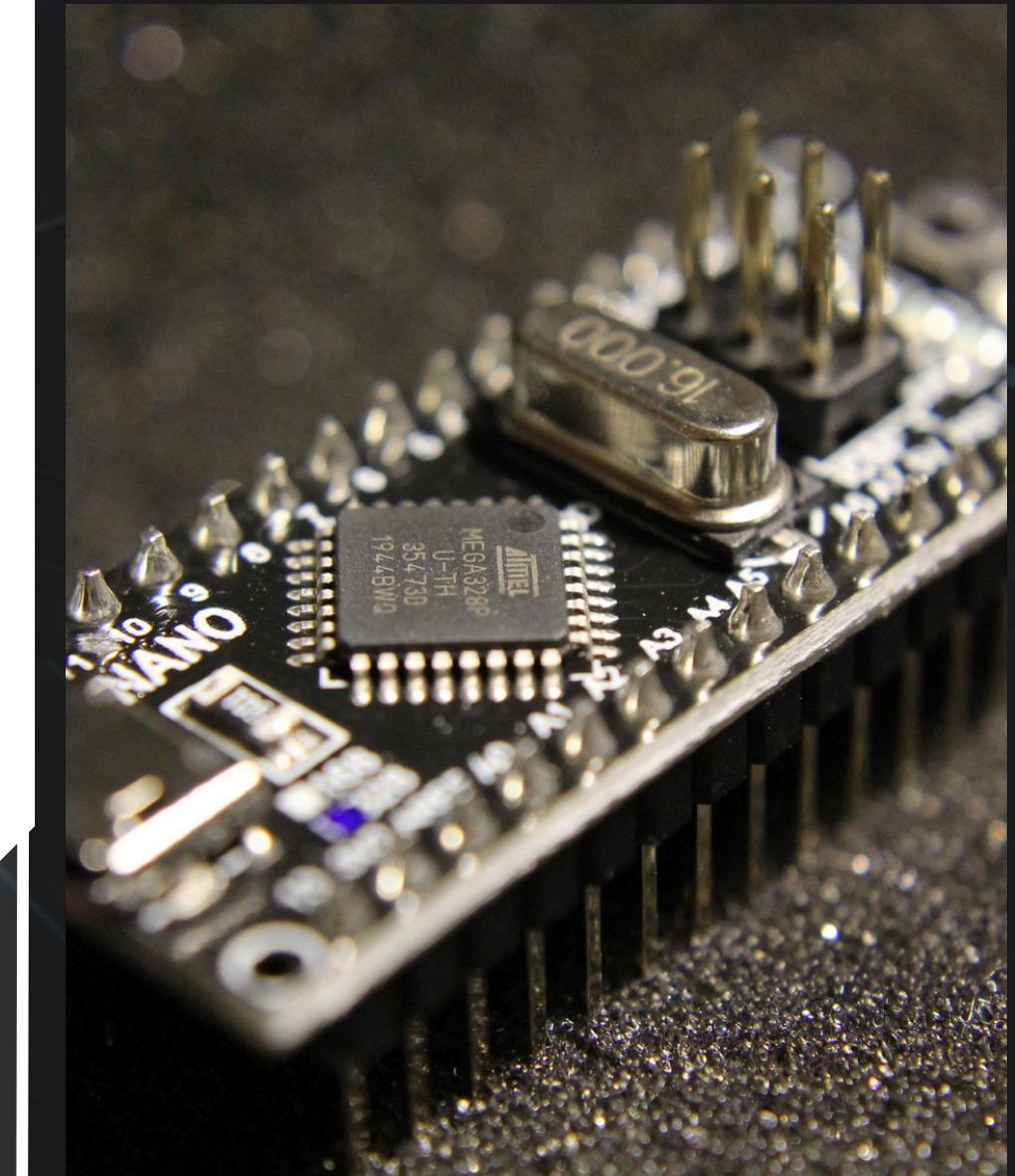




EMBEDDED SYSTEMS

The foundation of modern innovation, connecting the world through digital systems.





MCU CLOCKS

The clock system in a Microcontroller Unit (MCU) is responsible for generating the timing signals (clock pulses) that synchronize all operations inside the MCU — from instruction execution to peripheral communication.

A clock is a square wave signal (0/1) that ticks at a constant rate (frequency in Hz). Each tick tells the MCU when to fetch, decode, or execute instructions.





CLOCK TREE

Definition

A clock tree is the hierarchical structure that distributes the clock signal throughout the microcontroller. It defines how clock signals are generated, scaled, and routed from the clock source (like a crystal oscillator) to different parts of the system such as the CPU, memory, buses, and peripherals.

Functionalities

- Selects the clock source (e.g., internal oscillator, external crystal).
- Modifies the clock frequency using PLL (Phase-Locked Loop) or prescalers.
- Distributes the resulting clock to various internal modules.
- Manages clock gating to reduce power consumption (turning clocks off when not needed).



CLOCK SOURCES

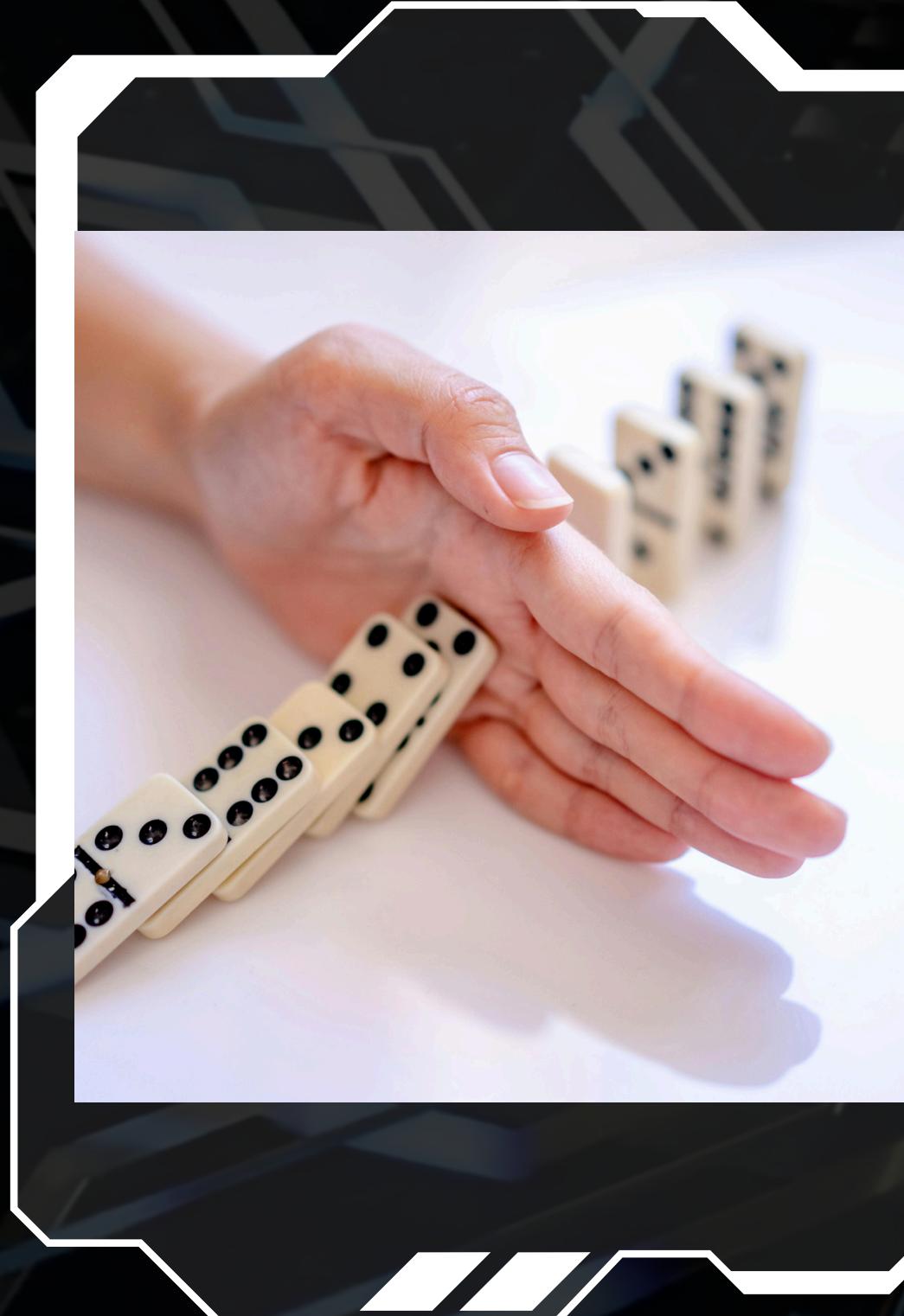
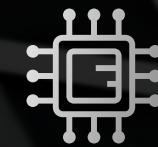
Internal

- Built-in (e.g., 8 MHz RC oscillator).
- Lower accuracy but doesn't require external parts.

External

- High accuracy.
- Often used in timing-critical applications (e.g., UART, USB).





MCU INTERRUPT

An interrupt is a mechanism that allows a microcontroller (MCU) to pause its current task, respond immediately to an important event (like a button press or sensor signal), and then return to what it was doing.



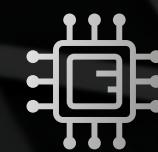
External



Software



Internal



INTERRUPT ARCHITECTURE



Interrupt
Sources



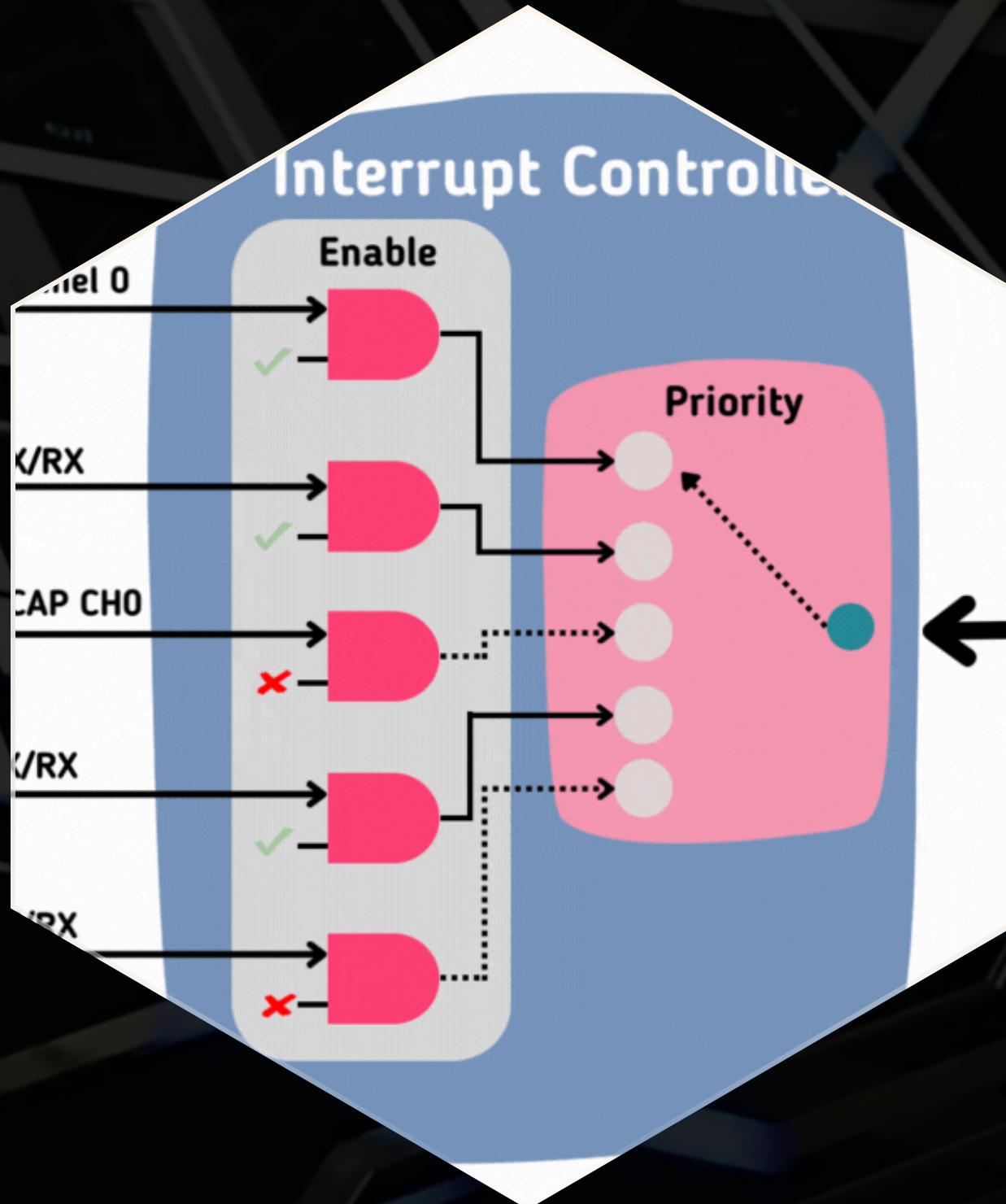
Interrupt Vector
Table



Interrupt
Controller



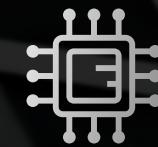
CPU Action



ISR DEFINITION

A small function that executes in response to an interrupt which must be short and fast.

- Interrupts can be disabled or masked to prevent unwanted triggers.
- Some MCUs support nested interrupts (handling multiple levels of priority).
- Poorly designed ISRs can cause bugs like missed events, lock-ups, or priority inversion.



INTERRUPT CONCEPTS



Determinism

Knowing what happens at every point in the timeline of the system.



Responsiveness

how fast will the system response for external events.



SYSTEM TYPES

Super loop

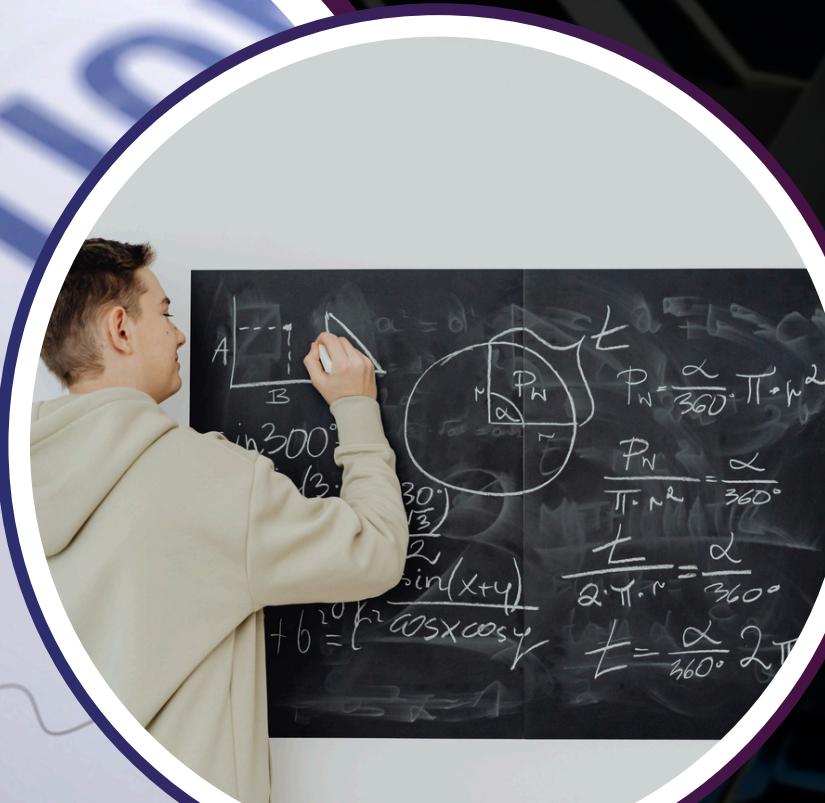
A Super Loop System is a simple, non-interrupt-based programming structure used in bare-metal embedded systems, where the entire program runs inside an infinite loop (`while(1)` or `for(;;)`), repeatedly executing tasks.

Foreground / Background

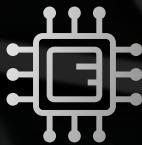
A Foreground/Background system is an embedded software design pattern that combines a background task that runs in a continuous super loop, and one or more foreground tasks that are executed through interrupts.



SOFTWARE MECHANISMS

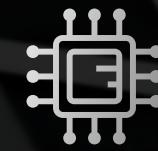


-  **Interrupt Service
Routine (ISR)**
-  **Interrupt Vector
Table (IVT)**
-  **Interrupt Priority
and Nesting**
-  **Interrupt Flags and
Acknowledgment**
-  **Disabling/Enabling
Interrupts**



STARTUP CODE

- Load the data from FLASH to RAM
 - Initialize vector table for Dynamic vector table only
 - Register Initializations
 - Calling main



THANK YOU!

Thank you for exploring computer technology—a field shaping our future!