ARM CORTEX-M ARCHITECTURE

- **ARM Cortex-M** is a **32-bit RISC microcontroller processor** widely used in embedded systems.
- It uses the **Harvard Architecture**, which has **separate buses for instructions and data**.
- This allows the processor to **fetch instructions and access data simultaneously**, improving execution speed.
- The Cortex-M processor provides the **high performance of a 32-bit processor** while maintaining the **compact code size** of 8-bit and 16-bit microcontrollers.

Major Components of ARM Cortex-M Architecture

- **Flash ROM**
  - Stores the program instructions permanently.
  - Instructions are fetched through the **ICode Bus**.
- **ICode Bus (Instruction Bus)**
  - Transfers program instructions from Flash ROM to the CPU.
  - Used only for instruction fetching.
- **System Bus**
  - Transfers data between the processor, RAM, and I/O peripherals.
  - Supports both read and write operations.
- **DCode Bus (Debug Bus)**
  - Used for debugging operations.
  - Helps in program testing and debugging.
- **Advanced High-Performance Bus (AHB)**
  - Available in Cortex-M4.
  - Provides high-speed communication for peripherals such as **USB**.
- **Private Peripheral Bus (PPB)**
  - Connects internal peripherals directly to the processor.
  - Used by components like the **Nested Vectored Interrupt Controller (NVIC)**.
- **Nested Vectored Interrupt Controller (NVIC)**
  - Manages all interrupts in the processor.
  - Supports nested interrupts and priority-based interrupt handling.
  - Reduces interrupt response time (low interrupt latency).

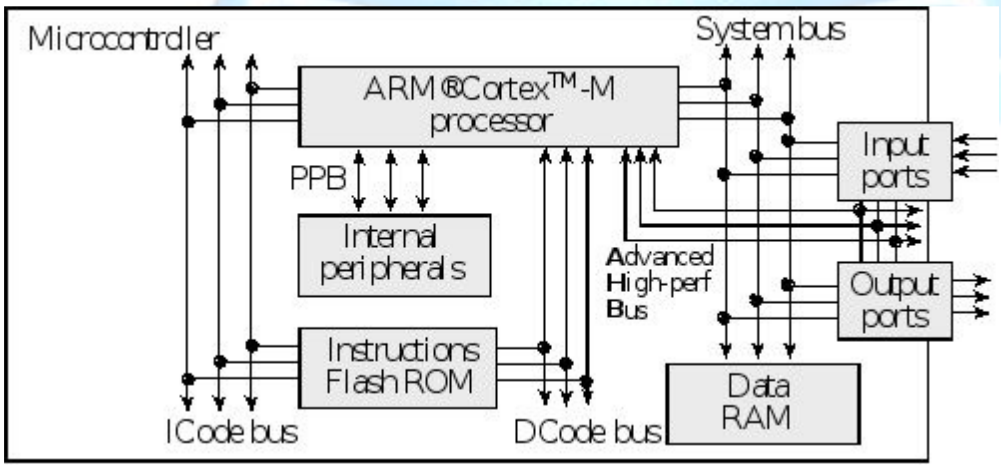


Figure 2.1. Harvard architecture of an ARM ® CortexTM-M-based microcontroller.

REGISTERS IN ARM CORTEX-M

Registers are small, high-speed memory locations inside the processor used to store data, addresses, and instructions temporarily.

General Purpose Registers (R0–R12)

- ARM Cortex-M has **13 general-purpose registers (R0–R12)**.
- They are used to:
  - Store data
  - Store memory addresses
  - Perform arithmetic and logical operations
- These registers can be used by the programmer during program execution.

Stack Pointer (R13 or SP)

- **R13** is called the **Stack Pointer (SP)**.
- It always points to the **top of the stack**.
- The stack is used to store:
  - Local variables
  - Function parameters
  - Return addresses
  - Temporary data

Types of Stack Pointer

1. **Main Stack Pointer (MSP)**
   - Used by the operating system.
   - Also used during interrupt handling.
   - Mainly used in embedded systems.
2. **Process Stack Pointer (PSP)**
   - Used by user applications.
   - Helps protect the operating system if the user program crashes.

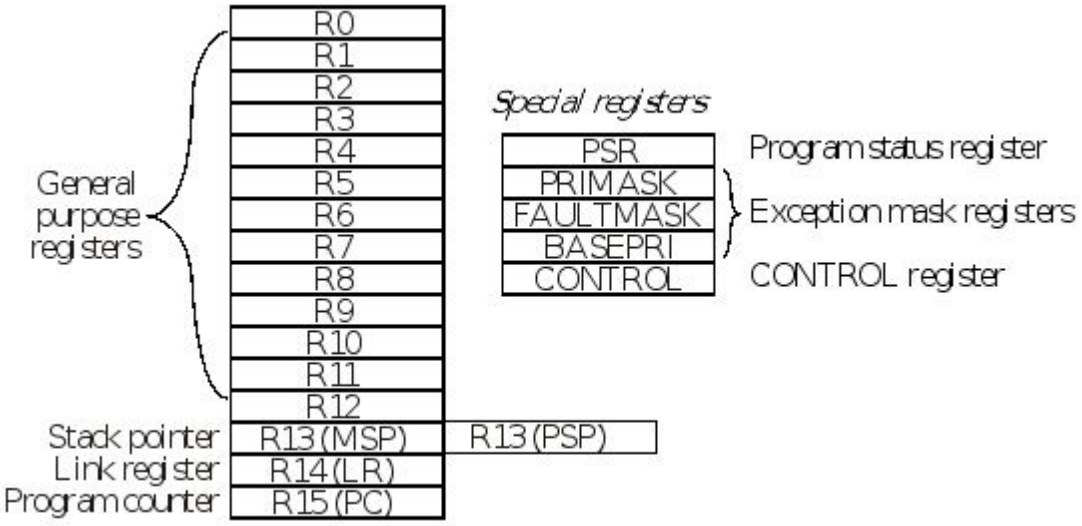
Note: Only **one stack pointer (MSP or PSP)** is active at a time.

Link Register (R14 or LR)

- **R14** is called the **Link Register (LR)**.
- It stores the **return address** when a function is called.
- After the function execution is completed, the processor returns to this stored address.
- LR is also used during **interrupts and exceptions**.

Program Counter (R15 or PC)

- **R15** is called the **Program Counter (PC)**.
- It stores the address of the **next instruction** to be executed.
- After fetching an instruction, the PC automatically increments by **2 or 4 bytes**.



Procedure Call Standard (AAPCS)

ARM follows the **ARM Architecture Procedure Call Standard (AAPCS)** for function calls.

Parameter Passing

- **R0–R3** are used to pass input arguments to a function.
- The **return value** of the function is stored in **R0**.

Register	Purpose
R0–R3	Function arguments
R0	Return value

Program Status Registers

ARM Cortex-M has **three status registers**.

1. APSR (Application Program Status Register)

- Stores the result of arithmetic and logical operations.
- Contains condition flags.

2. IPSR (Interrupt Program Status Register)

- Indicates which interrupt is currently executing.

3. EPSR (Execution Program Status Register)

- Stores execution state information.
- Indicates that the processor is executing **Thumb instructions**.

Program Status Register (PSR)

The **PSR** is a combination of:

- APSR
- IPSR
- EPSR

Condition Flags in APSR

Flag	Meaning
N	Negative result
Z	Zero result
C	Carry generated

Flag	Meaning
V	Signed overflow
Q	Saturation occurred

Explanation

- **N Flag** → Set when the result is **negative**.
- **Z Flag** → Set when the result is **zero**.
- **C Flag** → Set when an **unsigned carry/overflow** occurs.
- **V Flag** → Set when a **signed overflow** occurs.
- **Q Flag** → Indicates a **saturation operation** has occurred.

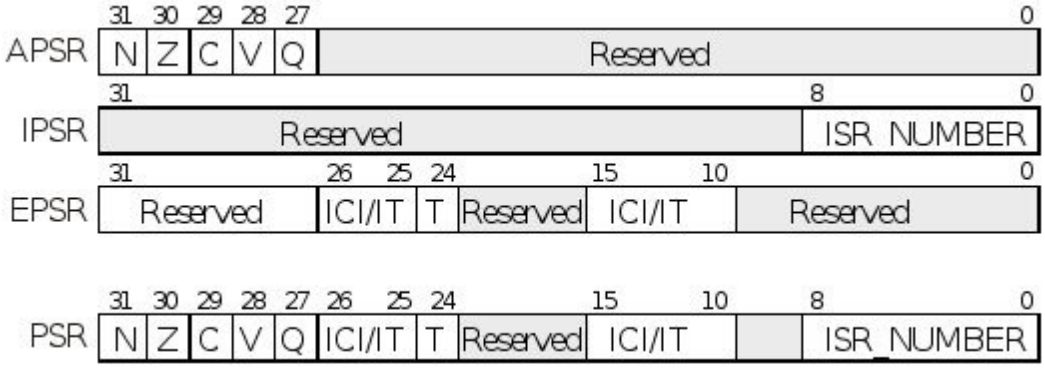


Figure 2.3. The program status register of the ARM ® CortexTM-M processor.

Special Registers

Thumb Bit (T)

- The **T bit** is always **1**.
- Indicates that the processor executes **Thumb instructions**.

ISR Number

- Indicates the **interrupt currently being executed**.
- If no interrupt is active, the value is **0**.

PRIMASK Register

- Used to **enable or disable interrupts**.

Bit Value	Operation
0	Interrupts enabled
1	Most interrupts disabled

FAULTMASK Register

- Used to enable or disable **fault exceptions**.

Bit Value	Operation
0	Faults enabled
1	Faults disabled

Note: The **Non-Maskable Interrupt (NMI)** cannot be disabled.

BASEPRI Register

- Used to set the **priority level of interrupts**.
- Higher-priority interrupts are allowed.
- Lower-priority interrupts are blocked.

Example:

- If **BASEPRI = 3**
  - Priority **0, 1, and 2** interrupts are allowed.
  - Priority **3 and above** are blocked until the current task finishes.

OPERATING MODES

ARM Cortex-M processor operates in **two privilege levels** and **two execution modes**.

1. Privilege Levels

The processor has **two privilege levels**.

a) Privileged Mode

- Has **full access** to all processor resources.
- Can access:
  - System timer
  - Interrupt controller (NVIC)
  - System control registers
- Mainly used by the **Operating System (OS)**.

b) Unprivileged Mode

- Has **limited access** to system resources.
- Cannot directly access:
  - System timer
  - Interrupt controller
  - Some system registers
- Mainly used by **user applications**.
- Improves system security by preventing unauthorized access.

CONTROL Register

The **CONTROL register** decides the processor's privilege level and stack pointer.

Bit 0 – Thread Mode Privilege Level (TPL)

Bit Value	Processor Mode
1	Privileged Mode
0	Unprivileged Mode

Bit 1 – Active Stack Pointer Selection (ASPSEL)

This bit selects which stack pointer is used.

Bit Value	Stack Pointer Used
0	Main Stack Pointer (MSP)
1	Process Stack Pointer (PSP)

Stack Pointer Usage

Main Stack Pointer (MSP)

- Used by the **Operating System**.
- Used during **interrupt handling**.
- Provides reliable system operation.

Process Stack Pointer (PSP)

- Used by **user applications**.
- Keeps user programs separate from the operating system.
- Prevents OS failure if the user program crashes.

Note: In high-reliability systems:

- **User programs** use **PSP**.
- **Operating System** uses **MSP**.

2. Execution Modes

ARM Cortex-M has **two execution modes**.

a) Thread Mode

- Also called the **Foreground Mode**.
- Executes the **main program**.
- Processor starts in Thread Mode after reset.
- **ISR_NUMBER = 0** indicates Thread Mode.

Features

- Executes normal application code.
- Can run in either:
  - Privileged mode
  - Unprivileged mode

b) Handler Mode

- Also called the **Background Mode**.
- Executes **Interrupt Service Routines (ISRs)**.
- Entered automatically when an interrupt occurs.

Features

- Handles interrupts and exceptions.
- Always uses the **Main Stack Pointer (MSP)**.
- Returns to Thread Mode after interrupt execution.