Question Bank

EE3018-Embedded Processsor

DEPARTMENT OF

ELECTRICAL AND ELECTRONICS

ENGINEERING

Question Bank

Unit-1

1. What is an embedded system? What are the components of embedded system?

An embedded system employs a combination of hardware & software (a "computational engine") to perform a specific function; is part of a larger system that may not be a "computer"; works in a reactive and time-constrained environment.

The three main components of an embedded system are

- Hardware
- Main application software
- RTOS

System:

A way of doing one or more tasks according to a program.

2. What is ARM?

ARM (stylised in lowercase as arm, formerly an acronym for Advanced RISC Machines and originally Acorn RISC Machine) is **a family of RISC instruction set architectures (ISAs) for computer processors**

3. What is ALU?

In computing, an arithmetic logic unit (ALU) is **a combinational digital circuit that performs arithmetic and bitwise operations on integer binary numbers.** This is in contrast to a floating-point unit (FPU), which operates on floating point numbers.

4.What are the types of register modes?

1.USER Mode
2.FIQ MODE
3.IRQ MODE
4.SVC MODE
5.UNDEFINED MODE
6.ABORT MODE
7.MONITOR MODE

5.What are the various classifications of embedded systems?

1. Small scale embedded systems
2. Medium scale embedded systems
3. Sophisticated embedded systems

6.What are the two essential units of a processor on an embedded system?

4. Program flow control unit (CU)
5. Execution unit (EU)

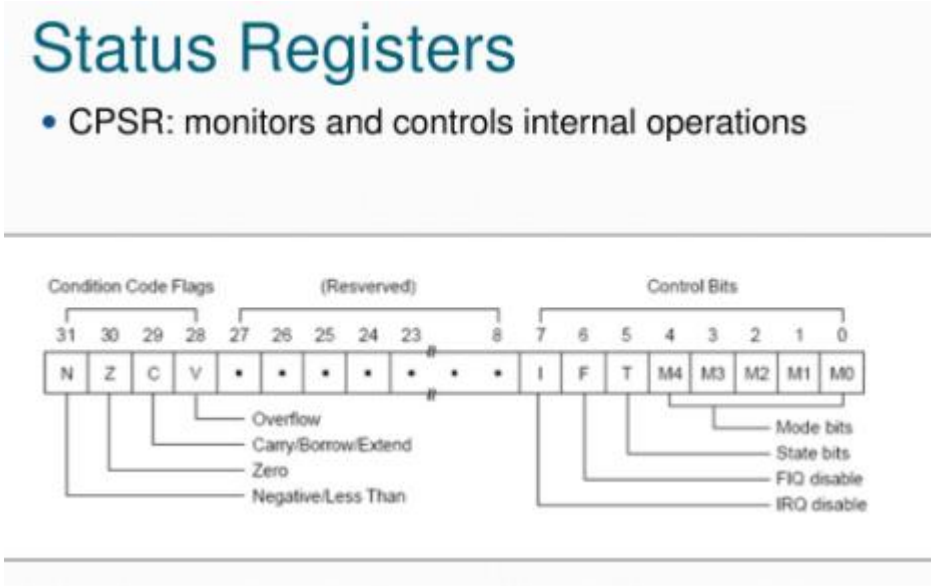
7.Classify the processors in embedded system?

6. General purpose processor
    - Microprocessor
    - Microcontroller
    - Embedded processor
    - Digital signal processor
    - Media processor

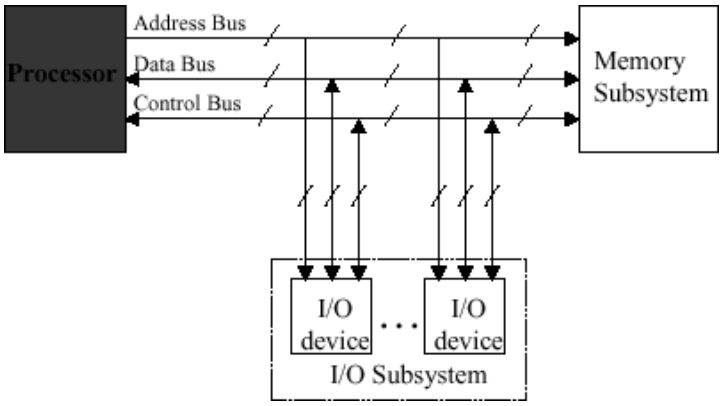
8.What is pipelining?

Pipelining is the process of storing and prioritizing computer instructions that the processor executes. The pipeline is a "logical pipeline" that lets the processor perform an instruction in multiple steps. The processing happens in a continuous, orderly, somewhat overlapped manner.

9.Draw the format of CPSR?



10.Draw the simple view of organization of processor and memory in a system.



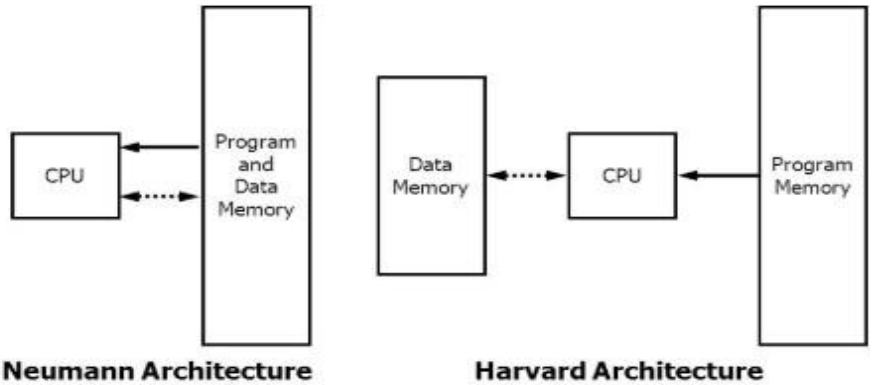
In what ways CISC and RISC processors differ?

CISC	RISC
1. It provides number of addressing modes	It provides very few number of addressing modes
2. It has a micro programmed unit with a control memory	It has a hard wired unit without a control memory
3. An easy compiler design	Complex compiler design
4. Provides precise and intensive calculations slower than a RISC	Provides precise and intensive calculations faster than a RISC

11.Differentiate between timers and counters.

Timer	Counter
The register incremented for every machine cycle.	The register is incremented considering 1 to 0 transition at its corresponding to an external input pin (T0, T1).
Maximum count rate is 1/12 of the oscillator frequency.	Maximum count rate is 1/24 of the oscillator frequency.
A timer uses the frequency of the internal clock, and generates delay.	A counter uses an external signal to count pulses.

12.Draw and compare von-Neumann and Harvard architecture.



von Neumann Architecture **Harvard Architecture**

13.Define interrupt latency? How to avoid it.

The interrupt latency refers to the amount of time taken by a system to respond to an interrupt.

Following factors causes interrupt latency:

- Interrupt disabling
- Longer time taken by higher priority interrupts
- Time taken by processor for bookkeeping
- Time taken by processor for context saving

Methods for avoiding it:

Using multiple arrays for critical data's. , Using circular queues

14.List the important considerations when selecting a processor.

- Instruction set
- Maximum bits in an operand
- Clock frequency
- Processor ability

15.Name some of the hardware parts of embedded systems?

- Power source
- Clock oscillator circuit
- Timers
- Memory units
- DAC and ADC
- LCD and LED displays
- Keyboard/Keypad

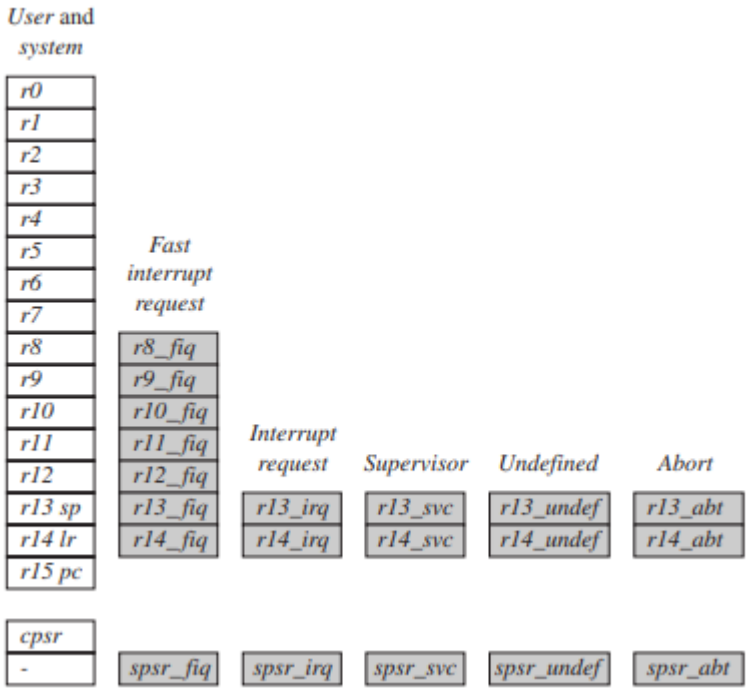
16.What are the various types of memory in embedded systems?

- RAM internal External
- ROM/PROM/EEPROM/Flash
- Cache memory

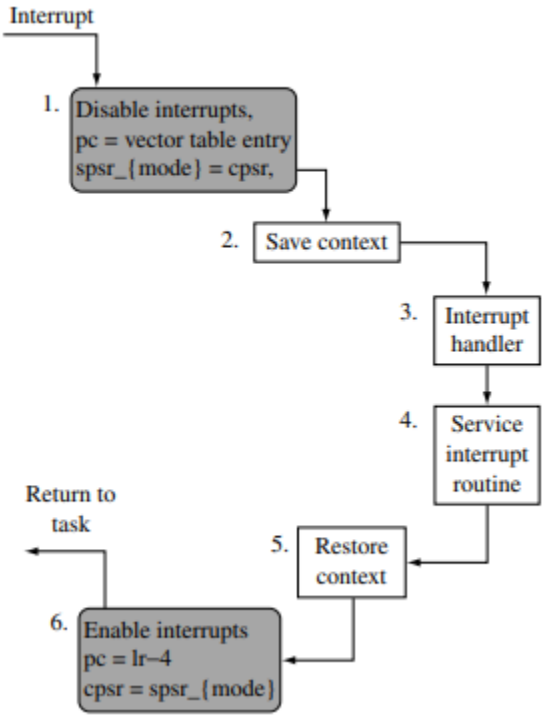
17.What are the important embedded processor chips?

- ARM 7 and ARM 9
- i 960
- AMD 29050

18.Draw the structure of register address modes.



19.Draw the structure of interrupt.



20.What is watch dog timer?

Watch dog timer is a timing device that resets after a predefined timeout.

21.What are the 3 types of operating modes?

- Normal mode
- Idle mode
- Power down mode

21. **What is a coprocessor ?**

- A co-processor is many times referred as a Math Processor. As the coprocessor performs routine mathematical tasks, the core processor is freed up from this computation and its time is saved. By taking specialized processing tasks from core CPU, coprocessor reduces the strain on the main microprocessor, so that it can run at a greater speed.

Unit-2

1. Draw the exception table.

Exception/interrupt	Shorthand	Address	High address
Reset	RESET	0x00000000	0xffff0000
Undefined instruction	UNDEF	0x00000004	0xffff0004
Software interrupt	SWI	0x00000008	0xffff0008
Prefetch abort	PABT	0x0000000c	0xffff000c
Data abort	DABT	0x00000010	0xffff0010
Reserved	—	0x00000014	0xffff0014
Interrupt request	IRQ	0x00000018	0xffff0018
Fast interrupt request	FIQ	0x0000001c	0xffff001c

2. What is Software interrupt vector.

Software interrupt vector is called when you execute a SWI instruction. The SWI instruction is frequently used as the mechanism to invoke an operating system routine.

3. What is Prefetch abort.

Prefetch abort vector occurs when the processor attempts to fetch an instruction from an address without the correct access permissions. The actual abort occurs in the decode stage**.**

4. Define data abort vector.

Data abort vectoris similar to a prefetch abort but is raised when an instruction attempts to access data memory without the correct access permissions

5. Define IRQ.

Interrupt request vector is used by external hardware to interrupt the normal execution flow of the processor. It can only be raised if IRQs are not masked in the cpsr.

6. Write the syntax of data processing instruction.

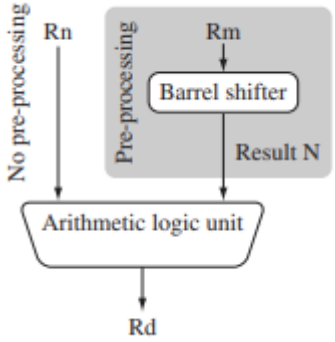
```
Syntax: <instruction>{<cond>}{S} Rd, N
```

MOV	Move a 32-bit value into a register	$Rd = N$
MVN	move the NOT of the 32-bit value into a register	$Rd = \sim N$

7. Define barrel shifter.

Data processing instructions are processed within the arithmetic logic unit (ALU). A unique and powerful feature of the ARM processor is the ability to shift the 32-bit binary pattern in

one of the source registers left or right by a specific number of positions before it enters the ALU. This shift increases the power and flexibility of many data processing operations.



8. Expalin the barrel shifter operation.

Mnemonic	Description	Shift	Result	Shift amount y
LSL	logical shift left	*x*LSL *y*	$x \ll y$	#0–31 or *Rs*
LSR	logical shift right	*x*LSR *y*	(unsigned)$x \gg y$	#1–32 or *Rs*
ASR	arithmetic right shift	*x*ASR *y*	(signed)$x \gg y$	#1–32 or *Rs*
ROR	rotate right	*x*ROR *y*	((unsigned)$x \gg y$) | ($x \ll (32 - y)$)	#1–31 or *Rs*
RRX	rotate right extended	*x*RRX	(*c* flag $\ll 31$) | ((unsigned)$x \gg 1$)	none

9. Write the example of a MOVS instruction shifts register r1 left by one bit. This multiplies register r1 by a value 21. As you can see, the C flag is updated in the cpsr because the S suffix is present in the instruction mnemonic.

```
PRE    cpsr = nzcvqiFt_USER
       r0 = 0x00000000
       r1 = 0x80000004

       MOVS    r0, r1, LSL #1

POST   cpsr = nzCvqiFt_USER
       r0 = 0x00000008
       r1 = 0x80000004
```

10. Write down the arithematic instructions.

Syntax: <instruction>{<cond>}{S} Rd, Rn, N

ADC	add two 32-bit values and carry	$Rd = Rn + N +$ carry
ADD	add two 32-bit values	$Rd = Rn + N$
RSB	reverse subtract of two 32-bit values	$Rd = N - Rn$
RSC	reverse subtract with carry of two 32-bit values	$Rd = N - Rn -$!(carry flag)
SBC	subtract with carry of two 32-bit values	$Rd = Rn - N -$!(carry flag)
SUB	subtract two 32-bit values	$Rd = Rn - N$

11. Write down the logical instructions.

AND logical bitwise AND of two 32-bit values
Rd = Rn & N ORR logical bitwise OR of two 32-bit values
Rd = Rn | N EOR logical exclusive OR of two 32-bit values
Rd = Rn ∧ N BIC logical bit clear (AND NOT) Rd = Rn & ~N

12. Write down the comparative instructions.

```
Syntax: <instruction>{<cond>} Rn, N
```

CMN	compare negated	flags set as a result of $Rn + N$
CMP	compare	flags set as a result of $Rn - N$
TEQ	test for equality of two 32-bit values	flags set as a result of $Rn \ \hat{} \ N$
TST	test bits of a 32-bit value	flags set as a result of $Rn \ \& \ N$

13. Write down the multiply instructions.

```
Syntax: <instruction>{<cond>}{S} RdLo, RdHi, Rm, Rs
```

SMLAL	signed multiply accumulate long	$[RdHi, RdLo] = [RdHi, RdLo] + (Rm * Rs)$
SMULL	signed multiply long	$[RdHi, RdLo] = Rm * Rs$
UMLAL	unsigned multiply accumulate long	$[RdHi, RdLo] = [RdHi, RdLo] + (Rm * Rs)$
UMULL	unsigned multiply long	$[RdHi, RdLo] = Rm * Rs$

14. What are the branch instructions.

```
Syntax: B{<cond>} label
        BL{<cond>} label
        BX{<cond>} Rm
        BLX{<cond>} label | Rm
```

B	branch	*pc* = *label*
BL	branch with link	*pc* = *label* *lr* = address of the next instruction after the BL
BX	branch exchange	*pc* = *Rm* & 0xfffffffe, *T* = *Rm* & 1
BLX	branch exchange with link	*pc* = *label*, *T* = 1 *pc* = *Rm* & 0xfffffffe, *T* = *Rm* & 1 *lr* = address of the next instruction after the BLX

A device can be attached, configured and used, reset, reconfigured and used, share the bandwidth with other devices, detached and reattached.

15. Write the example shows an STM increment before instruction followed by an LDM decrement after instruction.

```
PRE     r0 = 0x00009000
        r1 = 0x00000009
        r2 = 0x00000008
        r3 = 0x00000007

        STMIB   r0!, {r1-r3}

        MOV     r1, #1
        MOV     r2, #2
        MOV     r3, #3

PRE(2)  r0 = 0x0000900c
        r1 = 0x00000001
        r2 = 0x00000002
        r3 = 0x00000003

        LDMDA r0!, {r1-r3}

POST    r0 = 0x00009000
        r1 = 0x00000009
        r2 = 0x00000008
        r3 = 0x00000007
```

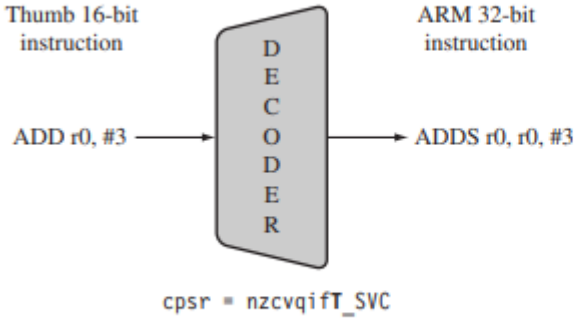
16. What are the stack operations.

The ARM architecture uses the load-store multiple instructions to carry out stack operations. The pop operation (removing data from a stack) uses a load multiple instruction; similarly, the push operation (placing data onto the stack) uses a store multiple instruction.

17. Addressing methods for stack operations.

Addressing mode	Description	Pop	= LDM	Push	= STM
FA	full ascending	LDMFA	LDMDA	STMFA	STMIB
FD	full descending	LDMFD	LDMIA	STMFD	STMDB
EA	empty ascending	LDMEA	LDMDB	STMEA	STMIA
ED	empty descending	LDMED	LDMIB	STMED	STMDA

18. Define thumb instruction decoding.



2 Thumb instruction decoding.

19. Single-Register Load-Store Instructions

```
Syntax: <LDR|STR>{<B|H>} Rd, [Rn,#immediate]
        LDR{<H|SB|SH>} Rd,[Rn,Rm]
        STR{<B|H>} Rd,[Rn,Rm]
        LDR Rd,[pc,#immediate]
        <LDR|STR> Rd,[sp,#immediate]
```

LDR	load word into a register	*Rd <- mem32[address]*
STR	save word from a register	*Rd -> mem32[address]*
LDRB	load byte into a register	*Rd <- mem8[address]*
STRB	save byte from a register	*Rd -> mem8[address]*
LDRH	load halfword into a register	*Rd <- mem16[address]*
STRH	save halfword into a register	*Rd -> mem16[address]*
LDRSB	load signed byte into a register	*Rd <- SignExtend(mem8[address])*
LDRSH	load signed halfword into a register	*Rd <- SignExtend(mem16[address])*

20. Write the example shows two Thumb instructions that use a preindex addressing mode. Both use the same pre-condition.

```
PRE     mem32[0x90000] = 0x00000001
        mem32[0x90004] = 0x00000002
        mem32[0x90008] = 0x00000003
        r0 = 0x00000000
        r1 = 0x00090000
        r4 = 0x00000004

        LDR  r0, [r1, r4]    ; register

POST    r0 = 0x00000002
        r1 = 0x00090000
        r4 = 0x00000004

        LDR    r0, [r1, #0x4]    ; immediate

POST    r0 = 0x00000002
```

Both instructions carry out the same operation. The only difference is the second LDR uses a fixed offset, whereas the first one depends on the value in register *r4*.

21. Write down Multiple-Register Load-Store Instructions

The Thumb versions of the load-store multiple instructions are reduced forms of the ARM load-store multiple instructions. They only support the increment after (IA) addressing mode.

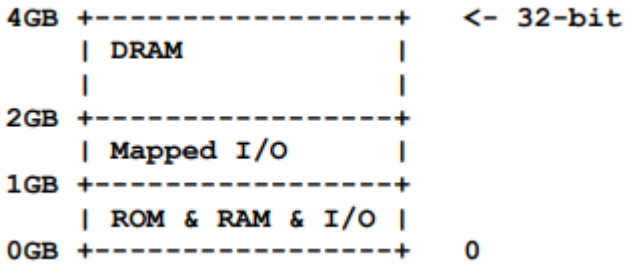
```
Syntax : <LDM|STM>IA Rn!, {low Register list}
```

LDMIA	load multiple registers	{Rd}^{*N} <- mem32[Rn + 4*N], Rn = Rn + 4*N
STMIA	save multiple registers	{Rd}^{*N} -> mem32[Rn + 4*N], Rn = Rn + 4*N

Here *N* is the number of registers in the list of registers. You can see that these instructions always update the base register *Rn* after execution. The base register and list of registers are limited to the low registers *r0* to *r7*.

Unit-3

1. What are the I/O memory



2.What is watchdog timer

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BODEN(1)	CP1	CP0	PWRTE(1)	WDTE	FOSC1	FOSC0
81h,181h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

3. What is TIMER 0 ?

The Timer0 module is a simple 8-bit overflow counter. The clock source can be either the internal system clock or an external clock When the clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge. The Timer0 module also has a programmable prescaler option. This prescaler can be assigned to either the Timer0 module or the Watchdog Timer. Bit PSA (OPTION) assigns the prescaler, and bits PS2:PS0 (OPTION) determine the prescaler value. TMR0 can increment at the following rates: 1:1 when the prescaler is assigned to Watchdog Timer, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, and 1:256. Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device‘s frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

4. What is the TIMER 1?

Timer1 can operate in one of two modes: • As a timer • As a counter. The clock source can be either the internal system clock (Fosc/4), an external clock, or an external crystal. Timer1 can operate as either a timer or a counter. When operating as a counter , the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows Timer1 to operate during sleep, which is useful for applications that require a real-time clock as well as the power savings of SLEEP mode. TImer1 also has a presale option which allows TMR1 to increment at the following rates: 1:1, 1:2, 1:4, and 1:8. TMR1 can be used in conjunction with the Capture/ Compare/PWM module. When used with a CCP module, Timer1 is the time-base for 16-bit capture or 16-bit compare and must be synchronized to the deviceDescribe the system behaviour based on the states and states function

5. What is the function of capture mode?

The Basic function of capture mode by CCP1 or CCP2 module is that the exact point of time of occurrence of that input edge change can be detected. For this purpose, timer1 must be running in timer mode

6.Define EEPROM.

EEPROM (also called E2PROM) stands for electrically erasable programmable read-only memory and is a type of non-volatile memory used in computers, usually integrated in microcontrollers such as smart cards and remote keyless systems, or as a separate chip device to store relatively small amounts of data by allowing individual bytes to be erased and reprogrammed.

7.What is SRAM?

SRAM (static RAM) is a type of random access memory (RAM) that retains data bits in its memory as long as power is being supplied. Unlike dynamic RAM (DRAM), which must be continuously refreshed, SRAM does not have this requirement, resulting in better performance and lower power usage. However, SRAM is also more expensive than DRAM, and it requires a lot more space.

8.Define UART?

UART stands for Universal Asynchronous Receiver/Transmitter. It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART's main purpose is to transmit and receive serial data.

9.What is ADC?

Analog to digital converter is a circuit that converts a continuous voltage value (analog) to a binary value (digital) that can be understood by a digital device which could then be used for digital computation. These ADC circuits can be found as an individual **ADC ICs** by themselves or embedded into a **microcontroller**. They're called ADCs for short.

10.What is DAC?

DAC [Digital-to-Analog Converter] is a device that converts digital audio information (comprised of a series of 0s and 1s) into an analog audio signal that can be sent to a headphone amp.

11.What ADC and DAC interfacing?

The Analog to Digital Conversion is a quantizing process. Here the analog signal is represented by equivalent binary states. The A/D converters can be classified into two groups based on their conversion techniques. In the first technique it compares given analog signal with the initially generated equivalent signal. In this technique, it includes successive approximation, counter and flash type converters.

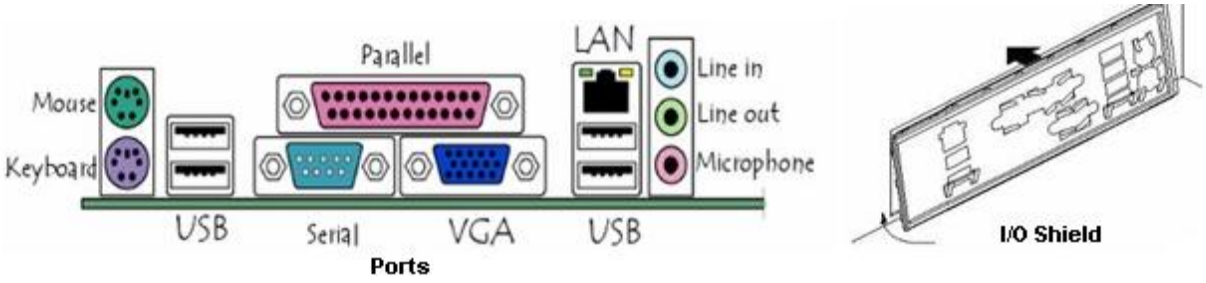
12. List some features of ADC 0808/0809:

The conversion speed is much higher
- The accuracy is also high
- It has minimal temperature dependence
- Excellent long term accuracy and repeatability
- Less power consumption

13. What is stepper motor interfacing?

A **stepper motor** is made up of a rotor, which is normally a permanent magnet and it is, as the name suggests the rotating component of the motor. A stator is another part which is in the form of winding. In the diagram below, the center is the rotor which is surrounded by the stator winding. This is called as four phase winding.

14.What are the I/O ports in ARM?



Ports come in various shapes like round (PS/2, etc.), rectangular (FireWire, etc.), square (Telephone plug), trapezoidal (D-Sub or old printer port), etc. Their physical properties and functions are standardized. Different ports and their terms are explained.

Hot-swappable ports can connect while equipment is running. PS/2 is not hot-swappable. Plug-and-play ports can automatically start the connected devices and are hot-swapping like USB and FireWire.

15.What is timer?

A timer is **a specialized type of clock used for measuring specific time intervals.** Timers can be categorized into two main types. The word "timer" is usually reserved for devices that count down from a specified time interval called a countdown timer,

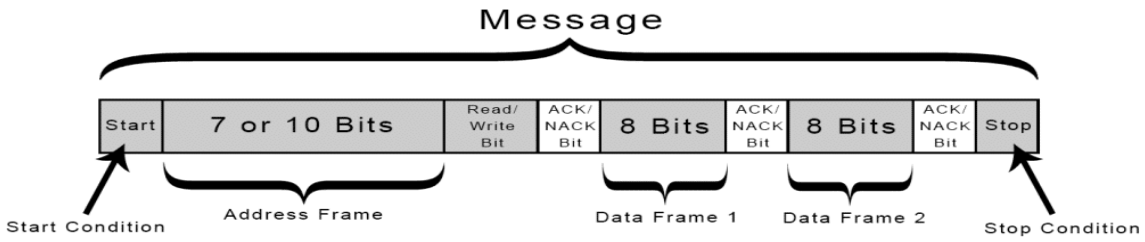
Unit-4

1.What is a CAN bus? Where is it used?

CAN is a serial bus for interconnecting a central Control network. It is mostly used in automobiles. It has fields for bus arbitration bits, control bits for address and data length data bits, CRC check bits, acknowledgement bits and ending bits.

2.I2C bus frame format:

With I2C, data is transferred in *messages.* Messages are broken up into *frames* of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame:



3.Define the term carrier sense in CSMA/CD?

Carrier Sense Multiple Access/Collision Detect (CSMA/CD) is the protocol for carrier transmission access in Ethernet networks. On Ethernet, any device can try to send a frame at any time. Each device senses whether the line is idle and therefore available to be used. Ifit is, the device begins to transmit its first frame. If another device has tried to send at the same time, a collision is said to occur and the frames are discarded. Each device then waitsa random amount of time and retries until successful in getting its transmission sent.

4.What are the features of the USB protocol?

A device can be attached, configured and used, reset, reconfigured and used, share the bandwidth with other devices, detached and reattached.

5.What is I2C?

I2C is a serial bus for interconnecting ICs .It has a start bit and a stop bit like an UART. It has seven fields for start,7 bit address, defining a read or a write, defining byte as acknowledging byte, data byte, NACK and end.

6.What is a CAN bus? Where is it used?

CAN is a serial bus for interconnecting a central Control network. It is mostly used in automobiles. It has fields for bus arbitration bits, control bits for address and data length data bits, CRC check bits, acknowledgement bits and ending bits.

7.State the special features on I2C?

- Low cost
- Easy implementation
- Moderate speed (upto 100 kbps).

8.What are disadvantages of I2C?

- Slave hardware does not provide much support
- Open collector drivers at the master leads to be confused

9.What is USB? Where is it used?

USB is a serial bus for interconnecting a system. It attaches and detaches a device from the network. It uses a root hub. Nodes containing the devices can be organized like a tree structure. It is mostly used in networking the IO devices like scanner in a computer system.

10.What are the features of the USB protocol?

A device can be attached, configured and used, reset, reconfigured and used, share the bandwidth with other devices, detached and reattached.

11.What are the features of SPI?

- SPI has programmable clock rates
- Full-duplex mode
- Crystal clock frequency is 8MHz
- Open drain or totempole output from master to slave

12.What are the four types of data transfer used in USB?

- Controlled transfer
- Bulk transfer
- Interrupt driven data transfer

Iso-synchronous transfer

13.Define multitasking.

Multitasking is the process of scheduling and switching the CPU between several tasks. A single CPU switches its attention between several sequential tasks.

It maximizes the utilization of the CPU and also provides modular construction of application.

14.What is ARM Core:

- ARM processors are known for their energy efficiency and high performance, making them suitable for a wide range of applications.
- ARM cores come in various configurations and performance levels, allowing you to choose the right core for your specific needs.

15.Define CAN Controller.

- CAN is a widely used communication protocol in automotive, industrial automation, and other industries for real-time data exchange.
- The CAN controller integrated into the ARM-based system allows for communication with other CAN nodes or devices on the network.

16.What is power efficiency?

ARM processors are known for their power efficiency, making them suitable for battery-powered or energy-conscious applications that use CAN.

17.Compare OS and RTOS.

OS	RTOS
1. It is an OS for systems do not have any time and deadline constrains. 2. control is given first to OS 3. Protects themselves from application 4. it needs large memory space EX: Windows, Linux, etc..,	1. it is a OS for systems which has time deadlines and real time dead lines 2. Application programs take control first. 3. RTOS can't protect themselves from application programs. 4. it needs less memory space EX:µC/OS-II, PSOS,VRTx, Vx works

18.Describe in brief about types of RTOS.

Preemptive RTOS→in this type of RTOS, new higher priority task will gain control of the CPU only when the current task gives up the CPU.

Non Preemptive RTOS→ In this type of RTOS, if any higher priority task or ISR is ready to run, the current task is preempted (suspended) and higher task is immediately giventhe control of CPU.

19.What is Task scheduler?

A part of the RTOS called the scheduler keeps the track the state of which one task should go into the running state.

20.What is shared data problem?

If a variable is used in two different processes and another task if interrupts before the operation on that data is completed then the value of the variable may differ from the one expected if the earlier operation had been completed .This ids known as shared data problem.

Unit-5

1. What is a "Thing" in the context of Internet of Things (IoT)?

Answer: The "Thing" commonly referred to by the concept of the Internet of Things is any item that can contain an embedded, connected computing device. A "Thing" in the IoT could be a shipping container with an RFID tag or a consumer's watch with a WiFi chip that sends fitness data or short messages to a server somewhere on the Internet.

2. Define Raspberry pi Architecture.

The Raspberry Pi is a series of small, affordable single-board computers (SBCs) designed to promote computer science education and provide a platform for various DIY projects. As of my last knowledge update in September 2021, the Raspberry Pi has gone through several iterations, each with its own architecture.

3. *What are the raspberry pi model.*

1. Raspberry Pi 1 Model A/B:
   - Architecture: ARM1176JZF-S (ARMv6)
   - CPU: Single-core ARM CPU
2. Raspberry Pi 2 Model B:
   - Architecture: Cortex-A7 (ARMv7)
   - CPU: Quad-core ARM Cortex-A7 CPU
3. Raspberry Pi 3 Model B:
   - Architecture: Cortex-A53 (ARMv8-A)
   - CPU: Quad-core ARM Cortex-A53 CPU
4. Raspberry Pi 4 Model B:
   - Architecture: Cortex-A72 (ARMv8-A)
   - CPU: Quad-core ARM Cortex-A72 CPU
5. Raspberry Pi Zero:
   - Architecture: ARM1176JZF-S (ARMv6)
   - CPU: Single-core ARM CPU
6. Raspberry Pi Zero W:
   - Architecture: ARM1176JZF-S (ARMv6)

- CPU: Single-core ARM CPU

7. Raspberry Pi Pico:
   - Architecture: RP2040 (dual ARM Cortex-M0+ cores)
   - CPU: Dual-core ARM Cortex-M0+ CPU
8. Raspberry Pi 400:
   - Architecture: Cortex-A72 (ARMv8-A)
   - CPU: Quad-core ARM Cortex-A72 CPU

4.Define booting up RPI.

Booting up a Raspberry Pi involves several steps, including preparing the microSDcard with the operating system, connecting peripherals, and powering on the device.

5.What is Connect Peripherals :

- Depending on your Raspberry Pi model and intended use, you may need to connect various peripherals. At a minimum, you'll need a USB keyboard and mouse, an HDMI display or a compatible display adapter, and a power supply.
- Optionally, you can connect other peripherals like a USB Wi-Fi adapter, Ethernet cablefor a wired connection, or external storage devices.

6. What is the difference between the Internet of Things (IoT) and Machine to Machine(M2M)?

Answer: Generally speaking, M2M could be considered a subset of IoT. M2M is like a line connecting 2 points, and IoT is like a network, a system composed of lots of M2M and triggering lots of interactions/activities.

Giving a simple definition to M2M which is transferring data from one machine to another one. It's been used everywhere in our daily life. For example, entrance security. Just like using your employee card to unlock a door. When the security detector receives the ID from the employee card and then unlock the door once the ID is approved. This is M2M.

7. *What is difference between Wireless Sensor Network (WSN) and Internet of Things(IoT) network (sensor)?*

Answer: About WSN:
Wireless sensor network is the foundation of IoT applications.

WSN is the network of motes, formed to observe, to study or to monitor physical parameters of desired application.

8. *What is IoT?*

Answer: IoT stands for Internet of Things. It is basically a network using which things can communicate with each other using internet as means of communication between them. All the things should be IP protocol enabled in order to have this concept possible. Not onebut multiple technologies are involved to make IoT a great success.

9. *Define GPIO.*

GPIO stands for "General-Purpose Input/Output." It refers to a type of interface or

pin on a microcontroller, microprocessor, or single-board computer (such as theRaspberry Pi) that can be configured to serve as either a digital input or a digitaloutput.

10. Applications of GPIO.

GPIO pins are commonly used in embedded systems, robotics, electronics projects, and IoT (Internet of Things) applications. They provide a simple way to interface with the physical world, allowing devices to interact with and control external hardware. The specific number and capabilities of GPIO pins vary depending on the microcontroller or single-board computer in use.

11. *Application of IoT:*

- Smart home
- Connected car
- Smart city
- Smart Retail
- Connected Health
- Smart grids
- Smart Farming

12. Define OS?

An "OS," or Operating System, is a fundamental software component that manages and controls computer hardware resources and provides essential services to software applications. It acts as an intermediary between users, applications, and the hardware of a computer or computing device.

13.Define **Hardware Abstraction**:

The OS abstracts the underlying hardware, presenting a consistent and standardized interface to software applications.

14.Define**User Interface** :

Most modern operating systems include graphical user interfaces (GUIs) that allow users to interact with the computer through windows, icons, menus, and pointing devices like mice or touchscreens.

15.What is **Security and Authentication**: Operating systems often include security features like user authentication, access control, and encryption to protect the system and user data from unauthorized access and threats.

16. Define **Networking**: The OS provides networking capabilities, enabling devices to connect to networks and the internet. It manages network interfaces, protocols, and services like DHCP and DNS.

17. Define **Error Handling**: It handles system errors, exceptions, and crashes gracefully to prevent catastrophic failures, recover from errors when possible, and notify users or administrators of issues

****