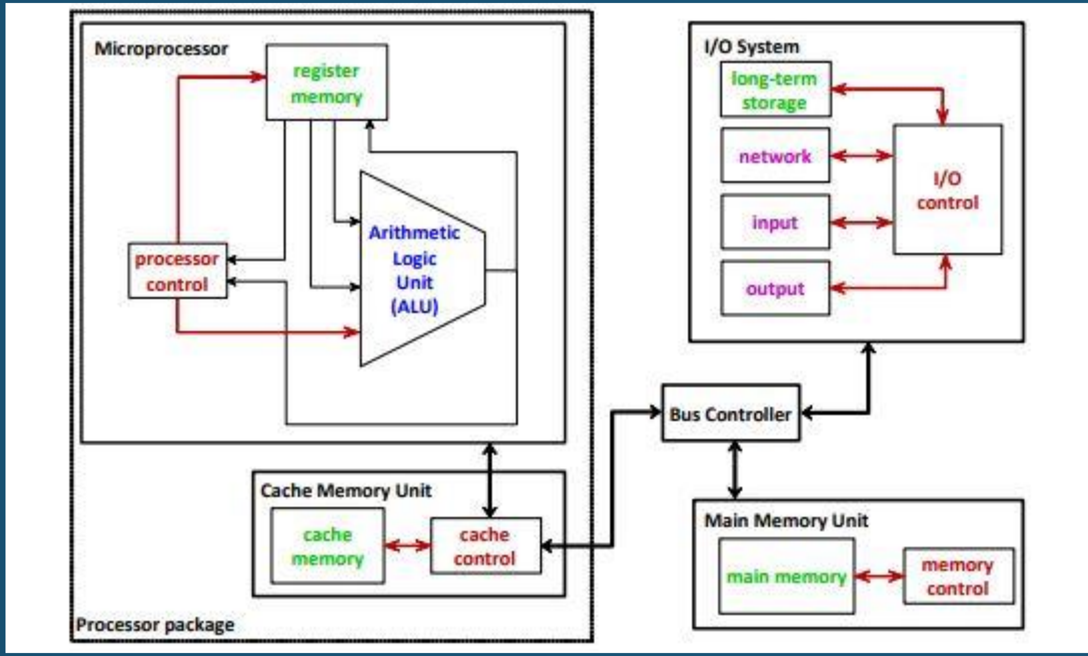
AMERICAN INTERNATIONAL UNIVERSITY – BANGLADESH(AIUB)

Where leaders are created

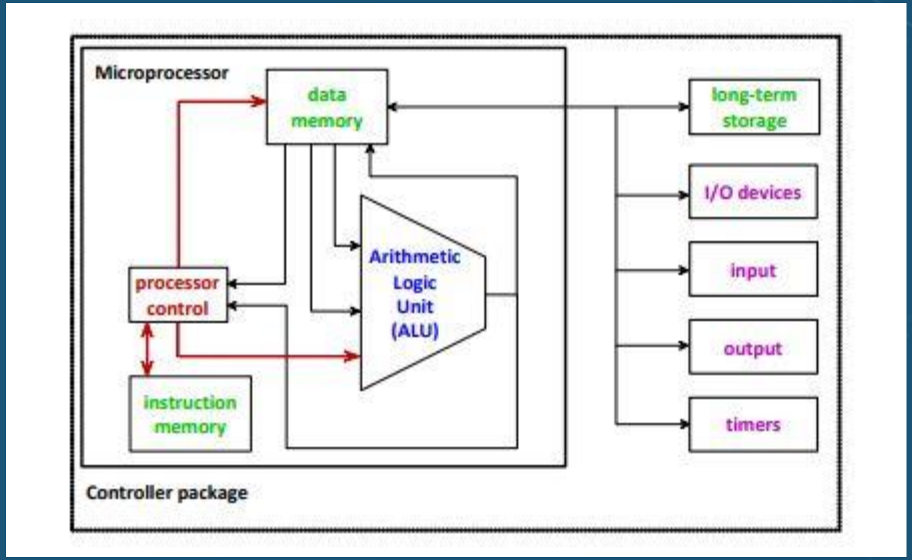
FUNDAMENTALS OF MICROPROCESSOR AND EMBEDDED SYSTEM

SOME IMPORTANT DIFFERENCES :

Microprocessor



Microcontroller.



SOME IMPORTANT DIFFERENCES :

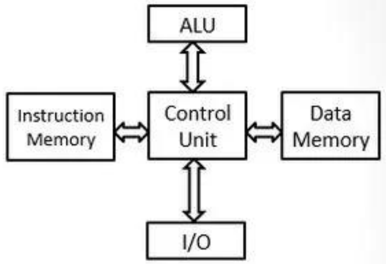
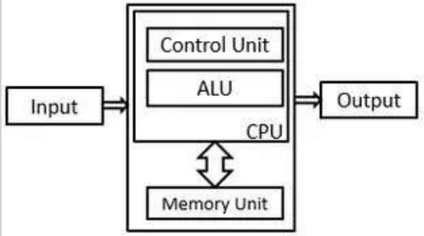
Microprocessor

- Used in personal computers or laptops.
- More expensive than microcontrollers.
- Power consumption is high as clock speed is low.

Microcontroller

- Used in home appliances like refrigerator, washing machine, etc. along with PC.
- Less expensive than microprocessors.
- Power consumption is lower as clock speed is high.

<u>SOME IMPORTANT DIFFERENCES :</u>

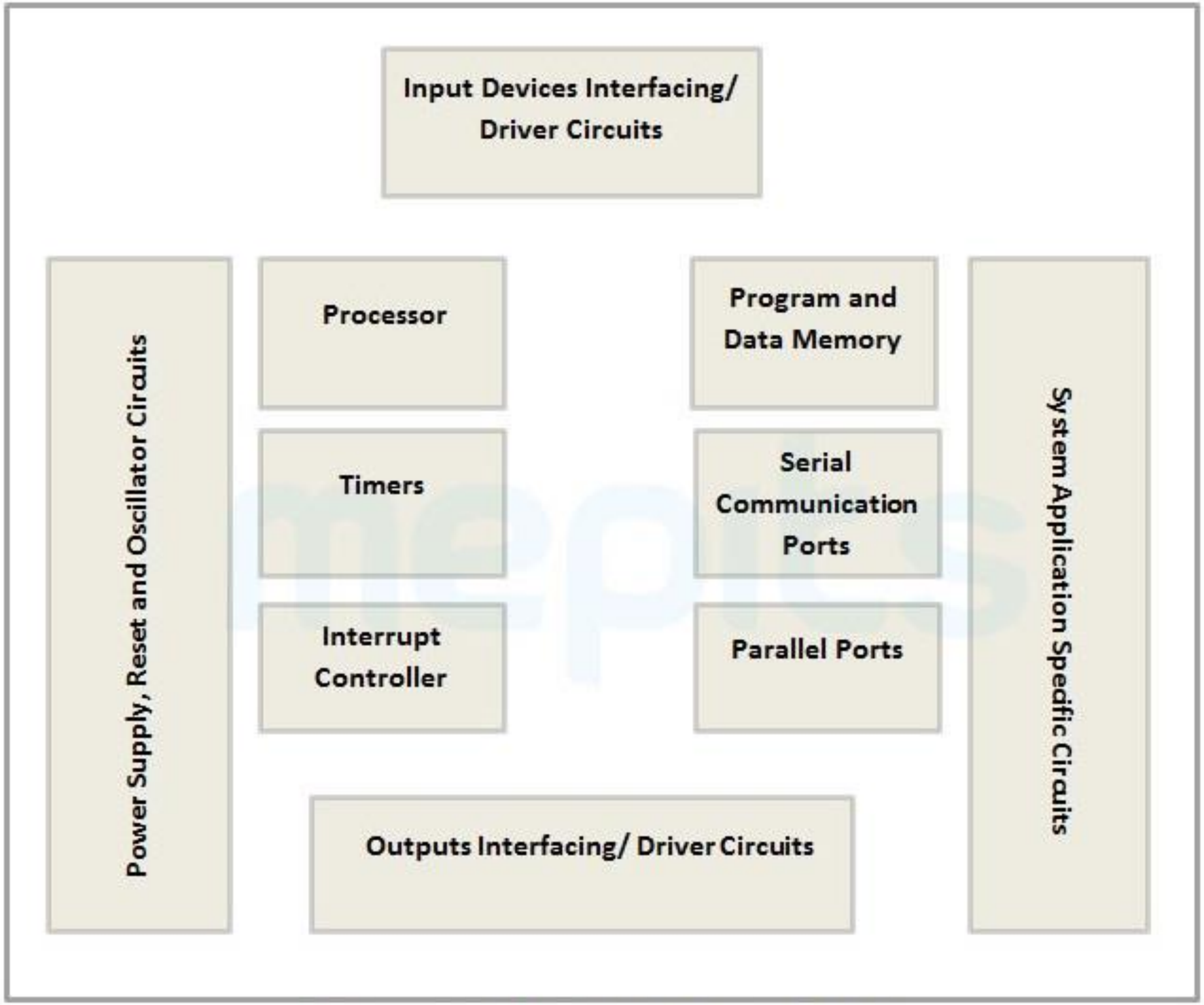
Point of Comparison	Harvard Architecture	Von Neumann Architecture
Arrangement	In Harvard architecture, the CPU is connected with both the data memory (RAM) and program memory (ROM), separately. Some of the earlier microcontrollers used this architectures.  Harvard Model	In Von-Neumann architecture, there is no separate data and program memory. Instead, a single memory connection is given to the CPU. Some of the earlier microprocessors used in this architecture.  Von Neumann Model
Hardware requirements	It requires more hardware since it will be requiring separate data and address bus for each memory.	In contrast to the Harvard architecture, this requires less hardware since only a common memory needs to be reached.

SOME IMPORTANT DIFFERENCES :

Point of Comparison	Harvard Architecture	Von Neumann Architecture
Space requirements	This requires more space.	Von-Neumann Architecture requires less space.
Speed of execution	Speed of execution is faster because the processor fetches data and instructions simultaneously .	Speed of execution is slower since it cannot fetch the data and instructions at the same time.
Space usage	It results in wastage of space since if the space is left in the data memory then the instructions memory cannot use the space of the data memory and vice-versa.	Space is not wasted because the space of the data memory can be utilized by the instructions memory and vice-versa.
Controlling	Controlling becomes complex since data and instructions are to be fetched simultaneously.	Controlling becomes simpler since either data or instructions are to be fetched at a time.

EMBEDDED SYSTEM

An embedded system is **a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system**. At the core is an integrated circuit designed to carry out computation for real-time operations.



Components of Embedded System Hardware

EMBEDDED SYSTEM

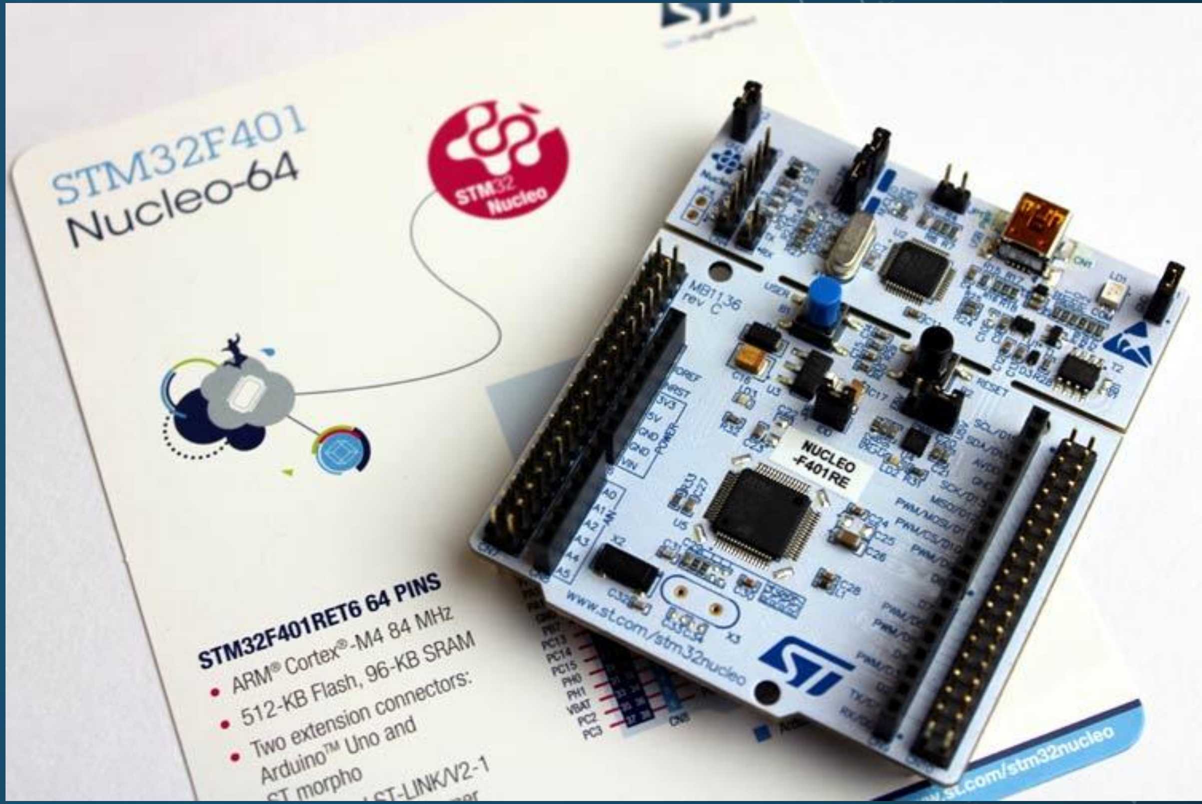
- An embedded System can be best described as a system that has both the hardware and software and is designed to do a specific task. Embedded systems may only have some specific range of operations and can be programmed to perform a wide range of tasks.

- The Internet of Things (IoT) refers to physical objects equipped with sensors, processing power, software, and other technologies that connect to and exchange data with other devices and systems over communication networks. The devices in IoT do not need to be connected to the public internet; rather must be connected to a network to be individually addressable.

- The Internet of Things has evolved as a result of the convergence of various technologies, such as ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning.

NEW HARDWARE OPTIONS EMERGED FOR EMBEDDED SYSTEMS AREAS FOLLOWS:

1) **STMicroelectronics offered joint usage of microcontroller (Arm Cortex A) and microprocessor (Cortex M CPUs.**
2) **Development boards consisting of graphical processing units (GPU) allow parallel processing via high-level programming languages.**
3) **Recents advances in deep learning and neural networks also led to devices consisting of neural processing units (NPU or TPU) dedicated to neural network implementations.**



CLASSIFICATIONS OF EMBEDDED SYSTEMS:

Based on processing power, cost, functionality, and architecture, classifications are as follows:

1) **Small-scale embedded system:** Mainly based on 8 or 16-bit architecture. The operating voltage is 5V. Commonly uses EEPROM to store programs and instructions. C language is used for programming.to develop such a system, a board-level design is preferred rather than a chip-level design.

2) **Medium-scale embedded system:** Mainly based on 16 or 32-bit architecture. Mostly used for digital signal processing (DSP). Reduced instruction set computing (RISC) is most preferable for supporting transmission control protocol (TCP) or internet protocol stacking and networking. Automation systems are examples.

3) **Sophisticated embedded systems:** both complex instruction set computing CISC and RISC are supported by these systems. It also supports a system-on-chip (SOC). The software running on these systems mostly a real-time operating system (RTOS).

NEW HARDWARE OPTIONS EMERGED FOR EMBEDDED SYSTEMS AREAS FOLLOWS:

- At present, some of the popular microcontroller families in the market are:
  - ATMega family: ATMega328P, ATMega32
  - Pic-chips: Pic24, Pic33 etc
  - ARM processors: Raspberry Pi, TM4C chips, STM32 F401

- In Bangladesh, ATMega based Arduino boards have gained wide popularity due to easy availability and low price

- We are going to mostly focus on the ATMega328P chip during midterm and then on the Arm processors in final term.

REAL-LIFE EXAMPLES OF EMBEDDED SYSTEMS:

Exceptionally versatile and adaptable, embedded systems can be found in all smart devices today. It is difficult to find a single portion of modern life that doesn't involve this technology. Here are some of the real-life examples of embedded system applications.

Central heating systems

GPS systems

Fitness trackers

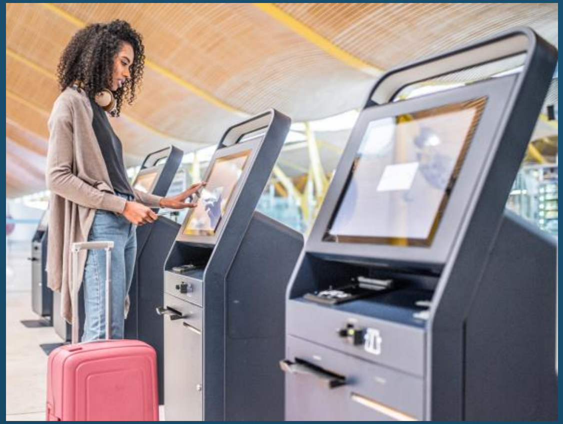
REAL-LIFE EXAMPLES OF EMBEDDED SYSTEMS:

Automatic fare collection(AFC)

ATM systems

Factory robots

Medical devices



Self-service kiosks

Electric vehicle charging stations

REFERENCES:

1. Embedded Systems and Robotics with Open Source Tools By Nilanjan Dey, Amartya Mukherjee
2. Embedded System Design with ARM Cortex-M Microcontrollers: Applications with C, C++ and MicroPython by Cem Unsalan
3. https://www.digi.com/blog/post/examples-of-embedded-systems