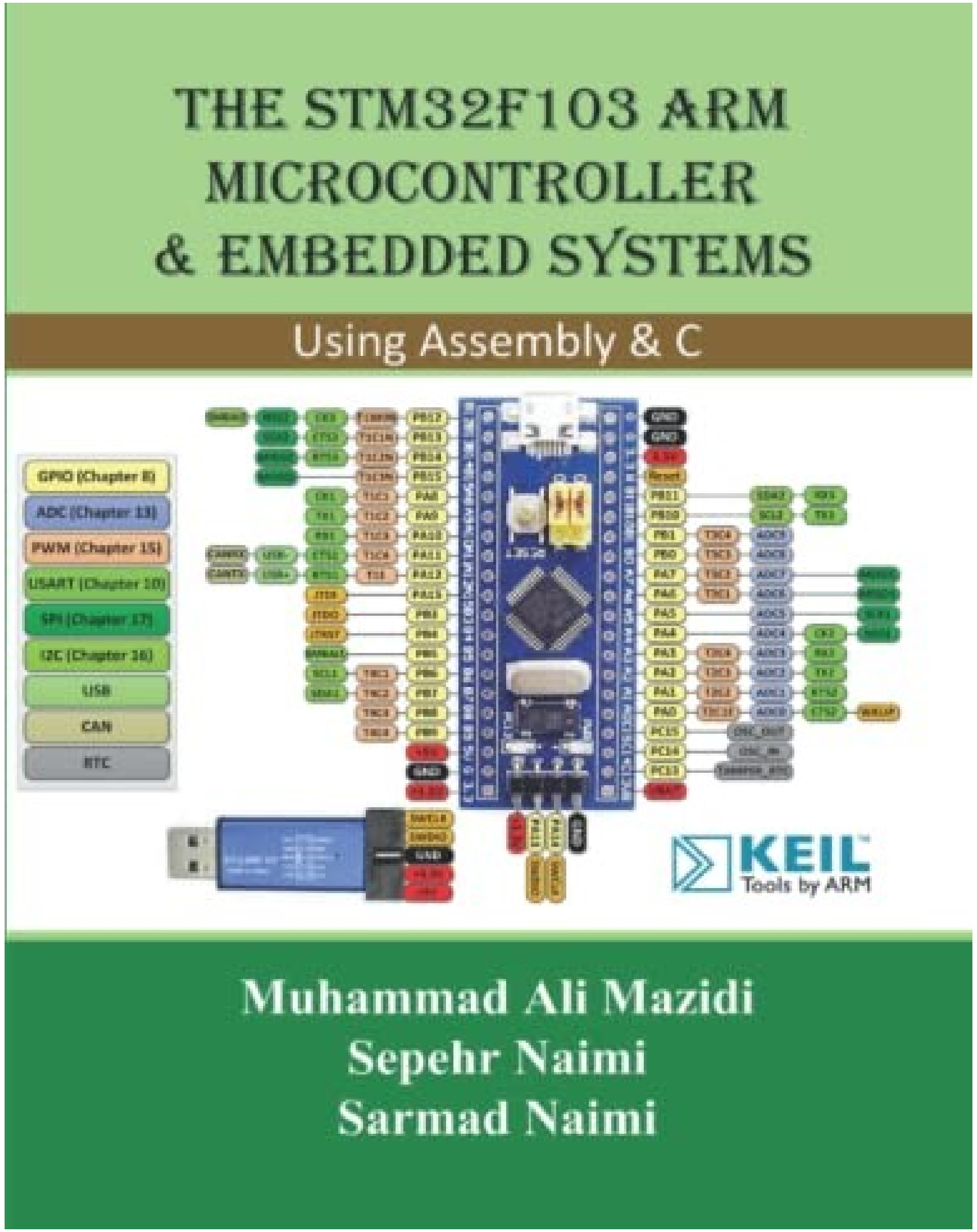
Introduction to Microcontroller

Modified By
Niloy Goswami

Prepared By
Prof. Dr. Engr. Muhibul Haque Bhuyan

Department of Electrical and Electronics Engineering
American International University-Bangladesh(AIUB)

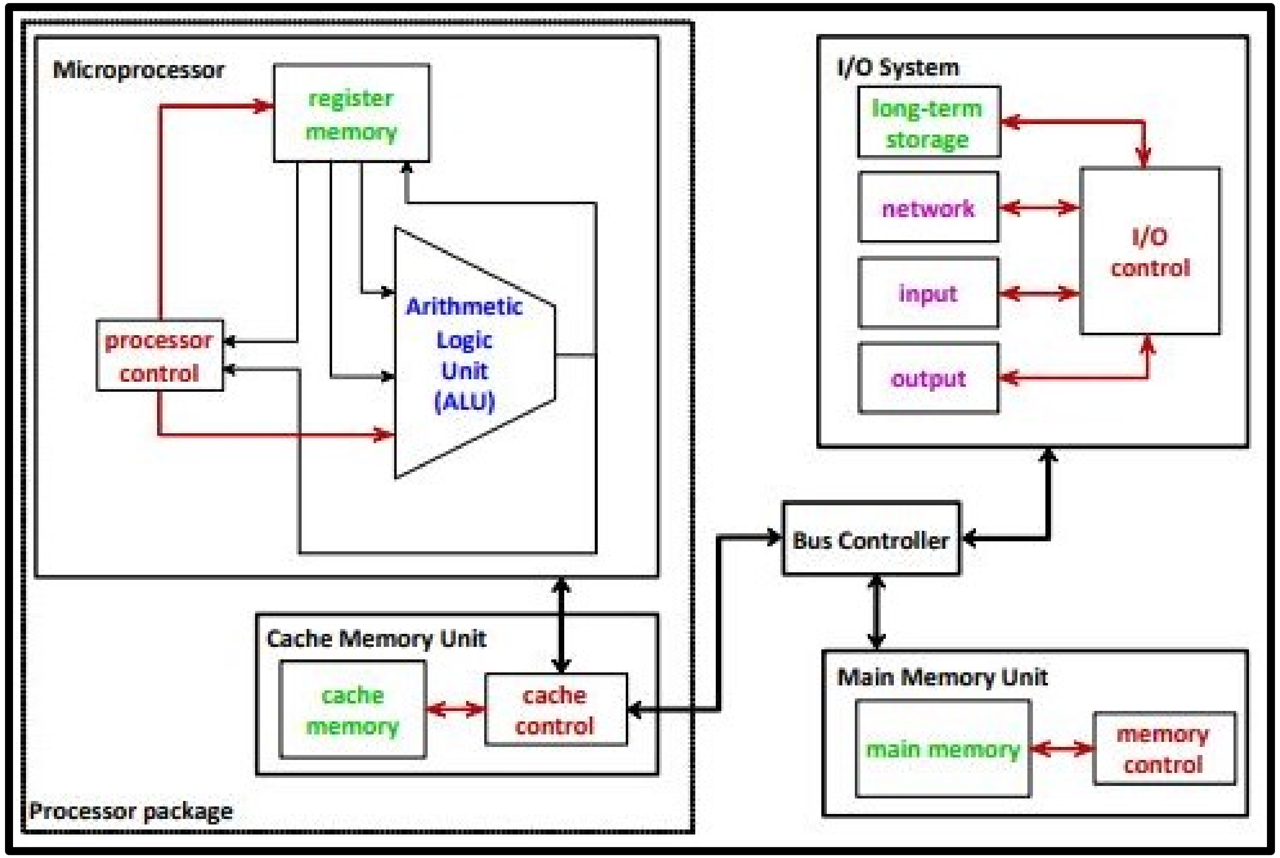

THE STM32F103 ARM
MICROCONTROLLER
& EMBEDDED SYSTEMS
Using Assembly & C
GPIO (Chapter 8)
ADC (Chapter 13)
PWM (Chapter 15)
USART (Chapter 10)
SPI (Chapter 17)
I2C (Chapter 16)
USB
CAN
RTC
KEIL
Tools by ARM
Muhammad Ali Mazidi
Sepehr Naimi
Sarmad Naimi

Outline

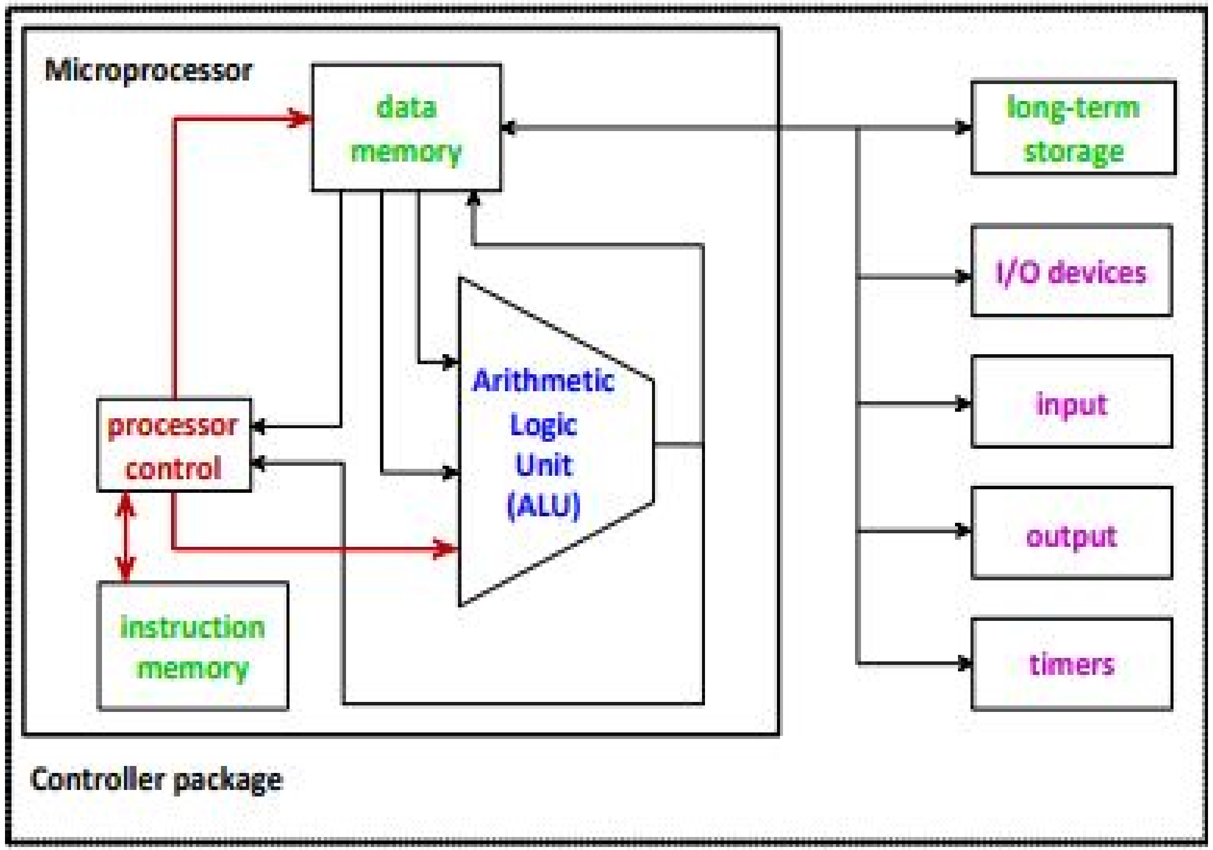
- Differences between microcontroller and microprocessor
- Differences between Harvard and Von Neumann Architectures
- Introducing STM32 Microcontroller Board
- Embedded Systems and IoT
- Architecture of a Modern Microcontroller

Some important Differences

Microprocessor



Microcontroller



Comparison of Microprocessor and Microcontroller

Microprocessor

- The integrated single-chip CPU of a computer system.
- Controls the memory and I/O through a series of connections called busses.
- CPU/Microprocessor:
  - ❖Performs arithmetic or logic operations
  - ❖Temporarily stores data and instructions in internal high-speed registers

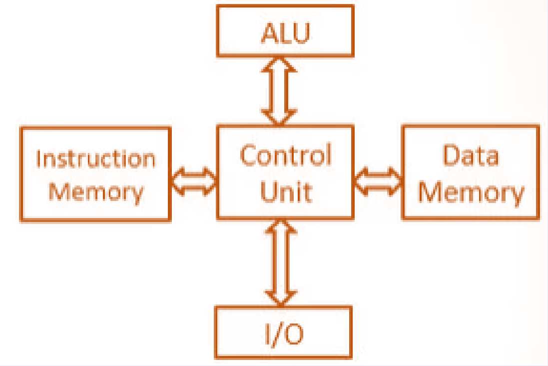
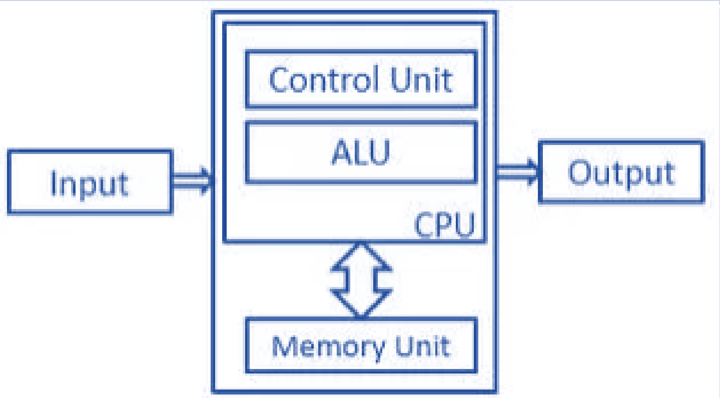
Examples: AMD Athlon, Intel Core i7

Microcontroller

- Single-chip microcomputers with limited capabilities.
- Microprocessor, memory, and I/O – all fabricated in the same chip.
- On-chip timers, A/D, and D/A converters.
- Appropriate for dedicated applications.

Examples: STM32, Arduino UNO, ATmega328

Architectures: Harvard vs. Von Neumann

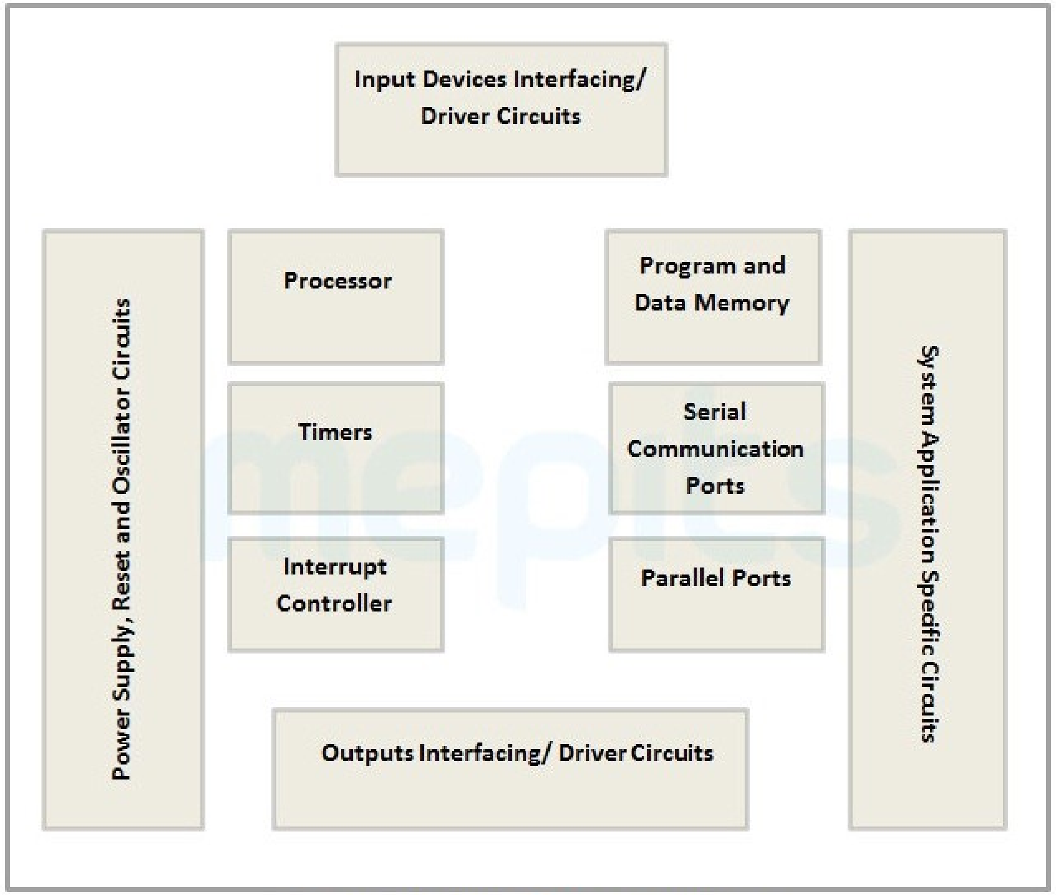
Point of Comparison	Harvard Architecture	Von Neumann Architecture
Arrangement	• Memory is divided into two memories: - Data memory - Instruction memory • Term originated from Harvard Mark I relay based computer system • Harvard architecture is often found in microcontrollers and digital signal processors (DSPs)	• All memory is capable of storing all program elements, data and instructions (single memory space). Also referred to as Princeton Architecture • Based on the “First Draft of a Report on the EDVAC” by John Von Neumann published in 1945 • Most general-purpose computers, including most personal computers, use this.
System diagrams		

Architectures: Harvard vs. Von Neumann

Point of Comparison	Harvard Architecture	Von Neumann Architecture
Addressing mode	• Distinct program and data address spaces • E.g. Instruction address zero is not the same as data address zero	• Same memory is capable of storing both data and instructions • Effectively no distinction between instructions and data
Speed of execution	• Separate pathways for data and instruction memories • Capable of executing instructions in a single clock cycle • Data and instructions do not contend for a single memory pathway • Faster even without cache memories	• Shared pathway for data and instruction memories • Requires at least 2 clock cycles to execute an instruction • Suffers from “Von Neumann Bottleneck” – the limited data transfer rate between CPU and memory • Introduction of cache memories can mitigate the bottleneck

Embedded System

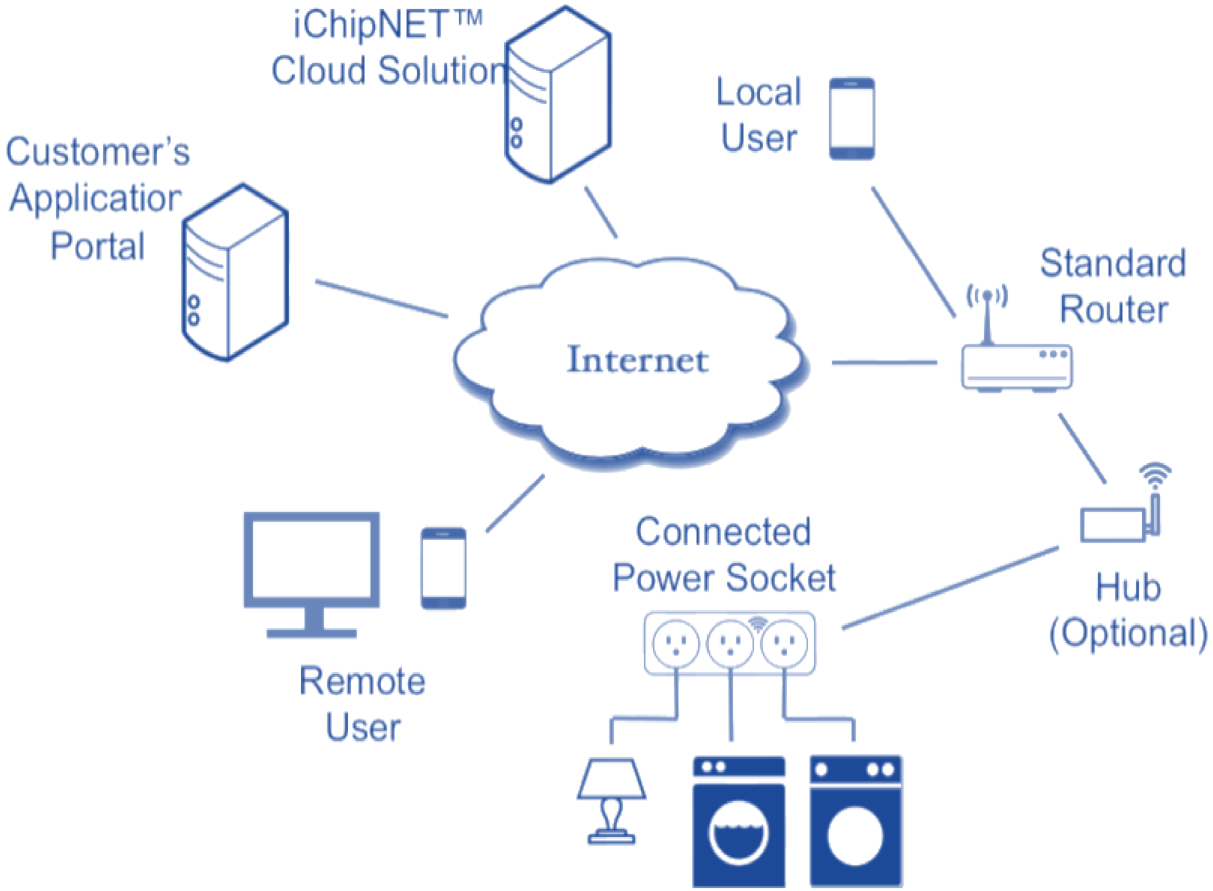
- A microprocessor-based system designed to control a dedicated function, or a range of functions as opposed to general-purpose computers
- A combination of computer hardware and software and additional parts, either mechanical or electronic
- Often a component within a larger system
- General-purpose computer interfaces to numerous embedded systems, such as keyboard and mouse
- Common examples from everyday household appliances include a washing machine and a microwave oven



Components of Embedded System Hardware

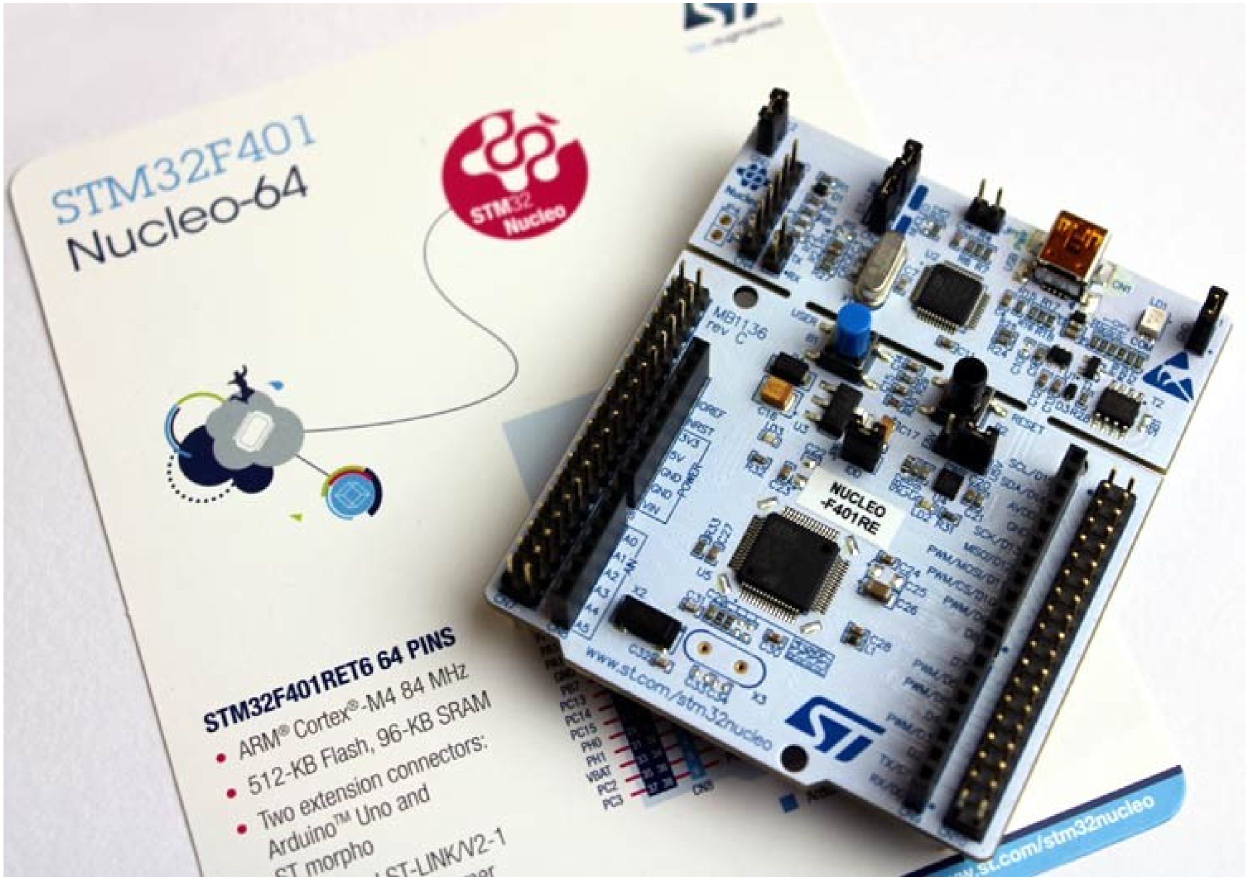
Internet of Things (IoT)

- A network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity, enabling these objects to connect and exchange data.
- Inter-networking of embedded devices, sensors, and computers
- Can collect and distribute a large amount of data
- Connected devices:
  - ❖ Only need to be connected to a network
  - ❖ Need to be individually addressable
  - ❖ Do not necessarily need to be connected to the public internet



New Hardware Emerged for Embedded Systems

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) Recent advances in deep learning and neural networks also led to devices consisting of neural processing units (NPU or TPU) dedicated to neural network implementations.



New Hardware Emerged for Embedded Systems

- 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 the midterm and then on the ARM processors in the final term.

Real-life Examples of embedded Systems

Automatic Fare Collection (AFC)

ATM Systems

Factory Robots

Medical Devices

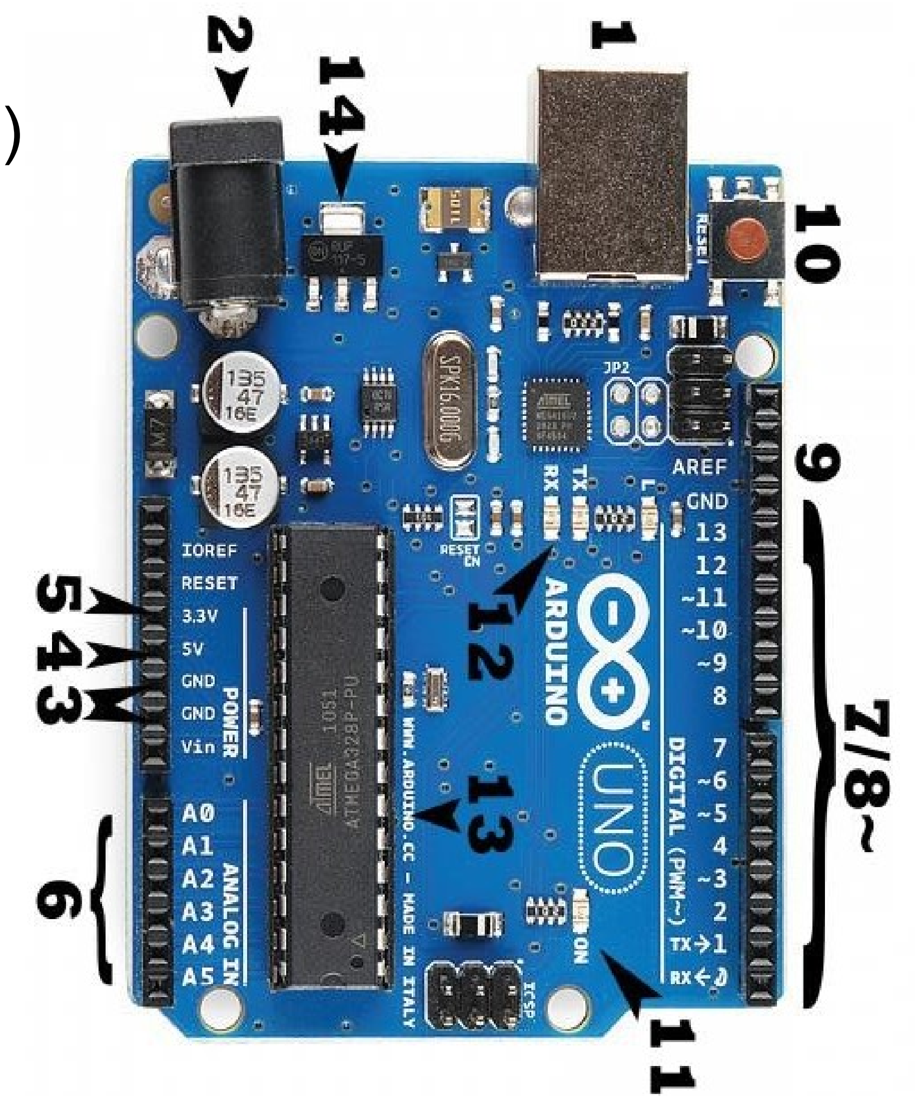
Self-Service Kiosks

Electric Vehicle Charging Stations

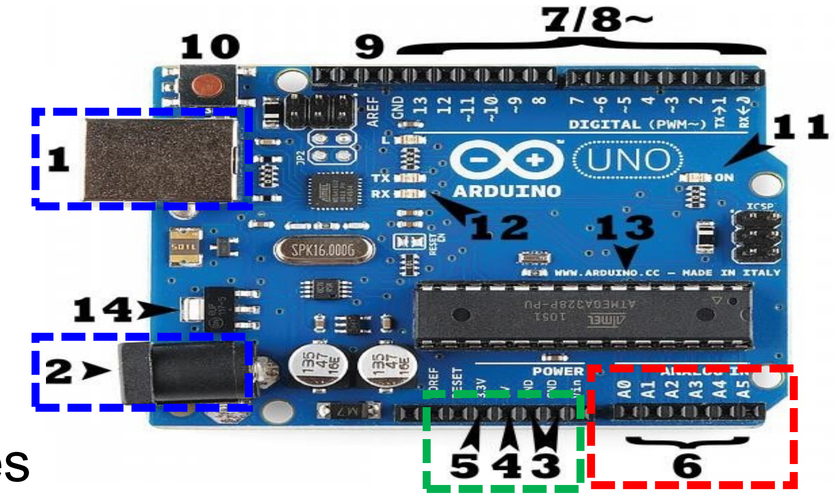
Architecture of a Modern Microcontroller

Basics Features of Arduino UNO

- ❑ Consists of:
  - A programmable microcontroller (ATMega328) and
  - Integrated Development Environment or IDE
- ❑ Uses a computer to write and upload code (written in the easier version of C++) to the microcontroller
- ❑ Doesn’t need any additional programmer/burner
- ❑ Operating voltage ranges from 1.8 V to 5.5 V
- ❑ Digital pins: 14 (of which 6 provide PWM output)
- ❑ DC Current per I/O pin: 20 mA



Arduino UNO Pin Configuration



1. Power (USB/Barrel Jack)
Every Arduino board needs a power source through a USB cable coming from your computer **(1)** or a wall power supply that is terminated in a barrel jack **(2)**.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF):

GND (3): Short for 'Ground' and **5V (4) & 3.3V (5)**: used as supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power.

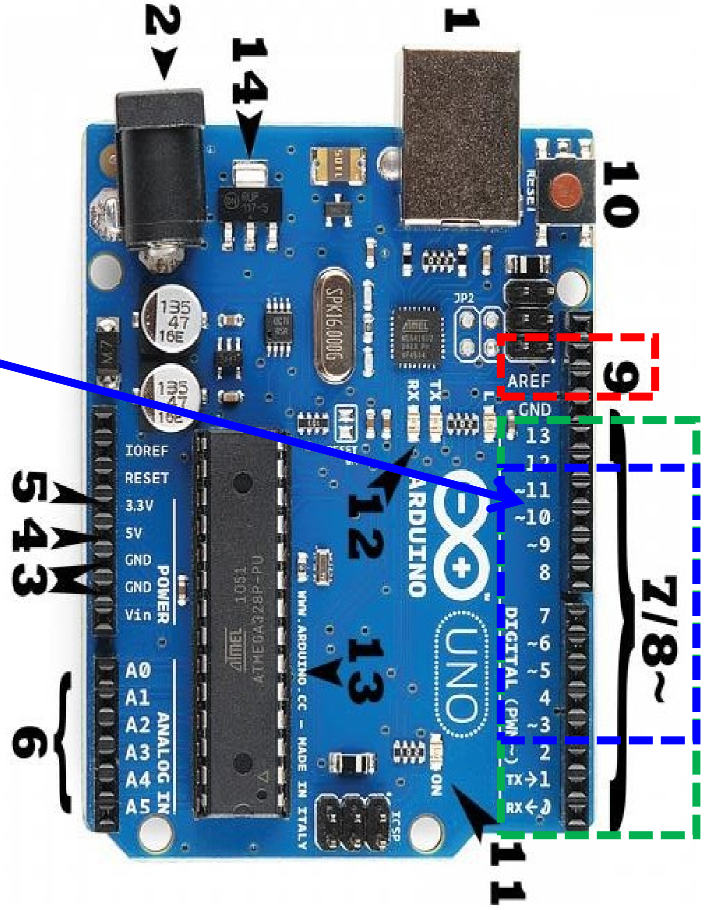
Analog Pins (6): The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog Input pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

Arduino UNO Pin Configuration

Digital (7): Across from the analog pins are the digital pins (0 through 13 on the UNO).

PWM (8): You may have noticed the tilde (~) sign next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as Pulse Width Modulation (PWM, think of these pins as being able to simulate analog output (like fading an LED in and out).

AREF (9): Analog reference input voltage pin to convert it into digital. It specifies the top value for the input range, and consequently each discrete step in the converted output. Most of the time you can leave this pin alone.



Arduino UNO Pin Configuration

Reset Button: Pushing it will temporarily connect the reset pin to the ground and restart any code that is loaded on the Arduino.

Power LED indicator (11): If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit.

TX RX LEDs (12): These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (for example, when we are loading a new program onto the board).

Main IC (Integrated Circuit) (13): The main IC on the Arduino is the ATmega328 microcontroller of the ATMEL company., where we have to upload the code.

Voltage Regulator: The voltage regulator **(14)** is used to control the amount of voltage that is let into the Arduino board. But Arduino should not be powered by more than 20 V.

Example of a Modern Microcontroller: ATmega328

ATmega-328 is basically an Advanced Virtual RISC (AVR) microcontroller.

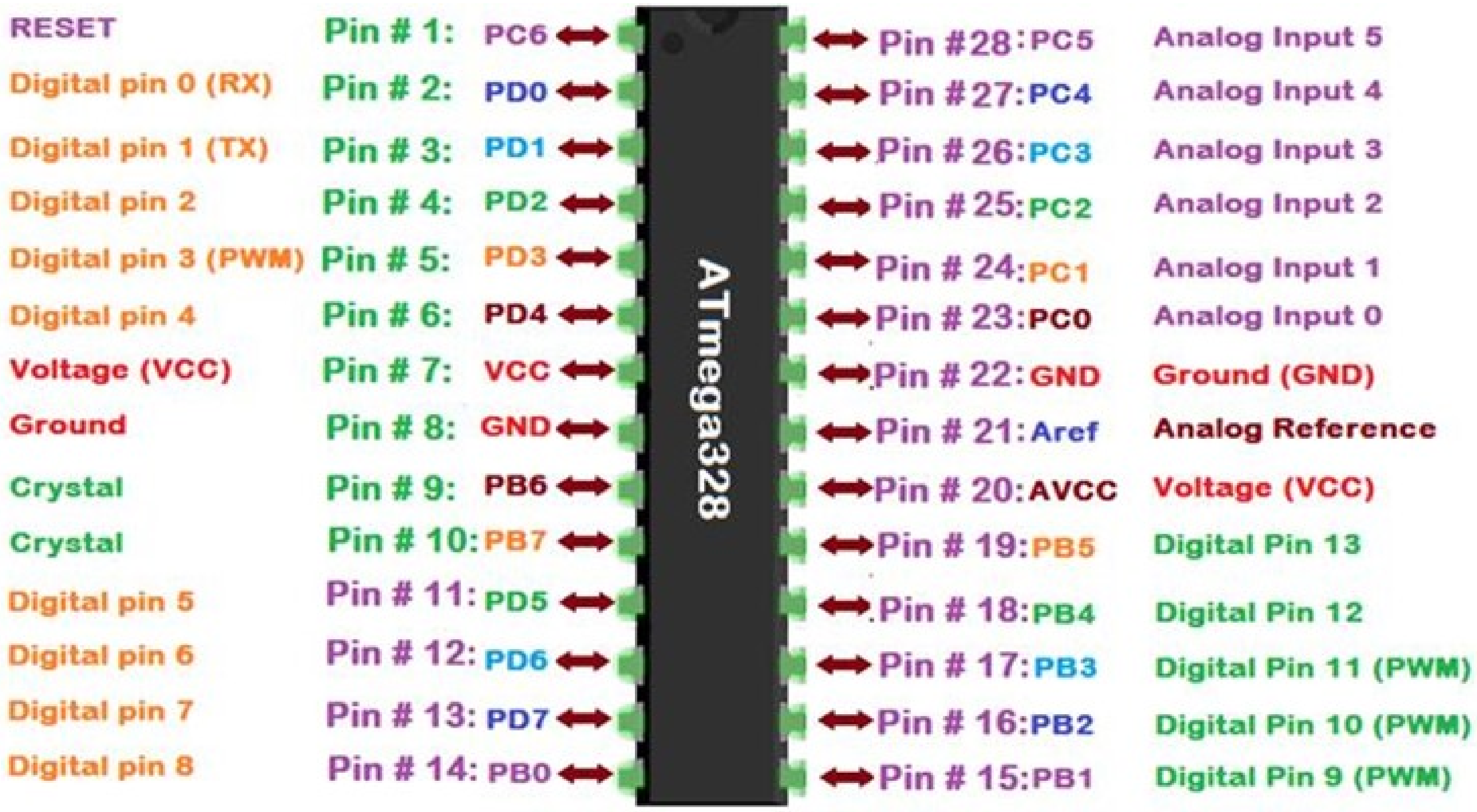
ATmega-328 has **32 KB internal built-in memory**.

- ATmega328 supports data up to **eight (8) bits and has 28 Pins.** AVR Microcontroller, manufactured by Microchip, follows RISC Architecture and **has a flash-type program memory of 32 KB**.
- ATmega328 has **1 KB Electrically Erasable Programmable Read Only Memory (EEPROM)**. For an EEPROM, if the electric supply is removed from the microcontroller, it can store the data. After providing an electric supply it can provide previous results.
- ATmega328 has **2 KB Static Random Access Memory** (SRAM).
- It has **8 pins for ADC operations**, which all combine to form Port A (PA0-PA7).

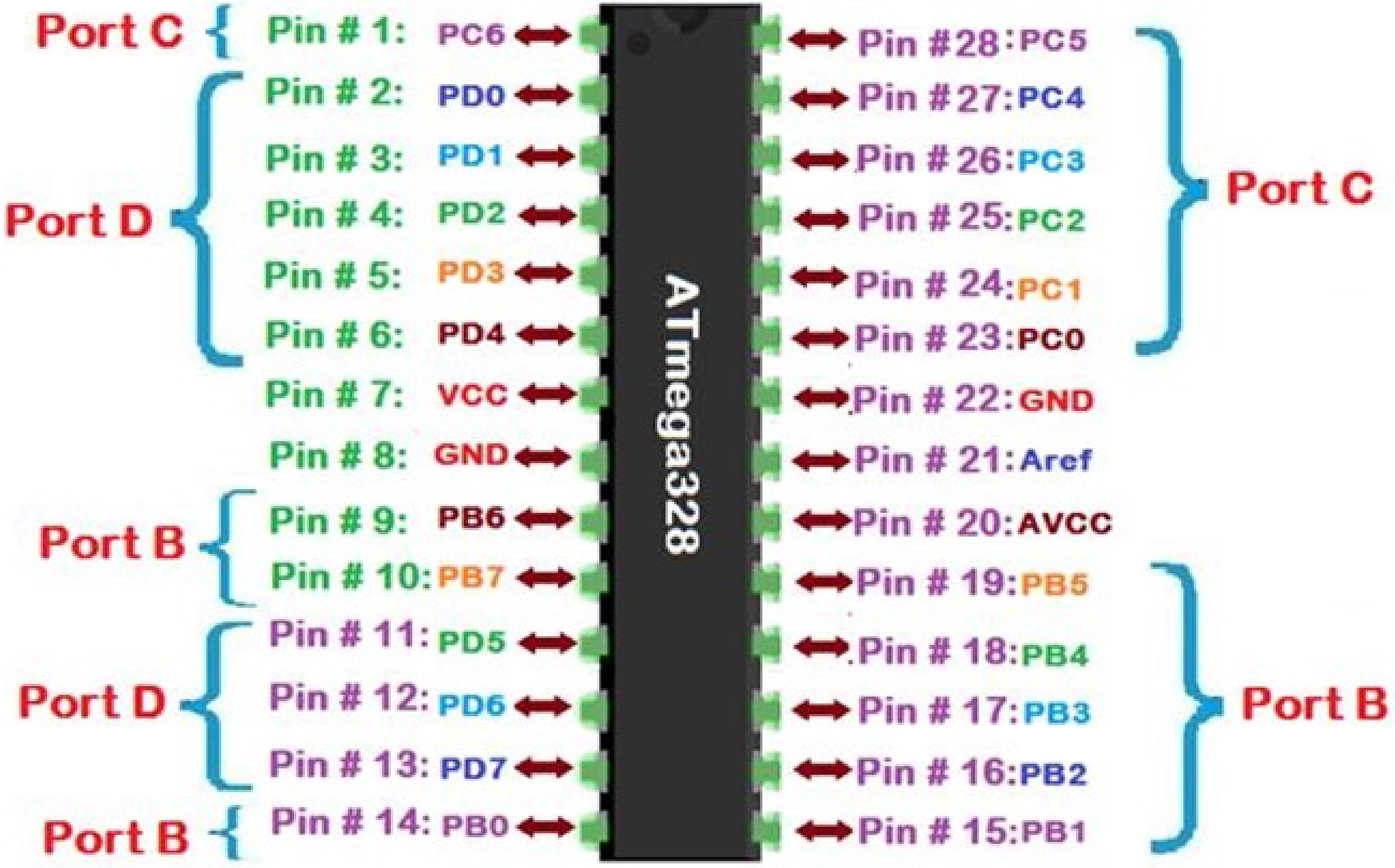
Example of a Modern Microcontroller: ATmega328

- It also has 3 built-in Timers, two of them are 8 Bit timers while the third one is a 16-Bit Timer.
- Arduino UNO is based on Atmega328 Microcontroller. It is UNO’s heart.
- It operates ranging from 3.3 V to 5.5 V but normally we use 5 V as a standard.
- Its excellent features include cost efficiency, low power dissipation, programming lock for security purposes, real timer counter with a separate oscillator.
- It is normally used in Embedded System applications.
- Moreover, ATmega328 has several different features which make it the most popular device in today’s market. These features consist of advanced RISC architecture, good performance, 6 PWM pins, programmable Serial USART, programming lock for software security, throughput up to 20 MIPS, etc.

Arduino ATmega328 Pin Configuration



Arduino ATmega328 Port Configuration



Arduino ATmega328 Pin Description

Functions associated with the pins must be known in order to use the device appropriately:

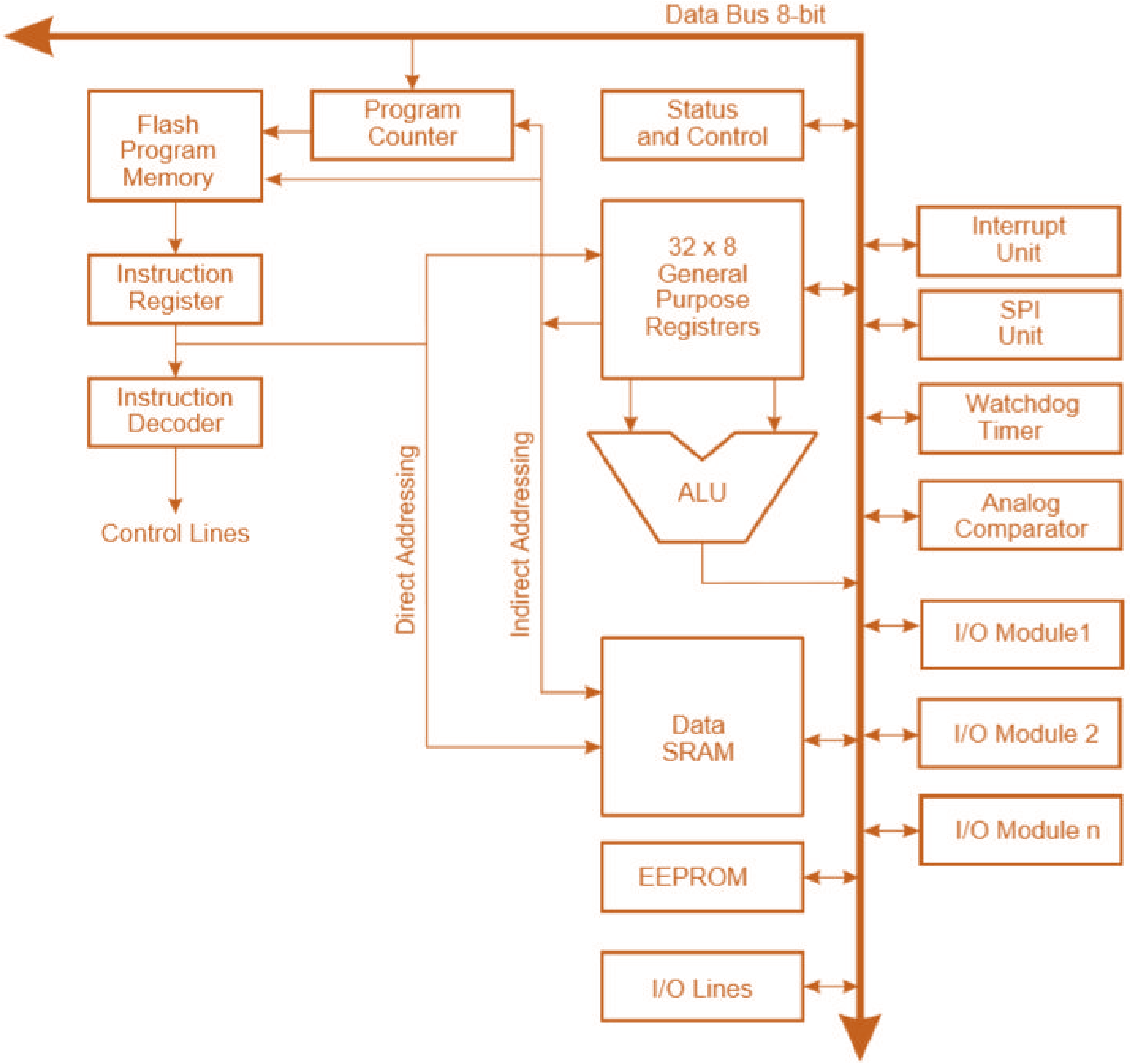
- VCC is a digital voltage supply.
- AVCC is a supply voltage pin for analog to digital converter.
- GND denotes Ground and it has a 0V.

ATmega-328 pins are divided into different ports which are given in detail below:

- ✓ Port A consists of the pins from PA0 to PA7. **These pins serve as an analog inputs to convert from analog to digital. Port A** acts as **an eight (8) bit bidirectional input/output port** if ADC is not required.
- ✓ Port B consists of the pins from PB0 to PB7. This port is an **8-bit bidirectional port having an internal pull-up resistor.**
- ✓ Port C consists of the pins from PC0 to PC7. The **output buffers of port C have symmetrical drive characteristics with source capability as a well-high sink.**
- ✓ Port D consists of the pins from PD0 to PD7. It is also **an 8-bit input/output port having an internal pull-up resistor.**
- ✓ AREF is an **analog reference pin for analog to digital converter**.

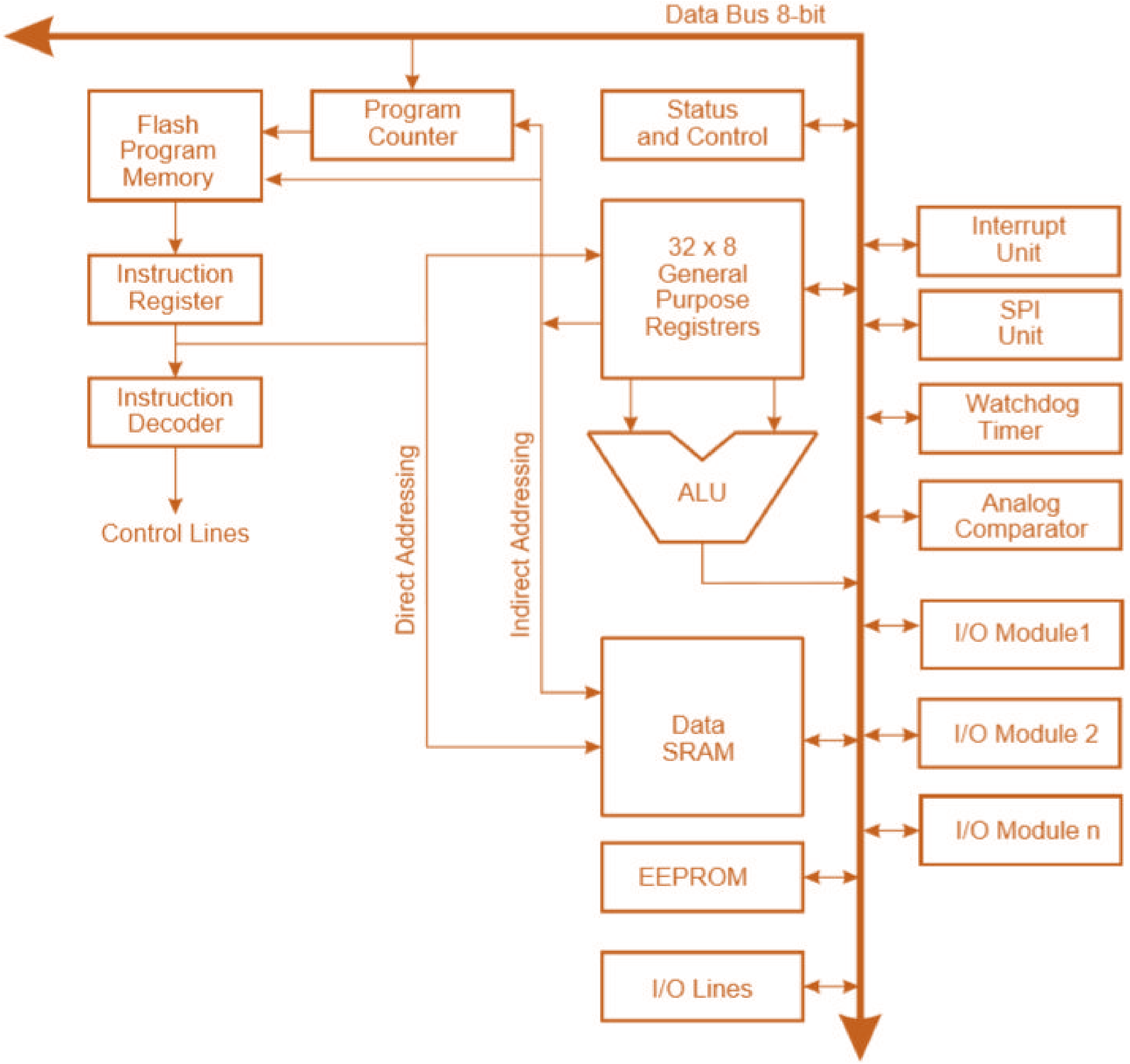
Internal Architecture of ATmega328

1. The ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC (reduced instruction set computer) architecture.
2. The clock is controlled by an external 16 MHz Crystal Oscillator.
3. The data is uploaded in serial via the port.
4. Instructions are sent to instruction register and it decodes the instructions on the same clock pulse.
5. In general purpose registers the registers are of 8-bit (used to store data for normal calculations and results) but there are 3 16-bit registers also (used to store data of timer counter).



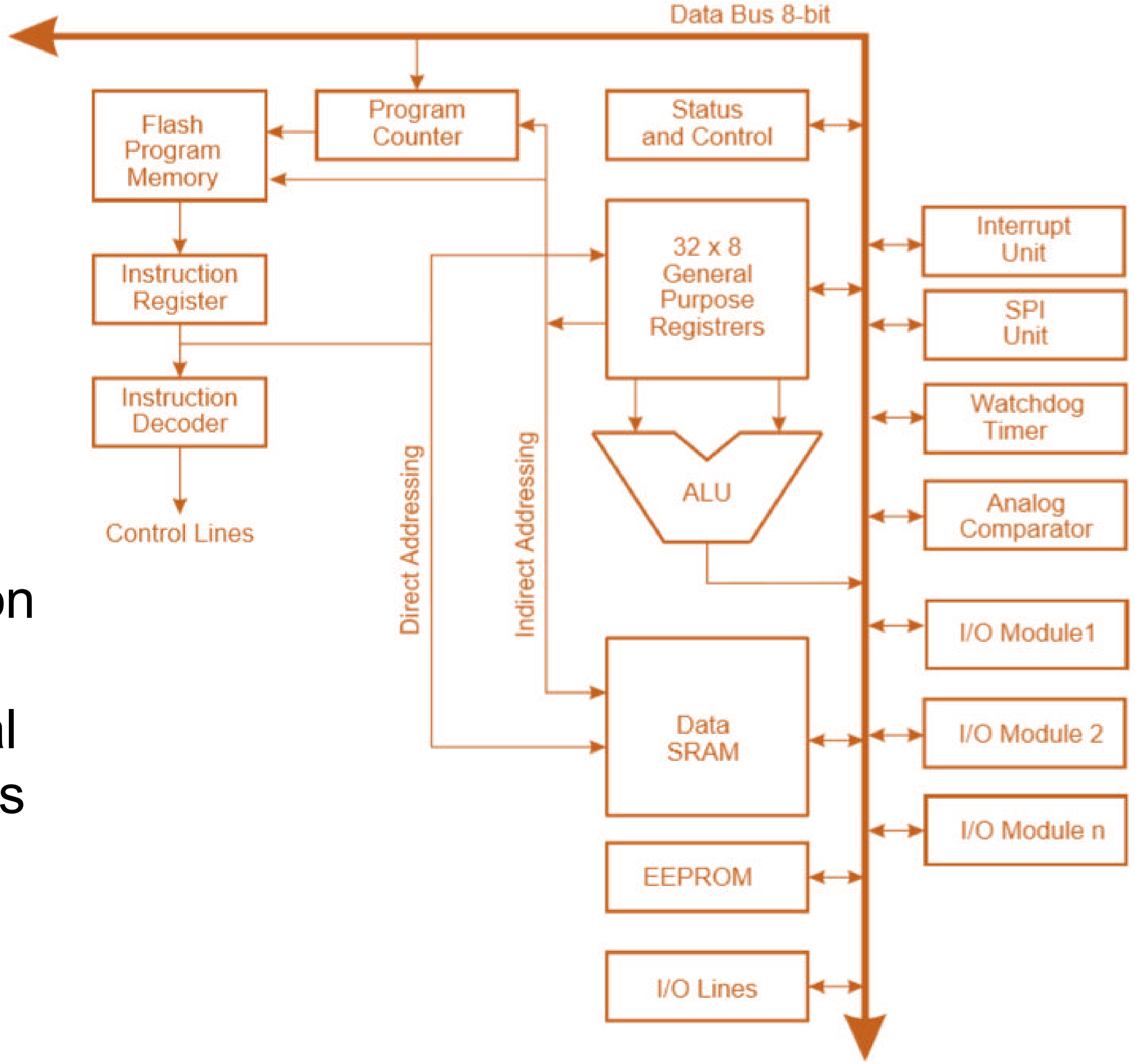
Internal Architecture of ATmega328

6. EEPROM stores data permanently even if the power is cut out.
7. Interrupt Unit checks whether there is an interrupt for the execution of instruction to be executed in ISR (Interrupt Service Routine).
8. Serial Peripheral Interface (SPI) is used to send data between microcontroller and small peripherals such as Camera, Display, SD cards, etc. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.
9. Watchdog timer is used to detect and recover from MCU malfunctioning.



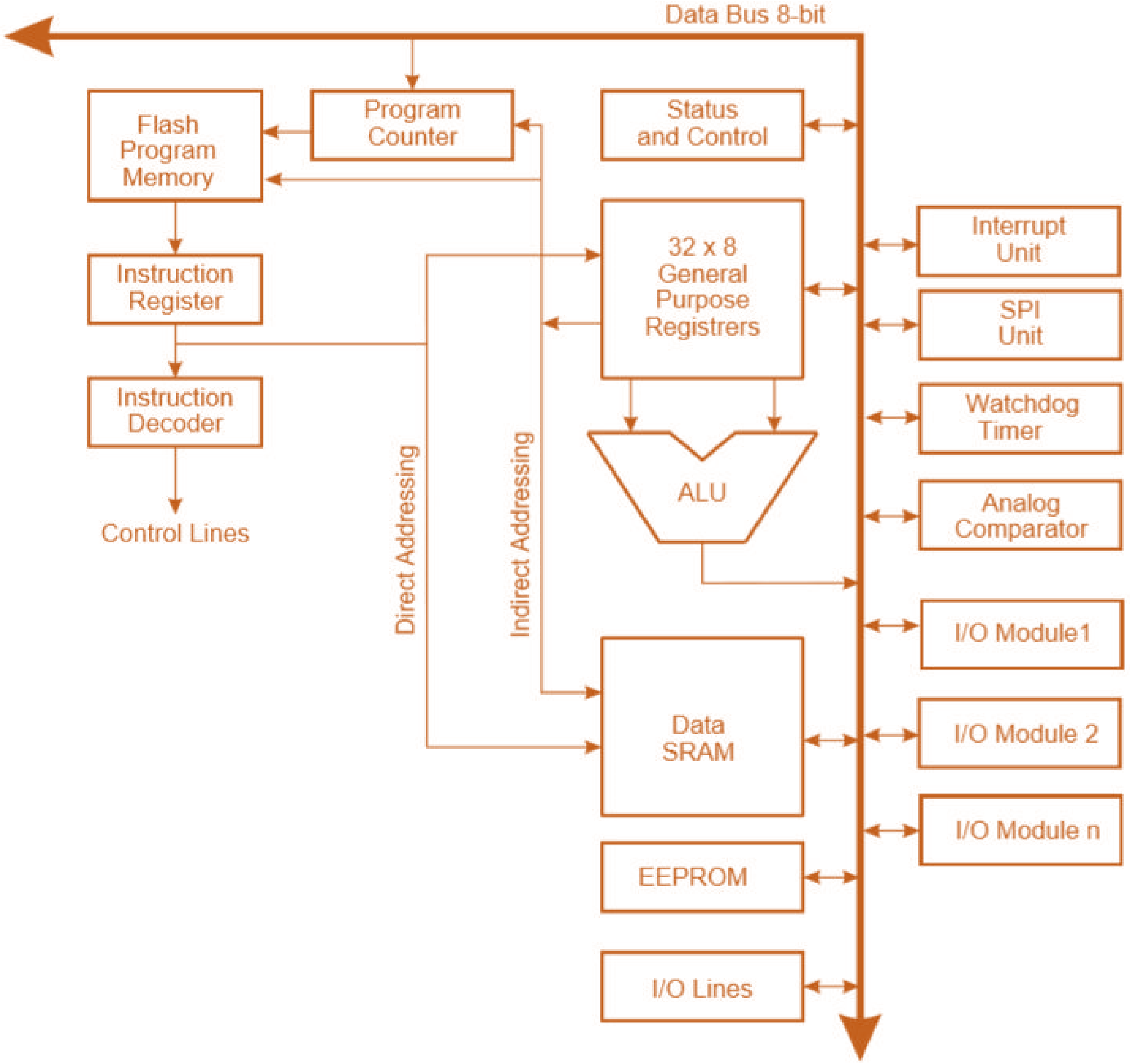
Internal Architecture of ATmega328

10. Analog comparator compares the input values on the positive and negative pin, when the value of positive pin is higher the output is set.
11. Status and control is used to control the flow of execution of commands by checking other blocks inside the CPU at regular intervals.
12. ALU (Arithmetic and Logical Unit): The high performance AVR ALU operates in direct connection with the all 32 general purpose registers. Within a single clock cycle, arithmetic operations b/w general purpose registers are executed. The ALU operations are divided into 3 main categories– arithmetic, logical and bit-function.



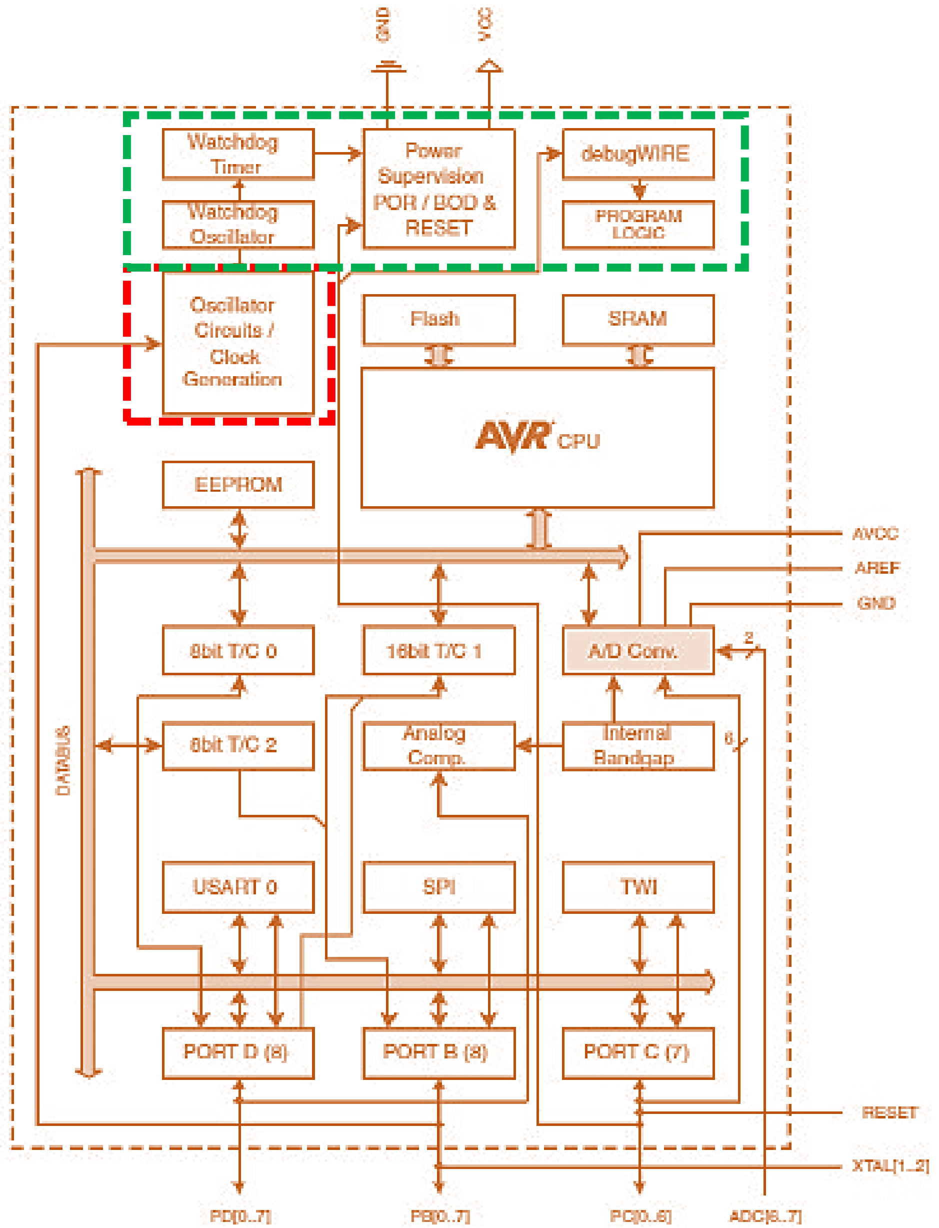
Internal Architecture of ATmega328

10. I/O pins The digital inputs and outputs (digital I/O) on the Arduino are what allow you to connect the Arduino sensors, actuators, and other ICs. Learning how to use them will allow you to use the Arduino to do some really useful things, such as reading switch inputs, lighting indicators, and controlling relay outputs.



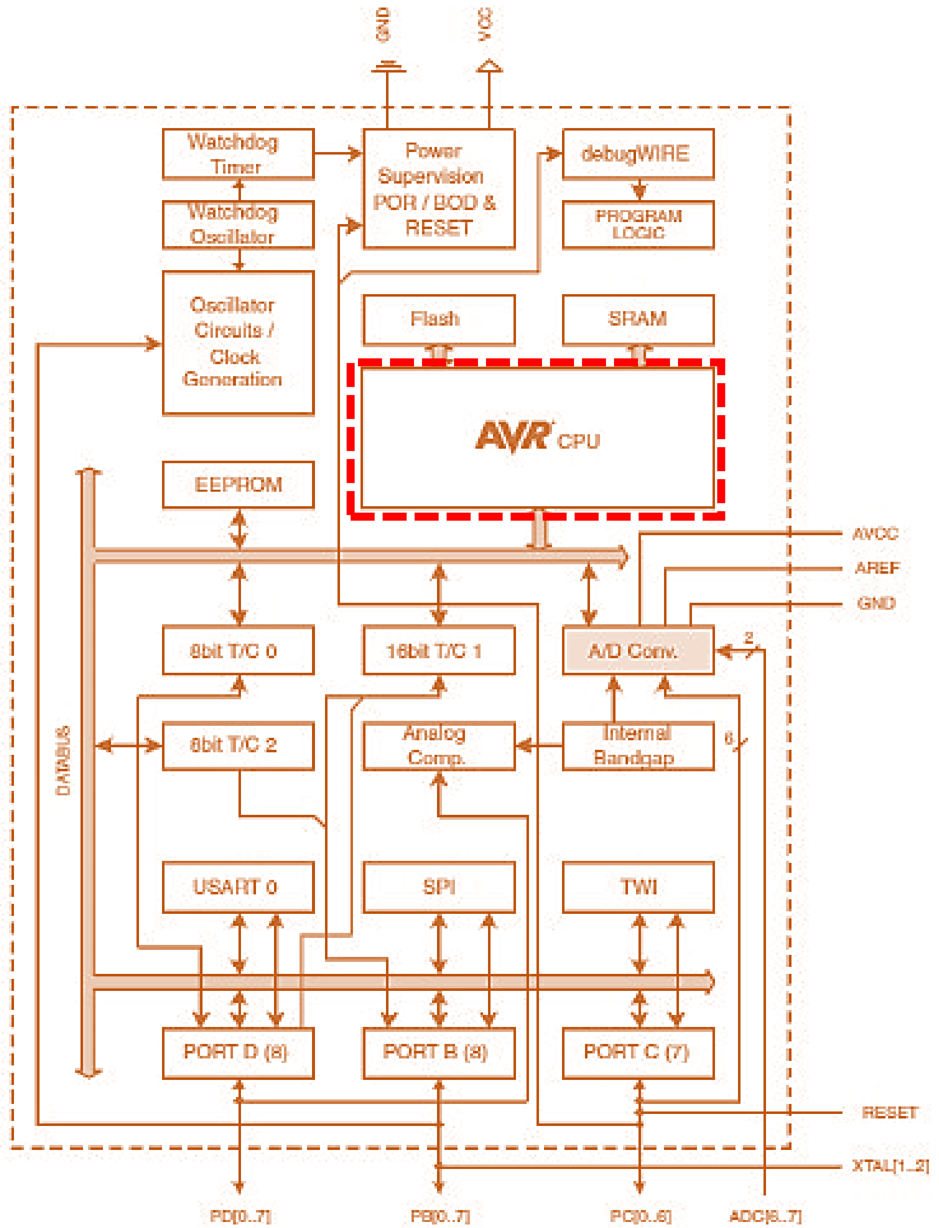
Internal Architecture of AVR

- ❑ 6 different oscillators to serve as clock sources
- ❑ 5 different clock outputs from the AVR clock control Unit
- ❑ CPU clock routed to part of the system concerned with the operation of AVR core
- ❑ I/O clock used by most I/O modules
- ❑ Flash clock controls operation of the flash interface
- ❑ Asynchronous timer clock allows asynchronous timer/counter to be clocked directly by external clock
- ❑ ADC Clock is dedicated to the ADC module



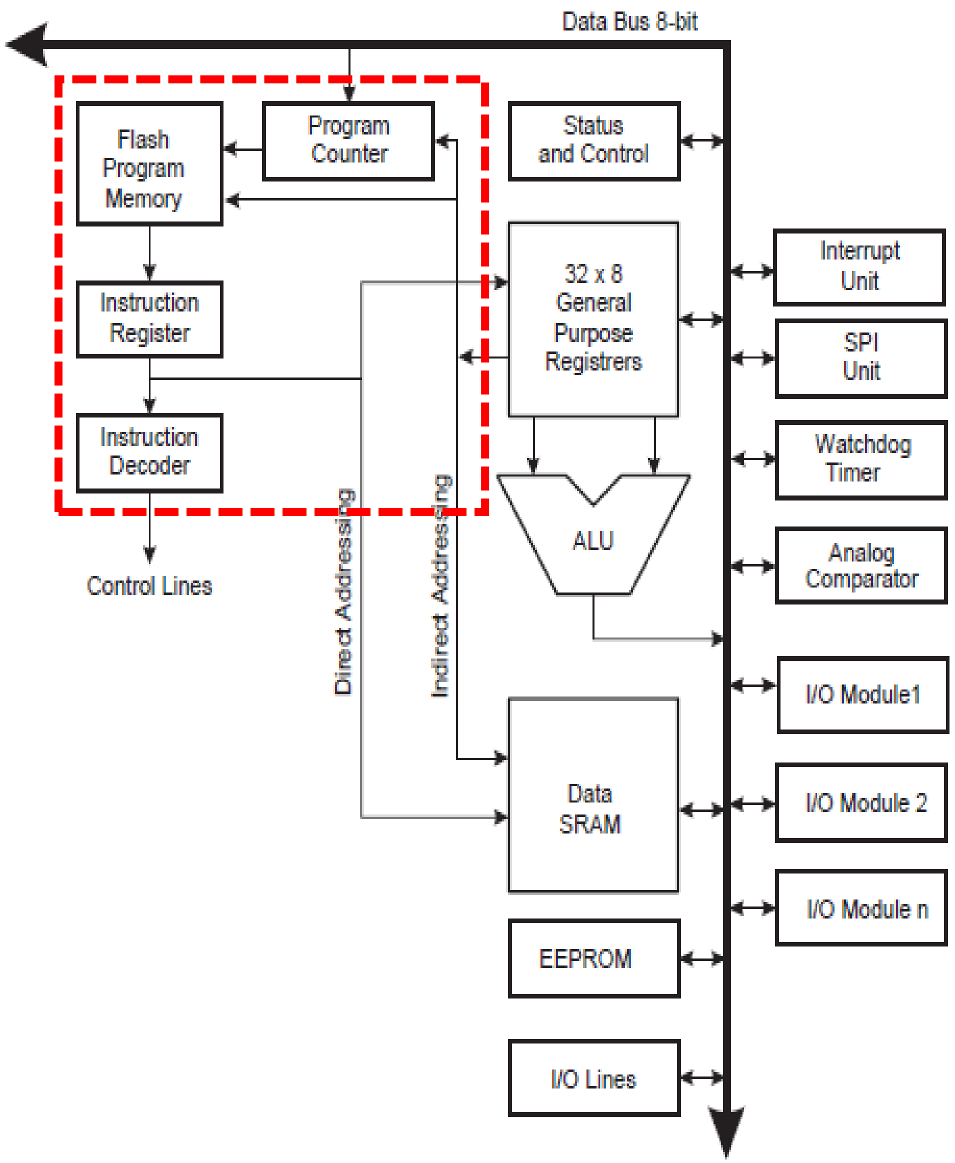
Internal Architecture of AVR

- CPU is the Central Processing Unit
- The main function of the CPU core is to ensure correct program execution.
- The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.
- In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data.



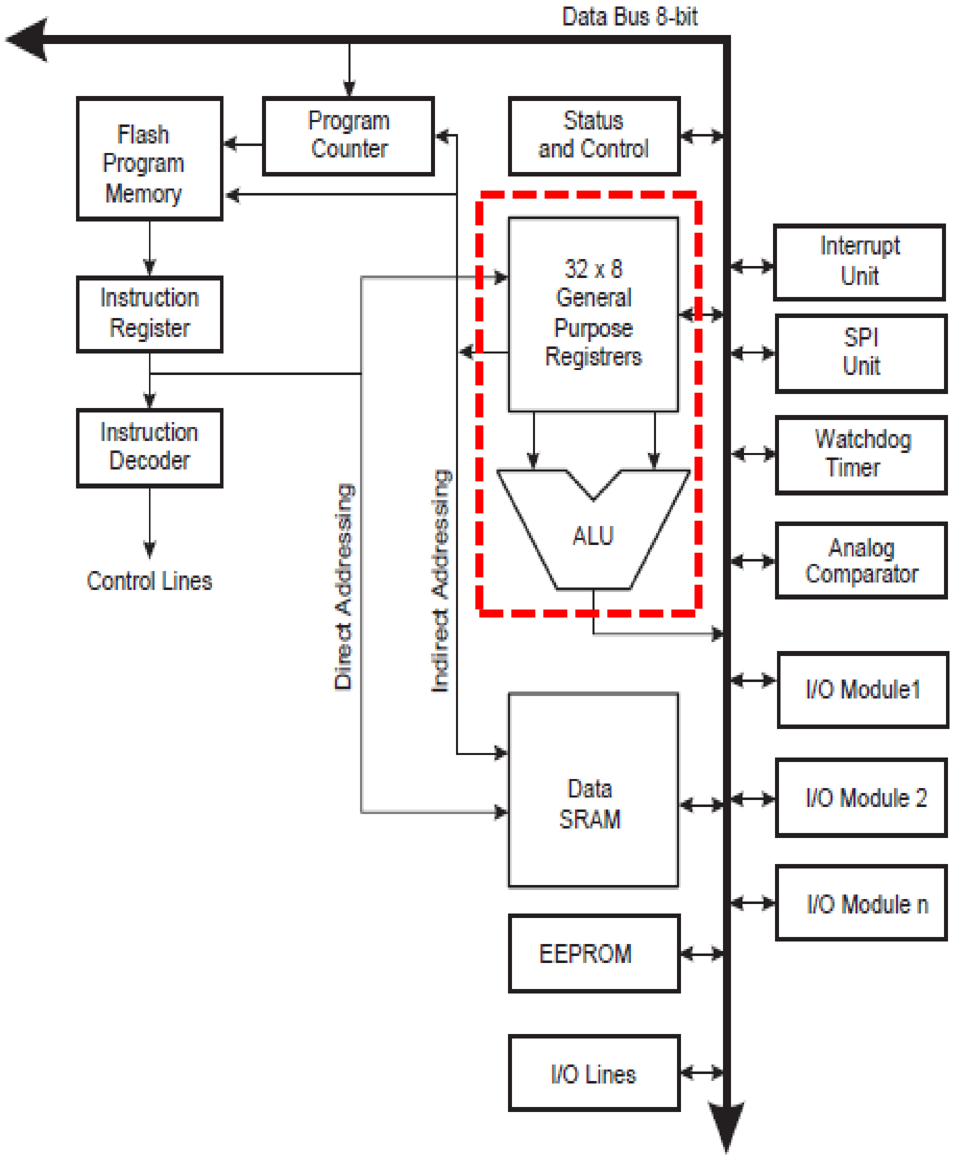
Instruction Fetch and Decode

- Separate memories and busses for program and data (Harvard Architecture)
- Instructions in program memory are executed with single-level pipelining
- The next instruction is pre-fetched from program memory while previous instruction is being executed
- Instructions are executed every clock cycle
- Program counter points towards the next instruction to be executed



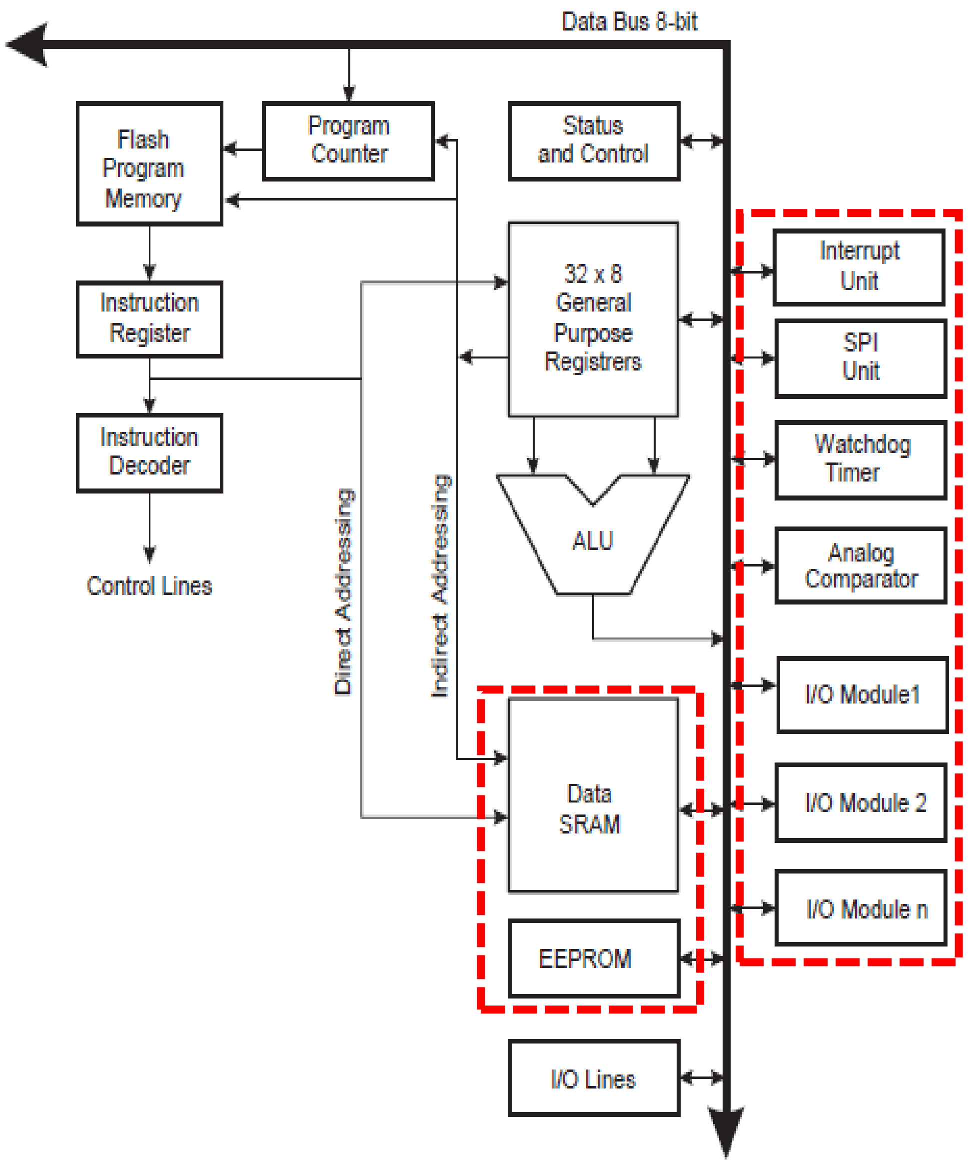
ALU Instructions

- ALU supports both arithmetic and logic operations between:
  - Registers
  - A constant and a register
- Single register operations can also be executed
- Status register updated after arithmetic operations to reflect information about the operation
- In typical ALU operations, in a single clock cycle:
  - Operation is executed
  - Result is stored back in the register



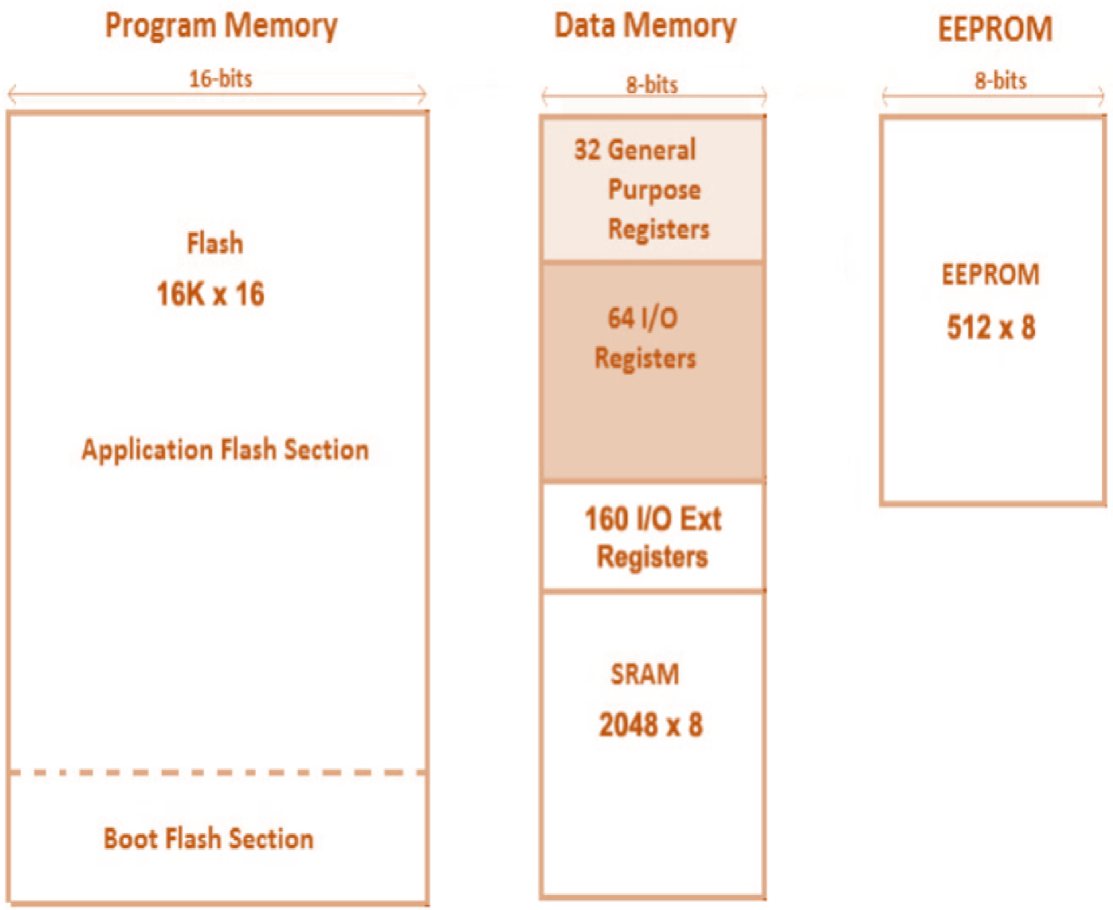
I/O and Special Functions

- When used as general digital I/O ports, all AVR ports have true read-modify-write functionality.
- General digital I/O pins are:
  - Bi-directional
  - Optional internal pull-ups
  - When configured as an input pin, pull up resistor is activated
  - When configured as an output pin, pull up resistor is turned off
- Most digital I/O pins have alternate functions



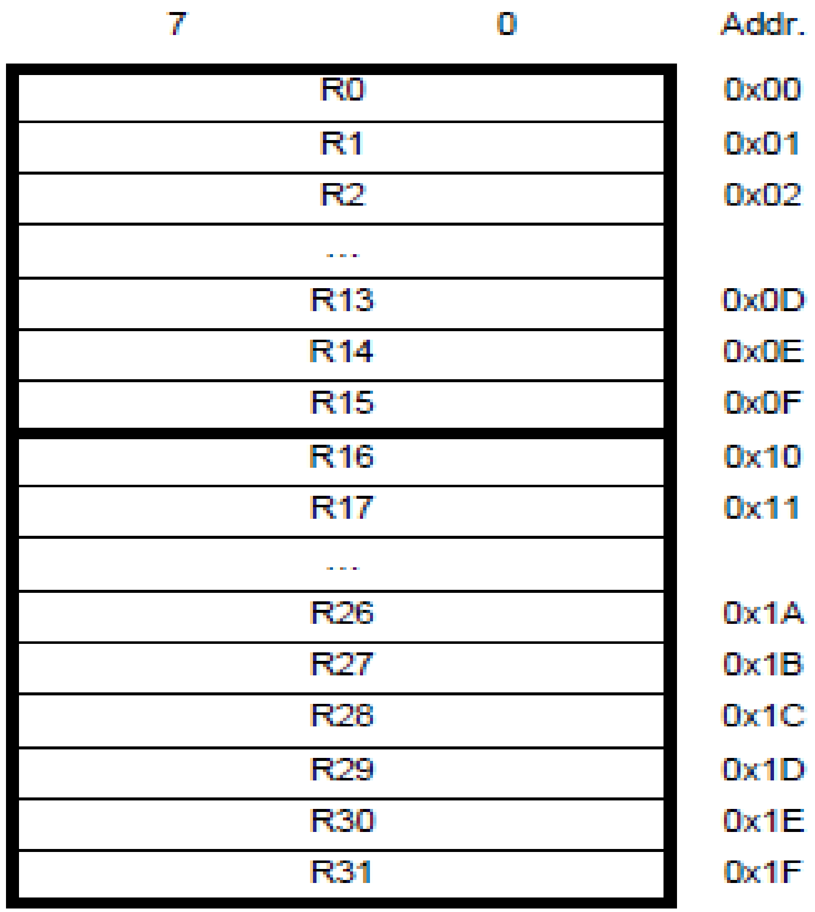
ATmega328 Memory

- ATmega 328 has **three types of memories:**
  - **Flash Memory (Program Memory)** has **32 KB** capacity. It is a non-volatile memory. It is used for the permanent saving program (CODE) being executed. The AVR executes programs stored in program memory only.
  - SRAM stands for **Static Random Access Memory. It is also called Data Memory. It has a 2 KB capacity.** It is a volatile memory i.e., data will be removed after removing the power supply.
  - EEPROM stands for Electrically Erasable Programmable Read Only Memory. **It has a 1 KB capacity. It has long-term data.**



ATmega328 General Purpose Registers

- ATmega-328 has thirty-two **(32) General Purpose Registers (GPR).**
- All these registers are **part of Static Random Access Memory (SRAM).**
- These **GPRs** are 8-bit in size.
- Most instructions operating on the Register file:
  - ✓ Have direct access to all registers
  - ✓ Most are single-clock cycle execution



References:

1. The Intel microprocessors By Barry B. Brey
2. Microprocessors and Microcomputer-based System Design By Mohamed Rafiquzzaman
3. Implementing a One Address CPU in Logisim By Charles W. Kann
4. Security in Embedded Systems By J. Rosenberg
5. External Search By Stefan Edelkamp and Stefan Schrödl
6. Embedded Systems Design By Steve Heath
7. Programming Embedded Systems with C and GNU Development Tools By Michael Barr and Anthony Massa
8. The Rise of "Internet of Things": Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks, By Muhammad Shafiq, ,Zhaoquan Gu, Omar Cheikhrouhou, Wajdi Alhakami, and Habib Hamam, *Wireless Communications and Mobile Computing,* 2022
9. Internet of Things and Big Data Analytics Toward Next-Generation Intelligence By Nilanjan Dey, Aboul Ella Hassanien, Chintan Bhatt, Amira S. Ashour, Suresh Chandra Satapathy
10. Family Datasheet: 8-bit AVR Microcontrollers
11. Arduino Uni R3 Datasheet

Thank You