

1) FPU

I. What is FPU?

FPU stands for Floating Point Unit. It is a specialized hardware unit in a microcontroller (MCU) that is designed to perform floating-point arithmetic operations. Floating-point arithmetic is a type of arithmetic that is used to represent and compute real numbers. It is more complex than integer arithmetic, but it is also more precise.

FPUs are typically used in MCU applications that require high-precision floating-point calculations, such as:

- Motor control
- Signal processing
- Image processing
- Scientific computing

FPUs can also be used to accelerate the execution of complex mathematical functions, such as trigonometric functions and exponential functions and decrease the CPU load.

Some MCUs have integrated FPUs, while others require an external FPU coprocessor. External FPUs are typically used in high-performance MCU applications, where the extra cost and size are justified.

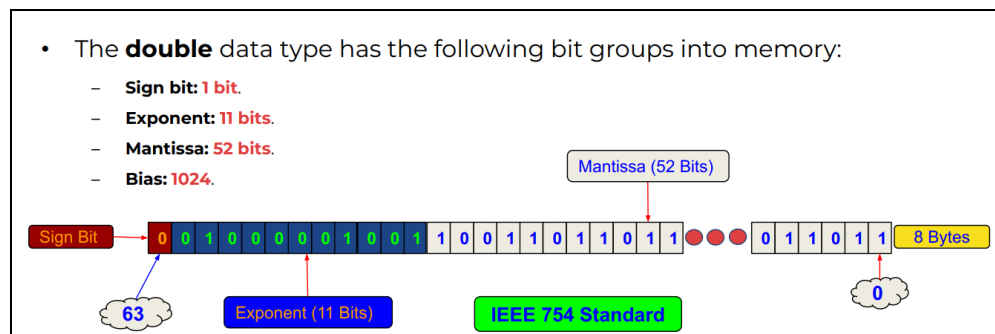
II. Benefits of using FPU in MCU:

- Higher performance for floating-point calculations
- Reduced code size
- Lower power consumption

III. Drawbacks of using FPU in MCU:

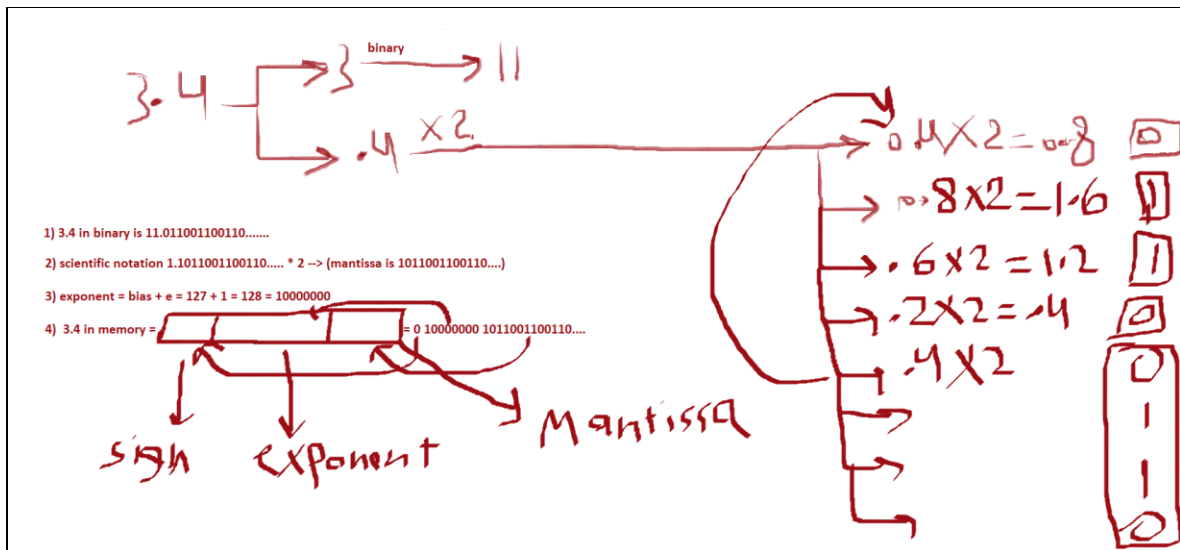
- Increased cost
- Increased size
- Increased complexity

Overall, the benefits of using an FPU in an MCU can outweigh the drawbacks, especially for applications that require high-precision floating-point calculations



II. How to represent a float number?

- Five steps to get the binary representation of a float number into memory:
 - Convert the **float value to binary**.
 - Convert the binary value to **scientific notation** form **$1.xxxx * 2^e$** .
 - If the value is **negative put 1 in the sign bit otherwise put 0**.
 - The **exponent** bits are the **binary representation of the Bias + e**.
 - **The Mantissa is the xxxx binary value**.



3) **Bus matrix**

I. What is bus matrix?

A bus matrix in a microcontroller (MCU) is a component that allows multiple bus masters to be connected to multiple bus slaves. This enables parallel access to a number of shared bus slaves from a number of different bus masters. The bus matrix determines the master that gains access to each slave, and routes the control signals and data signals between them.

Bus matrices are used in a variety of MCUs, including those with multiple processors, those with multiple peripherals, and those with complex memory hierarchies. They can provide a number of benefits, including:

Improved performance: By enabling parallel access to shared resources, bus matrices can improve the overall performance of an MCU.

Increased flexibility: Bus matrices make it easier to add and remove peripherals from an MCU, without having to redesign the MCU bus.

Reduced power consumption: By reducing the number of bus transactions, bus matrices can help to reduce the power consumption of an MCU.

A typical bus matrix in an MCU might connect the following bus masters and slaves:

Bus masters:

- Processor
- DMA controller
- Ethernet controller
- Bus slaves:
- Memory
- UART

I2C controller

When a bus master needs to access a bus slave, it sends a request to the bus matrix. The bus matrix arbitrates the requests and grants access to the slave to the master with the highest priority. Once a master has been granted access to a slave, the bus matrix routes the control signals and data signals between the master and the slave.

Bus matrices are an important component of many modern MCUs. They allow for the design of complex and high-performance systems, while still maintaining flexibility and low power consumption.

Here are some examples of MCUs that use bus matrices:

- STM32F4 series
- NXP Kinetis K series
- TI MSP432 series
- Microchip SAM4E series

These MCUs are all capable of running complex applications, such as embedded real-time operating systems, control systems, and multimedia applications. The bus matrix is an essential component that allows these MCUs to achieve the performance and flexibility required for these applications.

II. Example on bus matrix in STM32F401(XC/XB/XD/XE)

Figure 1. System architecture

