ET & IoT (CAT-1)

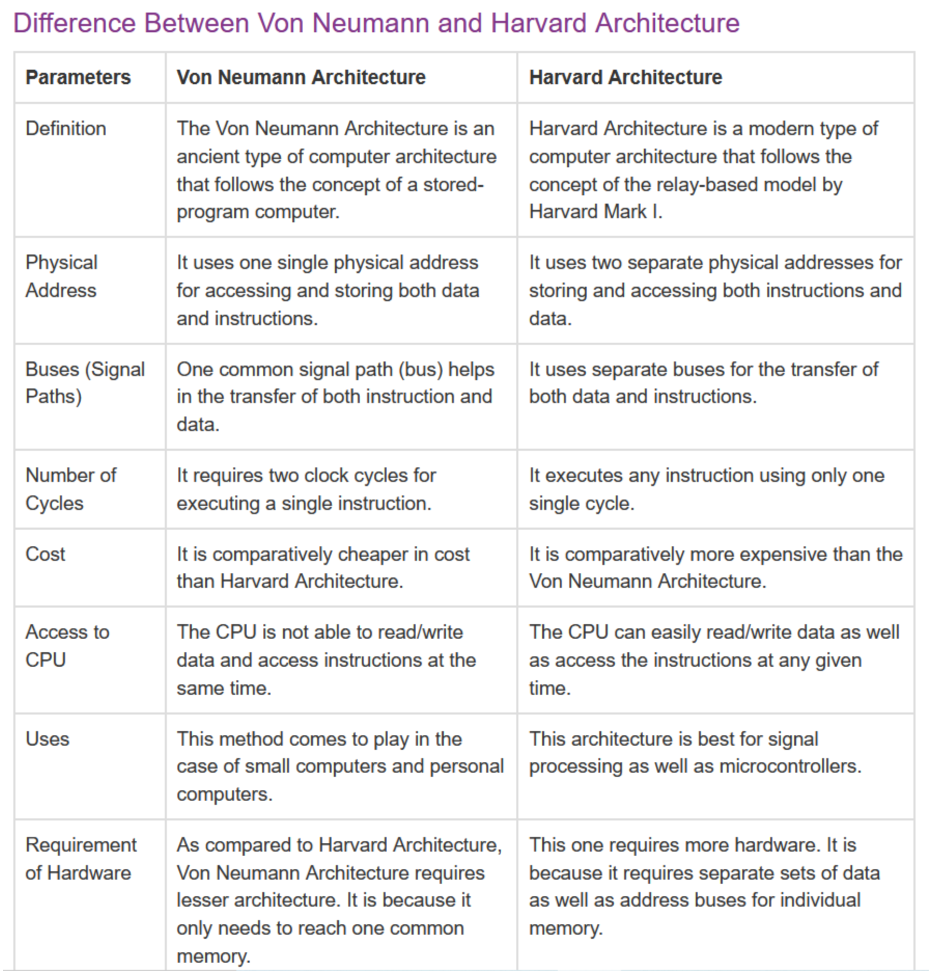
Question Bank solution by **JAAT**

For more information visit https://youtu.be/6DO24YfGPXI

1. What is UART?

ANS.. UART represents Universal Asynchronous Receiver Transmitter. It is dedicated to hardware related to serial communication. UART is one of the most generally used serial communication techniques. UART is being used in several applications like GPS Receivers, Bluetooth Modules, GSM and GPRS Modems, Wireless Communication Systems, RFID-based applications, etc.

2. What is the difference between Von Neumann architecture and Harvard architecture ?



Difference Between Von Neumann and Harvard Architecture

Parameters	Von Neumann Architecture	Harvard Architecture
Definition	The Von Neumann Architecture is an ancient type of computer architecture that follows the concept of a stored-program computer.	Harvard Architecture is a modern type of computer architecture that follows the concept of the relay-based model by Harvard Mark I.
Physical Address	It uses one single physical address for accessing and storing both data and instructions.	It uses two separate physical addresses for storing and accessing both instructions and data.
Buses (Signal Paths)	One common signal path (bus) helps in the transfer of both instruction and data.	It uses separate buses for the transfer of both data and instructions.
Number of Cycles	It requires two clock cycles for executing a single instruction.	It executes any instruction using only one single cycle.
Cost	It is comparatively cheaper in cost than Harvard Architecture.	It is comparatively more expensive than the Von Neumann Architecture.
Access to CPU	The CPU is not able to read/write data and access instructions at the same time.	The CPU can easily read/write data as well as access the instructions at any given time.
Uses	This method comes to play in the case of small computers and personal computers.	This architecture is best for signal processing as well as microcontrollers.
Requirement of Hardware	As compared to Harvard Architecture, Von Neumann Architecture requires lesser architecture. It is because it only needs to reach one common memory.	This one requires more hardware. It is because it requires separate sets of data as well as address buses for individual memory.

3. What do you mean by hard real time embedded system? Mention two applications where we can use this type of embedded system.

ANS.. Real time systems are those systems that work within strict time constraints and provide a worst case time estimate for critical situations. Embedded systems provide a specific function in a much larger system. When there is an embedded component in a real time system, it is known as a real time embedded system.

Applications of Real Time Embedded Systems

There are various applications of real time embedded systems. Some of these are −

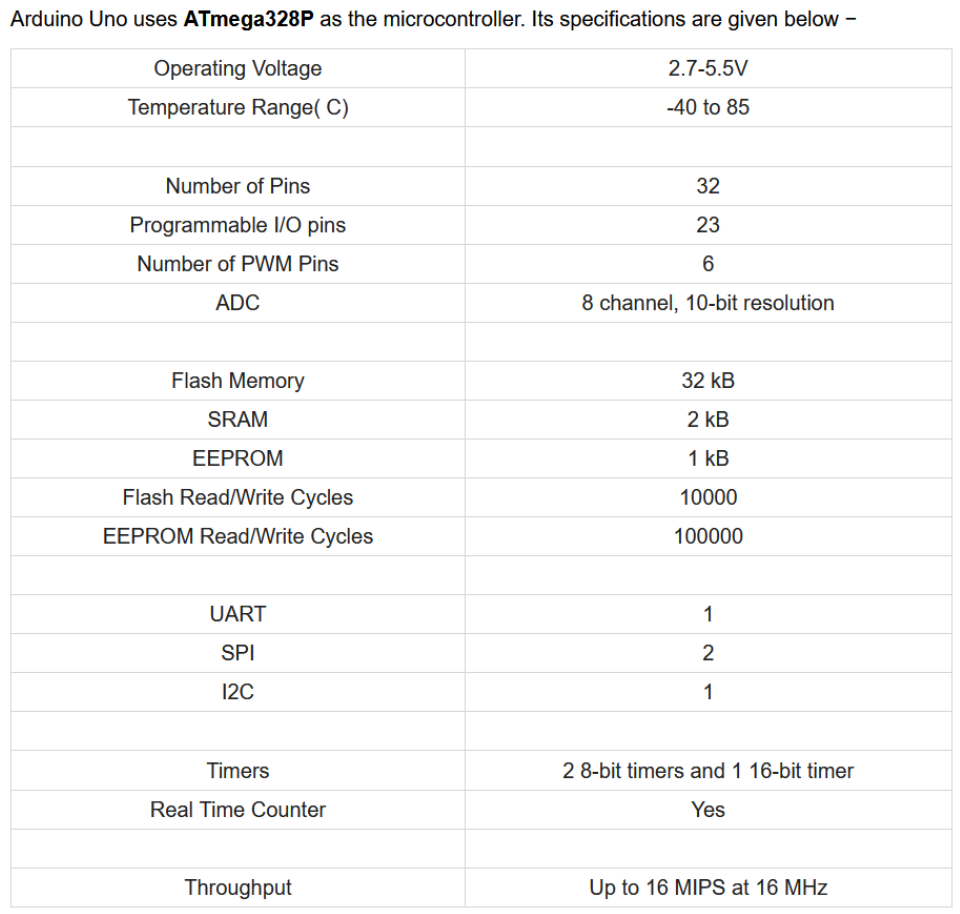
- Vehicle control systems for automobiles, ships, railways, airplanes etc.
- Telephones, radio and satellite communications.

4. What is large scale embedded systems ?

Sophisticated or Complex Embedded Systems :

Sophisticated or Complex Embedded Systems are designed using multiple 32-bit or 64-bit micro-controller. These systems are developed to perform large scale complex functions. These systems have high hardware and software complexities. We use both hardware and software components to design final systems or hardware products.

5. Architecture of Microcontroller used in Arduino UNO.



Arduino Uno uses **ATmega328P** as the microcontroller. Its specifications are given below –

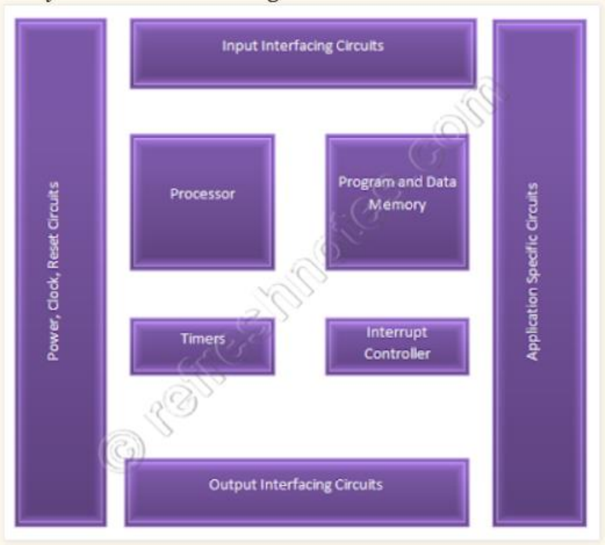
Operating Voltage	2.7-5.5V
Temperature Range(C)	-40 to 85
Number of Pins	32
Programmable I/O pins	23
Number of PWM Pins	6
ADC	8 channel, 10-bit resolution
Flash Memory	32 kB
SRAM	2 kB
EEPROM	1 kB
Flash Read/Write Cycles	10000
EEPROM Read/Write Cycles	100000
UART	1
SPI	2
I2C	1
Timers	2 8-bit timers and 1 16-bit timer
Real Time Counter	Yes
Throughput	Up to 16 MIPS at 16 MHz

6. Explain about the details of other hardware units available in embedded system

Hardware Units in Embedded System

Overview of Hardware Units in Embedded System

An Embedded System has the following essential units in it.


Input Interfacing Circuits
Power, Clock, Reset Circuits
Processor
Program and Data Memory
Application Specific Circuits
Timers
Interrupt Controller
Output Interfacing Circuits

Processor - Brain of an Embedded System. It is the one which has Control Unit and Execution Unit.

Control Unit:

- Controls Program flow and data path
- Includes a fetch unit - to fetch program instructions from memory

Execution Unit:

- Includes Arithmetic and Logic Unit
- Execute instructions for a program control task like interrupt, halt, reset, call jump
- Execute application program instructions

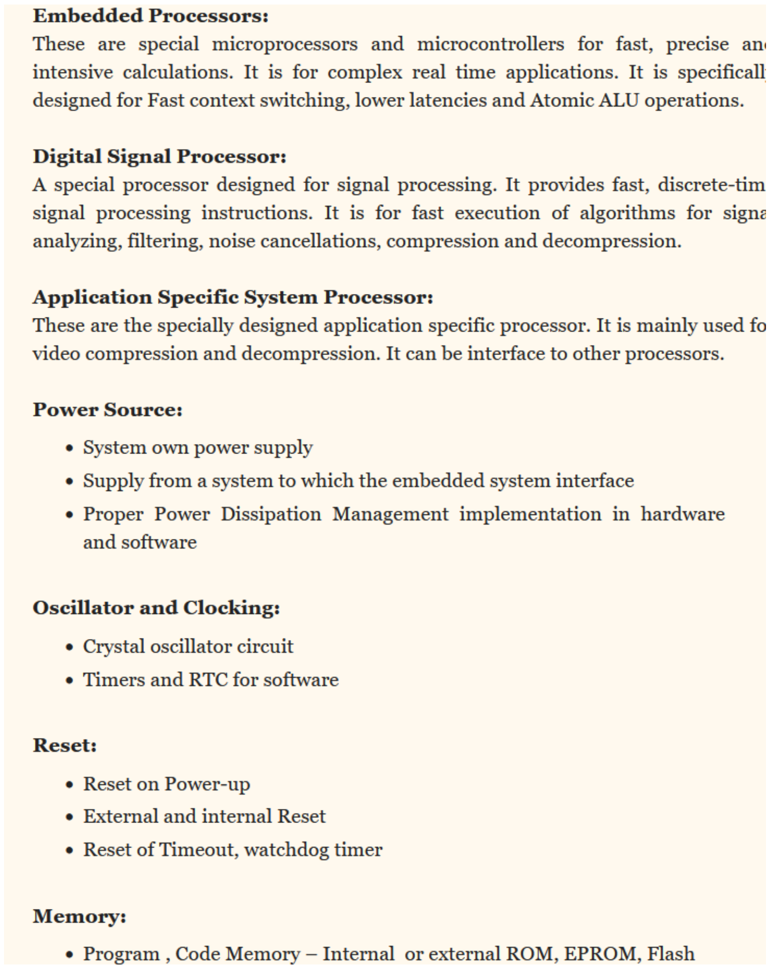
A processor is mostly in the form of an IC chip. It could be in the form of ASIC or SoC. Processor core is a part of functional circuit on a chip.
Processor chip or core can be,

- General Purpose Processor (GPP)
- Application Specific System Processor (ASSP)
- Multiprocessor using GPP and Application Specific Instruction Processor (ASIP)
- GPP cores or ASIP cores integrated in to an ASIC or VLSI chip
- FPGA core integrated with processor units in a VLSI chip

General Purpose Processor:
A processor having a general purpose instruction set and readily available compilers to enable programming in a high level language is called General Purpose Processor. It can be Microprocessor, Microcontroller, Embedded Processor, and Digital Signal Processor.

Microprocessor:
It has CPU on a chip. It may include additional units like cache memory and floating point processing units for faster processing.

Microcontroller:
It has CPU, memory and other functional units on a chip. It includes peripherals like interrupt handler, IO ports, Timer, ADC, etc...



Embedded Processors:
These are special microprocessors and microcontrollers for fast, precise an intensive calculations. It is for complex real time applications. It is specificall designed for Fast context switching, lower latencies and Atomic ALU operations.

Digital Signal Processor:
A special processor designed for signal processing. It provides fast, discrete-tim signal processing instructions. It is for fast execution of algorithms for signa analyzing, filtering, noise cancellations, compression and decompression.

Application Specific System Processor:
These are the specially designed application specific processor. It is mainly used fo video compression and decompression. It can be interface to other processors.

Power Source:

- System own power supply
- Supply from a system to which the embedded system interface
- Proper Power Dissipation Management implementation in hardware and software

Oscillator and Clocking:

- Crystal oscillator circuit
- Timers and RTC for software

Reset:

- Reset on Power-up
- External and internal Reset
- Reset of Timeout, watchdog timer

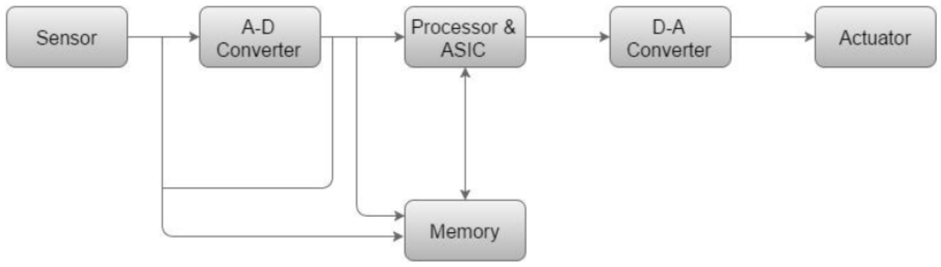
Memory:

- Program , Code Memory – Internal or external ROM, EPROM, Flash

7. Describe in detail about embedded system on-chip with necessary sketch.

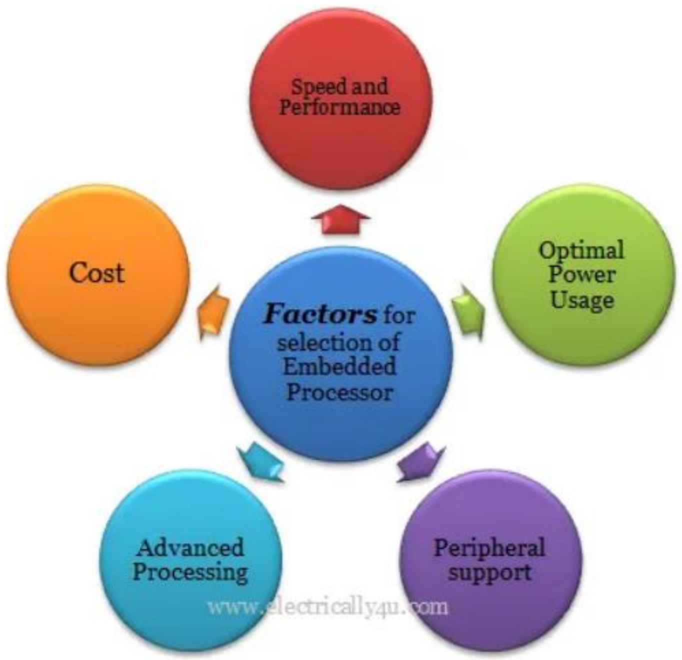
Basic Structure of an Embedded System

The following illustration shows the basic structure of an embedded system –



- **Sensor** – It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** – An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- **Processor & ASICs** – Processors process the data to measure the output and store it to the memory.
- **D-A Converter** – A digital-to-analog converter converts the digital data fed by the processor to analog data
- **Actuator** – An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

8. Discuss about the factors to be considered for selection of processor in embedded system.



Speed and Performance

The most important factor to consider when choosing a processor for an embedded system is its performance. A processor's speed is primarily determined by its architecture and silicon design.

The number of instructions executed per second and the number of operations per clock cycle must be evaluated for assessing the performance. At the same time, the efficiency of the computation units is also important while talking about the performance.

Optimal Power usage

Increasing the logic density and clock speed have an adverse impact on the power requirement of the processor. Faster charging and discharging cycles in the capacitor, leakage currents may lead to more power consumption.

More logic leads to higher power density thereby making the heat dissipation difficult. With more emphasis on greener technologies and since many systems are becoming battery operated, it is important to design the system for optimal power usage.

Peripheral support

Apart from the processor, the embedded system has many other peripherals to perform input and output operations. It is important to have the right peripherals to assist the processor in optimized performance.

Advanced Processing

Along with the core processor, the presence of various co-processors and specialized processing units can add more value to the processing performance. The instructions fetched by the core processor are executed by the co-processors in parallel, thereby reducing the processing load.

Cost

For all the required functionalities to be built up in a system, certainly the price will be the determining factor during the selection of processor for an embedded system.

9. Illustrate with example the techniques used for memory devices.

10. Write the need for software in embedded systems.

11. What is flash memory and EEPROM ?

Flash memory is a type of electronically-erasable programmable read-only memory (EEPROM), but it can also be a standalone memory storage device such as USB drive.

It is a non-volatile memory chip used for storage and for transferring data between a PC and other digital devices. It is often found in USB flash drives, MP3 players, digital cameras and solid-state drives

EEPROM is a type of data memory device that uses an electronic device to erase or write digital data. It has per byte erase-and-write capabilities, which makes it slow. Flash memory is a distinct type of EEPROM, which is programmed and erased in large blocks. Nonetheless, the trend seems to be of using AND flash for devices that only support large-block erasure.

Flash memory has many features. It is a lot cheaper than EEPROM and does not require batteries for solid-state storage such as static RAM.

To sum it up,

- Flash is just one type of EEPROM.
- Flash uses NAND-type memory, while EEPROM uses NOR type.
- Flash is block-wise erasable, while EEPROM is byte-wise erasable.
- Flash is constantly rewritten, while other EEPROMs are seldom rewritten.
- Flash is used when large amounts are needed, while EEPROM is used when only small amounts are needed.

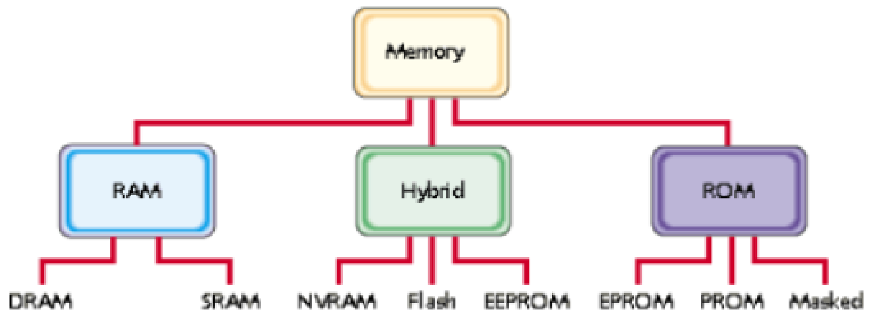
12. What do you mean by system-on-chip (SOC)? Mention one example.

A **system on a chip**, also known as an **SoC**, is essentially an integrated circuit or an IC that takes a single platform and integrates an entire electronic or computer system onto it. It is, exactly as its name suggests, an entire system on a single chip. The components that an SoC generally looks to incorporate within itself include a central processing unit, input and output ports, internal memory, as well as analog input and output blocks among other things.

You will tend to find System on a Chip hardware in embedded systems such as smartphones, tablets, and devices used for the Internet of Things, to name a few.

Intel has its own form of SoC, and it is called the **Intel NUC**, or **Next Unit of Computing**.

13. What are the different memory devices used in embedded systems?



14. Explain input output devices used in embedded systems.

Input/Output Devices

Introduction

A processor is not of much use unless you can pass data into and out of the CPU to external devices like serial ports, network interfaces, or displays. In this section I will attempt to talk about how I/O works on a microprocessor at a low-level.

The Address Bus

Recall from our discussion earlier about microprocessors, that every CPU has a number of pins, which work together, called an address bus. The address bus is normally used to read or write to memory, most often RAM chips. Most modern microprocessors use the address bus for more than just reading and writing to memory however.

The Data Bus

The data bus is nothing more than a series of pins on the processor that are used to get data into, or out of, the processor chip itself. All memory and I/O devices are connected to the data bus, but depending on the current state of the address bus and other control pins on the processor, only one chip can actually be connected to the data bus at any given moment.

Interrupt Requests

In addition to the processor using the data bus, address bus and special I/O pin to communicate with external devices; the external devices use another pin when they need the attention of the processor. This is referred to as an Interrupt Request Line or IRQ Line. For example, whenever you press a key on the keyboard, the keyboard controller device generally signals the main processor that a key is available by asserting the interrupt line. Some processors have a single IRQ line, which must be shared amongst all devices, while other have a series of IRQ lines.

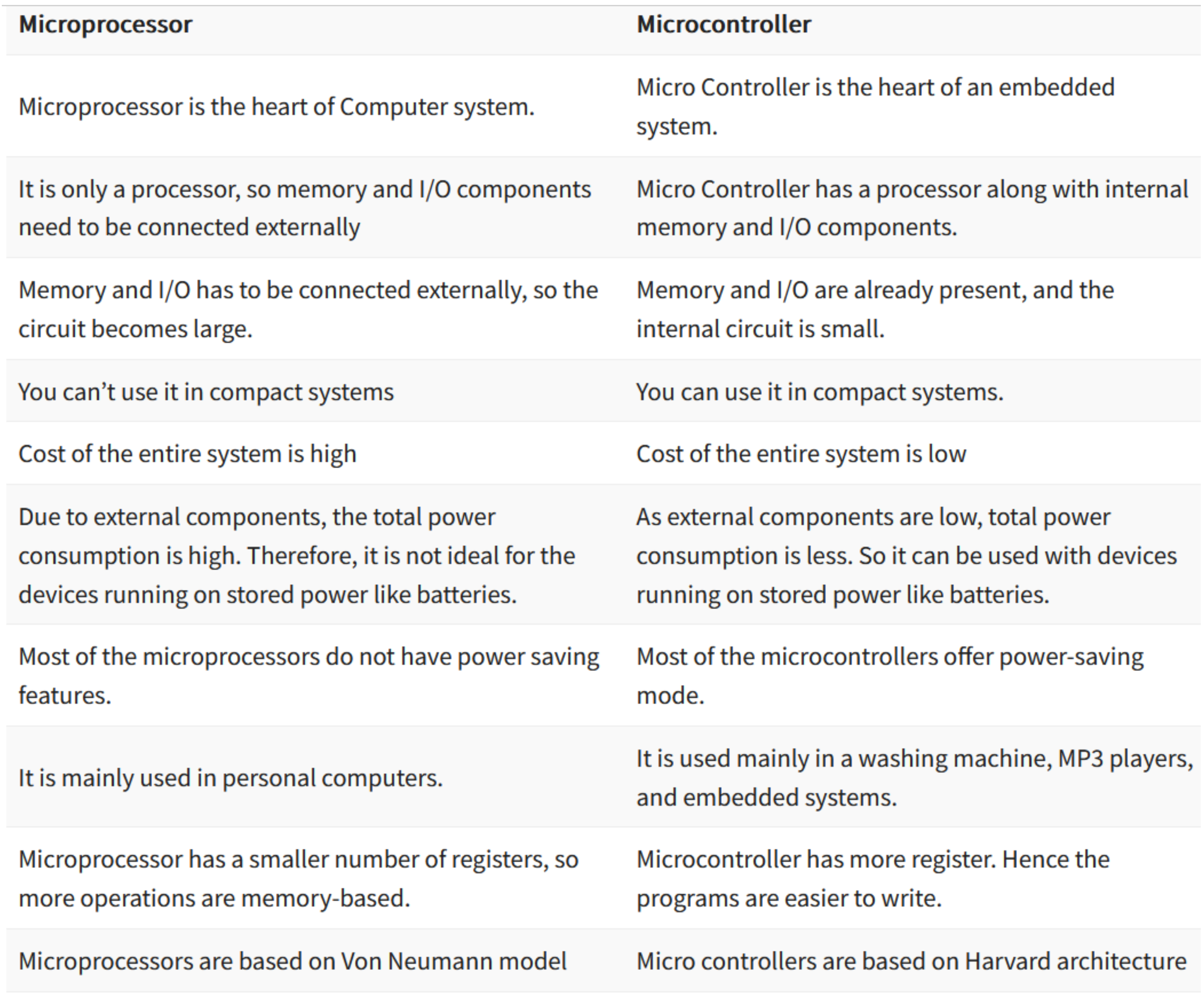
Memory Mapped I/O

I/O Port addressing is not the only way the processor can communicate with external devices however. Another commonly used technique is called memory mapped I/O. In this case, instead of asserting the I/O pin and addressing a data port, the processor just accesses a memory address directly. The external device can have a small amount of RAM or ROM that the processor just reads or writes as needed.

Direct Memory Access

One technique that has been used for years to speed transfer of data from main memory to an external device's memory is the direct memory access feature (DMA). The processor on the external device executes DMA transfers, without any assistance from the main processor. The processors must cooperate for this to work obviously. While the DMA transfer is in progress, the main processor is free to tend to other tasks, but should not attempt to modify the information in the buffer being transfer, until the transfer is complete

15. Distinguish between microprocessor and microcontroller ?



Microprocessor	**Microcontroller**
Microprocessor is the heart of Computer system.	Micro Controller is the heart of an embedded system.
It is only a processor, so memory and I/O components need to be connected externally	Micro Controller has a processor along with internal memory and I/O components.
Memory and I/O has to be connected externally, so the circuit becomes large.	Memory and I/O are already present, and the internal circuit is small.
You can't use it in compact systems	You can use it in compact systems.
Cost of the entire system is high	Cost of the entire system is low
Due to external components, the total power consumption is high. Therefore, it is not ideal for the devices running on stored power like batteries.	As external components are low, total power consumption is less. So it can be used with devices running on stored power like batteries.
Most of the microprocessors do not have power saving features.	Most of the microcontrollers offer power-saving mode.
It is mainly used in personal computers.	It is used mainly in a washing machine, MP3 players, and embedded systems.
Microprocessor has a smaller number of registers, so more operations are memory-based.	Microcontroller has more register. Hence the programs are easier to write.
Microprocessors are based on Von Neumann model	Micro controllers are based on Harvard architecture

16. What is system on chip? Explain embedded systems change with system on chip.

A **system on a chip**, also known as an **SoC**, is essentially an integrated circuit or an IC that takes a single platform and integrates an entire electronic or computer system onto it. It is, exactly as its name suggests, an entire system on a single chip. The components that an SoC generally looks to incorporate within itself include a central processing unit, input and output ports, internal memory, as well as analog input and output blocks among other things.

17. What is processor architecture? What are the different processor architectures available for processor design?

CPU architecture is basically the arrangement/layout of the electronic components that make up the cpu(like transistors).

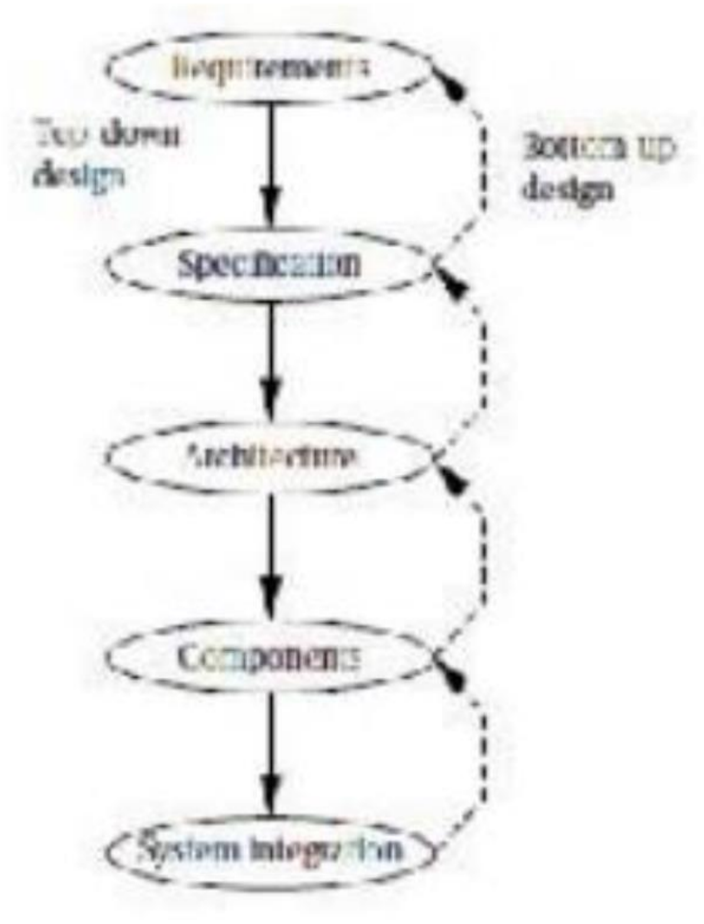
It can be devided into

- Instruction Set Architecture(ISA).
    - ISA is the set of instructions supported by a processor. Typically, a bunch of processors support the same ISA. For example, x86, ARM ISA, TI DSPs ISA are different ISAs.
- Microarchitecture or ISA implementation
    - Microarchitecture concepts deal with how the ISA is implemented. Concepts such as instruction pipelining, branch prediction, out of order execution are all employed to achieve an efficient (fast and power-effective(?)) realization of the ISA.

the different processor architectures available for processor design are:

- (1) Von-Neumann (or stored program computer) architecture.
- (2) Harvard architecture.
- (3) Harvard Architecture Derivatives.
- (4) CISC (Complex Instruction Set Computer)
- (5) RISC (Reduced Instruction Set)
- (6) DSPs.
- (7) VLIW architecture.

18. Explain the design process of embedded systems.



Top–down Design—we will begin with the most abstract description of the system and conclude with concrete details. The alternative is a **bottom–up** view in which we start with components to build a system. Bottom–up design steps are shown in the figure as dashed-line arrows. We need bottom–up design because we do not have perfect insight into how later stages of the design process will turn out. During the design process we have to consider the major goals of the design such as

■ manufacturing cost;

■ performance (both overall speed and deadlines); and

■ power consumption.

19. What are the programming languages used in embedded systems?

1. C

- The loose data typing policy of C makes it quite suitable.
- It is simple to port the embedded programs from one device to the next as compared to other languages.
- The widespread community in C provides vast support for Embedded Systems Programming.

2. C++

- C++ is more secure than C because of its use of string literals, enumeration constants, templates etc.
- Overloaded functions and constructors in C++ are an asset for embedded systems programming.
- The object oriented nature of C++ is also quite useful for complex embedded systems programming

3. Java

- Java can be used to write extensible, portable, and downloadable embedded systems applications.
- There are many DevOps tools and libraries in Java that make it suitable for Embedded Systems Programming.
- The Java Virtual Machine ensures that embedded systems programmed in Java are portable and can be used for different IoT platforms.

4. Python

- Python is a popular language and known for its writability, concise, readable coding style, and error deduction.
- Python is much handier in the case of complicated embedded systems such as those using neural networks.
- Real time embedded systems use Python quite often. *MicroPython* is a good example of a lean and efficient implementation for this.

5. Rust

- Rust allows memory management using both dynamic and static methods using various tools.

- Rust can be used to program a wide range of embedded systems from small micro controllers to large multifaceted systems.
- There is a large community support in Rust for embedded systems programming.

6. Ada

- Ada is useful for embedded systems programming because of strong typing, run-time checking, parallel processing, exception handling, generics etc.
- Ada packages can be compiled separately as it was created for the development of large software systems.
- Ada is used in critical systems as it supports run-time checks for bugs such as unallocated memory range violations, off-by-one errors, array access errors etc.

7. Lua

- The base language in Lua is quite light as it has various meta-features that can be extended as required.
- Lua can implement object oriented programming using first-class functions and tables.
- Lua is cross-platform and it supports a C API that can be embedded into applications.

8. B#

- B# is well suited for small scale embedded systems programming because of its tiny core and small memory constraints.
- B# is type safe and does not allow any pointer manipulation which is an asset in embedded systems programming.
- The code written in B# is directly mapped to a tight instruction set. This reduces the runtime in low resource embedded devices.

20. Explain about significance of embedded system and classification of the Embedded systems.

An Embedded System is an integrated system which is formed as an combination of computer hardware and software for a specific function. It can be said as a dedicated computer system which has been developed for some particular reason. But it is not our traditional computer system or general purpose computers, these are the Embedded systems which may work independently or attached to a larger system to work on few specific functions.

Embedded Systems are classified based on the two factors i.e.

1. Performance and Functional Requirements
2. Performance of Micro-controllers

Based on Performance and Functional Requirements it is divided into 4 types as follows :

1. **Real-Time Embedded Systems :**

A Real-Time Embedded System is strictly time specific which means these embedded systems provides output in a particular/defined time interval. These type of embedded systems provide quick response in critical situations which gives most priority to time based task performance and generation of output.

- **Soft Real Time Embedded Systems –**
- In these types of embedded systems time/deadline is not so strictly followed. If deadline of the task is passed (means the system didn't give result in the defined time) still result or output is accepted.
- **Hard Real-Time Embedded Systems –**
- In these types of embedded systems time/deadline of task is strictly followed. Task must be completed in between time frame (defined time interval) otherwise result/output may not be accepted.

Examples :

- Traffic control system
- Military usage in defense sector

2. **Stand Alone Embedded Systems :**

Stand Alone Embedded Systems are independent systems which can work by themselves they don't depend on a host system. It takes input in digital or analog form and provides the output.

Examples :

- MP3 players
- Microwave ovens

3. **Networked Embedded Systems :**

Networked Embedded Systems are connected to a network which may be wired or wireless to provide output to the attached device. They communicate with embedded web server through network.

Examples :

- Home security systems
- ATM machine

4. **Mobile Embedded Systems :**

Mobile embedded systems are small and easy to use and requires less resources. They are the most preferred embedded systems. In portability point of view mobile embedded systems are also best.

Examples :

- MP3 player
- Mobile phone

Based on Performance and micro-controller it is divided into 3 types as follows :

1. **Small Scale Embedded Systems :**

Small Scale Embedded Systems are designed using an 8-bit or 16-bit micro-controller. They can be powered by a battery. The processor uses very less/limited resources of memory and processing speed. Mainly these systems does not act as an independent system they act as any component of computer system but they did not compute and dedicated for a specific task.

2. **Medium Scale Embedded Systems :**

Medium Scale Embedded Systems are designed using an 16-bit or 32-bit micro-controller. These medium Scale Embedded Systems are faster than that of small Scale Embedded Systems. Integration of hardware and software is complex in these systems. Java, C, C++ are the programming languages are used to develop medium scale embedded systems.

3. **Sophisticated or Complex Embedded Systems :**

Sophisticated or Complex Embedded Systems are designed using multiple 32-bit or 64-bit micro-controller. These systems are developed to perform large scale complex functions. These systems have high hardware and software complexities. We use both hardware and software components to design final systems or hardware products

21. Explain about the components used as core of an embedded system. Also mention their commonly used application.

Components of the Embedded System

As the embedded system is made up of hardware and software components. In below section hardware components are described below:

1. Power supply
For the embedded system the power supply is the key component to provide the power to the embedded system circuit. Usually, the embedded system requires 5 V supply or can be range from 1.8 to 3.3. V. The power supply source can be battery or can be provided by a wall adaptor. The power supply is selected as per user requirements and application requirements

2. Processor
For any embedded system the processor acts as the brain of the system. The processor is responsible for deciding the performance of the embedded system. In the market there are multiple types of processors available and can be selected as per user requirement. The embedded system can act as a microcontroller and microprocessor. The processor can be an 8-bit processor, a 16-bit processor, and a 32-bit processor. The lesser the bit the smaller the application is for embedded systems.

3. Memory
As there are different microcontrollers is used in the embedded system the memory is present in the microcontroller itself. There are basically two types of memory RAM(Random access memory) and ROM (Read-only memory). As the RAM is volatile type memory the data can be stored temporarily in the memory and when system is switch off the data is lost from the memory.

5. Communication ports
The communication port is the type of interface that is used to communicate with other types of embedded systems. In the embedded system there is multiple types of communication ports like UART, USB, Ethernet, RS-485, and many more. When

6. Output and Input
When the embedded system is used the input is needed to interact with the system. The input to the embedded system can be provided by the sensor or by the user itself. The processor used in the embedded system can be based on input and output.

Hardware Components of the Embedded System
When all the hardware components are selected for an embedded system the next task is to select software components for designing an embedded system.

1. Assembler
The assembler is sued when the programming language sued for designing the application is assembly language. The assembly language program is then converted into the HEX code so that it can be further processed. And after writing the code the programmer is used for writing the program in the chip.

2. Emulator
An emulator is a software tool that is used to execute the functions of the host system. All the components can be controlled by the emulator tool.

3. Compiler
The compiler is a type of software that is used to convert the programming language into some language that the target machine can understand and execute the functions.

24. What is an embedded system? List out its applications. Explain why the processors play a vital role in embedded systems.

Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke.

Its applications

- Central heating systems
- GPS systems
- Fitness trackers
- Medical devices
- Automotive systems
- Transit and fare collection
- ATMs
- Factory robots
- Electric vehicle charging stations

Processor is the heart of an embedded system. It is the basic unit that takes inputs and produces an output after processing the data. For an embedded system designer, it is necessary to have the knowledge of both microprocessors and microcontrollers.

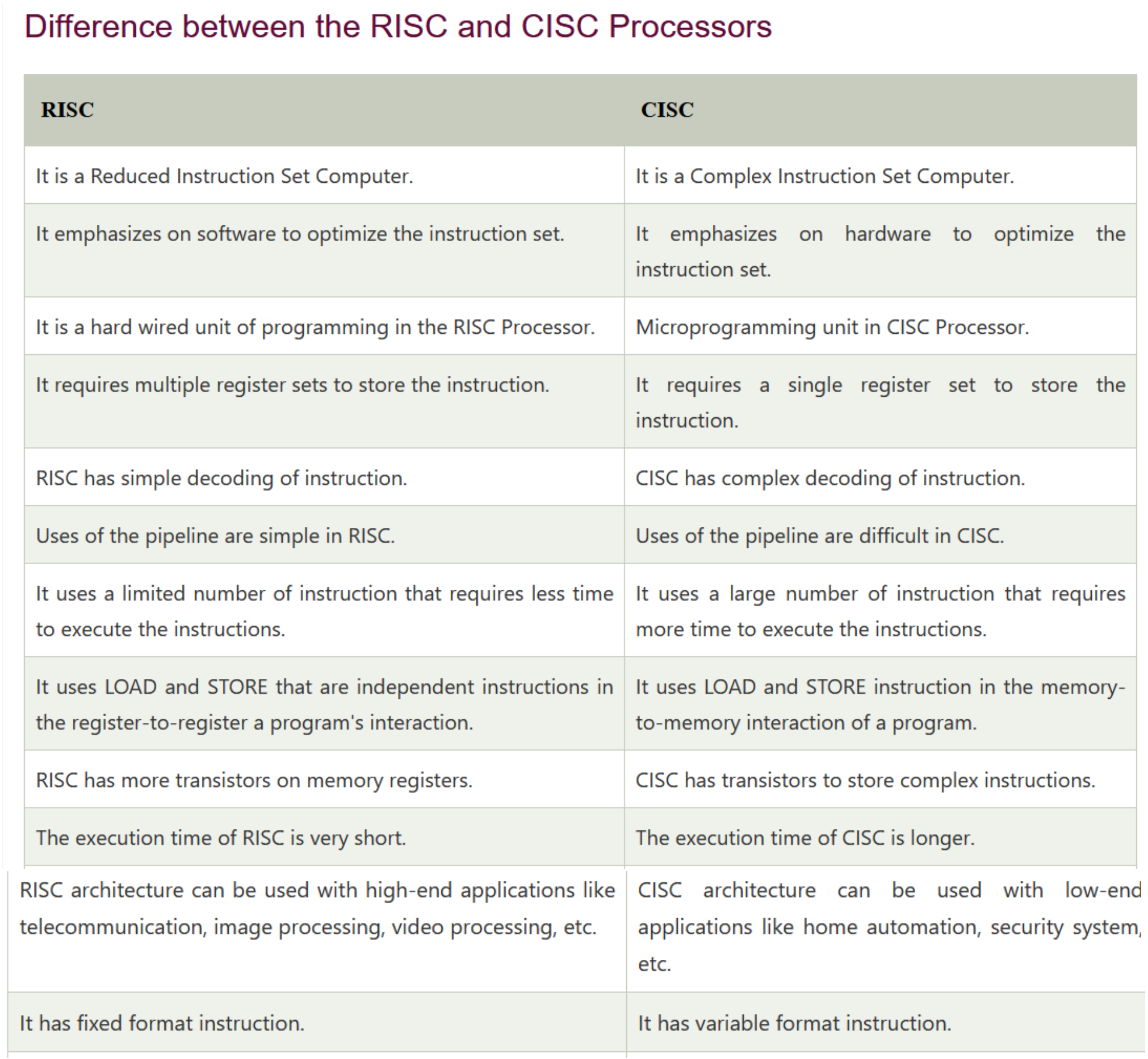
Processors in a System
A processor has two essential units −

- Program Flow Control Unit (CU)
- Execution Unit (EU)

The CU includes a fetch unit for fetching instructions from the memory. The EU has circuits that implement the instructions pertaining to data transfer operation and data conversion from one form to another.

The EU includes the Arithmetic and Logical Unit (ALU) and also the circuits that execute instructions for a program control task such as interrupt, or jump to another set of instructions.

25. What is the difference between RISC and CISC ?



Difference between the RISC and CISC Processors

RISC	CISC
It is a Reduced Instruction Set Computer.	It is a Complex Instruction Set Computer.
It emphasizes on software to optimize the instruction set.	It emphasizes on hardware to optimize the instruction set.
It is a hard wired unit of programming in the RISC Processor.	Microprogramming unit in CISC Processor.
It requires multiple register sets to store the instruction.	It requires a single register set to store the instruction.
RISC has simple decoding of instruction.	CISC has complex decoding of instruction.
Uses of the pipeline are simple in RISC.	Uses of the pipeline are difficult in CISC.
It uses a limited number of instruction that requires less time to execute the instructions.	It uses a large number of instruction that requires more time to execute the instructions.
It uses LOAD and STORE that are independent instructions in the register-to-register a program's interaction.	It uses LOAD and STORE instruction in the memory-to-memory interaction of a program.
RISC has more transistors on memory registers.	CISC has transistors to store complex instructions.
The execution time of RISC is very short.	The execution time of CISC is longer.
RISC architecture can be used with high-end applications like telecommunication, image processing, video processing, etc.	CISC architecture can be used with low-end applications like home automation, security system, etc.
It has fixed format instruction.	It has variable format instruction.

27. Explain the techniques used for selection of memory in embedded systems.

Memory Selection

Selection of suitable memory is very much essential step in high performance applications, because the challenges and limitations of the system performance are often decided upon the type of memory architecture.

Systems memory requirement depend primarily on the nature of the application that is planned to run on the system. Memory performance and capacity requirement for low cost systems are small, whereas memory throughput can be the most critical requirement in a complex, high performance system.

Following are the factors that are to be considered while selecting the memory devices,

- Speed
- Data storage size and capacity
- Bus width
- Latency
- Power consumption
- Cost

SRAM's have lower data storage and capacity hence they are suitable for lower end systems where as SDRAM for higher end systems with complex requirements.

Among the high speed types of SDRAM, DDR2 memory modules can have memory capacities from 256MB to 4GB capacities. Most of the DDR2 memory chips come in FBGA (Fine Ball Grid Array) package. The package allows higher memory densities in smaller space with better electrical properties. DDR2 memory uses 1.8V for power, resulting in lower power and cooler operation, whereas the DDR uses 2.5V.