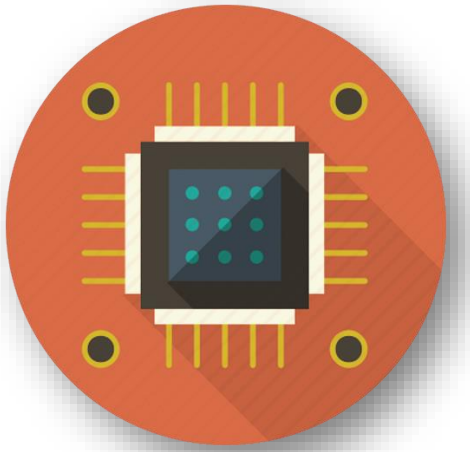
Embedded Systems Diploma

AMIT



Automotive bus technology 4H

ISTQB 18H

RTOS 20H

Microcontroller & Microprocessor (AVR - ARM) interfacing 85H

Embedded System Tools 6H

Software Engineering 4H

C Programming & Data Structure 30H 10H

Introduction to Embedded Systems 4H

AMIT

Diploma OUTLINES

- Intro. Session
- HELLO C
- Control Flow
- Functions
- Pointers
- Arrays
- UDDT
- Algorithms
- Data Structure
- Tooling_1
- GITHUB
- Session_01
- Session_02
- Session_03
- Session_04
- EMBEDDED C
- Computer Arch.
- DIO
- LCD
- Key Pad
- INTERRUPT
- ADC
- Timer_1
- Timer_2
- UART
- SPI
- I2C
- ARM
- Session_01
- Session_02
- Session_03
- Session_04
- CAN
- LIN

AMIT

Contents

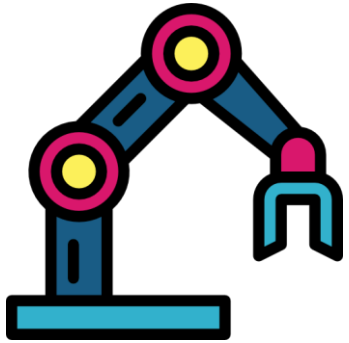
AMIT

What's Embedded Systems?

> A Combination of multiple systems merged together at one system

Example 1: **PCB** + **CODE** = **TV**

Hardware System Software System Embedded Systems



Example2: **PCB** + **CODE** + **Motors** = **Robotic Arm**

Hardware System Software System + Mechanical System Embedded Systems

AMIT

Embedded Systems applications

Robots

ROV

Drones

Robots

Home APP.

Home Devices

Smart Homes

Smart Cities

Military

Jets

Weapons

Satellite

Bio Medical

X- Rays

ECO

Ventilator

Automotive

Motor Control

Dashboard

Self-Drive Cars

Others

ATM Machines

Cameras

Routers & switches

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics

Efficiency:

- Considered in real time applications.
- A real time application is one in which must be able to act at a speed corresponding with the occurrence of an actual process. Must compute certain results in certain real-time without delay.

Efficiency

Some Embedded systems may have real-time performance constraints that must be met, for reasons such as safety and usability.

Others may have low or no timing performance requirements, allowing the system hardware to be simplified to reduce costs.

AMIT

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics

Tightly-constrained

Embedded Systems normally come with constraints in hardware resources:

- Processing(speed)
- Storage (Code Size)
- Memory(Data)
- Most of the time it targets real time objectives, this means,
- It needs to be fast and efficient (Response Time).
- It needs to be predictable (execution time known ahead, and almost constant)
- Power limited (battery operated devices).
- Cost
- Size.
- The developer has to deal with all of these constraints
- Development should take into consideration code efficiency and code foot print.
- Debugging tools are “closer to the metal”
- Special attention to power consumption in some cases.

AMIT

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

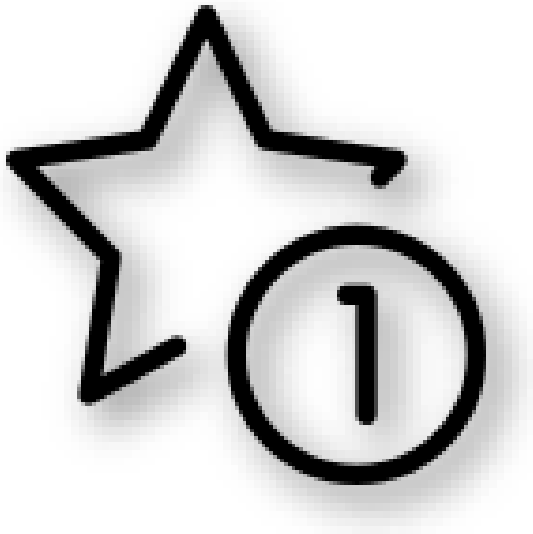
Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics



Dedicated to perform a specific function and include processors dedicated to specific function.

Single-functionality

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

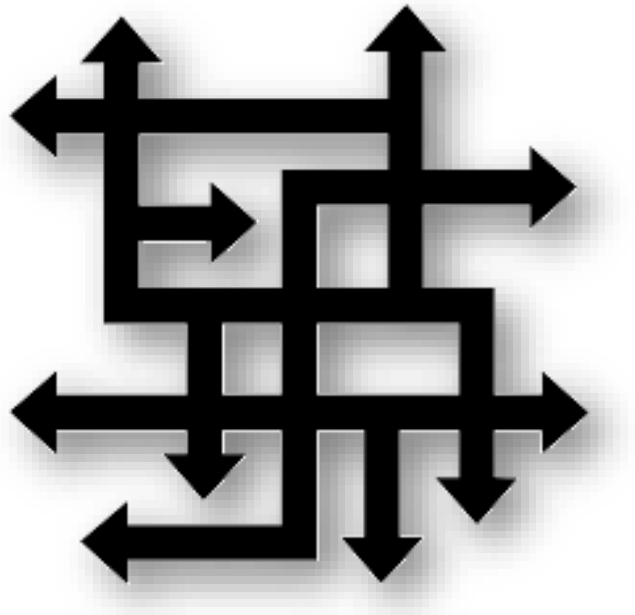
Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics



Often have to run sophisticated algorithms or multiple algorithms Cell phone, laser printer, automotive, etc.

Complex functionality

AMIT

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics

Must not endanger human life and the environment.

Interactive System

AMIT

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics

The ability to modify the system after its initial release and enhance its performance like execution time, code and memory size.

Maintainability

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

Maintainability

Single-functionality

Complex functionality

AMIT

Embedded Systems Characteristics

Safety-critical

Must not endanger human life and the environment.

AMIT

Embedded Systems Characteristics

Reliability

Safety-critical

Tightly-constrained

Strong association between the HW and SW

Efficiency

Interactive System

Maintainability

Single-functionality

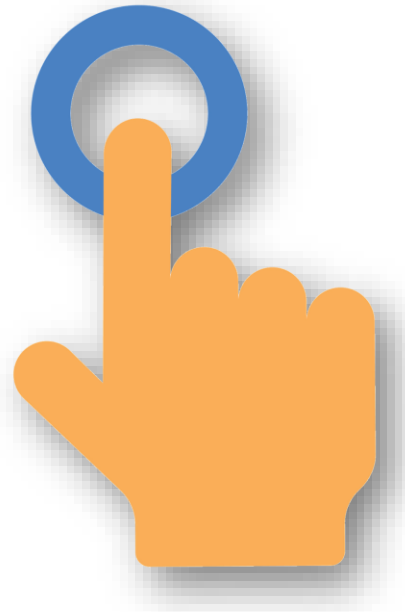
Complex functionality

AMIT

Embedded Systems Characteristics

An Embedded system should be as interactive as possible so that it is easily understandable, Operated and Handled by a user and provide ease of task.

Interactive System



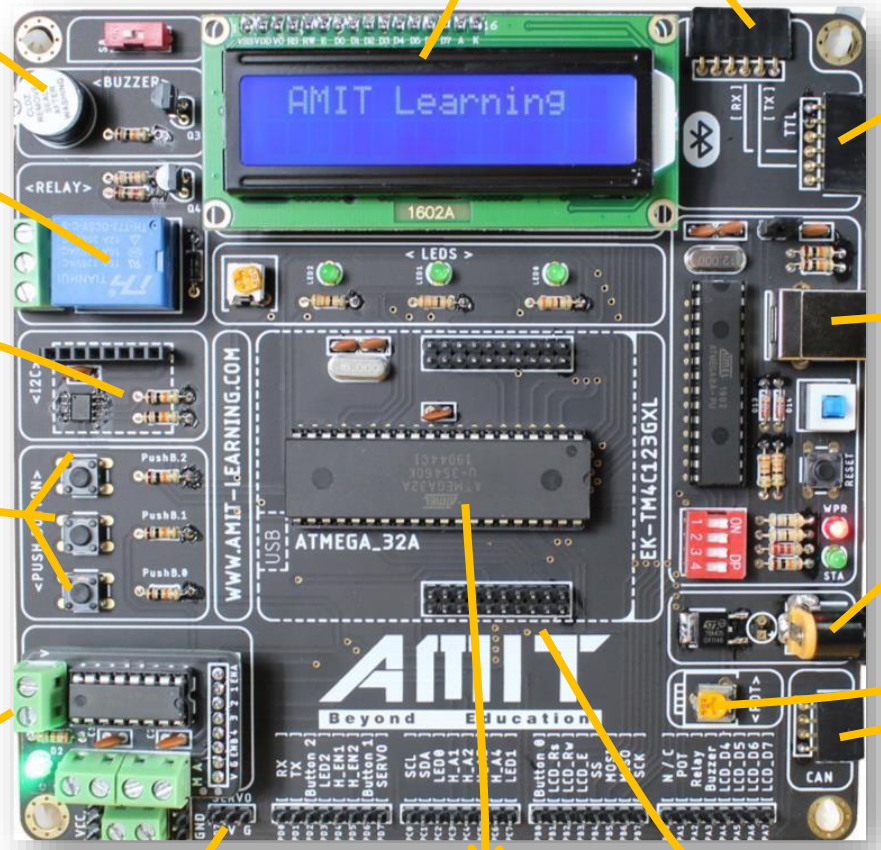
AMIT

Embedded Systems Companies

ROBOTS	Home APP.	MILITARY	BIO MEDICAL	AUTOMOTIVE	Others
HONDA	Google	ELC	PHILIPS	BOSCH	Honeywell
Boston Dynamics	SAMSUNG	NORTHROP GRUMMAN	SIEMENS	Mentor Graphics	SONY
ABB ROBOTICS	LG	AOI الهيئة العربية للتصنيع ARAB ORGANIZATION FOR INDUSTRIALIZATION	BIB	avelabs	Canon

AMIT

Atmega Evaluation KIT System



P LCD / 7-Segments
2*16LCD / (2)7SEG.

A Buzzer
5V

B Relay
10A / 220 VAC

C I2C
COMM. Protocol

D Buttons
3 Push buttons

E H-Bridge
For DC-Motor Control

F Servo
OUTLET

G Microcontroller
3 Push buttons

H TIVA-C
OUTLET

N BLUETOOTH Mod.
OUTLET

M TTL
OUTLET

L Programmer
With USB Cable

K Power Input
External

J POT
10K

T CAN
OUTLET

AMIT

Embedded Systems vs General Purpose Systems

Systems

Embedded Systems

> Perform ONLY Few Applications that are known at design-time

> Not Programmable by the user

> Fixed run-time requirements

General Purpose Systems

> Computing a lot of Applications

> Programmable by the end user

> Faster is better

Criteria

> Cost 

> Low Power Consumption 

> Deterministic behavior 

> Meeting the time bounds 

> Size 

> Cost 

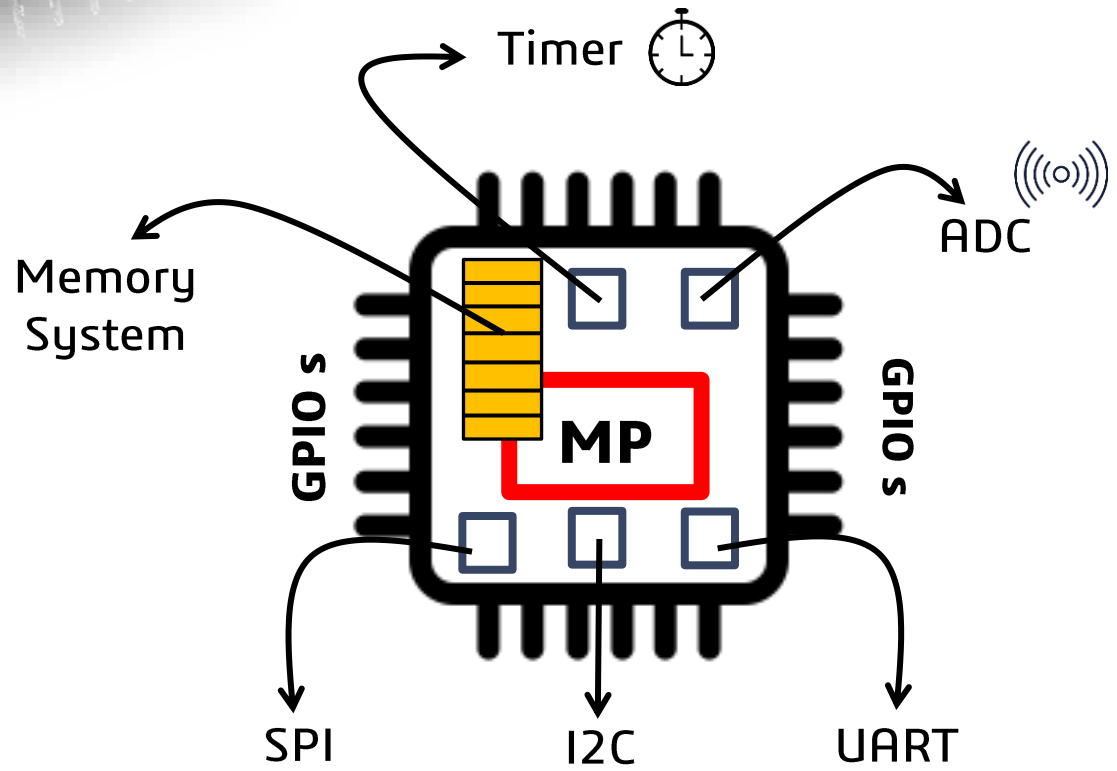
> Average Speed 

AMIT

MICROCONTROLER

Components

1. **CPU**
   - ALU
   - CU
   - ACCUMELATOR Register
   - General Purpose Registers (GPRs)
   - Special Purpose Registers (SPRs)
     - IR
     - PC
     - SP
     - PSW
2. **Memory System**
   - RAM
   - ROM
3. **Peripherals**
   - ADC
   - DAC
   - Timers
   - UART & SPI &I2C
4. **GPIO s**

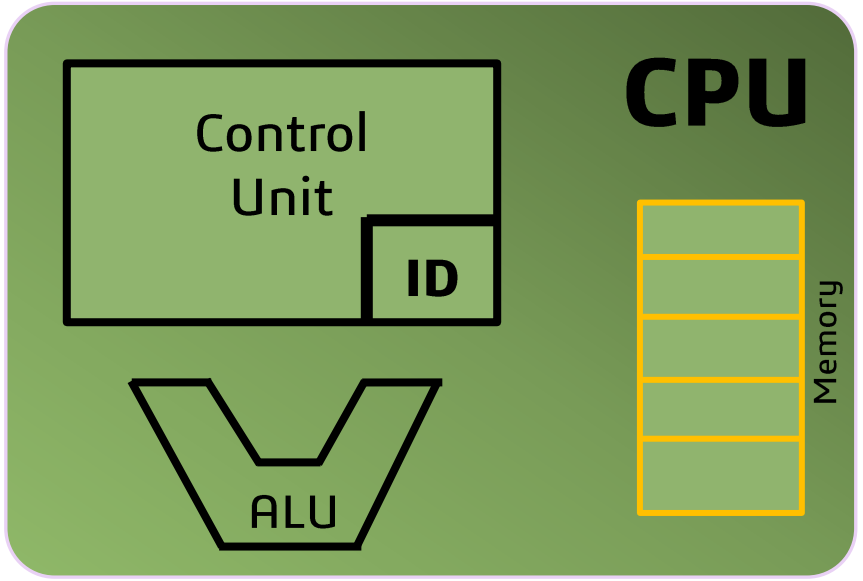


AMIT

1- CPU

CPU = Central Processing Unit = Processor

- Responsible of all the computing process
- Fetch, decode and execute all program instructions
- Direct the flow of data to / from memory unit
- Most CPUs are synchronous, this means it depends on external clock



ALU — Arithmetic Logic Unit

- Responsible of all the arithmetic and logic and shifting instructions

CU — Control Unit

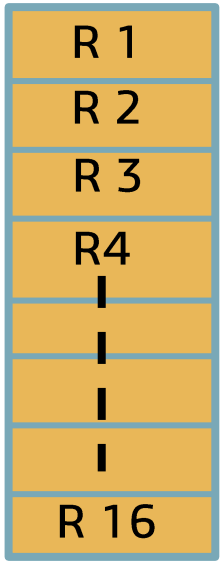
- Convert the instructions stored in program memory (ROM) into codes which ALU can understands.
- Generate the control signal of this instruction.

AMIT

1- CPU

General Purpose Registers (GPRs)

- Also called Register File
- A group of registers the processor use them to store the data fetched from memory



Depends on the Architecture of the **CPU**

- 16 in AVR
- 32 in ARM

ACC — Accumulator Register

- Is an SPR (Special Purpose Register) closely related to the operation of ALU

AMIT

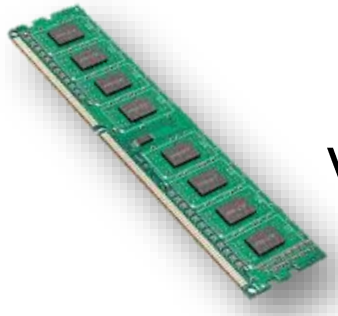
1- CPU

Special Purpose Registers (SPRs)

IR	**Instruction Register**	Contain the instruction after getting it from program memory
PC	**Program counter**	Contain the address of the next instruction in program memory which the CPU will process
SP	**Stack Pointer**	Hold the memory address of the next available location on the stack to store data
PSW	**Program Status Word**	Contains information about the result of the most recently executed arithmetic instruction

AMIT

2- Memory



Volatile
RAM

Static RAM

Dynamic RAM

Non-Volatile
ROM

Masked
ROM

PROM
(OTP)

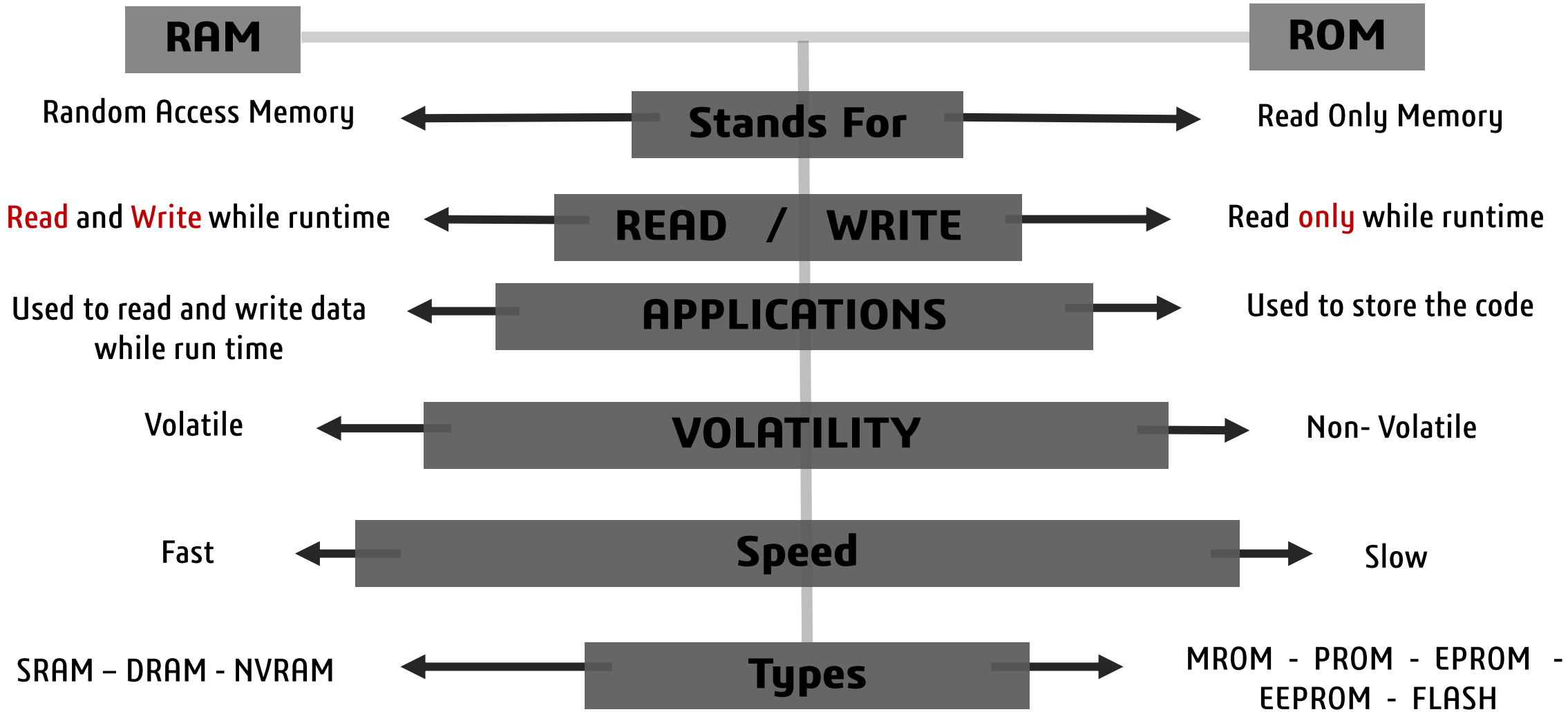
EPROM

EEPROM

Flash

AMIT

2- Memory



AMIT

2- Memory

RAM

SRAM

- Stands for **Static RAM**
- Consists of multiple Transistors for each memory cell
- **SRAM** does not required a periodic refreshing
- **Faster** than DRAM
- **Low** power consumption
- More expensive than **DRAM**
- Most used in microcontrollers **because limited power source**

DRAM

- Stands for **Dynamic RAM**
- Consists of Transistors and Capacitors
- **DRAM** need a periodic refreshing "a charge every few milliseconds to make capacitors retain its charge "
- Slower than **SRAM**
- **High** power consumption
- Cheaper than **SRAM**
- Most used in PCs because open power source

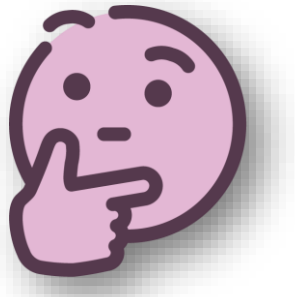
NVRAM

AMIT

2- Memory

NVRAM ----> NON- Volatile RAM

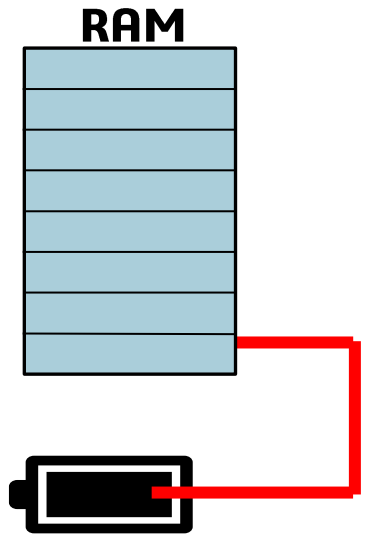
How to make a **Non-Volatile RAM ?!**



2 Ways

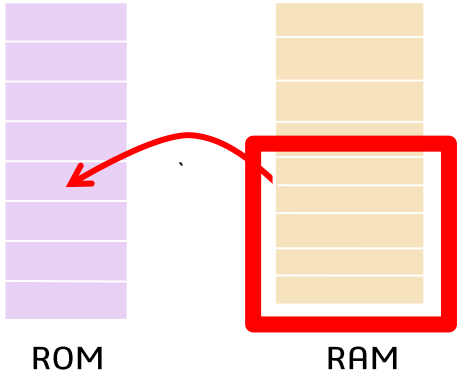
Hardware

We Will connect a battery to the **RAM**, **So we can hold data even when power to the memory chips has been turned off.**



Software

We will Apply a Complex Algorithm to take a copy of the desired data we want to keep **when power to the memory chips has been turned off.**



AMIT

2- Memory

ROM

1. MROM → **MASKED PROGRAMABLE ROM**
   - The microcontroller chip burned by the manufacture
2. PROM OR (OTP) → **ONE TIME PROGRAMMABLE ROM**
   - Can be burned one time by the developer
3. EROM → **ERASABLE PROGRAMMABLE ROM**
   - Can be erased and programmed by the developer
   - The erasing process done using **Ultraviolet Light**
4. EPROM → **ELECTRICALLY ERASABLE PROGRAMMABLE ROM**
   - Can be erased and programmed by the developer
   - The erasing process done using Electric Charge
   - EEPROM write the data by **byte**.

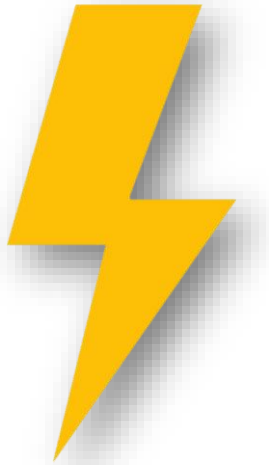
For example in (ATMEGA32) the finite number of writing / erasing is 100,000 times.

AMIT

4- GPIOs

5 Flash → **FLASH ELECTRICALLY ERASABLE PROGRAMMABLE ROM**

- Can be erased and programmed by the developer
- The erasing process done using Electric Charge.
- Developed from **EEPROM**
- EEPROM write the data by **Sector** / **Block**

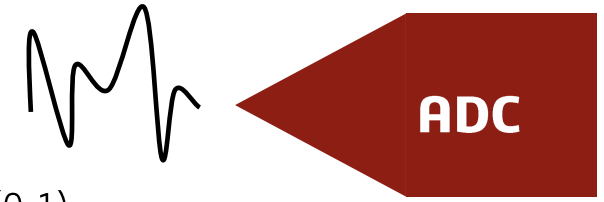


AMIT

4- GPIOs

ADC ANALOG DIGITALCONVERTER

To convert any analog signal like sound waves to a digital signal(0,1)

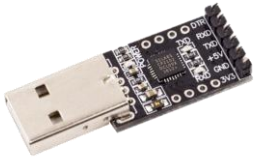


DAC DIGITAL ANALOG CONVERTER

To convert the digital signal into analog

UART UNIVERSAL ASYNCHRONOUS RECEIVER /TRANSMITTER

A communication protocol to send or receive data from microcontrollers



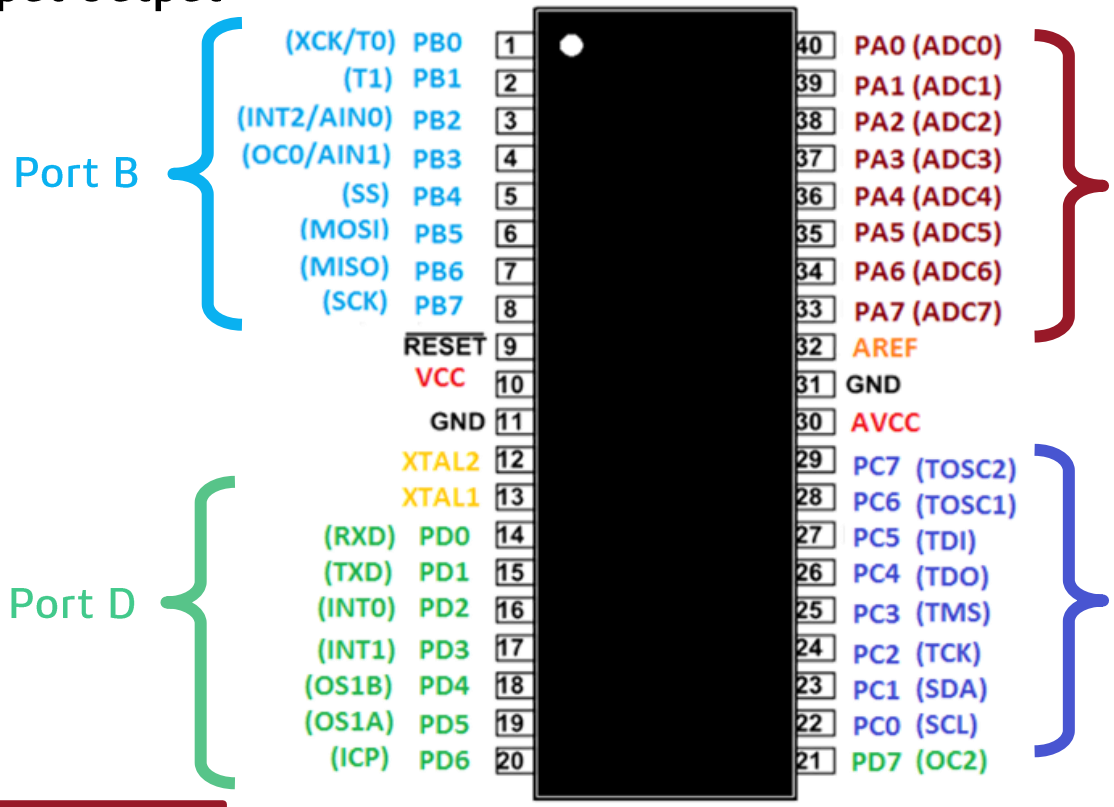
TIMERS what time is it?

Very important peripheral in any embedded system application

AMIT

4- GPIOs

General Purpose Input Output



These pins can be configured as

Input | Output

using the three I/O registers for each port

In this Microcontroller (ATMEGA32) The pins of these **four ports** can be used as general-purpose **inputs/outputs**.

AMIT

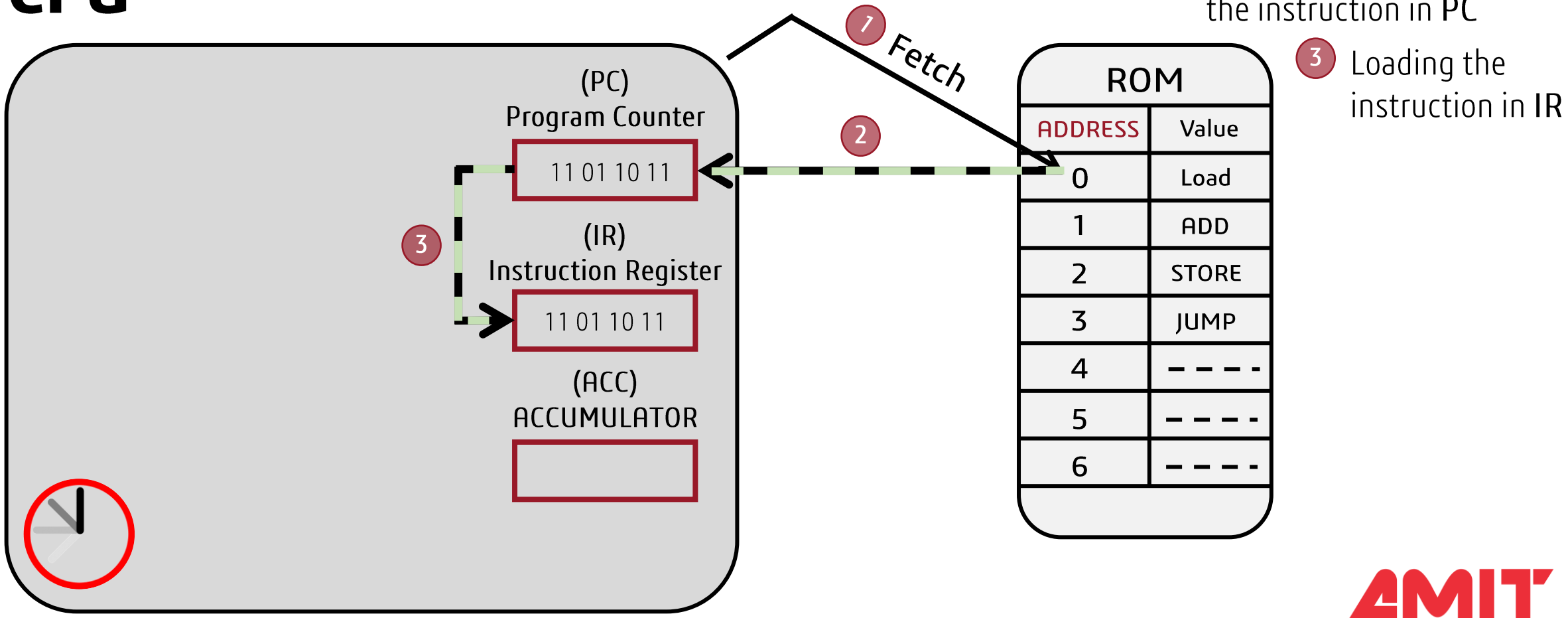
Instruction Life Cycle

AMIT

Instruction Life Cycle

Fetch

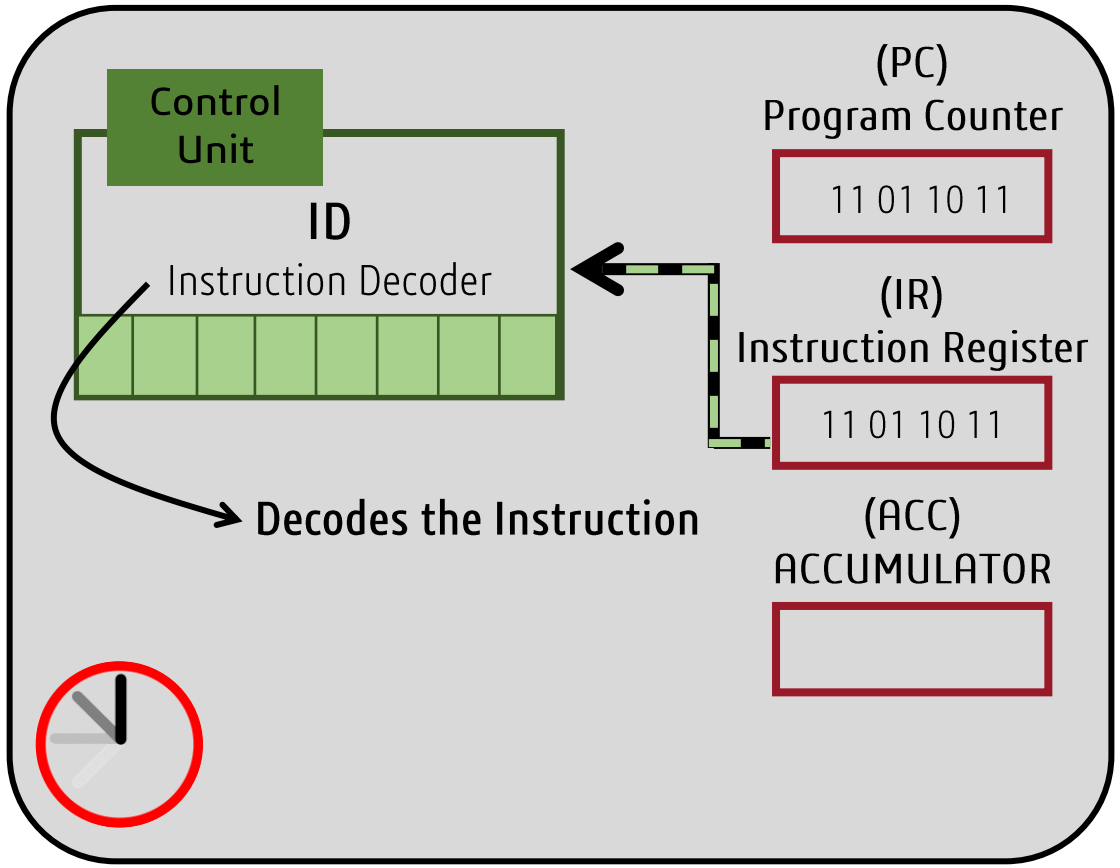
1. Fetching (Catching) the first Instruction
2. loading the value of the instruction in PC
3. Loading the instruction in IR



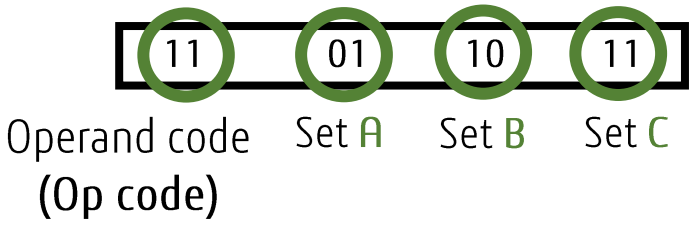
AMIT

Instruction Life Cycle

Decode



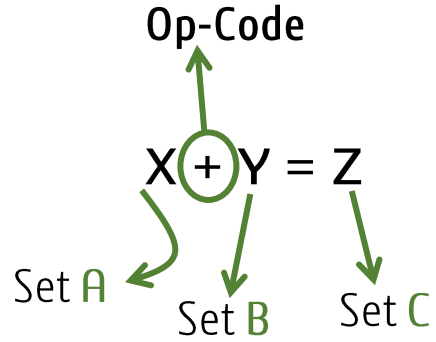
Control Word — Here we can actually decode the instruction



(ISA) Instruction Set Architecture

Is part of the abstract model of a computer that defines how the CPU is controlled by the software.

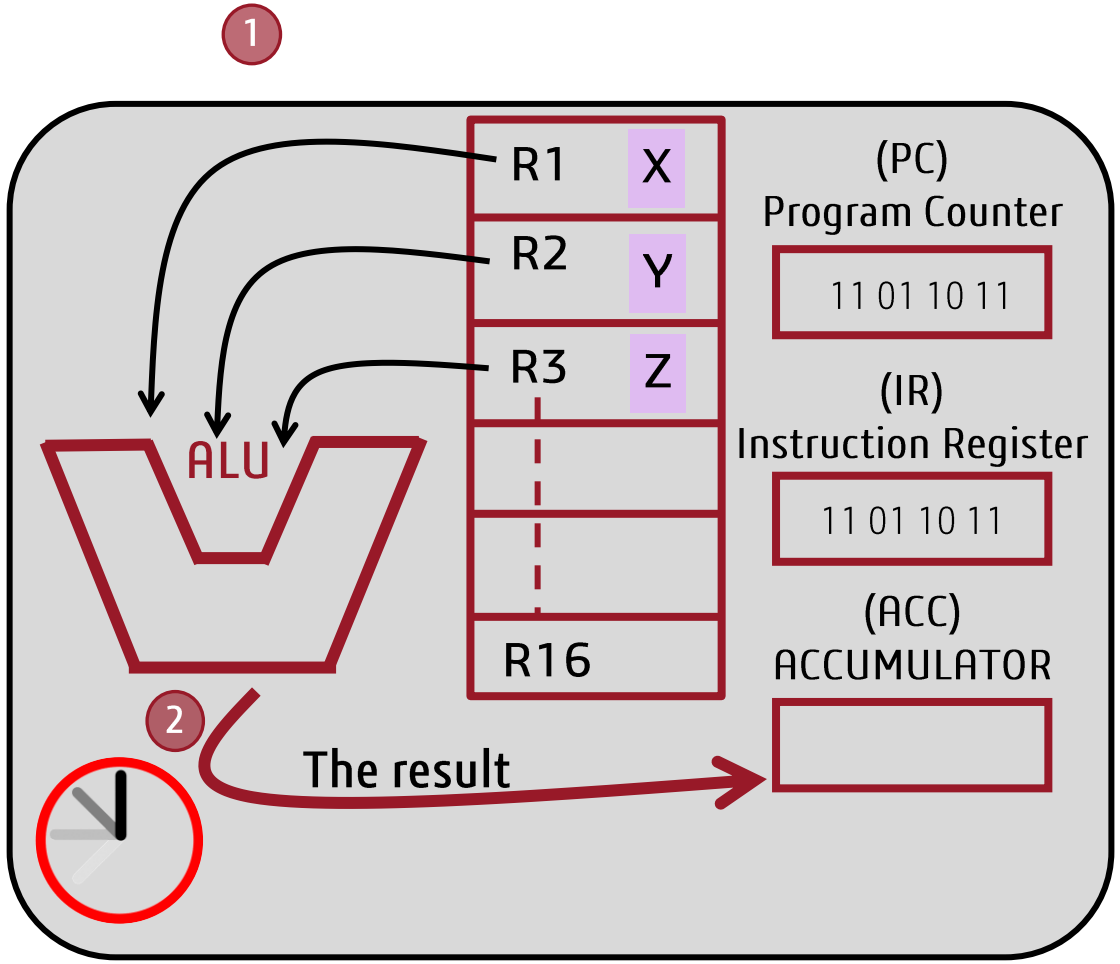
X + Y = Z



AMIT

Instruction Life Cycle

Execute



GPRs Depends on the CPU architecture.

AVR have 1 byte Register Structure

ARM have 4 byte Register Structure

1. Loading the arguments to the **ALU**, and the logic gates will do the instruction decoded.

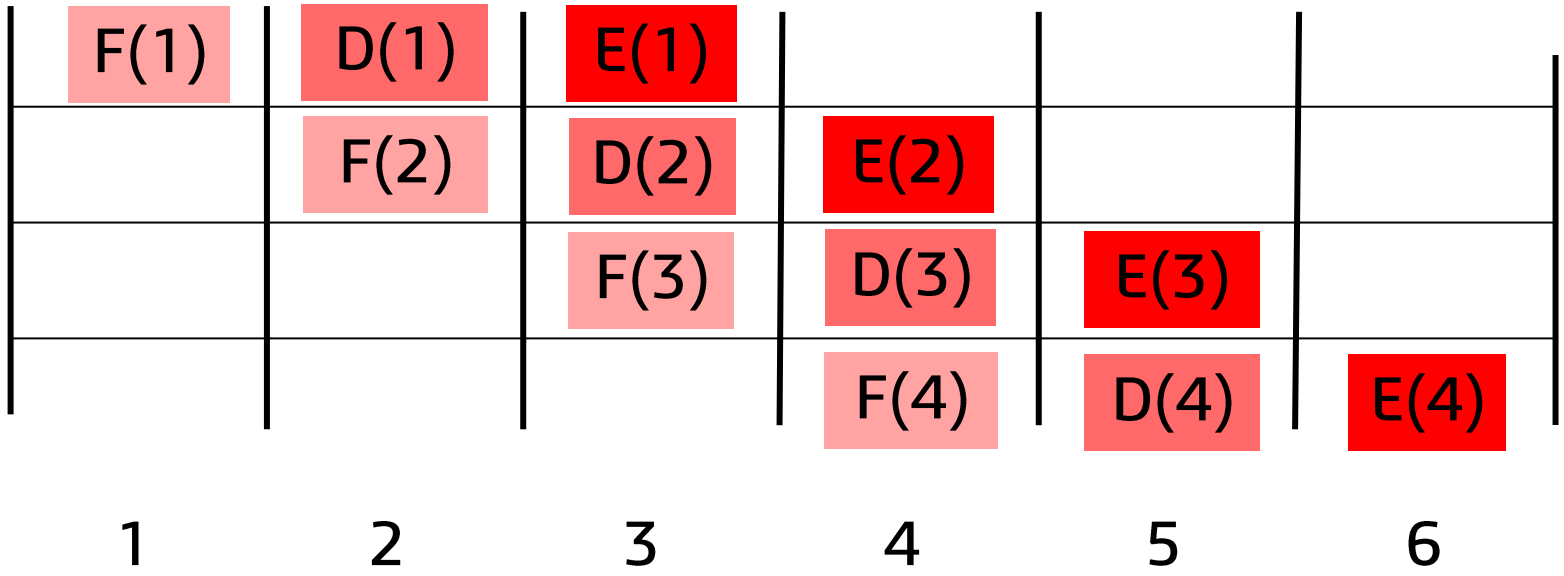
2. Loading the result of the operation in the **Accumulator**.

AMIT

Pipelining Process

Pipelining is a technique where multiple instructions are overlapped during execution.

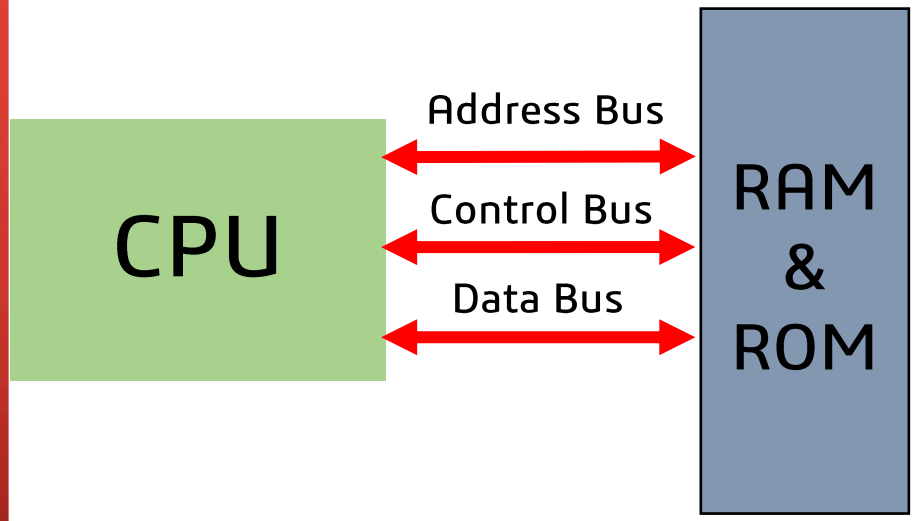
Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.



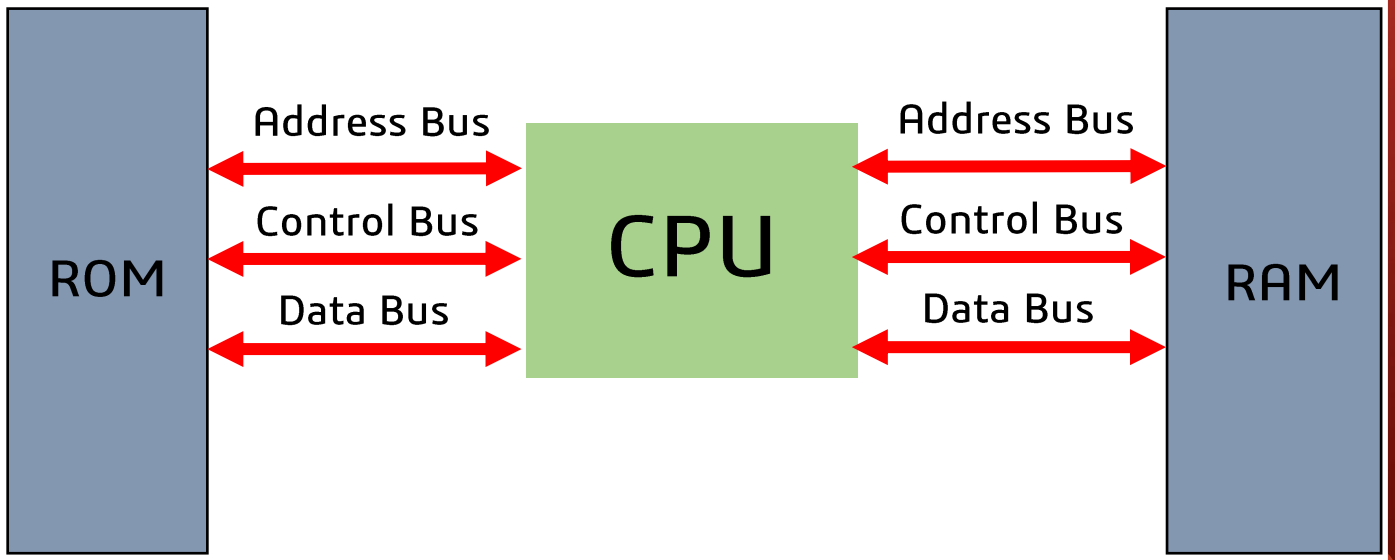
AMIT

CPU Architecture

Von-Neumann Architecture

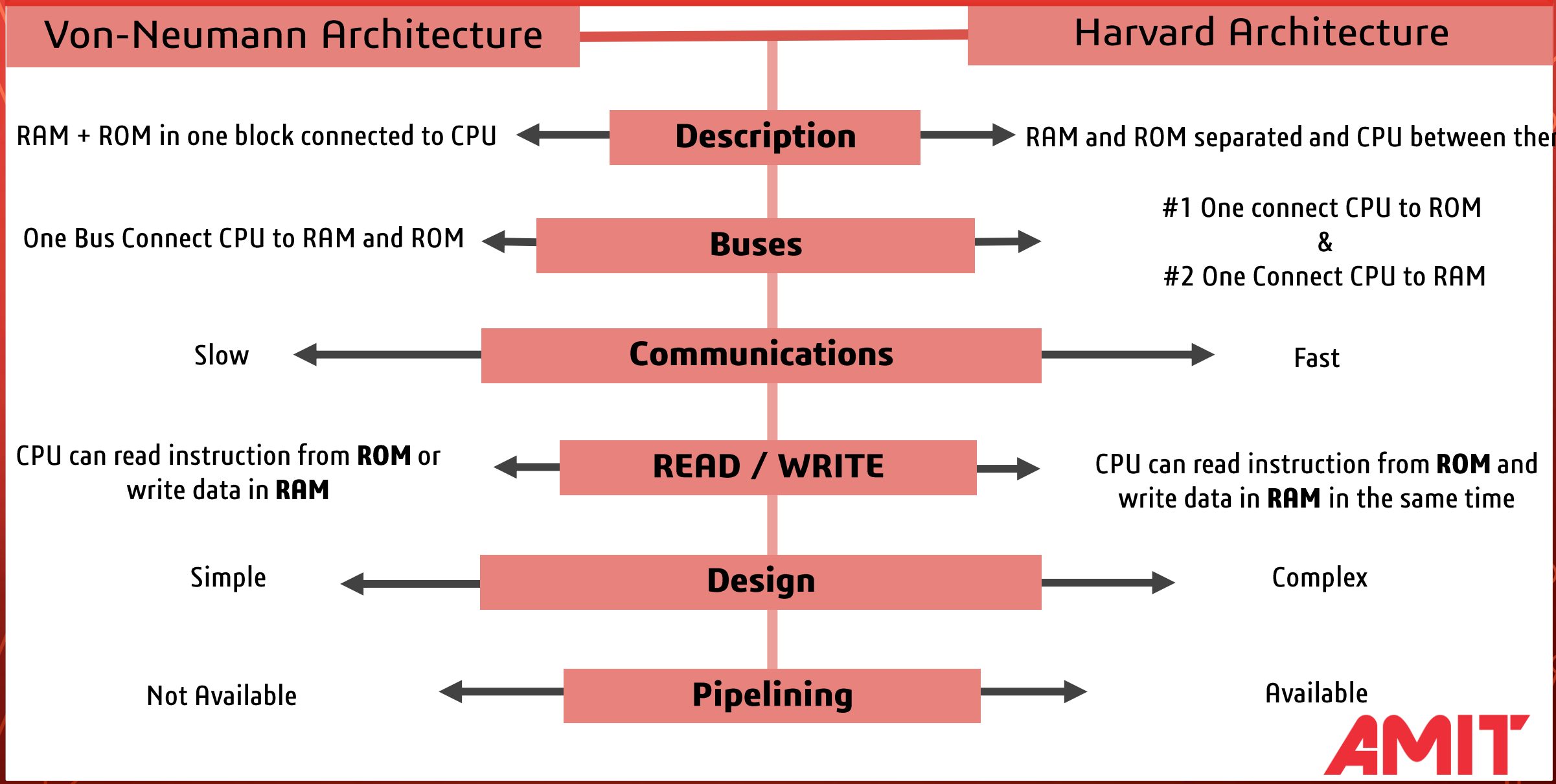


Harvard Architecture

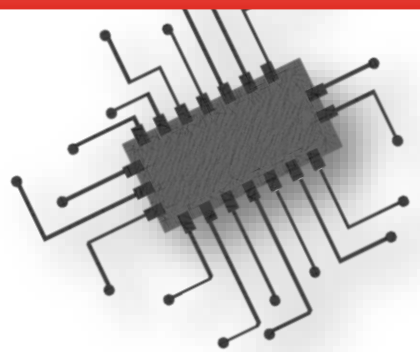


AMIT

CPU Architecture



Vendors



Infineon

Microchip

Texas Instruments

NXP

ST life.augmented

Cypress Embedded in Tomorrow™

Silicon Labs

AMIT

IDEs

Code::Blocks

eclipse

AMIT

Imagine How many lines of code of real products

Mercedes-benz-s-class
100 Million

Boeing 787
6.5 Million

F-35 jet
25 Million

AMIT

THANK YOU!

AMIT