Q.1 what do you understand by the embedded system ? Also explain Characteristics and applications of embedded systems.

Solution:

An embedded device is a machine that has software program embedded in PC hardware. It makes a device devoted to a selected a part of an application or fabricated from a bigger device. Depending on the application, embedded device can be programmable or non-programmable. Examples of embedded systems consist of diverse merchandise together with washing system, microwave ovens, cameras, printers, and motors. you can check [**the application of embedded systems in the medical field**](http://microcontrollerslab.com/embedded-systems-medical-applications/). They use [**microprocessors and microcontrollers**](http://microcontrollerslab.com/difference-between-microprocessor-and-microcontroller/) in addition to specially designed[**processors**](http://microcontrollerslab.com/embedded-processors-types/) consisting of digital signal processors (DSP**).**

CHARACTERISTICS OF EMBEDDED SYSTEM

* Using the embedded gadget definition it’s far possible to recognize the numerous basic traits one. Normally they’re:
* Embedded systems are designed for a specific challenge. Even though they use laptop strategies, they cannot be used as a widespread reason computer the use of a spread of different programs for one of a kind project. On this manner, their function can be focused on what they need to do, and they can consequently be made inexpensive and greater correctly.
* The software program for embedded systems is usually called firmware. In preference to being saved on a disc, where many applications can be stored, the single programmed for an embedded gadget is usually saved on the chip and its miles called firmware.
* Commonly, an embedded device executes a particular operation and does the similar always. As an example: a pager is continuously functioning as a pager.
* All of the computing structures have boundaries on layout metrics, but the ones can be in particular tight. Design calculation is a measure of executive functions like length, strength, price and also performance.
* It should carry out rapid sufficient and eat much less strength to increase battery existence.
* Several embedded systems have to continuously react to modifications inside the system and also calculate precise outcomes in actual time without any put-off. For instance, an automobile cruise controller; it constantly shows and responds to hurry & brake sensors. It should calculate acceleration/de-accelerations often in a restrained time; a behind schedule computation can outcome in letdown to manipulate the automobile.
* It has to be based totally on a microcontroller or microprocessor primarily based.
* An embedded system is built in with hardware and software program where the hardware is used for safety and performance and software is used for more flexibility and capabilities.

EMBEDDED SYSTEM APPLICATIONS

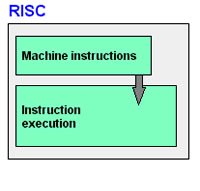
The programs of an embedded system fundamentals consist of smart cards, PC networking, satellites, telecommunications, digital consumer electronics, missiles, etc.

* Embedded systems in cars include motor control, Cruise manipulates, frame safety, engine safety, robotics in a meeting line, automobile multimedia, automobile enjoyment, e-com get entry to, mobiles etc.
* Embedded structures in telecommunications consist of networking, mobile computing, and wireless communications, and so on.
* Embedded structures in smart playing cards consist of banking, smartphone and protection systems.
* Embedded systems in satellites and missiles include defense, conversation, and aerospace.
* Embedded structures in computer networking & peripherals consist of photo processing, networking systems, printers, community cards, monitors, and shows.
* Embedded structures in virtual patron electronics include set-pinnacle boxes, dads, high definition TVs and digital cameras.

Q.2 Write down the difference between RISC and CISC.

Solution:

##### Key difference: The main difference between RISC and CISC is in the number of computing cycles each of their instructions take. The difference the number of cycles is based on the complexity and the goal of their instructions.

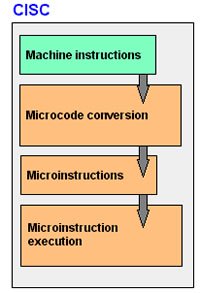
The term RISC stands for ‘Reduced Instruction Set Computer’. It is a CPU design strategy based on simple instructions and fast performance.

RISC is small or reduced set of instructions. Here, each instruction is meant to achieve very small tasks. In a RISC machine, the instruction sets are simple and basic, which help in composing more complex instructions. Each instruction is of the same length; the instructions are strung together to get complex tasks done in a single operation. Most instructions are completed in one machine cycle. This pipelining is a key technique used to speed up RISC machines.

RISC is a microprocessor that is designed to carry out few instructions at the same time. Based on small instructions, these chips require fewer transistors, which make the transistors cheaper to design and produce. Some other features of RISC include:

* Less decoding demand
* Uniform instruction set
* Identical general purpose register
* Simple addressing nodes
* Few data types in hardware

Also, while writing codes, RISC makes it easier by allowing the programmer to remove unnecessary codes and prevents wasting of cycles.



The term CISC stands for ‘Complex Instruction Set Computer’. It is a CPU design strategy based on single instructions, which are capable of performing multi-step operations.

CISC computers have shorted programs. It has a large number of complex instructions, which takes long time to execute. Here, a single set of instruction is covered in multiple steps; each instruction set has more than three hundred separate instructions. Most instructions are completed in two to ten machine cycles. In CISC, instruction pipelining is not easily implemented.

The CISC machines have good performances, based on the simplification of program compilers; as the range of advanced instructions are easily available in one instruction set. They design complex instructions in one simple set of instructions. They perform low level operations such as an arithmetic operation, or a load from memory and memory store. CISC makes it easier to have large addressing nodes and more data types in the machine hardware. However, CISC is considered less efficient than RISC, because of it inefficiency to remove codes which leads to wasting of cycles. Also, microprocessor chips are difficult to understand and program for, because of the complexity of the hardware.

Q.3 Justify the Utilization of ARM Processor and explain with its Architecture.

Solution: ARM, previously Advanced RISC Machine, originally Acorn RISC Machine, is a family of [reduced instruction set computing](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing) (RISC) [architectures](https://en.wikipedia.org/wiki/Instruction_set) for [computer processors](https://en.wikipedia.org/wiki/Central_processing_unit), configured for various environments. British company [ARM Holdings](https://en.wikipedia.org/wiki/ARM_Holdings) develops the architecture and licenses it to other companies, who design their own products that implement one of those architectures‍—‌including [systems-on-chips](https://en.wikipedia.org/wiki/System_on_a_chip) (SoC) and [systems-on-modules](https://en.wikipedia.org/wiki/System_on_module) (SoM) that incorporate memory, interfaces, radios, etc. It also designs [cores](https://en.wikipedia.org/wiki/Semiconductor_intellectual_property_core) that implement this [instruction set](https://en.wikipedia.org/wiki/Instruction_set) and licenses these designs to a number of companies that incorporate those core designs into their own products.

Introduction to The Arm Architecture

Enhancements to a basic RISC architecture enable Arm processors to achieve a good balance of high performance, small code size, low power consumption and small silicon area. The Arm Achitecture has evolved over time, introducing several architecture extensions throughout its history. These include:

* Security Extensions ([TrustZone](https://developer.arm.com/technologies/trustzone) technology)
* Advanced SIMD ([NEON](https://developer.arm.com/technologies/neon) technology)
* Virtualization Extensions, introduced in [Armv7-A](https://developer.arm.com/products/architecture/a-profile).
* Cryptographic Extensions, introduced in [Armv8-A](https://developer.arm.com/products/architecture/a-profile).

Arm produces a whole family of processors that share common instruction sets and programmer’s models and have some degree of backward compatibility. Processors implementing the Arm Architecture conform to a particular version of the architecture. These are:

* The [Architecture (‘A’) profile](https://developer.arm.com/products/architecture/a-profile) for high performance markets such as mobile and enterprise.
* The [Real-Time (‘R’) profile](https://developer.arm.com/products/architecture/r-profile) for embedded applications in automotive and industrial control.
* The [Microcontroller (‘M’) profile](https://developer.arm.com/products/architecture/m-profile)for the microcontroller market, meeting a broad range of gate count critical, real time and performance requirements.

Latest versions (Armv8 architectures)

The latest architecture Armv8 architecture has three variants of the architecture describing processors targeting different markets:

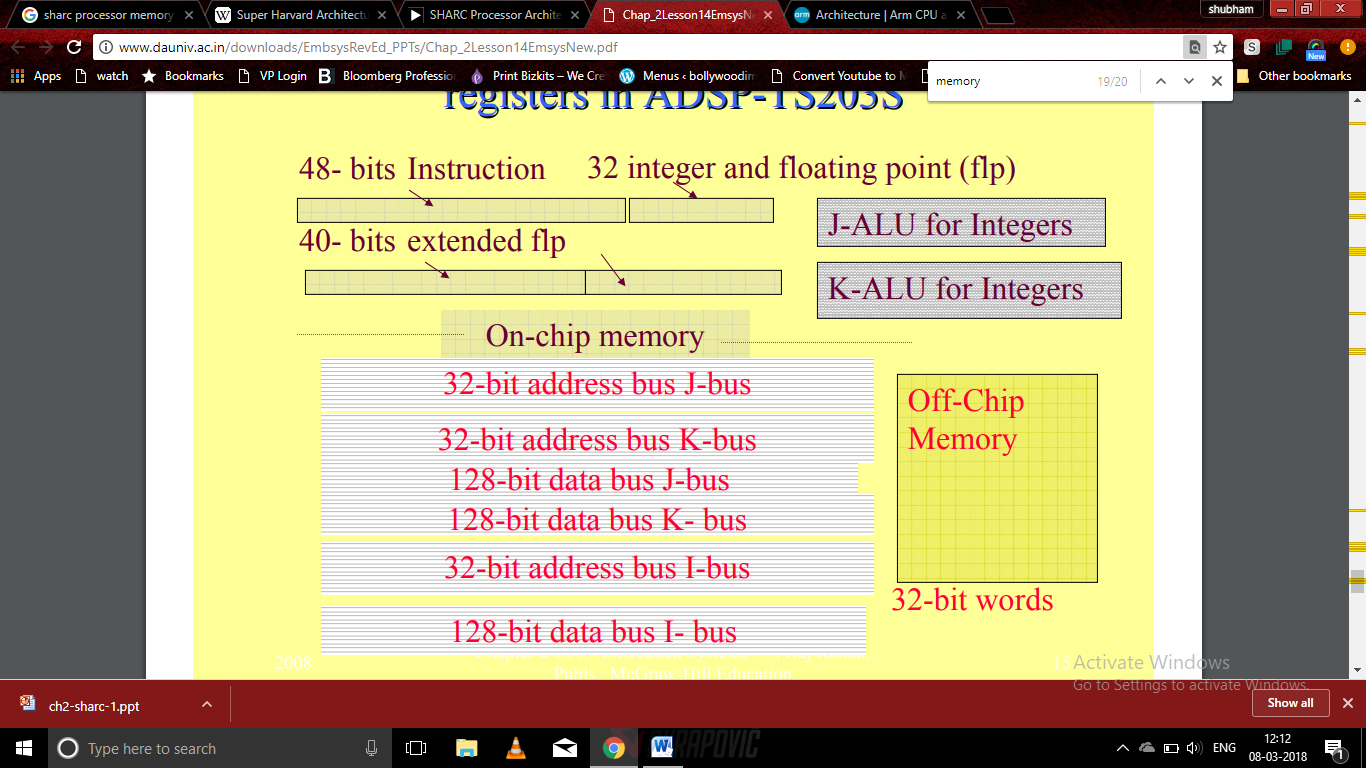
* The [Armv8-A](https://developer.arm.com/products/architecture/a-profile) architecture is the latest generation Arm architecture in the [A-profile](https://developer.arm.com/products/architecture/a-profile). It heralded the introduction of a 64-bit (AArch64) architecture alongside the well-established 32-bit (AArch32) architecture, and allows different levels of AArch64 and AArch32 support.
* The [Armv8-R](https://developer.arm.com/products/architecture/r-profile) architecture is the latest generation Arm architecture in the [R-profile](https://developer.arm.com/products/architecture/r-profile). This architecture includes a deterministic memory structure and a Memory Protection Unit (MPU), and supports the [A32](https://developer.arm.com/products/architecture/instruction-sets/a32-and-t32-instruction-sets)and [T32](https://developer.arm.com/products/architecture/instruction-sets/a32-and-t32-instruction-sets)instruction sets.
* The [Armv8-M](https://developer.arm.com/products/architecture/m-profile) architecture is the latest generation Arm architecture in the [M-profile](https://developer.arm.com/products/architecture/m-profile). It defines an architecture aimed at low cost deeply embedded systems, where low-latency interrupt processing is vital. It uses a different exception handling model to the other profiles and supports the [T32](https://developer.arm.com/products/architecture/instruction-sets/a32-and-t32-instruction-sets)instruction set.

Q.4 Give a brief introduction about the SHARC Processor and its memory organization.

Solution:

The SHARC is a [Harvard architecture](https://en.wikipedia.org/wiki/Harvard_architecture) [word-addressed](https://en.wikipedia.org/wiki/Word_addressing) [VLIW](https://en.wikipedia.org/wiki/VLIW) processor; it knows nothing of 8-bit or 16-bit values since each address is used to point to a whole 32-bit word, not just an [octet](https://en.wikipedia.org/wiki/Octet_(computing)). It is thus neither little-endian nor big-endian, though a compiler may use either convention if it implements 64-bit data and/or some way to pack multiple 8-bit or 16-bit values into a single 32-bit word. Analog Devices chose to avoid the issue by using a 32-bit char in their C compiler.

The word size is [48-bit](https://en.wikipedia.org/wiki/48-bit) for instructions, [32-bit](https://en.wikipedia.org/wiki/32-bit) for integers and normal floating-point, and 40-bit for extended floating-point. Code and data are normally fetched from on-chip memory, which the user must split into regions of different word sizes as desired. Small data types may be stored in wider memory, simply wasting the extra space. A system that does not use 40-bit extended floating-point might divide the on-chip memory into two sections, a 48-bit one for code and a 32-bit one for everything else. Most memory-related CPU instructions can not access all the bits of 48-bit memory, but a special 48-bit register is provided for this purpose. The special 48-bit register may be accessed as a pair of smaller registers, allowing movement to and from the normal registers.



Q.5 Write a short note on the following:

a. pipelining

solution:

Pipelining is an implementation technique where multiple instructions are overlapped in execution. The computer pipeline is divided instages. Each stage completes a part of an instruction in parallel. The stages are connected one to the next to form a pipe - instructions enter at one end, progress through the stages, and exit at the other end.

Pipelining does not decrease the time for individual instruction execution. Instead, it increases instruction throughput. The throughput of the instruction pipeline is determined by how often an instruction exits the pipeline.

Because the pipe stages are hooked together, all the stages must be ready to proceed at the same time. We call the time required to move an instruction one step further in the pipeline a machine cycle . The length of the machine cycle is determined by the time required for the slowest pipe stage.

The pipeline designer's goal is to balance the length of each pipeline stage . If the stages are perfectly balanced, then the time per instruction on the pipelined machine is equal to

b. Execption

solution:

An exception is a synchronous event that occurs during the execution of a thread that disrupts the normal flow of instructions. If exceptions are not properly processed during program execution, severe consequences, such as system failures, can occur. Exception handling is extremely important, especially in embedded systems, to avoid these failures, and improves the robustness of the software. Properly implemented exception handling can also aid in software execution recovery so that an application can proceed with its task after an exception has occurred.