1. What is interfacing?

Interface is the path for communication between two components.

Interfacing is of two types:

1. memory interfacing
2. I/O interfacing.

- Memory Interfacing

When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory. For this, both the memory and the microprocessor requires some signals to read from and write to registers.

The interfacing process includes some key factors to match with the memory requirements and microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.

- I/O Interfacing

There are various communication devices like the keyboard, mouse, printer, etc. So, we need to interface the keyboard and other devices with the microprocessor by using latches and buffers. This type of interfacing is known as I/O interfacing.

2. Direct Access Memory (DMA)

Direct memory access:

Direct memory access (DMA) is a feature of computer systems that allows certain hardware subsystems to access main system memory (random-access memory) independent of the central processing unit (CPU).

Without DMA, when the CPU is using programmed input/output, it is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work. With DMA, the CPU first initiates the transfer, then it does other operations while the transfer is in progress, and it finally receives an interrupt from the DMA controller (DMAC) when the operation is done. This feature is useful at any time that the CPU cannot keep up with the rate of data transfer, or when the CPU needs to perform work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards. DMA is also used for intra-chip data transfer in multi-core processors. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without DMA channels. Similarly, a processing element inside a multi-core processor can transfer data to and from its local memory without occupying its processor time, allowing computation and data transfer to proceed in parallel.

DMA can also be used for "memory to memory" copying or moving of data within memory. DMA can offload expensive memory operations, such as large copies or scatter-gather operations, from the CPU to a dedicated DMA engine. An implementation example is the I/O Acceleration Technology. DMA is of interest in network-on-chip and in-memory computing architectures.

Principles:

- **Third-party**

Standard DMA, also called third-party DMA, uses a DMA controller. A DMA controller can generate memory addresses and initiate memory read or write cycles. It contains several hardware registers that can be written and read by the CPU. These include a memory address register, a byte count register, and one or more control registers. Depending on what features the DMA controller provides, these control registers might specify some combination of the source, the destination, the direction of the transfer (reading from the I/O device or writing to the I/O device), the size of the transfer unit, and/or the number of bytes to transfer in one burst.

To carry out an input, output or memory-to-memory operation, the host processor initializes the DMA controller with a count of the number of words to transfer, and the memory address to use. The CPU then commands the peripheral device to initiate a data transfer. The DMA controller then provides addresses and read/write control lines to the system memory. Each time a byte of data is ready to be transferred between the peripheral device and memory, the DMA controller increments its internal address register until the full block of data is transferred.

- **Bus mastering**

In a bus mastering system, also known as a first-party DMA system, the CPU and peripherals can each be granted control of the memory bus. Where a peripheral can become a bus master, it can directly write to system memory without the involvement of the CPU, providing memory address and control signals as required. Some measures must be provided to put the processor into a hold condition so that bus contention does not occur

- ✓ **Modes of operation :**

1) Burst mode
2) Cycle stealing mode
3) Transparent mode

3. Explain the functions of Handshake signals.

In telecommunications, a handshake is an automated process of negotiation between two participants (example "Alice and Bob") through the exchange of information that establishes the protocols of a communication link at the start of the communication, before full communication begins. The handshaking process usually takes place in order to establish rules for communication when a computer attempts to communicate with another device. Signals are usually exchanged between two devices to establish a communication link. For example, when a computer communicates with another device such as a modem, the two devices will signal each other that they are switched on and ready to work, as well as to agree to which protocols are being used

Handshaking can negotiate parameters that are acceptable to equipment and systems at both ends of the communication channel, including information transfer rate, coding alphabet, parity, interrupt procedure, and other protocol or hardware features. Handshaking is a technique of communication between two entities. However, within

TCP/IP RFCs, the term "handshake" is most commonly used to reference the TCP three-way handshake. For example, the term "handshake" is not present in RFCs covering FTP or SMTP. One exception is Transport Layer Security, TLS, setup, FTP RFC 4217. In place of the term "handshake", FTP RFC 3659 substitutes the term "conversation" for the passing of commands.

4. List the operating modes of the 8155A programmable device

8155A microprocessor operating modes

- MODE 0: IN THIS MODE, TIMER GIVES ONLY ONE CYCLE OF SQUARE WAVE, THE OUTPUT REMAINS HIGH FOR 1/2 COUNT AND REMAIN S LOW FOR 1/2 COUNT. IF COUNT IS ODD IT REMAINS HIGH FOR (N+1)/2 AND LOW FOR (N-1)/2. WHERE N IS COUNT VALUE. WAVE WIDTH DEPENDS ON TWO FACTOR: ONE IS INPUT CLOCK PULSE FREQUENCY, AND THE OTHER IS COUNT LOADED IN COUNTER.
- MODE 1: THIS MODE IS SIMILAR TO SINGLE SQUARE WAVE IN OPERATION BUT THE WHEN COUNTER BECOMES ZERO, THE COUNT VALUE IS AUTOMATICALLY RELOADED. THUS IT PROVIDES CONTINUOUS SQUARE WAVE.
- MODE 2: THIS MODE GIVES A SINGLE CLOCK PULSE AS A OUTPUT OF THE END OF THE COUNT THE OUTPUT IS HIGH NORMALLY, BUT IT BECOMES LOW FOR 1 CLOCK PULSE AND AGAIN IT WILL BECOME HIGH AND REMAIN HIGH.
- MODE 3: THIS MODE IS SIMILAR TO MODE 2 BUT WHEN THE COUNTER BECOMES ZERO THE COUNT VALUE IS AUTOMATICALLY RELOADED. THUS IT PROVIDES CONTINUOUS PULSES.

5. Write a note on any of the applications of microprocessors.

Microprocessor Applications

A microprocessor makes daily life easier because of its low cost, low power, small weight, and vast application in every field. There are several applications of microprocessors. Some of the important applications are:

- Household Devices
- Industrial Applications of Microprocessors
- Transportation Industry
- Computers and Electronics
- In Medicals
- Instrumentation
- Entertainment
- Embedded Systems at Home
- Office Automation and Publication
- Communication

6. How many address lines are necessary on the chip of 2K byte memory?

11 address lines

It is 2k in size. **11 address lines** are needed to address all the addresses inside the EPROM. A similar calculation reveals that the 2K RAM also needs **11 address lines**. The PIO chip only has 4 bytes inside, so it only needs **2 address lines**.

7. If the memory chip size is 1024 X 4 bits, how many chips are required to make up 2K bytes of memory

You need 4 Chips

Explanation:

One Byte is 8 bits, so two 1024 x 4 chips would make 1KB memory

8. The memory map of a 4K byte memory chip begins at the location 2000 H. Specify the address of the last location on The chip and the number of pages on the chip

Assuming by 4K you mean 4096. Assuming 2000H means hexadecimal 0x2000. Assuming its byte addressable.

What is the specific address of the last location on the chip?

4096 = 0x1000.

0x2000 + 0x1000 = 0x3000

Minus 1 because it starts at 0, so the answer is 0x2fff

The number of pages on the chip?

Not enough information, how big is a page? Assuming the chip is page aligned.

If a page is 4096 bytes, then 4096 bytes is 1 page.

If a page is 2048 bytes, then 4096 bytes is 2 pages.

If a page is 1024 bytes, then 4096 bytes is 4 pages.

………

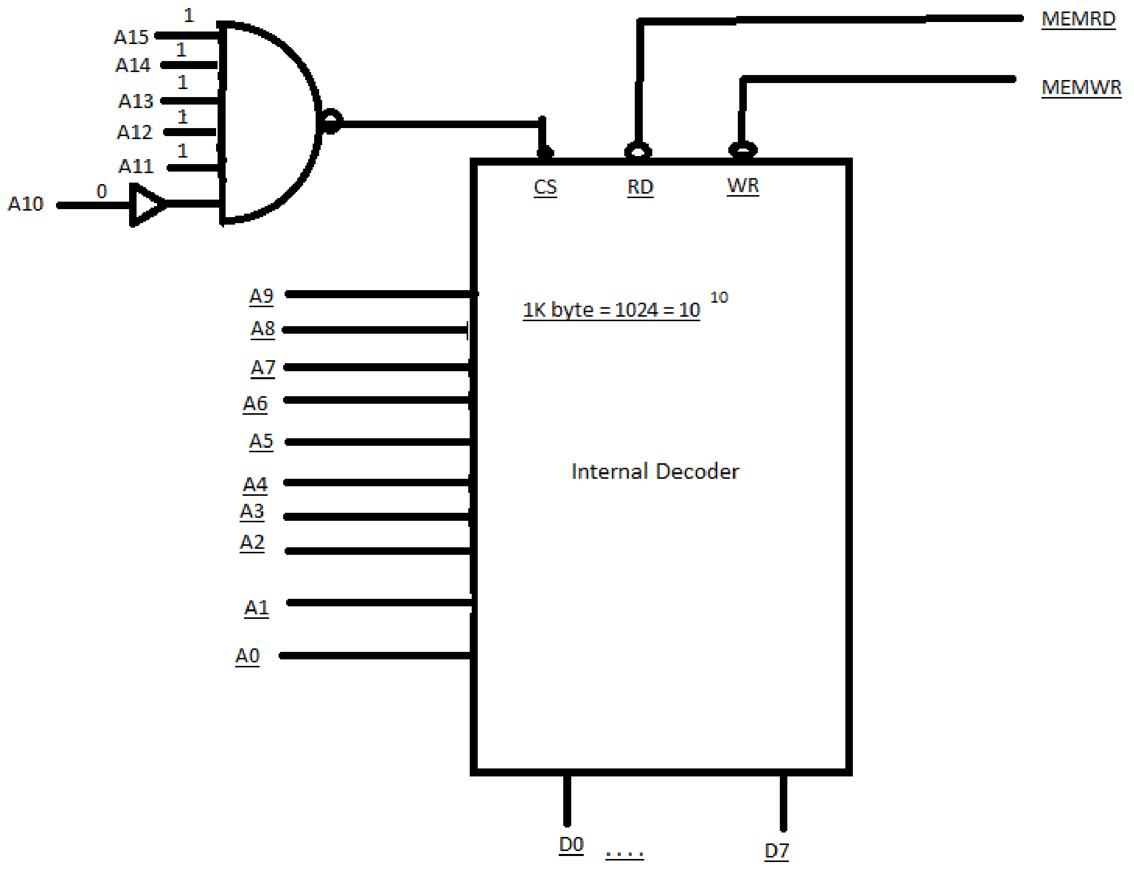
If a page is 1 byte, then 4096 bytes is 4096 pages.

9. The memory address of the last location of an 8K byte memory chip is FFFF H. Find the starting address.

The starting address of an 8K byte memory chip that ends at FFFFH is E000H.

8K is 8192 (8 * 1024) which is 2000H. Subtract 2000H from FFFFH, and add 1, and you get E000H.

10. The memory address of the last location of a 1 K byte memory chip is given as FBFF H. Specify the memory map



Address Range :

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
						1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

F800 H to FBFF H

This is the memory map of 1k bytes memory chip which last location FBFF H