arm Education

Embedded Systems Essentials with Arm: Getting Started

Module 4

KV2 (4): Types of Interrupts

There are different types of interrupts. The primary types are ‘regular’ and ‘exceptional’ interrupts.

Regular interrupts are based on external events. For example, they are created by analog-to-digital conversions, or data received at serial ports. Because they are not related to what code the processor is currently executing, they are asynchronous.

Exceptional interrupts are caused by undesirable behavior of executed instructions. For example, undefined instructions, or when an overflow occurs for a given instruction. Because they happen at the same time as the code, they are synchronous.

Most interrupts can be masked or disabled. The term ‘masked’ comes from the bit mask which is defined to filter unneeded or unwanted interrupts. But not all interrupts are maskable, for example, the watchdog interrupt. In this case, they are called non-maskable interrupts or NMIs.

External interrupts are managed and prioritized by the nested vector interrupt controller or NVIC. A “nested interrupt” is an interrupt routine call that is ‘nested’ or contained within another active interrupt.

The NVIC register has two possible states for managing the interrupts: the “enable state”, which means that interrupts are recognized, and the “pending state”, which means that an interrupt has been requested and detected but not yet implemented.

The Nested Vectored Interrupt Controller or NVIC manages and prioritizes external interrupts, facilitates low-latency exception and interrupt handling, and controls power management.

The NVIC also provides the vector table as a set of real vectors or addresses. In addition, the NVIC saves and automatically restores a set of the CPU registers, performs a quick entry to the next pending interrupt without a complete pop/push sequence, and serves a complete set of 255 internal or 240 external interrupts.

The Priority Mask Register or PRIMASK register prevents activation of all exceptions with configurable priority.

This eliminates the problem of so-called data races. A data race occurs when two independent threads may access the same data, for example, if one thread is manipulating data, and the other tries to access that data at the same time. To prevent situations like this, the priority mask or PM flag is either set to ‘1’ to prevent activation of all exceptions with configurable priority, or cleared to ‘0’ to allow activation of all exceptions.

The Cortex Microcontroller Software Interface Standard, or CMSIS, is a vendor-independent interface for debugging that also provides commands for controlling the NVIC. The CMSIS-Core API provides dedicated commands to manipulate the PRIMASK and the PM flag.

Exceptions are prioritized to order the response of simultaneous requests. Smaller numbered requests receive higher priority. The priorities of some exceptions are fixed. For example, reset is negative 3, and has the highest priority, followed by non-maskable interrupts or NMIs at negative 2, and hard fault at negative 1.

© Arm 2020

The priorities of other, external exceptions are adjustable and their values are stored in the interrupt priority register.

When multiple exceptions are valid at the same time, that is, more than one exception occurs during execution of an instruction, they are handled by the processor according to their priority.

In the case of simultaneous exception requests, the lowest exception type number is serviced first.

When a new exception is requested while a handler is executing, if the new priority is higher than current priority, then the new exception handler preempts the current exception handler.

If the new priority is lower than or equal to current priority, the new exception is held in a pending state while the current handler continues and completes execution. Following completion, the previous priority level is restored and the new exception is handled, if the priority level allows it.

© Arm 2020