USN										

Internal Assessment Test 2 – January 2022

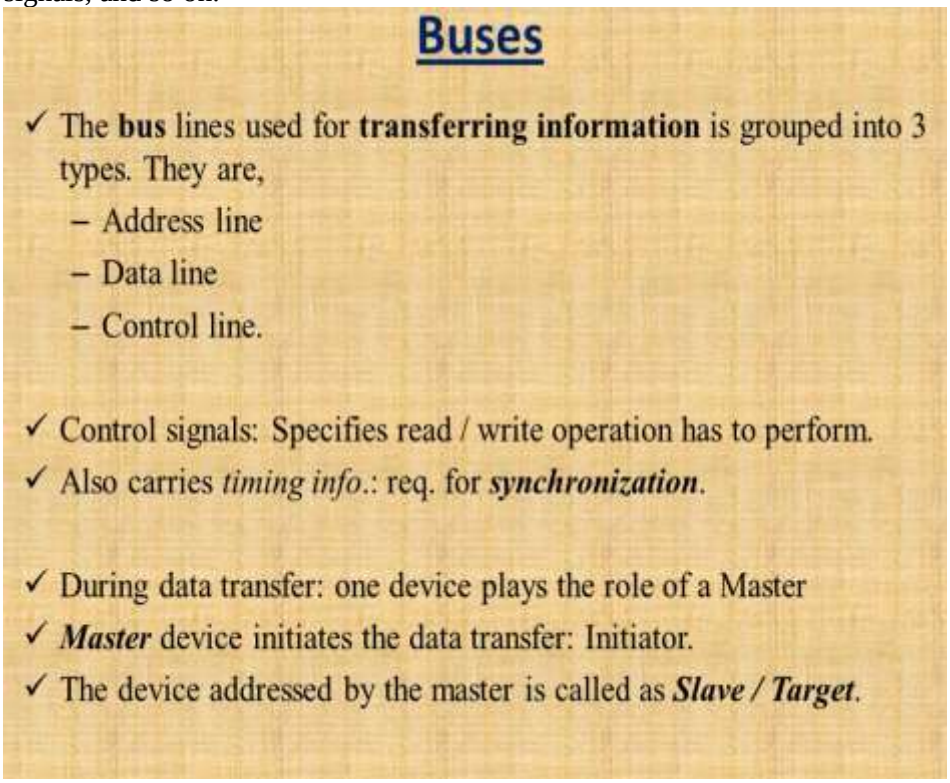
Sub:	COMPUTER ORGANIZATION					Sub Code:	18CS34	Branch:	ISE	
Date:	25/1/2022	Duration:	90 min's	Max Marks:	50	Sem/Sec:	III / A,B & C		OBE	

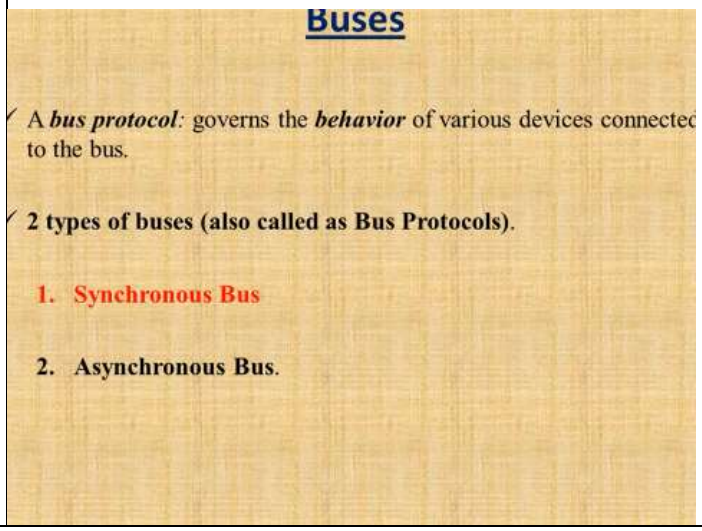
Answer any FIVE FULL Questions

		MARKS	CO	RBT
1.	a. Discuss in detail about the following bus types with timing diagram. i) Synchronous Bus ii) Asynchronous Bus	[10]	CO2	L2

Solution:

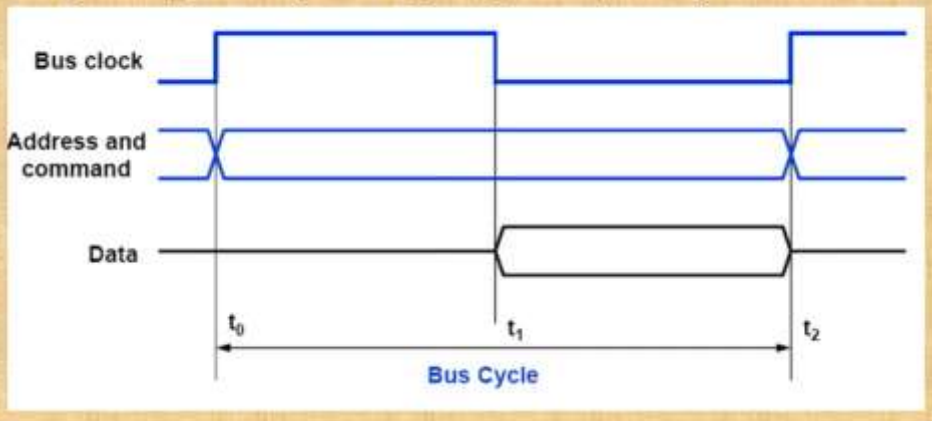
- A bus protocol is the set of rules that govern the behavior of various devices connected to the bus as to when to place information on the bus, assert control signals, and so on.



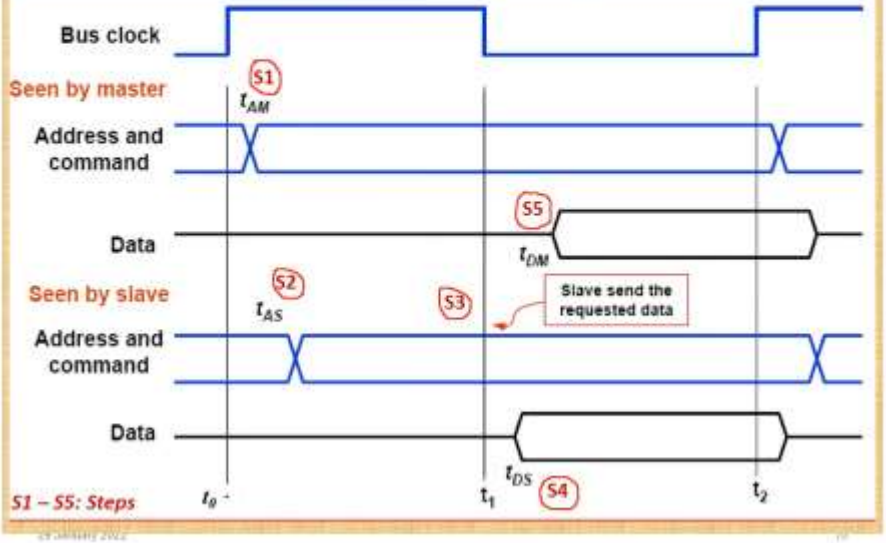


Synchronous Buses

- ✓ All devices derive timing information from a **common clock line**.
- ✓ Equally spaced pulses on clock line define **equal time**.
- ✓ During a "bus cycle", one data transfer take place: *Theoretically*
- ✓ Fig: Timing of an ***input transfer (operation)*** of a synchronous bus.

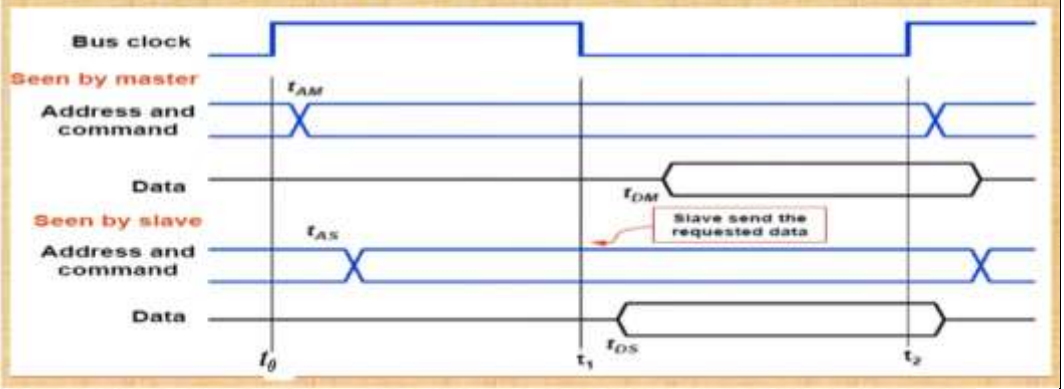


Detailed timing diagram for the input transfer: SB



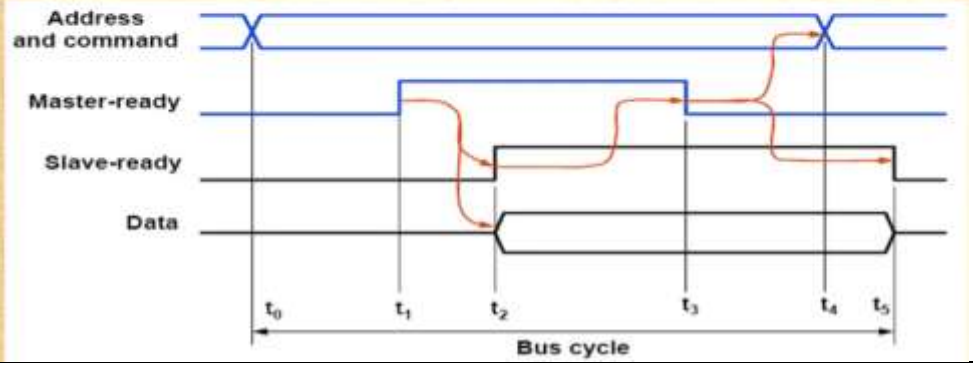
Detailed timing diagram for the input transfer: SB

- ✓ TD: two views of the signal except the clock.: Master & Salve.
- ✓ The master sends the address & cmd. signals: at t_0.
  - But actually appear on the bus at t_{AM}.
- ✓ At t_{AS} the signals reach the ***slave***.
- ✓ Slave *decodes* address at t_1 & *sends* the requested *data* at t_{DS}.
- ✓ Master sees data: t_{DM} & *strobes* the data into its i/p buffer at t_2.

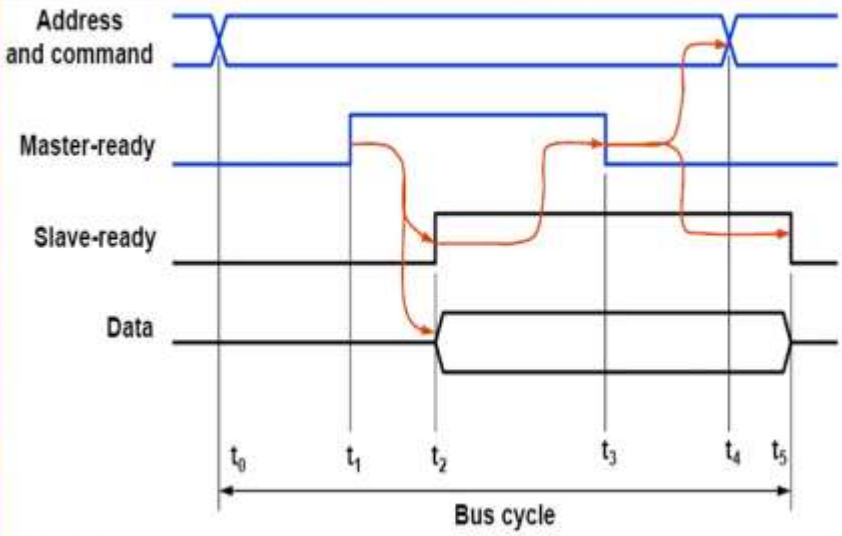


Asynchronous Buses

- ✓ Alternate scheme for **controlling data transfer on the bus**.
- ✓ The bus uses of "**handshake**" between **Master** & the **Slave**.
- ✓ The common clock is replaced by two timing control lines.
  - Master–ready
  - Slave-ready.
- Fig: Handshake control of data transfer during an **input operation**



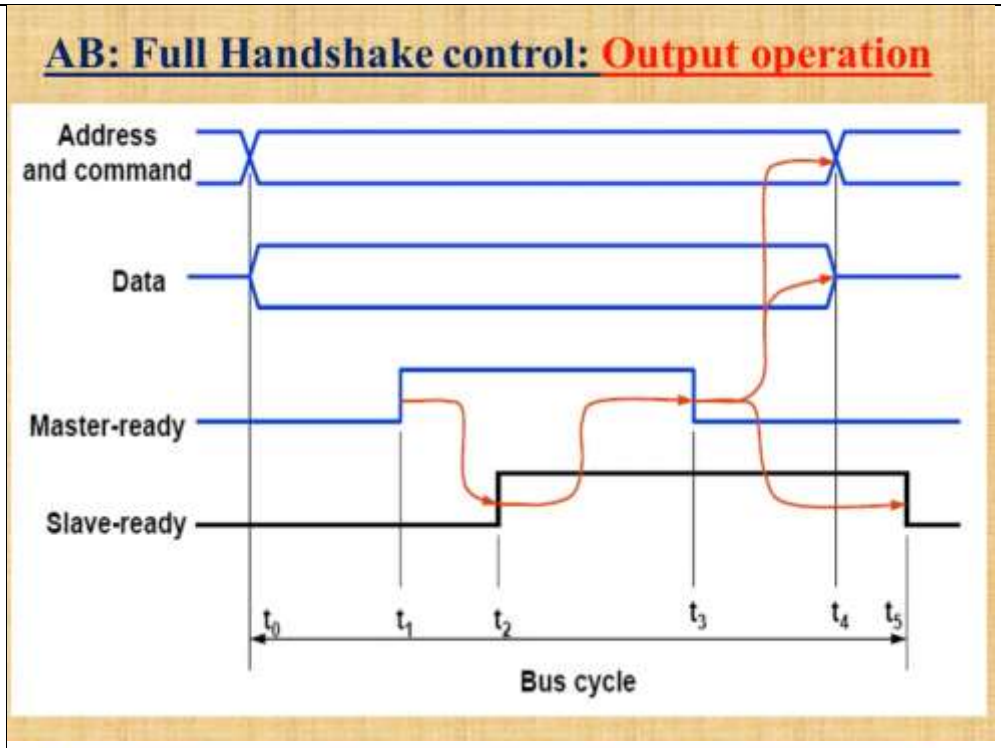
AB: Full Handshake control: Input operation

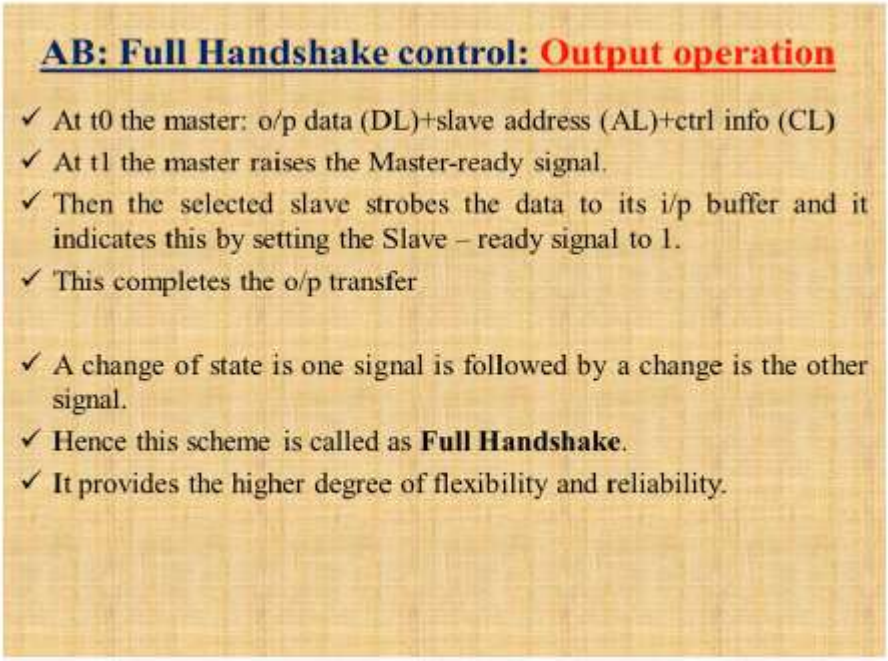


29 January 2022 18

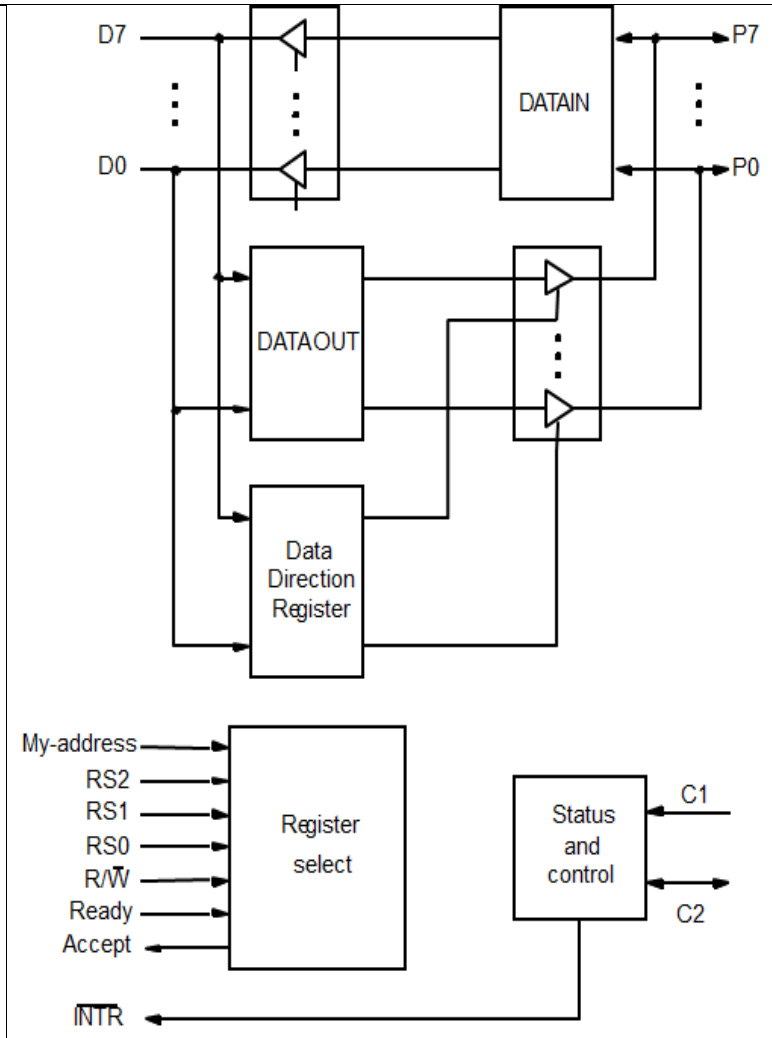
AB: Full Handshake control: Input operation

- ✓ t0: The master places the address and command information on the bus.
- ✓ t1: The master sets the Master-ready line=1: inform the I/O devices.
  - The delay t1 – t0 is intended to allow for any *skew* on the bus.
- ✓ t2: Slave puts the data on the data lines & sets Slave-ready=1.
- ✓ t3: Slave-ready signal arrives at the master: it strobes the data & remove the Master–ready signal.
- ✓ t4: The master removes the address and command information on the bus.
- ✓ t5: It removes the data and the Slave–ready signal from the bus.
- ✓ This completes the i/p transfer.





2.	a. Provide circuit diagram for parallel interface and write its functionality. **Solution:** **A General 8-Bit Parallel Interface** The processor can write any 8-bit pattern into Data Direction Register (DDR). For a given bit, if DDR value is 1, the corresponding data lines act as an output line; otherwise, it acts as an input line.	[5]	CO2	L2



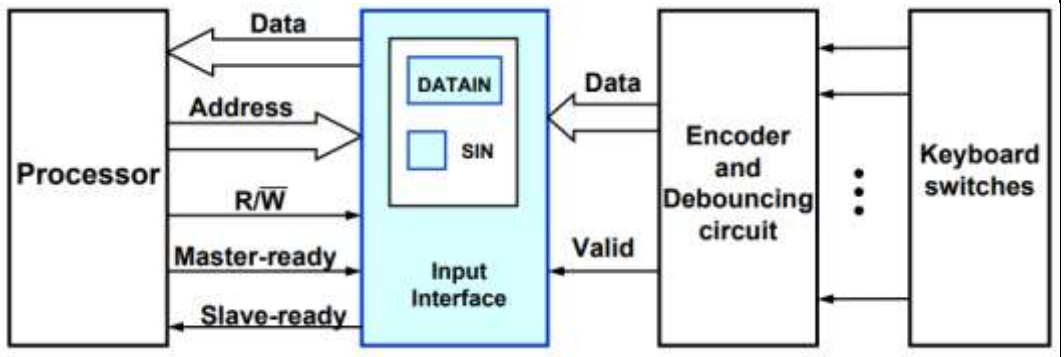
- *Data lines to I/O device are bidirectional.*
- *Data lines P7 through P0 can be used for both input, and output.*
- *In fact, some lines can be used for input & some for output depending on the pattern in the Data Direction Register (DDR).*
- *Processor places an 8-bit pattern into a DDR.*
- *If a given bit position in the DDR is 1, the corresponding data line acts as an output line, otherwise it acts as an input line.*
- *C1 and C2 control the interaction between the interface circuit and the I/O devices.*
- *Ready and Accept lines are the handshake control lines on the processor bus side, and are connected to Master-ready & Slave-ready.*
- *Input signal My-address is connected to the output of an address decoder.*
- *Three register select lines (RS2, RS1, RS0) that allow up to 8 registers to be selected.*

	b. Draw and explain keyboard input interface with block diagram.	[5]	CO2	L2

Solution:

Keyboard to processor connection

- When a key is pressed, the Valid signal changes from 0 o 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.

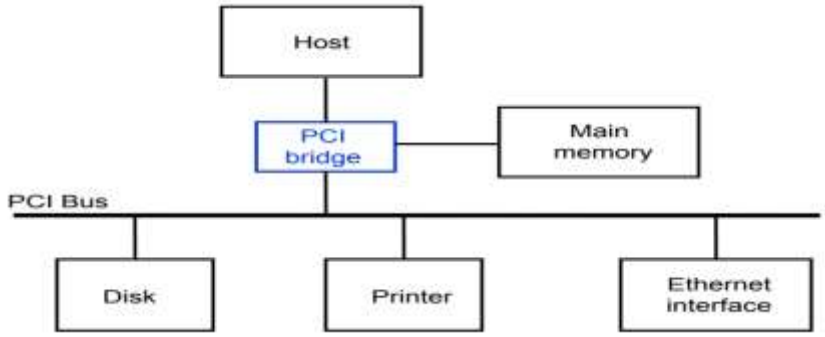


3.	a. Explain about PCI bus features, block diagram for host connection, Data transfer signals and Device configurations.	[10]	CO2	L2

Solution:

Use of a PCI bus in a computer system

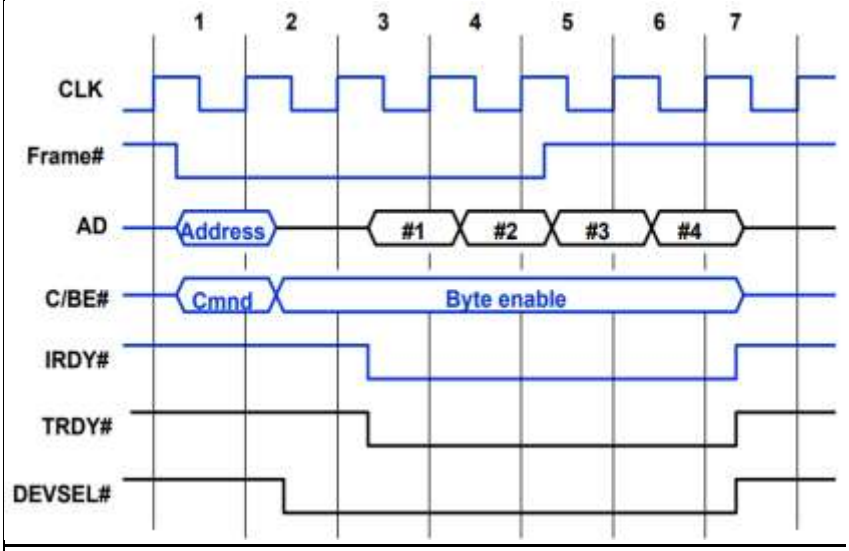
- PCI is developed as a low cost bus that is truly processor independent.
- PCI supports high speed disk, graphics and video devices.
- PCI has plug and play capability for connecting I/O devices.
- To connect new devices, the user simply connects the device interface board to the bus.



- The bus support three independent address spaces: memory, I/O, and configuration.
- The I/O address space is intended for use with processors, such Pentium, that have a separate I/O address space.
- However, the system designer may choose to use memory-mapped I/O even when a separate I/O address space is available
- The configuration space is intended to give the PCI its plug-and-play capability. ‹
  - A 4-bit command that accompanies the address identifies which of the three spaces is being used in a given data transfer operation

Name	Function
CLK	A 33-MHz or 66MHz clock
FRAME#	Sent by the initiator to indicate the duration of a transaction
AD	32 address/data lines, which may be optionally increased to 64
C/BE#	4 command/byte-enable lines (8 for 64-bit bus)
IRDY#, TRDY#	Initiator-ready and Target-ready signals
DEVSEL#	A response from the device indicating that it has recognized its Address and is ready for a data transfer transaction
IDSEL#	Initialization Device Select

A Read Operation on the PCI Bus



4. a. With a neat diagram, explain the organization of 16*8 memory chip. [7] CO4 L3

Solution:

Internal organization of 16*8 memory chips

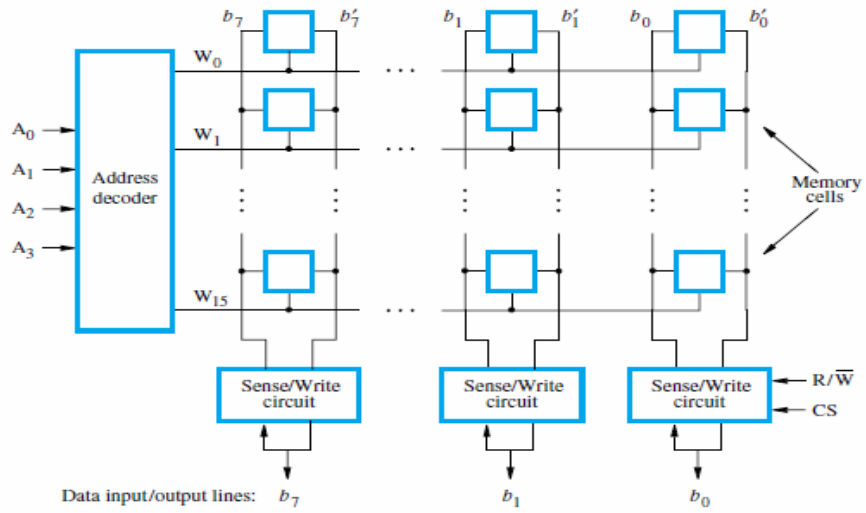
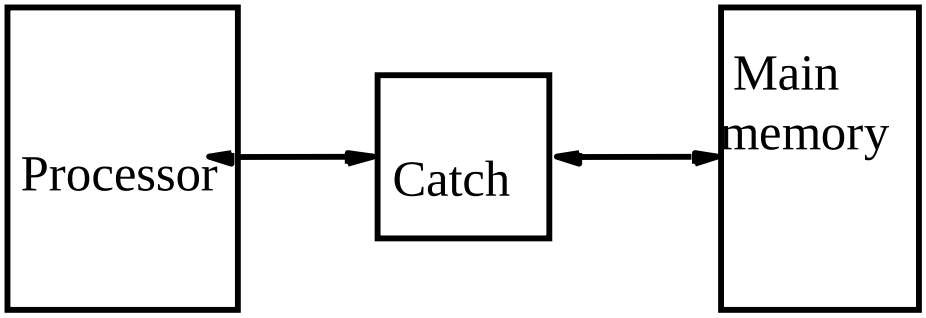


Figure 8.2 Organization of bit cells in a memory chip.

- Each memory cell can hold one bit of information.
- Memory cells are organized in the form of an array.
- One row is one memory word.
- All cells of a row are connected to a common line, known as the "word line".
- Word line is connected to the address decoder.

	• Sense/write circuits are connected to the data input/output lines of the memory chip.			
	b. Differentiate between SRAM and DRAM. **Solution:** • **Static RAMs (SRAMs):** ◦ Consist of circuits that are capable of retaining their state as long as the power is applied. ◦ Volatile memories, because their contents are lost when power is interrupted. ◦ Access times of static RAMs are in the range of few nanoseconds. ◦ However, the cost is usually high. • **Dynamic RAMs (DRAMs):** ◦ Do not retain their state indefinitely. ◦ Contents must be periodically refreshed. ◦ Contents may be refreshed while accessing them for reading.	[3]	CO4	L3
5.	a. What is Cache memory? Consider a cache consisting of 128 blocks of 16 words each, for total of 2048(2K) words and assume that the main memory is addressable by 16 bit address. Main memory is 64K which will be viewed as 4K blocks of 16 words each. Explain different cache memory mapping functions with the help of diagrams. **Solution:** ▪ **C**ache memory is an architectural arrangement which makes the main memory appear faster to the processor than it really is.  • *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".* **(i) Direct mapping**	[10]	CO4	L3

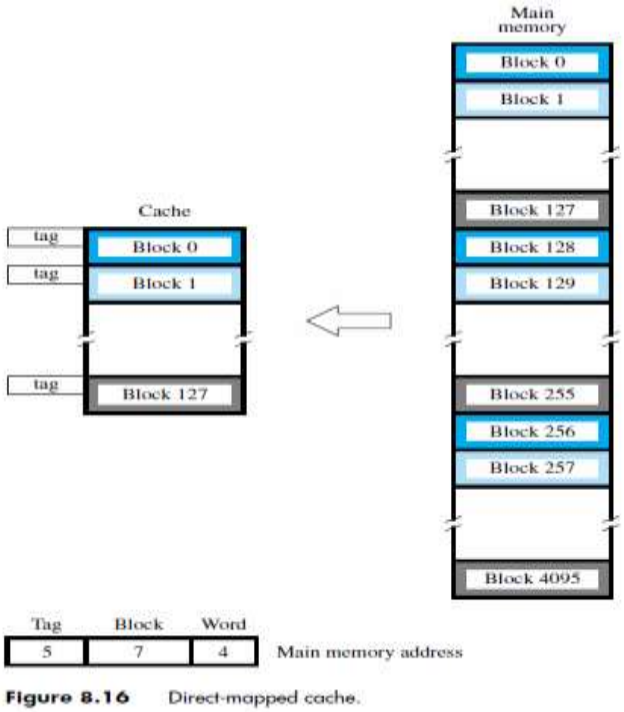


Figure 8.16 Direct-mapped cache.

- 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, leading to a trivial replacement algorithm.
- The placement of a block in the catch is determined from the memory address.
- 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.

(ii) <u>Associative mapping</u>

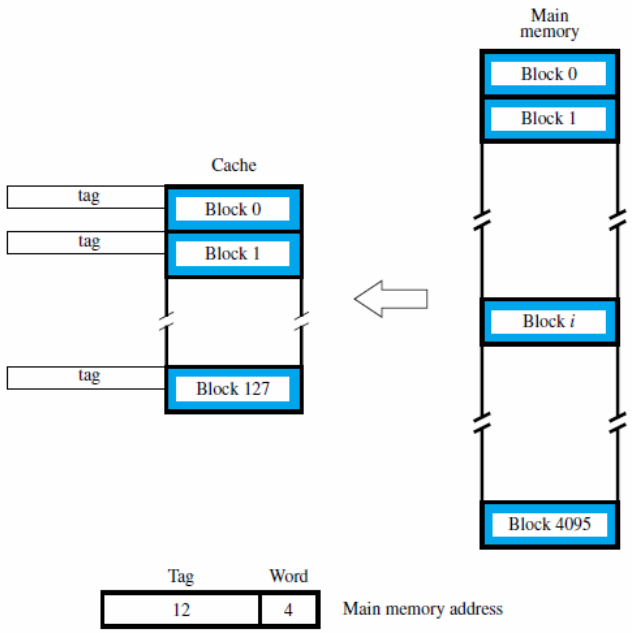


Figure 8.17 Associative-mapped cache.

- 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 lock 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.

Advantage of associative mapping:

There is flexibility when mapping a block to any block of the cache

Disadvantages of associative mapping:

1. A replacement algorithm must be used to determine which block of cache to swap out
2. More space is needed for the tag field
3. The most important disadvantage is the complex circuitry needed to examine all of the tags in parallel in the cache

Set-Associative mapping

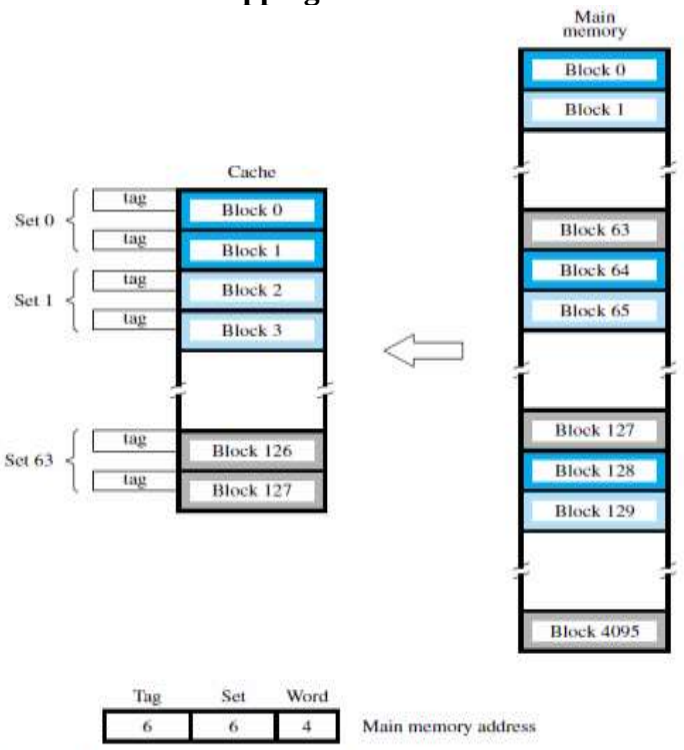
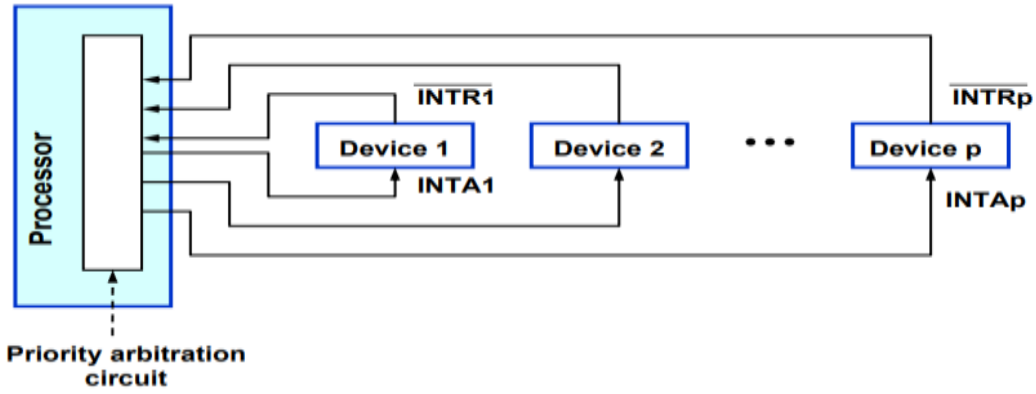
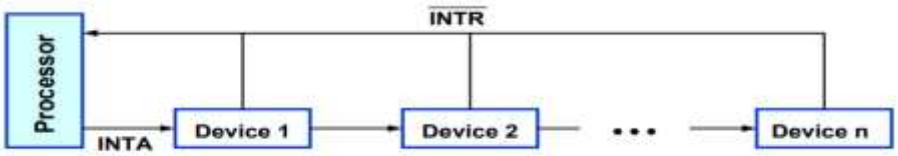


Figure 8.18 Set-associative-mapped cache with two blocks per set.

- 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.			
6.	• Explain the following methods of handling interrupts from multiple devices. (i) Interrupt Nesting/Priority Structure ii)Daisy Chain Method **Solution** **Handling Multiple Devices** • How can the processor recognize the device requesting an interrupt? • Given that different devices are likely to require different interrupt-service routines, how can the processor obtain the starting address of the appropriate routine 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? **Interrupt Nesting: Hardware Priority:- Multiple Requests over multiple interrupt request line** • Each device has a separate interrupt-request and interrupt-acknowledge line. • Each interrupt-request line is assigned a different priority level. • Interrupt requests received over these lines are sent to a priority arbitration circuit in the processor. • If the interrupt request has a higher priority level than the priority of the processor, then the request is accepted.  **INTR: Interrupt Request from Device to Processor** **INTA: Interrupt Acknowledgement From Processor to Device** **Daisy Chain** • A daisy chain is an interconnection of computer devices, peripherals, or network nodes in series, one after another. It is the computer equivalent of a series electrical circuit. **Daisy chain scheme:** • Devices are connected to form a daisy chain. • Devices share the interrupt-request line, and interrupt-acknowledge line is connected to form a daisy chain. • When devices raise an interrupt request, the interrupt-request line is activated. • The processor in response activates interrupt-acknowledge. • Received by device 1, if device 1 does not need service, it passes the signal to device 2. • Device that is electrically closest to the processor has the highest priority.	[10]	CO2	L2



General Structure: Combining Daisy Chain and Priority Groups

- When I/O devices were organized into a priority structure, each device had its own interrupt-request and interrupt-acknowledge line.
- When I/O devices were organized in a daisy chain fashion, the devices shared an interrupt-request line, and the interrupt-acknowledge propagated through the devices.
- A combination of priority structure and daisy chain scheme can also used.
- Devices are organized into groups.
- Each group is assigned a different priority level.
- All the devices within a single group share an interrupt-request line, and are connected to form a daisy chain.