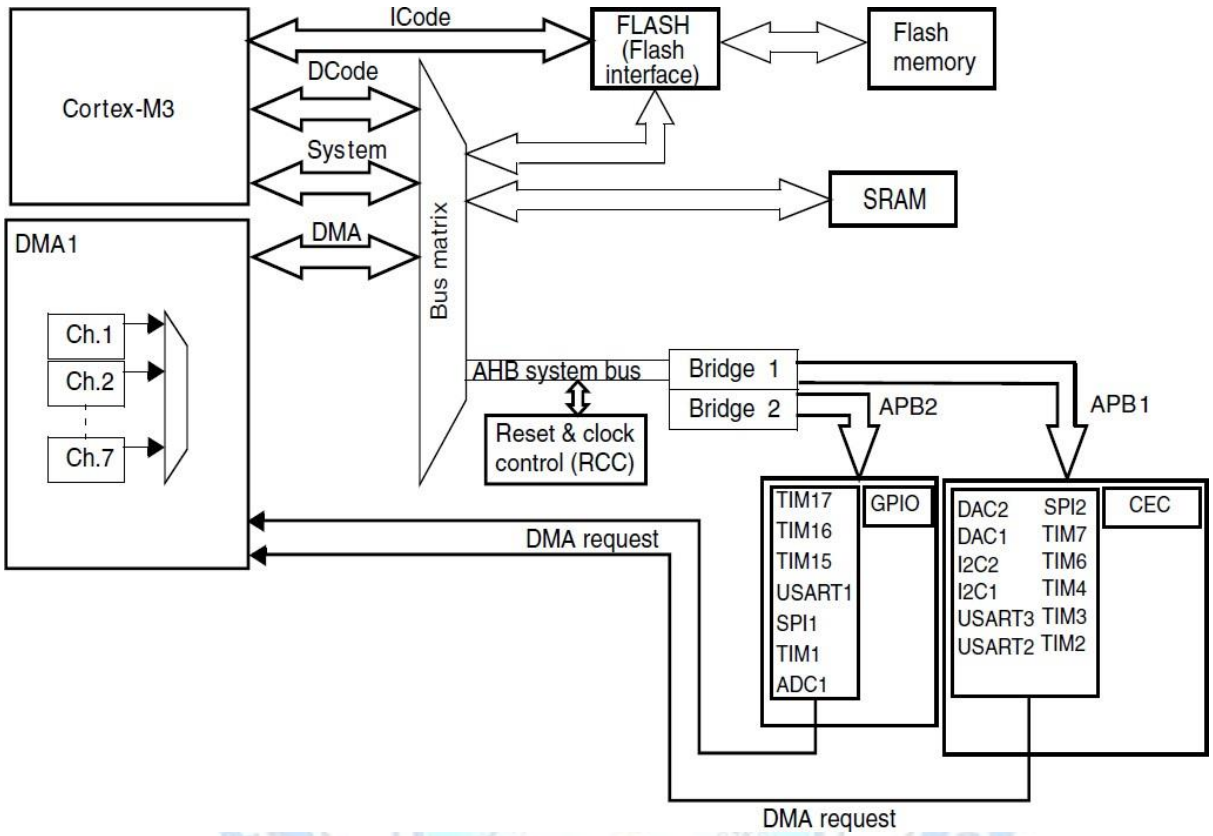
STM 32 ARCHITECTURE



MEMORY AND BUS ARCHITECTURE

- Three masters:
- Cortex™-M3 core DCode bus (D-bus) and System bus (S-bus)
- GP-DMA1 (general-purpose DMA)
- Three slaves:
- Internal SRAM
- Internal Flash memory
- AHB to APB bridges (AHB to APBx), which connect all the APB peripherals

ICode bus

Connects the instruction bus of the Cortex™-M3 core to the Flash memory instruction interface. Instruction fetches are performed on this bus.

DCode bus

Connects the DCode bus (literal load and debug access) of the Cortex™-M3 core to the Flash memory data interface

DMA bus

Connects the AHB master interface of the DMA to the bus matrix which manages the access of CPU DCode and DMA to the SRAM, Flash memory and peripherals

DMA Controller

DMA means **Direct Memory Access**.

Normally
Peripheral
↓
CPU
↓
Memory
The CPU performs all transfers.
With DMA Peripheral
↓
DMA Controller
↓
Memory
The CPU is free to execute other tasks.
The DMA shown here has multiple channels (Ch1 to Ch7), each serving different peripherals.
Example
ADC continuously stores conversion results into SRAM using DMA without CPU intervention.

Bus matrix
Manages the access arbitration between the core system bus and the
DMA master bus. The arbitration uses a round robin algorithm
AHB Bus
AHB means **Advanced High-performance Bus**.
High-speed bus connecting

- CPU
- Flash
- SRAM
- DMA

It carries large amounts of data quickly.
AHB/APB bridges (APB)
The two AHB/APB bridges provide full synchronous connections between the AHB and the two APB buses
APB buses operate at full speed (up to 24 MHz)
These bridges connect the high-speed AHB bus to the slower APB buses.
They perform speed conversion between the buses.
APB1 Bus
APB1 = Advanced Peripheral Bus 1
Used for **low-speed peripherals**.
APB2 Bus
APB2 is used for **high-speed peripherals**.
Flash Interface and Flash Memory
Flash memory stores
User program
Constants

Bootloader
The Flash Interface controls communication between the CPU and Flash memory.

SRAM

SRAM stores
Variables
Stack
Heap
Temporary data
SRAM is much faster than Flash but loses its contents when power is removed.

RCC (Reset and Clock Control)

RCC performs two important functions.

Clock Generation

Provides clock signals to

- CPU
- GPIO
- USART
- SPI
- Timers
- ADC

Without enabling the clock, peripherals cannot operate.

GPIO

GPIO allows connection with external hardware.

Timers

Timers perform
Delay generation
TIM1 – Advanced Control Timer **TIM2** – General Purpose Timer **TIM3** – General Purpose Timer **TIM4** – General Purpose Timer **TIM6** – Basic Timer **TIM7** – Basic Timer **TIM15, TIM16, TIM17** – General/Advanced Timers (depending on the STM32 series)

SPI

SPI is used for high-speed communication.

USART

USART performs serial communication.

I2C

Two-wire communication protocol.

WORKING:

The **STM32 microcontroller** operates by executing the user program stored in **Flash memory** using the **ARM Cortex-M3 processor core**.
The processor fetches instructions through the **I-Code bus**, accesses constant data through the **D-Code bus**, and communicates with **SRAM** and on-chip peripherals through the **System bus**.
A **Bus Matrix** efficiently manages data transfer between the CPU, memory, DMA, and peripherals, allowing multiple operations to occur simultaneously.

The **RCC (Reset and Clock Control)** module provides clock signals and reset control to all system components. High-speed components such as Flash, SRAM, and DMA communicate over the **AHB bus**, while peripheral devices like **GPIO, Timers, ADC, SPI, I²C, and USART** are connected through the **APB1 and APB2 buses**.

The **DMA controller** transfers data directly between memory and peripherals without CPU intervention, improving system performance.

Overall, the STM32 architecture enables fast, efficient, and reliable execution of embedded applications by integrating the ARM Cortex-M3 core with memory, buses, clock management, DMA, and a wide range of peripherals on a single chip.