At the system level, bus accesses are filtered by the Memory Protection Controller (MPC) and System Protection Unit (SPU) security components. These components trigger a BusFault using bus error response, not a SecureFault.

Access to a peripheral register that violates security permissions triggers the SPU event PERIPHACCERR. For more information about the SPU, see SPU — System protection unit on page 180. See also Peripherals with split security access on page 129.

If a peripheral DMA controller, or the the coprocessor, attempts to access a memory region that is not allowed by the MPC, the MPC will generate a bus error response that triggers a BusFault. The MPC also generates the MEMACCERR event. For more information, see MPC — Memory Privilege Controller on page 174.

7.6 Peripherals with split security access

Some peripherals have split security access, meaning they can handle both secure and non-secure access. A subset of the peripheral's functions can be secure, while another subset is non-secure. The security is configured using SPU registers.

The peripheral instantiation table in Instantiation on page 216 details the peripherals with split access.

Split security access is handled either on the register level or on the bit level, as explained in the following sections.

Register level split security access

For this group of peripherals, security is enforced at the register level. Split security settings apply for the entire register. Illegal access to the register will trigger a security fault. For example, if a register is configured as secure and the register is accessed from non-secure code, a security fault with a bus fault will be generated. A security fault due to an illegal access triggers the SPU event PERIPHACCERR.

Bit level split security access

For this group of peripherals, security is enforced at the register bit level. Split security settings are applied to individual bits of the register. The register supports access from both secure and non-secure code.

No exceptions are triggered for the access, however the following apply:

- Writing a secure bit from non-secure code will have no effect
- Reading a register from non-secure code will return `0` for all bits that are secure

For example, if bit `i` is configured as secure, then the following apply:

- Non-secure write access to the register will not change bit `i`
- Non-secure read access to the register will read `0` for the bit at position `i`

Interrupts

Some peripherals have split security interrupts. This means the interrupt can be configured with a security attribute.

An interrupt may be generated during secure or non-secure execution, and the interrupt handler is executed based on the interrupt's security attribute.

Interrupts implement split security at the register level. For instance, if interrupt 0 is configured as secure and there is a non-secure read/write access to registers INTEN0, INTENSET0, INTENCLR0, or INTPEND0, a security fault with a bus fault will be generated.