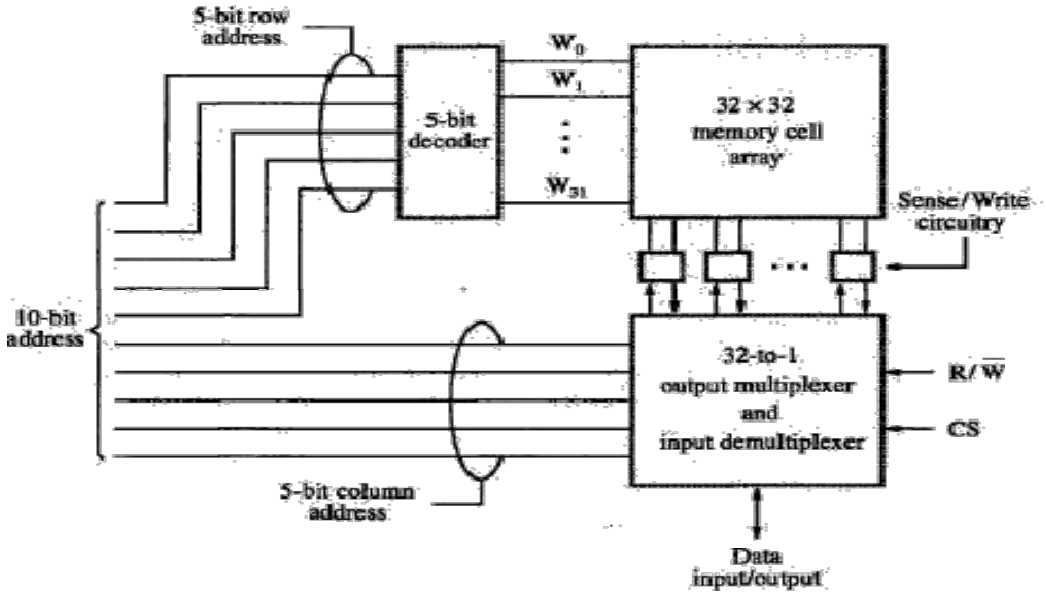
The memory circuit given above stores 128 bits and requires 14 external connections for address, data and control lines. Of course, it also needs two lines for power supply and ground connections.

Q. Organization of 1K X 1 Memory Chip

Consider now a slightly larger memory circuit, one that has a 1k (1024) memory cells.

- Arranged as:
  - 128x8 memory,
    - Rows = 128 = word Lines
    - Number of Address Lines= $\log_2 128=7$
    - Columns= 8= Number of Data Lines
    - Number of external connections: 19(7+8+2+2)
  - 1Kx1
    - Arranged as 32 X 32
    - Number of Address Lines= $\log_2 1024=10$
    - Columns= 1= Number of data lines
    - Number of external connections: 15 (10+1+2+2)

For a 1k×1 memory organization, the representation is given next. The required 10-bit address is divided into two groups of 5 bits each to form the row and column addresses for the cell array. A row address selects a row of 32 cells, all of which are accessed in parallel. However, according to the column address, only one of these cells is connected to the external data line by the 32 to 1 output multiplexer and 1 to 32 input de-multiplexer.



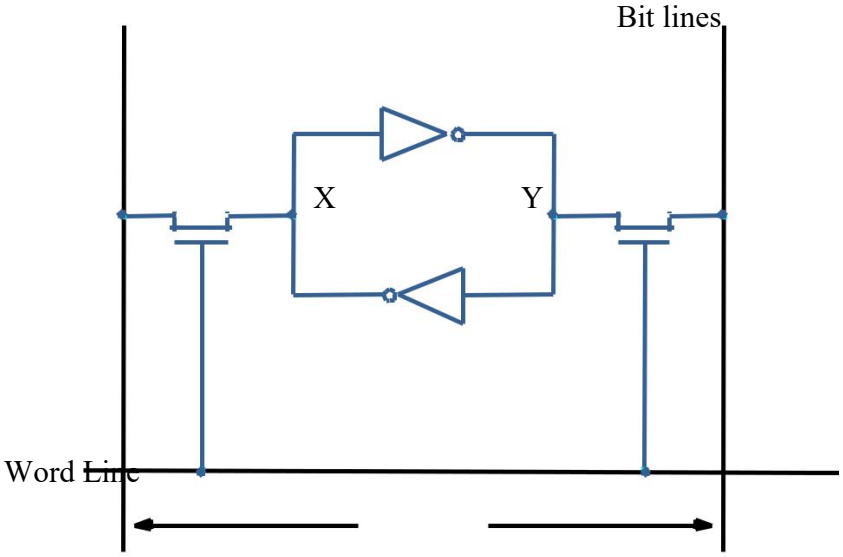
\

Figure 5.3 Organization of a 1K × 1 memory chip.

Q. Explain Static RAM Cell

5.2.2 Static Memories

Memories that consist of circuits capable of retaining their state as long as power is applied are known as static memories.



- The above figure illustrates how a static RAM (SRAM) cell may be implemented.
- Two inverters are cross- connected to form a latch.
- The latch is connected to two bit lines by transistors T_1 and T_2.
- These transistors act as switches that can be opened or closed under control of the word line.
- When the word line is at ground level, the transistors are turned off and the latch retains its state. For example, let us assume that the cell is in state 1 if the logic value at point X is 1 and at point Y is 0. This state is maintained as long as the signal on the word line is at ground level.

Read Operation

- In order to read the state of the 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 the bit line b is high and the signal on the bit line b‟ is low. The opposite is true if the cell is in state 0. Thus b and b'are compliments of each other.
- Sense/Write circuits at the end of the bit lines monitor the state of b and b‟ and set the output accordingly.

Write Operation

- The state of the cell is set by placing the appropriate value on bit line b and its complement b', and then activating the word line.
- This forces the cell into the corresponding state.
- The required signals on the bit lines are generated by the Sense/Write circuit.

CMOS Cell

A CMOS realization of the static RAM cell is given below:

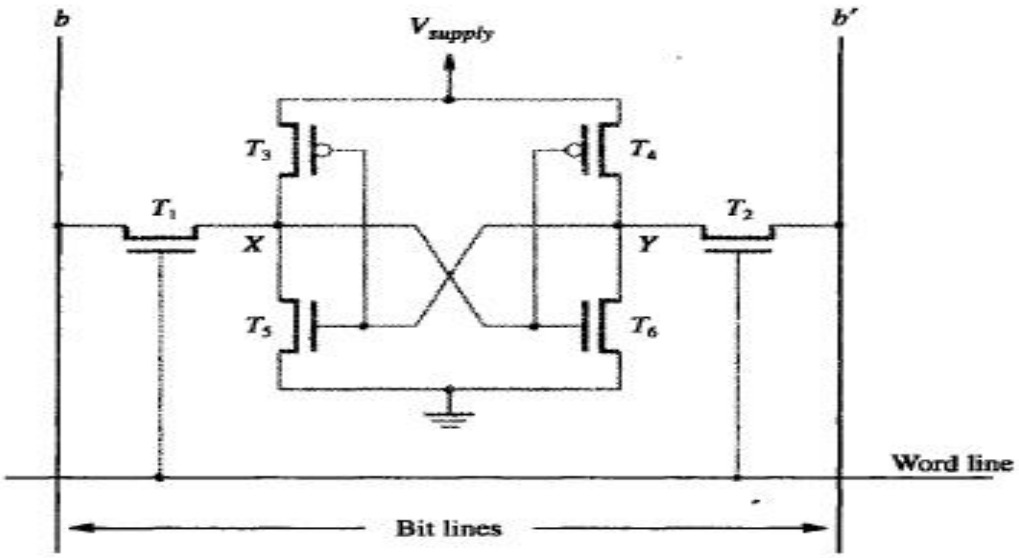


Figure 5.5 An example of a CMOS memory cell.

Transistor pairs (T_3, T_5) and (T_4, T_6) form the inverters in the latch. The state of the cell is read or written as just explained.

For example, in state 1, the voltage at point X is maintained high by having transistors T_3 and T_6 on, while T_4 and T_5 are off. Thus, if T_1 and T_2 are turned on (closed), bit lines b and b‟ will have high and low signals, respectively.

5.2.3 Asynchronous DRAMS

- An example of a dynamic memory cell that consists of a capacitor, C, and a transistor, T, is shown below.
- Information is stored in a dynamic memory cell in the form of a 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.

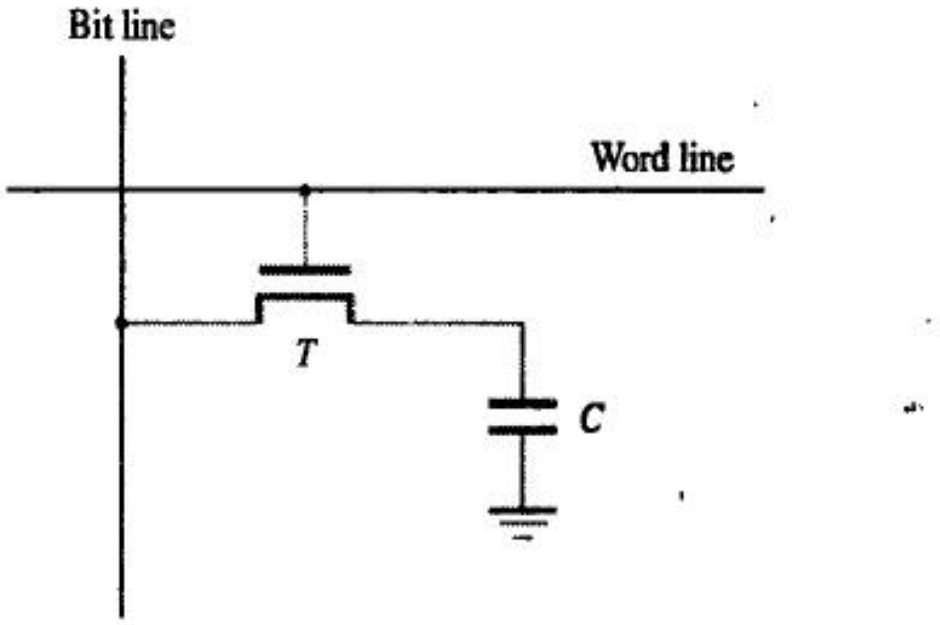


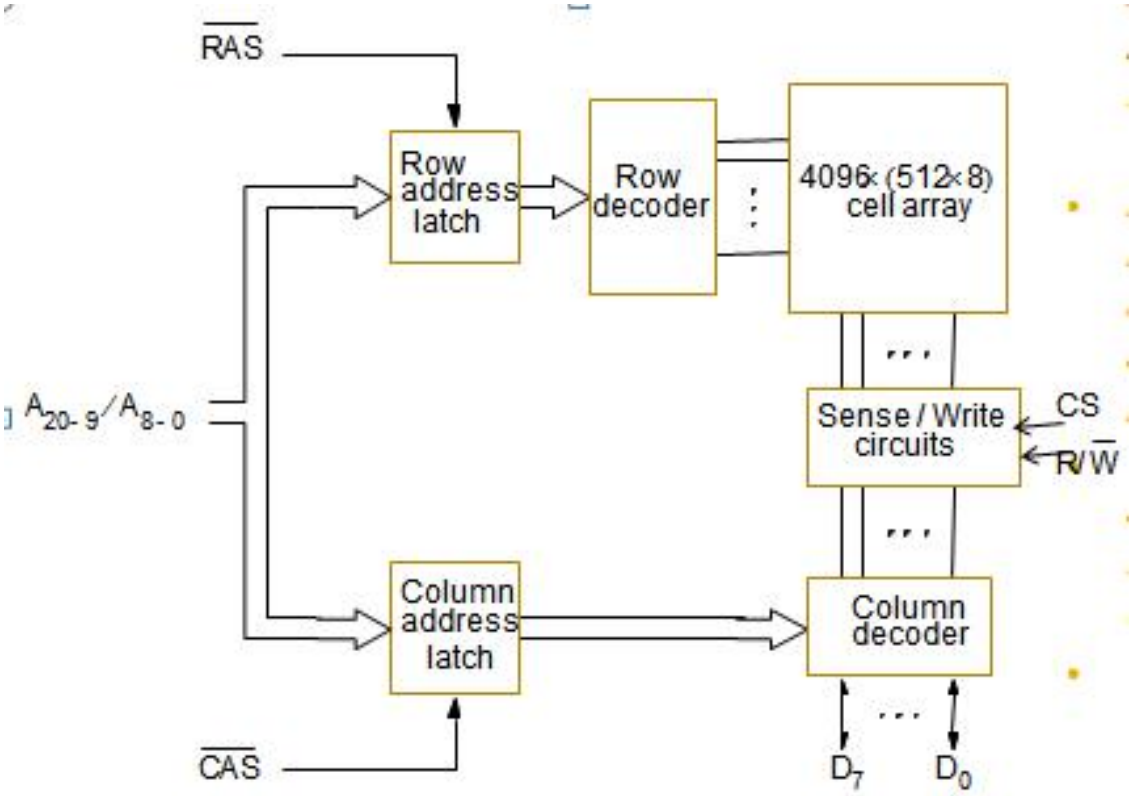
Figure 5.6 A single-transistor dynamic memory cell.

- A sense amplifier connected to the bit line detects whether the charge stored on the capacitor is above the threshold. If so, it drives the bit line to a full voltage that represents logic value 1.
- This voltage recharges the capacitor to full charge that corresponds to logic value 1. If the sense amplifier detects that the charge on the capacitor will have no charge, representing logic value 0.

A 16-megabit DRAM chip, configured as 2M×8, is shown below.

Q. With a neat diagram explain the internal organization of a 2M X 8 dynamic Memory Chip.

- 16 million bits-> 16 million cells
- Organized as 4kX4k array.
  - 4096X4096 = 4096X(512X8)=4096X(2MX8)
  - 4096 cells in each row divided into 512 groups of 8 cells each .
  - Each Rows stores 512 bytes = 4096 bits of data.
- 4096 column Configured as 2MX8
- High order 12 bits for Address line to select Row= log24096
- Low order 9 bits address lines to select group of 8 bits=log2512
- 12+9=21 address line/pins required.
- Reduce number of address lines = Row and col address multiplexed on 12 pins
  - Using Row Address Latch and Col Address latch
  - First apply the row address; RAS signal latches the row address. Then apply the column address, CAS signal latches the address.
  - Timing of the memory unit is controlled by a specialized unit which generates RAS and CAS.
  - This is asynchronous DRAM



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.

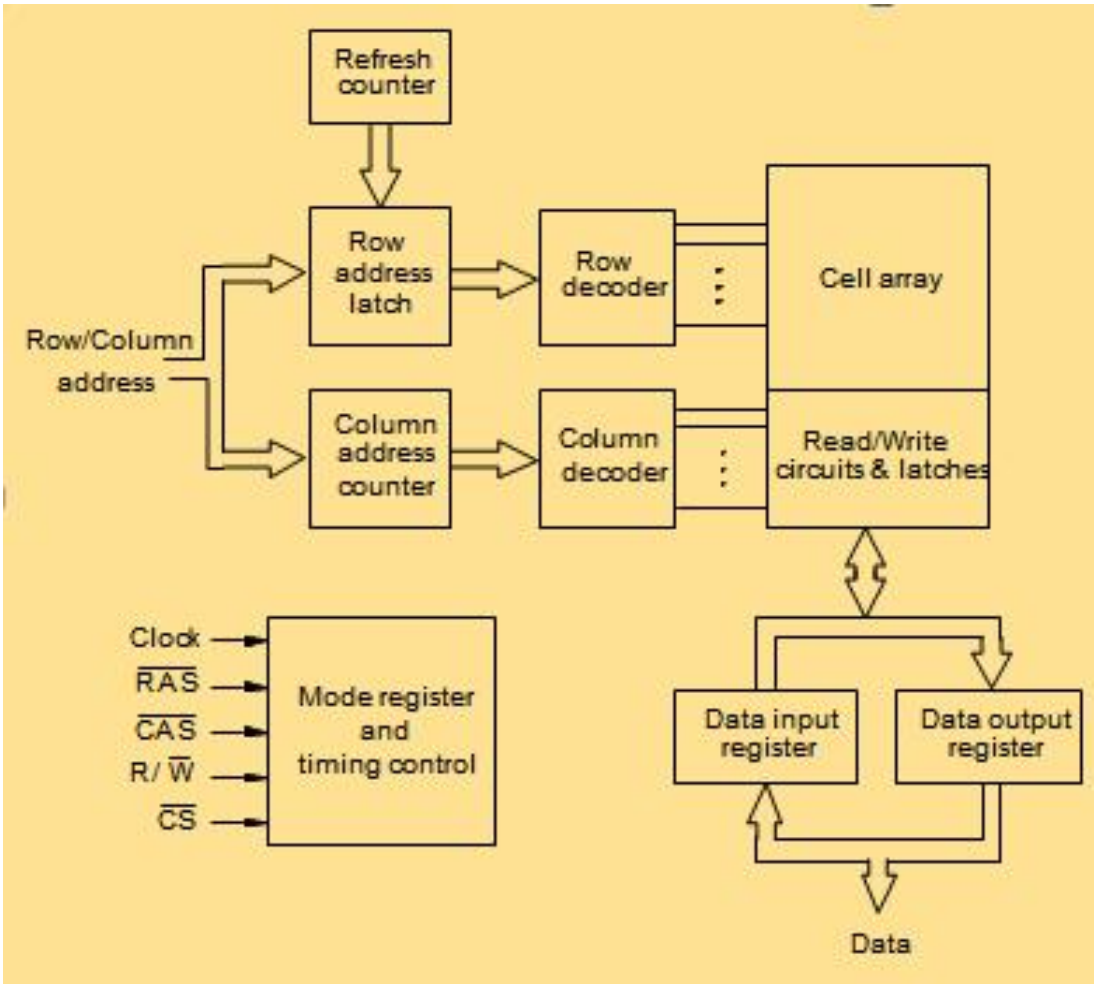
Add a latch at the output of the sense circuits in each row.
All the latches are loaded when the row is selected.

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.
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.

5.2.4 Synchronous DRAMs

In these DRAMs, operation is directly synchronized with a clock signal. The below given figure indicates the structure of an SDRAM.



- The output of each sense amplifier is connected to a latch.
- A Read operation causes the contents of all cells in the selected row to be loaded into these latches.
- But, if an access is made for refreshing purpose only, it will not change the contents of these latches; it will merely refresh the contents of the cells.
- Data held in the latches that correspond to the selected column(s) are transferred into the output register, thus becoming available on the data output pins.
- SDRAMs have several different modes of operation, which can be selected by writing control information into a mode register. For example, burst operations of different lengths are specified.
- The burst operations use the block transfer capability described before as fast page mode feature.
- In SDRAMs, it is not necessary to provide externally generated pulses on the CAS line to select successive columns. The necessary control signals are provided internally using a column counter and the clock signal. New data can be placed on the data lines in each clock cycles. All actions are triggered by the rising edge of the clock.

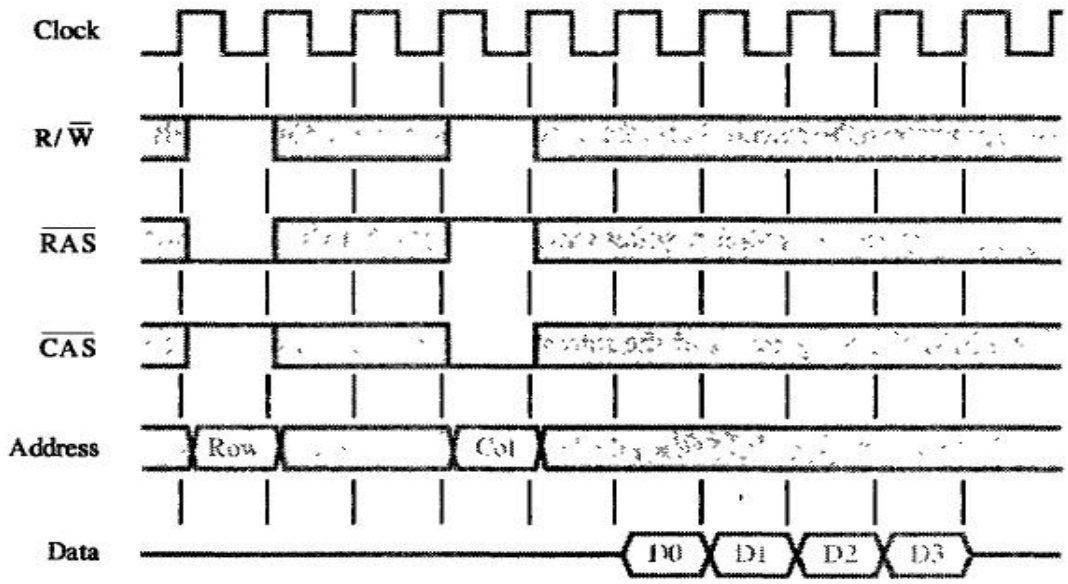


Figure 5.9 Burst read of length 4 in an SDRAM.

The above figure shows the timing diagram for a burst read of length 4.

First, the row address is latched under control of the RAS signal.

Then, column address latched under control of the CAS signal.

After a delay of one clock cycle, the first set of data bits is placed on the data lines.

The SDRAM automatically increments the column address to access next three sets of the bits in the selected row, which are placed on the data lines in the next clock cycles.

Q. Define Memory Latency and Memory Bandwidth

Latency and Bandwidth

- **Memory latency** is the time it takes to transfer a word of data to or from memory.
- In case of reading single word latency provides complete memory performance.
- In case of burst operation that transfers block of data the time needed to complete the operation depends on rate at which successive words can be transferred and size of block.
- It is denoted as time required transferring first word that is longer than time for subsequent word.

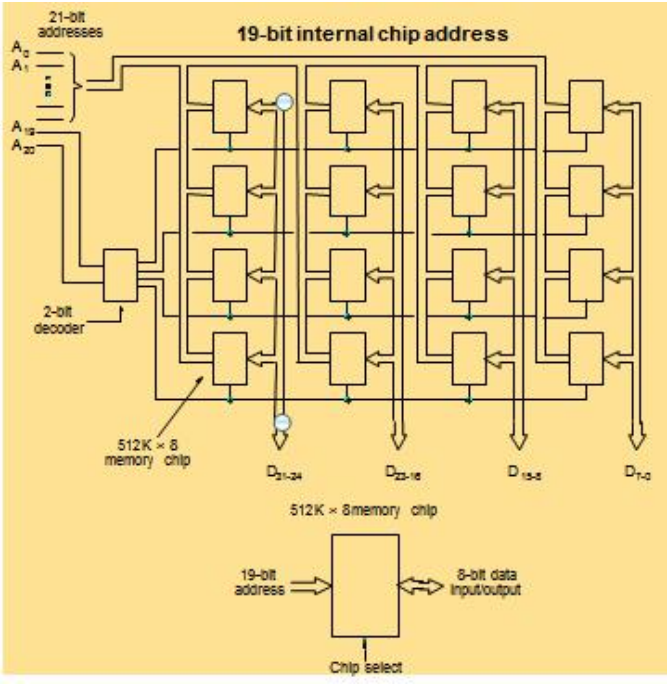
- **Memory bandwidth** is the number of bits or bytes that can be transferred in one second. As blocks can be of varying sizes it is better to define performance measure as number of bits or bytes transferred per second

Double Data Rate- Synchronous DRAMs (DDR- SDRAMs)

- To assist the processor in accessing data at high enough rate, the cell array is organized in two banks.
- Each bank can be accessed separately.
- Consecutive words of a given block are stored in different banks.
- Such interleaving of words allows simultaneous access to two words that are transferred on the successive edges of the clock. This type of SDRAM is called Double Data Rate SDRAM (DDR- SDRAM).

5.2.5 Structure of larger memories

Organization of 2MX32 Memory Chip using SRAM

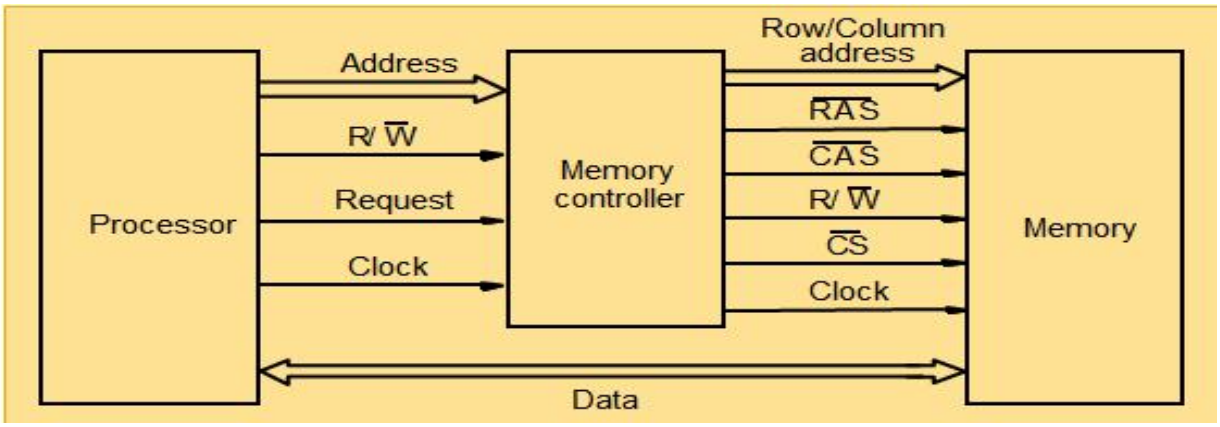


- Implementing a memory unit of 2M words of 32 bits each.
- Using 512Kx8 static memory chips arranged in 4 rows and 4 columns. Each chip implements one byte position.
- A chip is selected by setting its chip select control line to 1. Selected chip places its data on the data output line, outputs of other chips are in high impedance state.
- 21 bits to address a 32-bit word. 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.
- **Organization of 2MX32 Memory Chip using DRAM**

- 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.
- Occupy less space on the motherboard.
- Allows for easy expansion by replacement.

5.2.6 Memory System Considerations

- Recall that in a dynamic memory chip, to reduce the number of pins, multiplexed addresses are used.
- Address is divided into two parts:
  - High-order address bits select a row in the array.
  - They are provided first, and latched using RAS signal.
  - Low-order address bits select a column in the row.
- They are provided later, and latched using CAS signal
- However, a processor issues all address bits at the same time.
- In order to achieve the multiplexing, memory controller circuit is inserted between the processor and memory.



Refresh Operation:-

The Refresh control block periodically generates Refresh, requests, causing the access control block to start a memory cycle in the normal way. This block allows the refresh operation by activating the Refresh Grant line. The access control block arbitrates between Memory Access requests and Refresh requests, with priority to refresh requests in the case of a tie to ensure the integrity of the stored data.

As soon as the Refresh control block receives the Refresh Grant signal, it activates the Refresh line. This causes the address multiplexer to select the Refresh counter as the source and its contents are thus loaded into the row address latches of all memory chips when the RAS signal is activated.

Q. Explain Memory Hierarchy in terms of speed, size and cost

5.4 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**.**

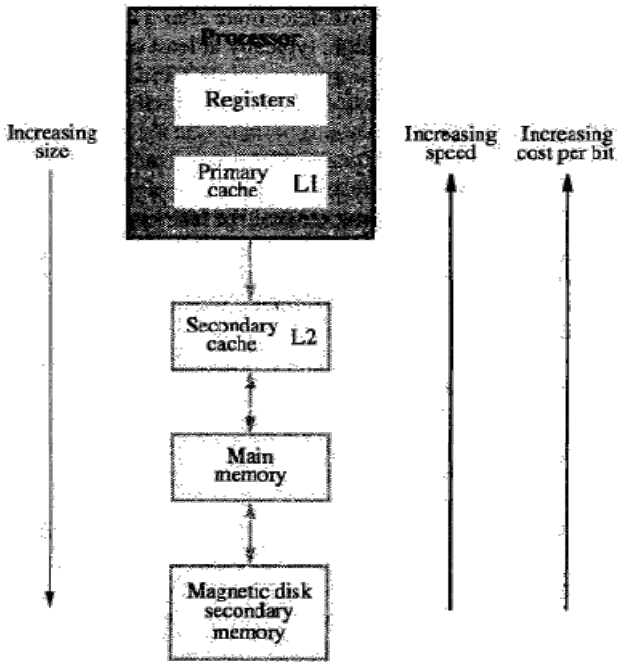


Figure 5.13 Memory hierarchy.

-
- Fastest access is to the data held in processor registers. Registers are at the top of the memory hierarchy.
- Relatively small amount of memory that can be implemented on the processor chip. This is processor cache. Two levels of cache. Level 1 (L1) cache is on the processor chip. Level 2 (L2) cache is in between main memory and processor. Implemented using SRAM.
- Next level is main memory, implemented as SIMMs. Much larger, but much slower than cache memory. Implemented using DRAM.
- Next level is magnetic disks. Huge amount of inexpensive storage. Speed of memory access is critical, the idea is to bring instructions and data that will be used in the near future as close to the processor as possible.

5.5 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. This serves as a major obstacle towards achieving good performance.
- Speed of the main memory cannot be increased beyond a certain point. So we use Cache memories.
- Cache memory is an architectural arrangement which makes the main memory appear faster to the processor than it really is. Cache memory is based on the property of computer programs known as "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". Its types are:

- **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.

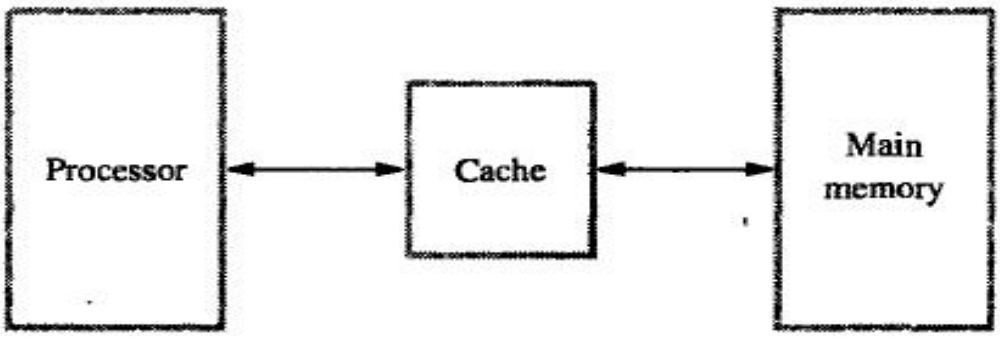


Figure 5.14 Use of a cache memory.

A simple arrangement of cache memory is as shown above.

- 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 are 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".

Q. Define Cache hit (read, write) and cache miss (read, write)

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, 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.

Q Explain all the cache mapping functions in detail

Mapping functions: Mapping functions determine how memory blocks are placed in the cache.

A simple processor example:

- Cache consisting of 128 blocks of 16 words each.
- Total size of cache is 2048 (2K) words.
- Main memory is addressable by a 16-bit address.
- Main memory has 64K words.

- Main memory has 4K blocks of 16 words each.

Three mapping functions can be used.

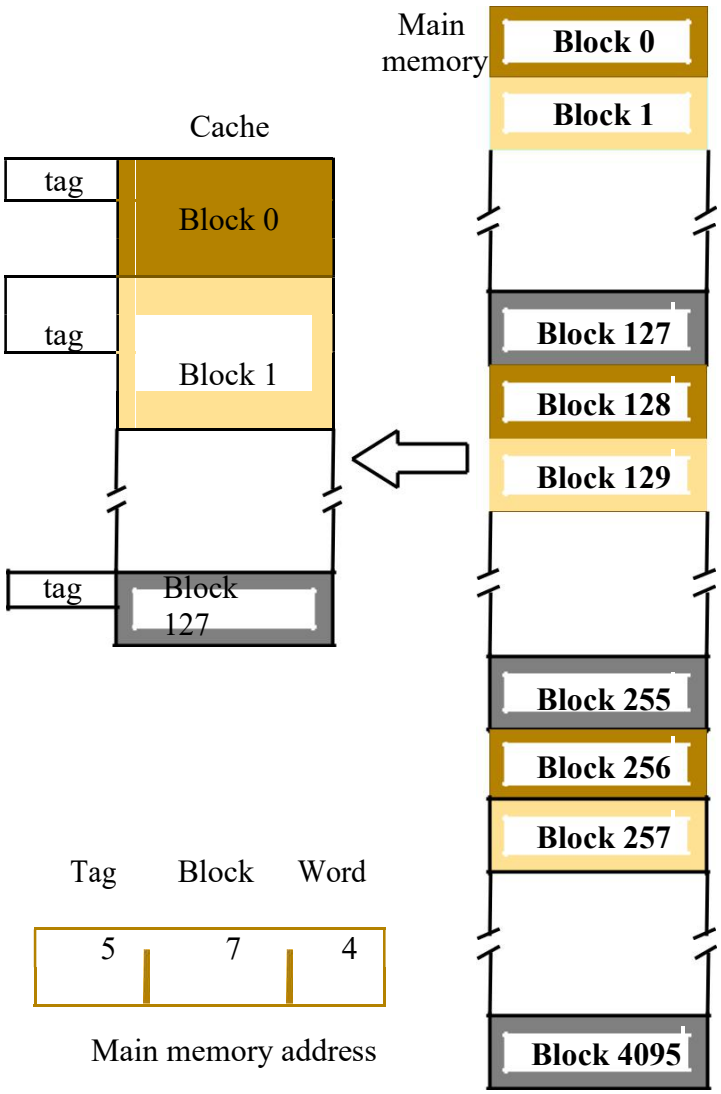
1. Direct mapping
2. Associative mapping
3. Set-associative mapping.

Direct Mapping

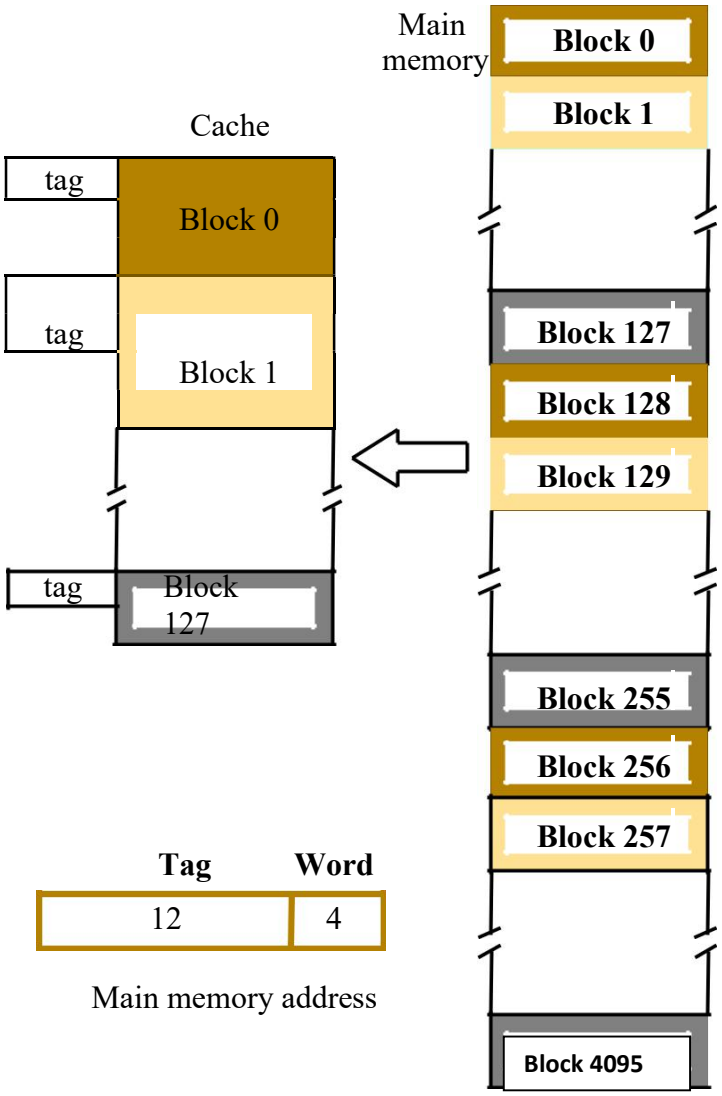
- Block j of the main memory maps to j modulo 128 of the cache. 0 maps to 0, 129 maps to 1.
- More than one memory block is mapped onto the same position in the cache.
- May lead to contention for cache blocks even if the cache is not full.
- Resolve the contention by allowing new block to replace the old block, using one of the replacement algorithms.
- 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 which cache block this new block is placed in.
  - High order 5 bits determine which of the possible 32 blocks is currently present in the cache. These are tag bits.
- Simple to implement but not very flexible leads to contention problem.

Refer to ppt if required
For Example Memory Address 1110111111111100 gets mapped to cache as follows:

- Tag: 11101
  - Identify which of the 32 blocks that are resident in the cache
- Block: 1111111=127, in the 127^{th} block of the cache
- Word:1100=12, the 12^{th} word of the 127^{th} block in the cache


Main memory
Block 0
Block 1
Block 127
Block 128
Block 129
Block 255
Block 256
Block 257
Block 4095
Cache
tag
Block 0
tag
Block 1
tag
Block 127
Tag
Block
Word
5
7
4
Main memory address

Associative Mapping



•Main memory block can be placed into any cache position.

•Memory address is divided into two fields:

- Low order 4 bits identify the word within a block.
- High order 12 bits or tag bits identify a memory block when it is resident in the cache.

•Flexible, and uses cache space efficiently.

•Replacement algorithms can be used to replace an existing block in the cache when the cache is full.

•Cost is higher than direct-mapped cache because of the need to search all 128 patterns to determine whether a given block is in the cache.

For Example Address: 1110111111111100

Mapped as

Tag: 111011111111

Word:1100=12, the 12th word of a block in the cacl

Advantage:

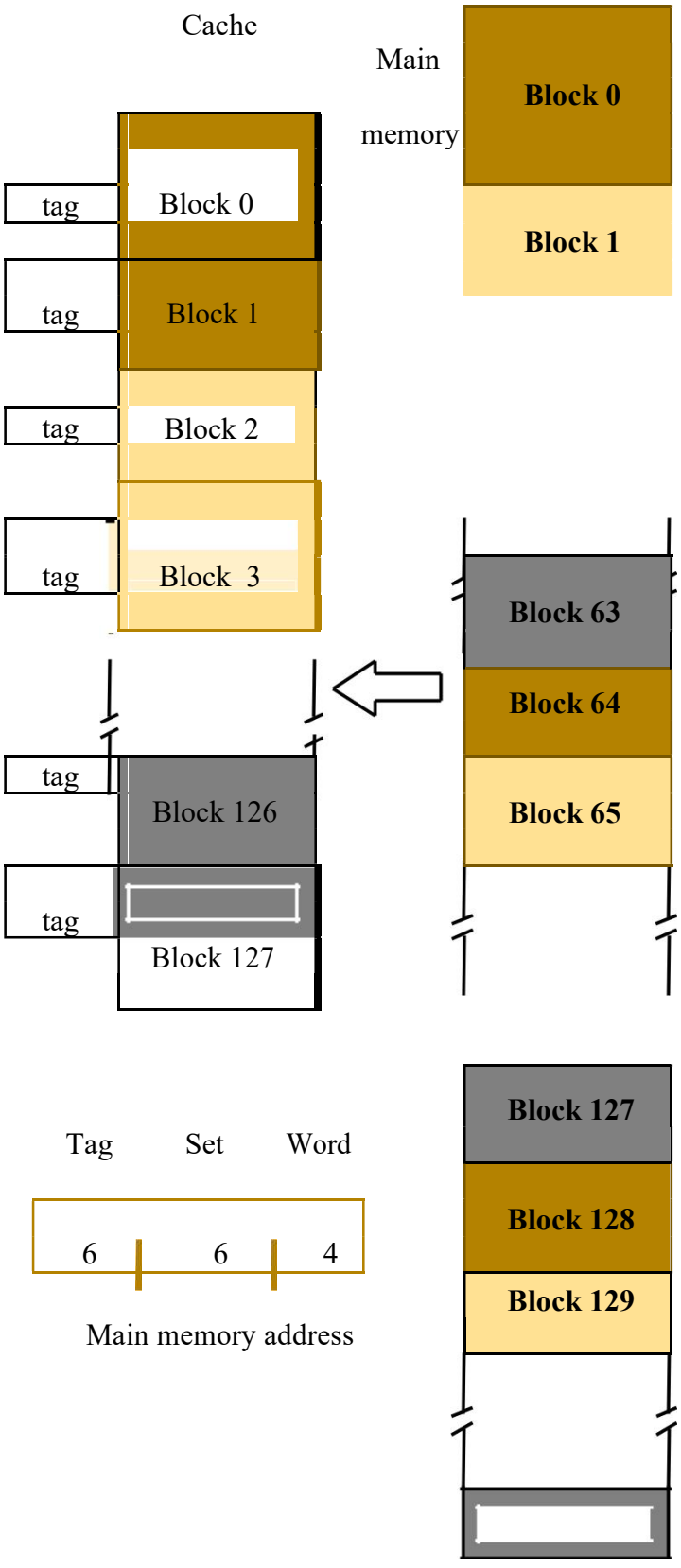
Complete freedom in choosing the cache location

Cache space more efficiently used

Disadvantage: Need to search all 128 *tags.*

Set Associative Mapping (For Diagram Refer to Text book)

Block 409



- Blocks of cache are grouped into sets. Mapping function allows a block of the main memory to reside in any block of a specific set.
- Divide the cache into 64 sets, with two blocks per set. Memory block 0, 64, 128 etc. map to block 0, and they can occupy either of the two positions. Memory address is divided into three fields:
- 6 bit field determines the set number.
- High order 6 bit fields are compared to the tag fields of the two blocks in a set.
- Set-associative mapping combination of direct and associative mapping.
- Number of blocks per set is a design parameter.
- One extreme is to have all the blocks in one set, requiring no set bits (fully associative mapping).
- Other extreme is to have one block per set, is the same as direct mapping.

- **Advantage**

- Reduce contention compared to Direct Mapping
- Reduce number of searches compared to Associative Mapping
  For Example Address 1110111111111100
  Mapped to cache as
  Tag: 111011
  Set: 111111=63, in the 63th set of the cache
  Word:1100=12, the 12th word of the 63th set in the cache

Replacement Algorithm

In a direct-mapped cache, the position of each block is fixed, hence no replacement strategy exists.

In associative and set-associative caches, when a new block is to be brought into the cache and all the Positions that it may occupy are full, the cache controller must decide which of the old blocks to overwrite. This is important issue because the decision can be factor in system performance.

The objective is to keep blocks in the cache that are likely to be referenced in the near future. Its not easy to determine which blocks are about to be referenced. The property of locality of reference gives a clue to a reasonable strategy.

First In First Out: When a block is to be over written, it is sensible to overwrite the one that was been referenced first. This block is called the First in block that will be first to go out.

For Example refer to ppt.

Least Recently Used: When a block is to be over written, it is sensible to overwrite the one that has gone the longest time without being referenced. This block is called the least recently used (LRU) block, and technique is called the LRU Replacement algorithm. The LRU algorithm has been used extensively for many access patterns, but it can lead to poor performance in some cases. For example, it produces disappointing results when accesses are made to sequential elements of an array that is slightly too large to fit into the cache. Performance of LRU algorithm can be improved by introducing a small amount of randomness in deciding which block to replace.