Usage fault status register (UFSR) 0xE000ED2A

Bits 31:26 Reserved, must be kept cleared
Bit 25 DIVBYZERO: Divide by zero usage fault
Set when processor has executed an SDIV or UDIV instruction with a divisor of 0. Enable trapping of divide by zero by setting CCR.DIV_0_TRP.
Bit 24 UNALIGNED: Unaligned access usage fault
Set when the processor has made an unaligned memory access. Enable trapping of unaligned accesses by setting CCR.UNALIGN_TRP. Unaligned LDM, STM, LDRD, and STRD instructions always fault irrespective of the setting of UNALIGN_TRP.
Bits 23:20 Reserved, must be kept cleared
Bit 19 NOCP: No coprocessor usage fault
Set when the processor has attempted to access a coprocessor, but it does not support coprocessor instructions.
Bit 18 INVPC: Invalid PC load usage fault
Set when the processor has attempted an illegal load of EXC_RETURN to the PC, as a result of an invalid context, or an invalid EXC_RETURN value.
Bit 17 INVSTATE: Invalid state usage fault
Set when the processor has attempted to execute an instruction that makes illegal use of the EPSR.
Bit 16 UNDEFINSTR: Undefined instruction usage fault
Set when the processor has attempted to execute an undefined instruction.

Bus fault status register (BFSR) 0xE000ED29

Bit 15 BFARVALID: Bus Fault Address Register (BFAR) valid flag
Set if BFAR holds a valid fault address.
Bit 14 Reserved, must be kept cleared
Bit 13 LSPERR: Bus fault on floating-point lazy state preservation
Bit 12 STKERR: Bus fault on stacking for exception entry
Bit 11 UNSTKERR: Bus fault on unstacking for a return from exception.
Bit 10 IMPRECISERR: Imprecise data bus error
When the processor sets this bit to 1, it does not write a fault address to the BFAR. This is an asynchronous fault. Therefore, if it is detected when the priority of the current process is higher than the bus fault priority, the bus fault becomes pending and becomes active only when the processor returns from all higher priority processes. If a precise fault occurs before the processor enters the handler for the imprecise bus fault, the handler detects both IMPRECISERR set to 1 and one of the precise fault status bits set to 1.
Bit 9 PRECISERR: Precise data bus error
Bit 8 IBUSERR: Instruction bus error
The processor detects the instruction bus error on prefetching an instruction, but it sets the IBUSERR flag to 1 only if it attempts to issue the faulting instruction. When the processor sets this bit is 1, it does not write a fault address to the BFAR.

Mem.mgmt fault status register (MMFSR) 0xE000ED28

Bit 7 MMARVALID: Memory Management Fault Address Register (MMAR) valid
Bit 6 Reserved, must be kept cleared
Bit 5 MLSPERR: MemMng fault occurred during floating-point lazy state preservation
Bit 4 MSTKERR: Memory manager fault on stacking for exception entry
Bit 3 MUNSTKERR: Mem. manager fault on unstacking for a return from exception
Bit 2 Reserved, must be kept cleared
Bit 1 DACCVIOL: Data access violation flag
Bit 0 IACCVIOL: Instruction access violation flag
This fault occurs on any access to an XN region, even the MPU is disabled or not present.

Only Reset and NMI can preempt the fixed priority hard fault. A hard fault can preempt any exception other than Reset, NMI, or another hard fault.

Hard fault status register (HFSR) 0xE000ED2C

Bit 31 DEBUG_VT: Reserved for Debug use
Write 0 to this bit, otherwise behavior is unpredictable.
Bit 30 FORCED: Forced hard fault
Indicates escalation of a fault with configurable priority that cannot be handles, either because of priority or **because it is disabled**.
Bits 29:2 Reserved, must be kept cleared
Bit 1 VECTTBL: Vector table hard fault
Indicates a bus fault on a vector table read during exception processing. This error is always handled by the hard fault handler.
Bit 0 Reserved, must be kept cleared

The processor enters a **lockup state** if a hard fault occurs when executing the NMI or hard fault handlers. When the processor is in lockup state it does not execute any instructions. The processor remains in lockup state until either it is reset, an NMI occurs, or it is halted by a debugger.

When in lockup state, the processor repeatedly fetches the same instruction, from a fixed address, the Lockup address, determined by the nature of the fault; for most faults 0xFFFFFFFE.

Exception stack frame layout

```
[SP] ⟶  R0    R1    R2    R3
        R12   LR    PC    xPSR
      ( S0    S1    ...   S15        FPSCR ) - see FPU and lazy stacking
(aligner - see CCR.STKALIGN)
```