

## Interrupts do not appear to work when debugging

When debugging a project after a power on reset (POR) it is sometimes observed that interrupts no longer appear to work. The interrupts work fine when not in debug mode.

This can happen as a result of the NMI pin being active at POR or active at the time the debug connection is made. If the NMI is active the program counter (PC) will be pointing to the NMI interrupt service routine (ISR). Some debug tools simply set the PC to the reset vector once the core is halted and the debug connection is established. If this happens, the NMI ISR is never entered. This results in the NMI exception being left active. The NMI exception has the highest priority level, other than reset. This blocks any lower priority exception/interrupt from being recognized by the ARM core until the NMI exception has been handled.

This situation typically only arises when the NMI pin is connected to some other signal in the application. This can cause the NMI to be active during a POR or other reset event. If possible, ensure the NMI pin is never allowed to be pulled low during a POR or other reset event and is only low during a real NMI condition.

If you find that you are running into this situation and the debugger is not setting the PC to the NMI ISR if the NMI is active when the debug connection is established, the following procedure may resolve the issue:

- 1) Set the debugger to halt the core at the reset vector (first line of code)
- 2) Write the PC to the NMI ISR vector (located in the map file or from memory address 0x8). Note that the PC must be written to the reported vector address minus 1
- 3) Re-write the stack pointer (SP) to its present value. This may not be required depending on the compiler/debugger being used.
- 4) Set the code executing.

This ensures that the NMI ISR will first be executed and then will return to the reset vector and begin code execution as normal. If the NMI ISR assumes a certain device initialization state then unexpected errors may occur and these need to be handled appropriately.

If you are not using the NMI function and are “disabling” it by simply changing the NMI pin port mux to a different function then this will not resolve this mode of operation. At reset, before any code executes, the NMI function will still be active up until the pin port mux selection is changed in the code. To correctly disable the NMI function it is recommended to do this by means of the NMI\_DIS bit in the FOPT register. Refer to the appropriate device reference manual for a description of how to disable the NMI function.