USN

Internal Assessment Test II – JULY 2022

Sub:	MICROCONTROLLER AND EMBEDDED SYSTEM			Sub Code: 18CS44		Branch: CSE	
Date:	08/08/22	Duration:	90 mins Max Marks: 50	Sem / Sec: IV Sem A/B/C		OBE	
			Answer any FIVE FULL Questions		MARKS	CO	RBT

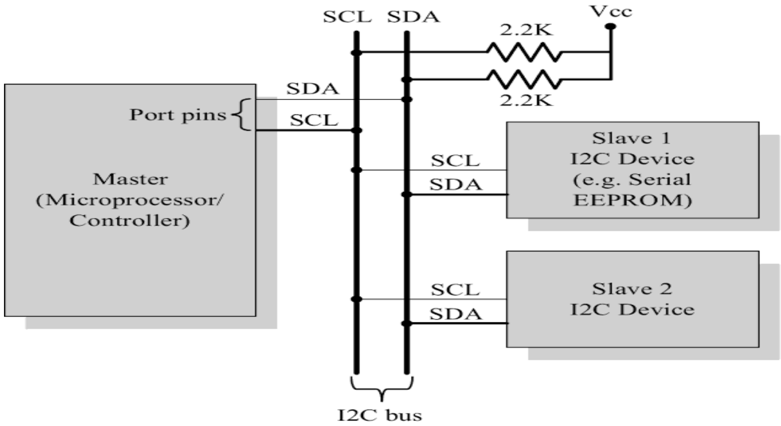
1 **With a neat interface diagram, illustrate the connection of Master-Slave devices on I^2C Bus.** [10] CO1 L1

Answer:

• The Inter Integrated Circuit Bus (I2C Pronounced 'I square C') is a synchronous bi-directional half duplex two wire serial interface bus. • (Half duplex - one-directional communication at a given point of time) • The concept of I2C bus was developed by Philips Semiconductors in the early 1980s. • The original intention of I2C was to provide an easy way of connection between a microprocessor/microcontroller system and the peripheral chips in television sets. • The I2C bus comprise of two buslines: • Serial Clock (SCL line) – responsible for generating synchronization clock pulses • Serial Data (SDA line) – responsible for transmitting the serial data across devices.

• I2C bus is a shared bus system to which many number of I2C devices can be connected. • Devices connected to the I2C bus can act as either 'Master' or'Slave'. • The 'Master' device is responsible for controlling the communication by initiating/terminating data transfer, sending data and generating necessary synchronization clock pulses. • 'Slave' devices wait for the commands from the master and respondupon receiving the commands. • 'Master' and 'Slave' devices can act as either transmitter or receiver. • Regardless whether a master is acting as transmitter or receiver, the synchronization clock signal is generated by the 'Master' device only. • I2C supports multi masters on the same bus.

The following bus interface diagram illustrates the connection of master and slave devices on the I2C bus.



The sequence of operations for communicating with an I2C slave device is listed below:

1. The master device pulls the clock line (SCL) of the bus to 'HIGH'
2. The master device pulls the data line (SDA) 'LOW', when the SCL line is at logic 'HIGH' (This is the 'Start' condition for data transfer)
3. The master device sends the address (7 bit or 10 bit wide) of the 'slave' device to which it wants to communicate, over the SDA line.

◦ Clock pulses are generated at the SCL line for synchronizing the bit reception by the slave device.

◦ The MSB of the data is always transmitted first.

◦ The data in the bus is valid during the 'HIGH' period of the clock signal

The master device sends the Read or Write bit (Bit value = 1 Read operation; Bit value = 0 Write operation) according to the requirement

	5. The master device waits for the acknowledgement bit from the slave device whose address is sent on the bus along with the Read/ Write operation command. ◦ Slave devices connected to the bus compares the address received with the address assigned to them 6. The slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value 1) over the SDA line 7. Upon receiving the acknowledge bit, the Master device sends the 8 bit data to the slave device over SDA line, if the requested operation is 'Write to device'. ◦ If the requested operation is 'Read from device', the slave device sends data to the master over the SDA line 8. The master device waits for the acknowledgement bit from the device upon byte transfer complete for a write operation and sends an acknowledge bit to the Slave device for a read operation 9. The master device terminates the transfer by pulling the SDA line 'HIGH' when the clock line SCL is at logic 'HIGH' (Indicating the 'STOP' condition) I2C bus supports three different data rates: • Standard mode (Data rate up to 100kbits/sec (100 kbps)) • Fast mode (Data rate up to 400kbits/sec (400 kbps)) • High speed mode (Data rate up to 3.4Mbits/sec (3.4 Mbps))			
2	**A. Explain the difference between Microprocessor and Microcontroller** **Answer:**	[5+5]	CO1	L2,L3

Microprocessor	Microcontroller
A silicon chip representing a central processing unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of instructions	A microcontroller is a highly integrated chip that contains a CPU, scratchpad RAM, special and general purpose register arrays, on chip ROM/ FLASH memory for program storage, timer and interrupt control units and dedicated I/O ports
It is a dependent unit. It requires the combination of other chips like timers, program and data memory chips, interrupt controllers, etc. for functioning	It is a self-contained unit and it doesn't require external interrupt controller, timer, UART, etc. for its functioning
Most of the time, general purpose in design and operation	Mostly application-oriented or domain-specific
Doesn't contain a built in I/O port. The I/O port functionality needs to be implemented with the help of external programmable peripheral interface chips like 8255	Most of the processors contain multiple built-in I/O ports which can be operated as a single 8 or 16 or 32 bit port or as individual port pins
Targeted for high end market where performance is important	Targeted for embedded market where performance is not so critical (At present this demarcation is invalid)
Limited power saving options compared to microcontrollers	Includes lot of power saving features

B. Explain Little Endian and Big Endian Architecture with suitable example.

Answer:

Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system.

• Suppose the word length is two byte then data can be stored in memory in two different ways:

1. Higher order of data byte at the higher memory and lower order of data byte at location just below the higher memory – Little-Endian
   ◦ E.g.: Intel x86 Processors
2. Lower order of data byte at the higher memory and higher order of data byte at location just below the higher memory – Big-Endian
   ◦ E.g.: Motorola 68000 Series Processors

Little-endian means the lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address. (The little end comes first.)

• For example, a 4 byte long integer Byte3 Byte2 Byte1 Byte0 will be stored in the memory as shown below:

Base Address + 0	Byte 0	Byte 0	0×20000 (Base Address)
Base Address + 1	Byte 1	Byte 1	0×20001 (Base Address + 1)
Base Address + 2	Byte 2	Byte 2	0×20002 (Base Address + 2)
Base Address + 3	Byte 3	Byte 3	0×20003 (Base Address + 3)

Big-endian means the higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address. (The big end comes first.)

• For example, a 4 byte long integer Byte3 Byte2 Byte1 Byte0 will be stored in the memory as shown below:

Base Address + 0	Byte 3	Byte 3	0×20000 (Base Address)
Base Address + 1	Byte 2	Byte 2	0×20001 (Base Address + 1)
Base Address + 2	Byte 1	Byte 1	0×20002 (Base Address + 2)
Base Address + 3	Byte 0	Byte 0	0×20003 (Base Address + 3)

3 **Explain the role of different types of memories used in the embedded system.** [10] CO1, L2

Answer:

Memory is an important part of a processor/controller based embedded systems.

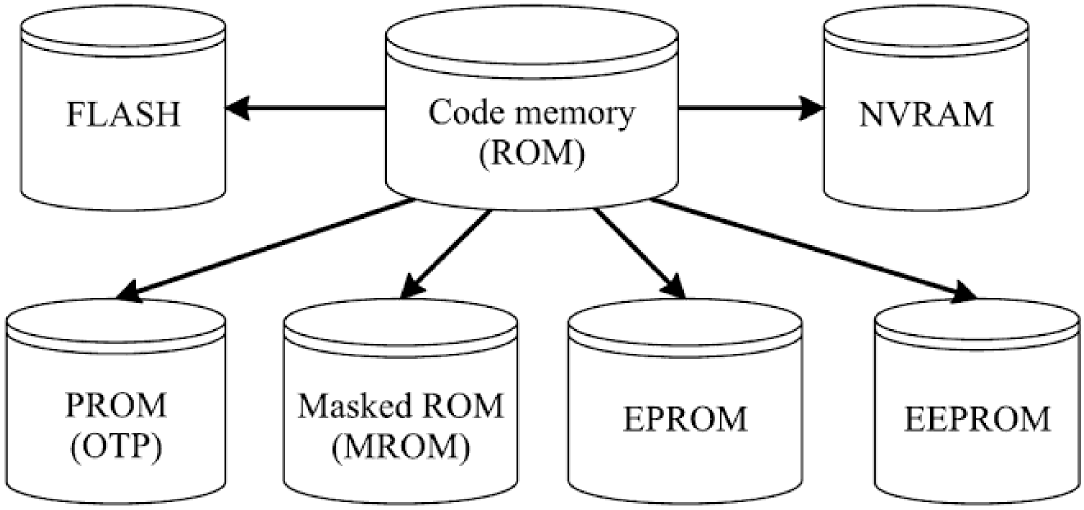
• Some of the processors/controllers contain built in memory and this memory is referred as on-chip memory.

• Others do not contain any memory inside the chip and requires external memory to be connected with the controller/processor to store the control algorithm. It is called off-chip memory.

• Also some working memory is required for holding data temporarily during certain operations.

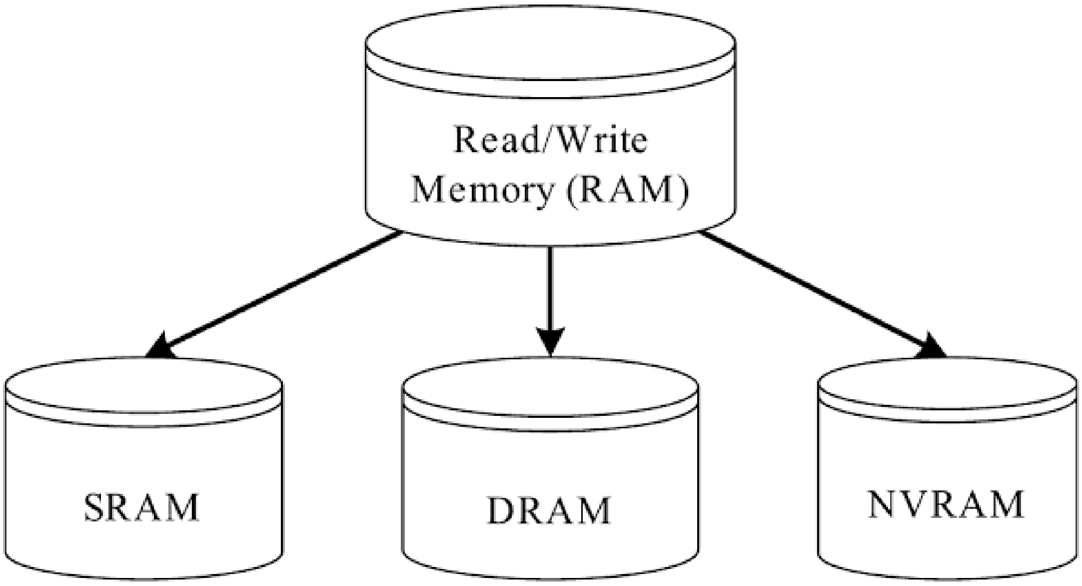
The program memory or code storage memory of an embedded system stores the program instructions.

• The code memory retains its contents even after the power is turned off. It is generally known as non-volatile storage memory.
• It can be classified into different types as shown:

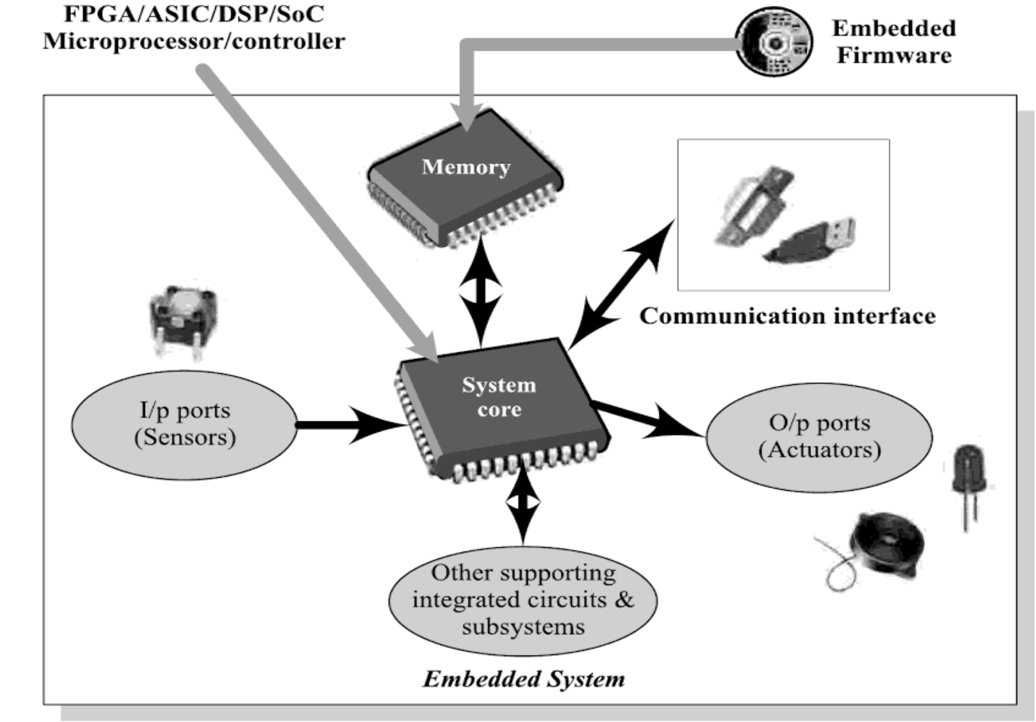


Masked ROM is a one-time programmable device.
• Masked ROM makes use of the hardwired technology for storing data.
• The device is factory programmed by masking and metallisation process at the time of production itself, according to the data provided by the end user.
• Advantage – low cost for high volume production.
• Limitation - inability to modify the device firmware against firmware upgrades.
• Since the MROM is permanent in bit storage, it is not possible to alter the bit information. the manufacturer.
• The end user is responsible for programming these devices.
• This memory has nichrome or polysilicon wires arranged in a matrix. These wires can be functionally viewed as fuses.
• It is programmed by a PROM programmer which selectively burns the fuses according to the bit pattern to be stored.
• Fuses which are not blown/burned represents a logic “1" whereas fuses which are blown/burned represents a logic 0 .
• The default state is logic "1".
• OTP is widely used for commercial production of embedded systems whose prototyped versions are proven and the code is finalised.
• It is a low cost solution for commercial production.
• OTPs cannot be reprogrammed.
Erasable Programmable Read Only Memory (EPROM) gives the flexibility to re-program the same chip.
• EPROM stores the bit information by charging the floating gate of an FET.
• Bit information is stored by using an EPROM programmer, which applies high voltage to charge the floating gate.
• EPROM contains a quartz crystal window for erasing the stored information.
• If the window is exposed to ultraviolet rays for a fixed duration, the entire memory will be erased.
• Even though the EPROM chip is flexible in terms of re-programmability, it needs to be taken out of the circuit board and put in a UV eraser device for 20 to 30 minutes.
• It is a tedious and time-consuming process.
The information contained in the EEPROM memory can be altered by using electrical signals at the register/byte level.
• They can be erased and reprogrammed in-circuit.
• These chips include a chip erase mode and in this mode they can be erased in a few milliseconds.
• It provides greater flexibility for system design.
• The only limitation is their capacity is limited (a few kilobytes) when compared with the standard ROM.

FLASH memory is a variation of EEPROM technology – It combines the re-programmability of EEPROM and the high capacity of standard ROMs.
• FLASH is the latest ROM technology.
• Most popular ROM technology used in today's embedded designs.
• FLASH memory is organised as sectors (blocks) or pages.
• FLASH memory stores information in an array of floating gate MOSFET transistors.
• The erasing of memory can be done at sector level or page level without affecting the other sectors or pages.
• Each sector/page should be erased before re-programming.
• The typical erasable capacity of FLASH is 1000 cycles.
• E.g.: W27C512 from WINBOND is an example of 64KB FLASH memory.
Non-volatile RAM is a random access memory with battery backup.
• It contains static RAM based memory and a minute battery for providing supply to the memory in the absence of extemal power supply.
• The memory and battery are packed together in a single package.
• The life span of NVRAM is expected to be around 10 years.
• E.g.: DS1644 from Maxim/Dallas is an example of 32KB NVRAM.
RAM is the data memory or working memory of the controller/processor.
• Controller/processor can read from it and write to it.
• RAM is volatile – when the power is turned off, all the contents are destroyed.
• RAM is a direct access memory – we can access the desired memory location directly without the need for traversing through the entire memory locations to reach the desired memory position (i.e. random access of memory location).
• This is in contrast to the Sequential Access Memory (SAM), where the desired memory location is accessed by either traversing through the entire memory or through a 'seek' method. Magnetic tapes, CD ROMs, etc. are examples of sequential access memories.
RAM generally falls into three categories: Static RAM (SRAM), Dynamic RAM (DRAM) and Non-Volatile RAM (NVRAM).

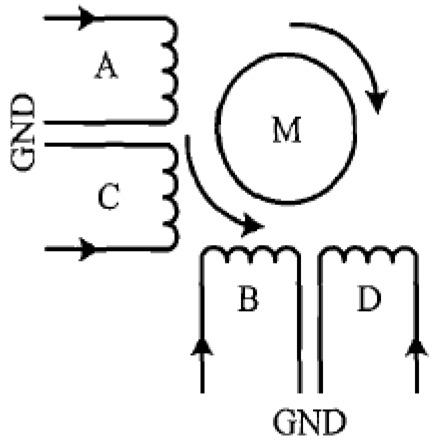


\Static RAM stores data in the form of voltage.
• They are made up of flip-flops.
• Static RAM is the fastest form of RAM available.
• Fast due to its resistive networking and switching capabilities.
• In typical implementation, an SRAM cell (bit) is realised using six transistors (or 6 MOSFETs).
• Four of the transistors are used for building the latch (flip-flop) part of the memory cell and two for controlling the access.
Dynamic RAM stores data in the form of charge.
• They are made up of MOS transistor gates.
• Advantages – high density and low cost compared to SRAM.
• Disadvantage – since the information is stored as charge it gets leaked off with time and to prevent this they need to be refreshed periodically.

	• Special circuits called DRAM controllers are used for the refreshing operation. • The refresh operation is done periodically in milliseconds interval.			
4	**With neat block diagram, explain the elements of the embedded system.** **Answer:**  *Real World* A typical embedded system contains a single chip controller, which acts as the master brain of the system. The controller can be microprocessor or a microcontroller or a digital signal processor or an application specific integrated circuit (ASIC). Embedded hardware/software systems are basically designed to regulate a physical variable or to manipulate the state of some device by sending some control signals to actuators or devices connected to the O/P ports of the systems, in response to the input signals provided by the end users on sensors which are connected to the input ports. Hence an embedded system can be viewed as a reactive system. Key boards, push buttons switches etc.. are examples for common user interface input devices whereas LED, LCD.. etc are examples for common user interface output devices for typical embedded system. The memory of the system is responsible for holding the control algorithm and other important configuration details. For most of embedded systems, the memory for storing the algorithm or configuration data is of fixed type, which is a kind of ROM and it is not available for end user for modification. The most common types of memories used in embedded systems for control algorithm storage are PROM, UVEPROM, EEPROM and FLASH. An embedded system without a control algorithm implemented memory is just like newborn baby. It is having all the peripherals but is not capable of making any decision depending on the situational as well as real world changes. Core of the Embedded System • Embedded systems are domain and application specific and are built around a central core. • The core of the embedded system falls into any one of the following categories: 1. General Purpose and Domain Specific Processors 1.1 Microprocessors 1.2 Microcontrollers 1.3 Digital Signal Processors 2. Application Specific Integrated Circuits (ASICs) 3. Programmable Logic Devices (PLDs) 4. Commercial off-the-shelf Components (COTS)	[10]	CO2	L2

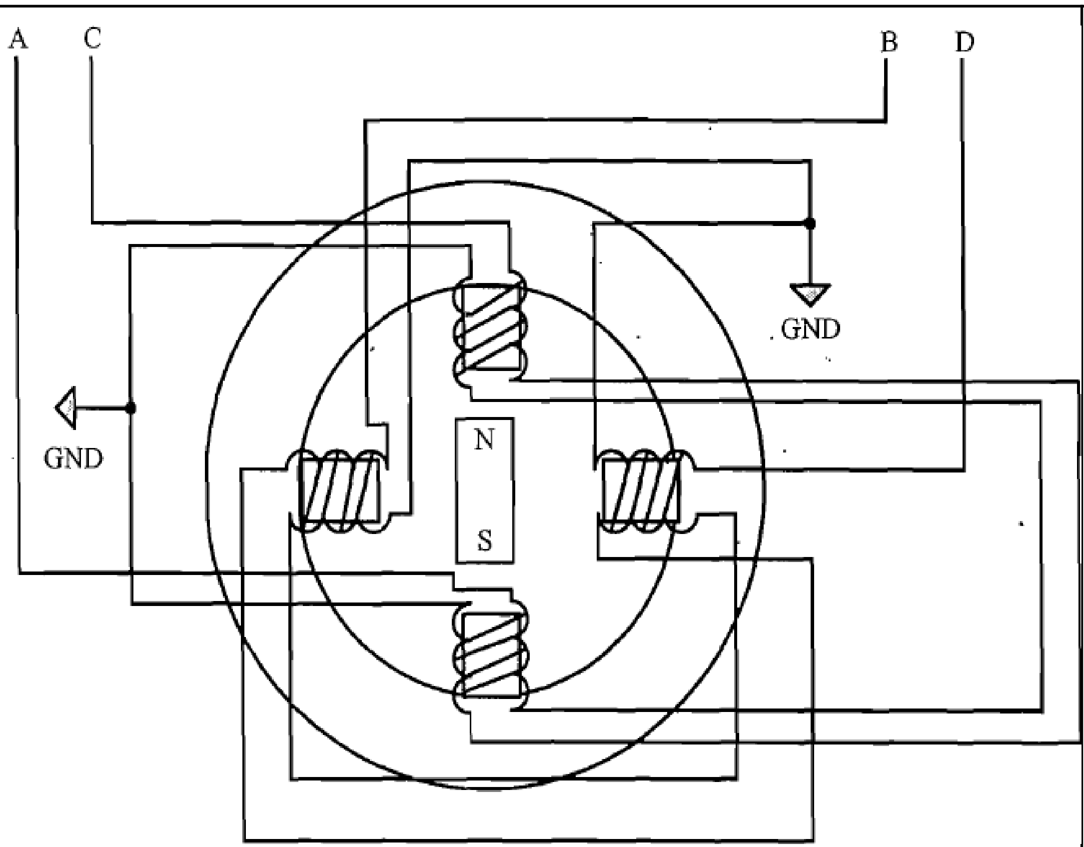
5	**With a neat diagram, explain the interfacing of Stepper Motor through the driver circuit to microcontroller.**	[10]	CO2	L3

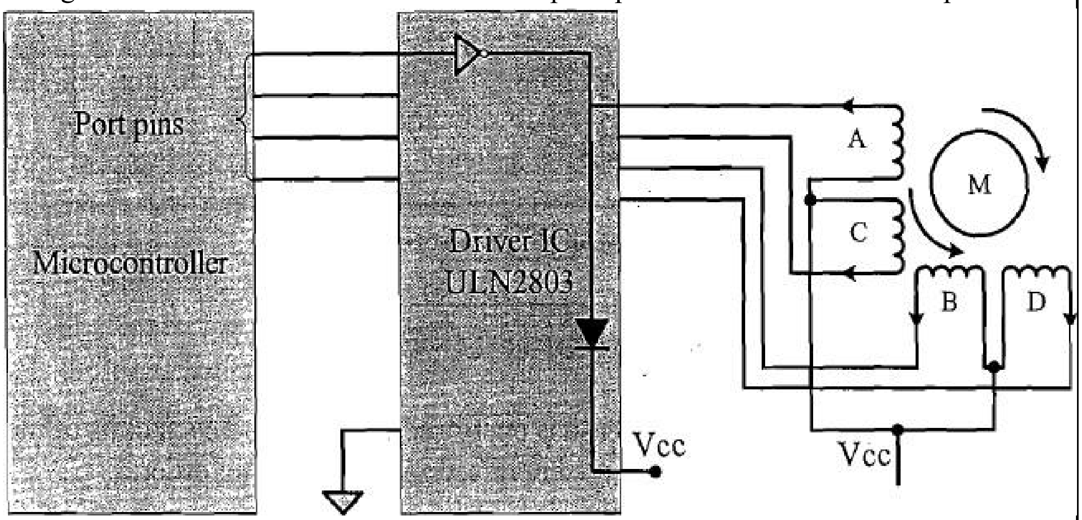
Answer:
A stepper motor is an electro-mechanical device which generates discrete displacement (motion) in response, to the electrical signals.
It differs from the normal DC motor in its operation. The DC motor produces continuous rotation on applying DC voltage whereas a stepper motor produces discrete rotation in response to the DC voltage applied to it.
Stepper motors are widely used in industrial embedded applications, consumer electronic products and robotics control systems.
Based on the coil winding arrangements, a two-phase stepper motor is classified into two.
1. Unipolar
2. Bipolar
1. Unipolar-A unipolar stepper motor contains two windings per phase. The direction of rotation (clockwise or anticlockwise) of a stepper motor is controlled by changing the direction of current flow.
Current in one direction flows through one coil and in the opposite direction flows through the other coil. It is easy to shift the direction of rotation by just switching the terminals to which the coils are connected.
Below figure illustrates the working of a two-phase unipolar stepper motor.
Figure below



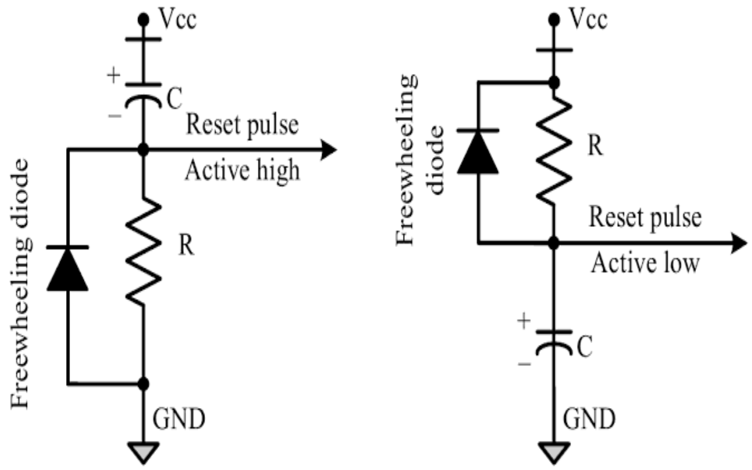
The coils are represented as A, B, C and D. Coils A and C carry current in opposite directions for phase 1. Similarly, B and D carry current in opposite directions for phase 2.

Bipolar- A bipolar stepper motor contains single winding per phase. For reversing the motor rotation the current flow through the windings is reversed dynamically.
The stator winding details for a two phase unipolar stepper motor is shown in below figure.

<table>
<tr>
<td></td>
<td>

The stepping of stepper motor can be implemented in different ways by changing the sequence of activation of the stator windings.
The different stepping modes supported by stepper motor are explained below.
i. Full Step- In the full step mode both the phases are energised simultaneously. The coils A, B, C and Dare energised in the following order:
ii. Wave Step - In the wave step mode only one phase is energised at a time and each coils of the phase is energised alternatively.
iii.Half Step - It uses the combination of wave and full step. It has the highest torque and stability.The following circuit diagram illustrates the interfacing of a stepper motor through a driver circuit connected to the port pins of a microcontroller/processor.

</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td> A. Write short note on i) Reset Circuit and ii) Watch Dog Timer.
Answer:
i) The reset circuit is essential to ensure that the device is not operating at a voltage level where the device is not guaranteed to operate, during system power ON.
• The reset signal brings the internal registers and the different hardware systems of the processor/controller to a known state and starts the firmware execution from the reset vector
• Normally from vector address 0x0000 for conventional processors/controllers. </td>
<td>[6+4]</td>
<td>CO2,CO1</td>
<td>L2, L1</td>
</tr>
</table>

• The reset signal can be either active high or active low.
• Since the processor operation is synchronised to a clock signal, the reset pulse should be wide enough to give time for the clock oscillator to stabilise before the internal reset state starts.
The reset signal to the processor can be applied at power ON through an external passive reset circuit comprising a Capacitor and Resistor or through a standard Reset IC like MAX810 from Maxim Dallas.
• Select the reset IC based on the type of reset signal and logic level (CMOS/TTL) supported by the processor/controller in use.
• Some microprocessors/controllers contain built-in internal reset circuitry and they don't require external reset circuitry.
Figure illustrates a resistor capacitor based passive reset circuit for active high and low configurations.
• The reset pulse width can be adjusted by changing the resistance value R and capacitance value C.



ii)A watchdog timer, or simply a watchdog, is a hardware timer for monitoring the firmware execution and resetting the system processor/microcontroller when the program execution hangs up.
• Depending on the internal implementation, the watchdog timer increments or decrements a free running counter with each clock pulse and generates a reset signal to reset the processor if the count reaches zero for a down counting watchdog, or the highest count value for an up counting watchdog.
If the watchdog counter is in the enabled state, the firmware can write a zero (for up counting watchdog implementation) to it before starting the execution of a piece of code (which is susceptible to execution hang up) and the watchdog will start counting.
• If the firmware execution doesn't complete due to malfunctioning, within the time required by the watchdog to reach the maximum count, the counter will generate a reset pulse and this will reset the processor.
• If the firmware execution completes before the expiration of the watchdog timer you can reset the count by writing a 0 (for an up counting watchdog timer) to the watchdog timer register.
Most of the processors implement watchdog as a built-in component and provides status register to control the watchdog timer (like enabling and disabling watchdog
functioning) and watchdog timer register for writing the count value.
• If the processor/controller doesn't contain a built in watchdog timer, the same can be

implemented using an external watchdog timer IC circuit.

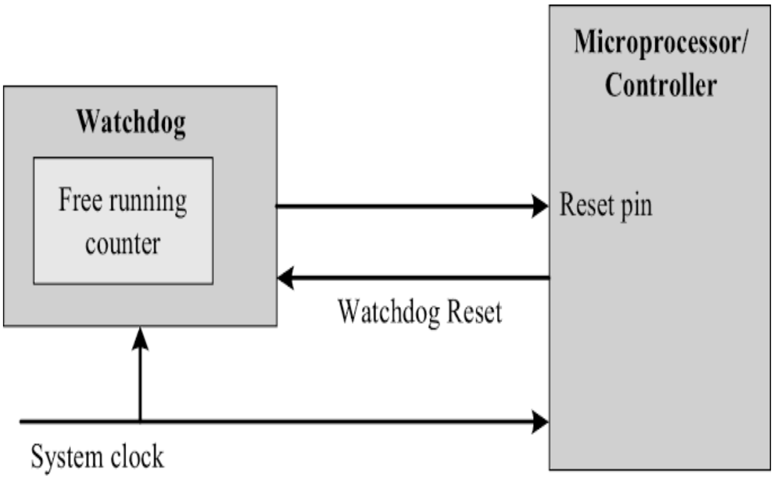
• The external watchdog timer uses hardware logic for enabling/disabling, resetting the
watchdog count, etc. instead of the firmware based 'writing' to the status and watchdog timer register.

• The Microprocessor supervisor IC DS1232 integrates a hardware watchdog timer in it.

• In modern systems running on embedded operating systems, the watchdog can be
implemented in such a way that when a watchdog timeout occurs, an interrupt is generated instead of resetting the processor.

• The interrupt handler for this handles the situation in an appropriate fashion.

Figure illustrates the implementation of an external watchdog timer based microprocessor supervisor circuit for a small scale embedded system.



B. Explain the classification of embedded system based on generation

Answer:

First Generation

• Early embedded systems were built around 8-bit microprocessors like 8085 and Z80 and 4-bit microcontrollers.

• Simple in hardware circuits with firmware developed in assembly code.

• E.g.: Digital telephone keypads, stepper motor control units, etc.

Second Generation

• Embedded systems built around 16-bit microprocessors and 8-bit or 16-bit microcontrollers.

• Instruction set were much more complex and powerful than the first generation.

• Some of the second generation embedded systems contained embedded operating systems for their operation.

• E.g.: Data acquisition systems, SCADA systems, etc.

Third Generation

• Embedded systems built around 32-bit microprocessors and 16-bit microcontrollers.

• Application and domain specific processors/controllers like Digital Signal Processors (DSP) and Application Specific Integrated Circuits (ASICs) came into picture.

• The instruction set of processors became more complex and powerful and the concept of instruction pipelining also evolved.

• Dedicated embedded real time and general purpose operating systems entered into the embedded market.

• Embedded systems spread its ground to areas like robotics, media, industrial process control, networking, etc.

Fourth Generation

	• The advent of System on Chips (SoC), reconfigurable processors and multicore processors are bringing high performance, tight integration and miniaturisation into the embedded device market. • The SoC technique implements a total system on a chip by implementing different functionalities with a processor core on an integrated circuit. • They make use of high performance real time embedded operating systems for their functioning. • E.g.: Smart phone devices, Mobile Internet Devices (MIDs), etc.			