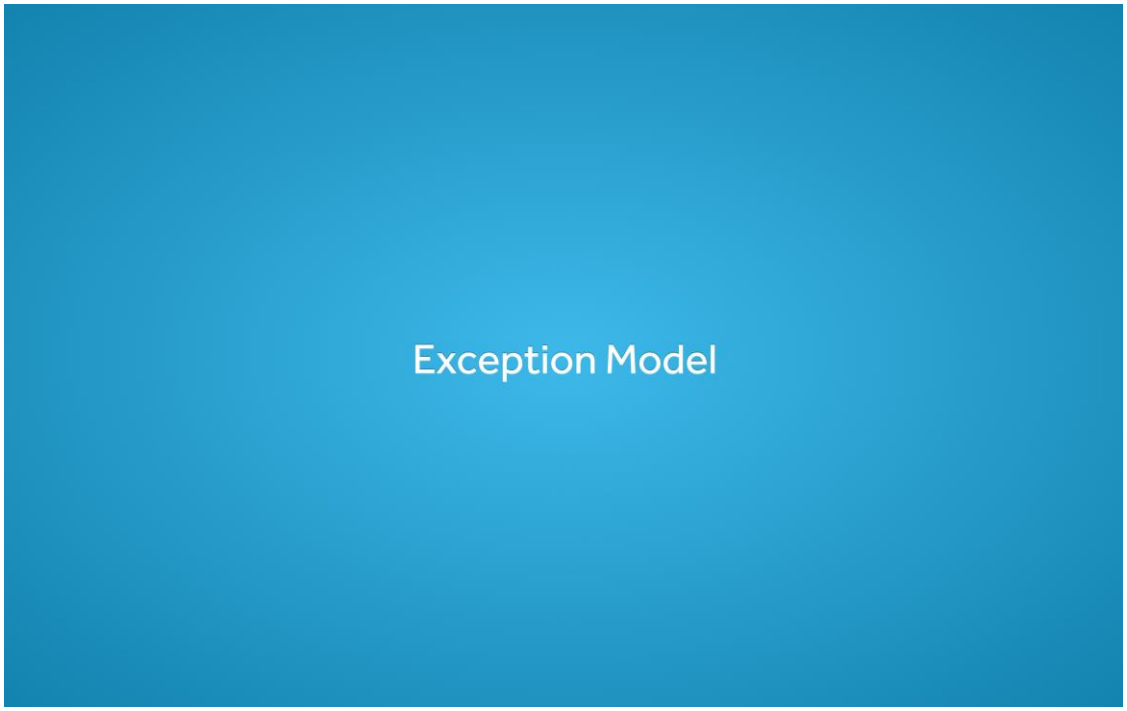


Exception States

Inactive - The exception is not active and not pending.

Pending - The exception is waiting to be serviced by the processor. An interrupt request has been made, but not processed yet.

Active - An exception is being serviced by the processor but has not completed.

Active and Pending - The exception is being serviced by the processor and there is a pending exception from the same source.

Exception Types

Reset -

- Reset is invoked on power up or a warm reset.
- The exception model treats reset as a special form of exception. When reset is asserted, the operation of the processor stops, potentially at any point in an instruction. When reset is deasserted, execution restarts from the address provided by the reset entry in the vector table.
- Execution restarts as privileged execution in Thread mode.

Exception Types

A **Non Maskable Interrupt (NMI)** can be signaled by a peripheral or triggered by software.
This is the highest priority exception other than reset.
It is permanently enabled and has a fixed priority of -2.

NMIs cannot be:

- Masked or prevented from activation by any other exception
- Preempted by any exception other than Reset

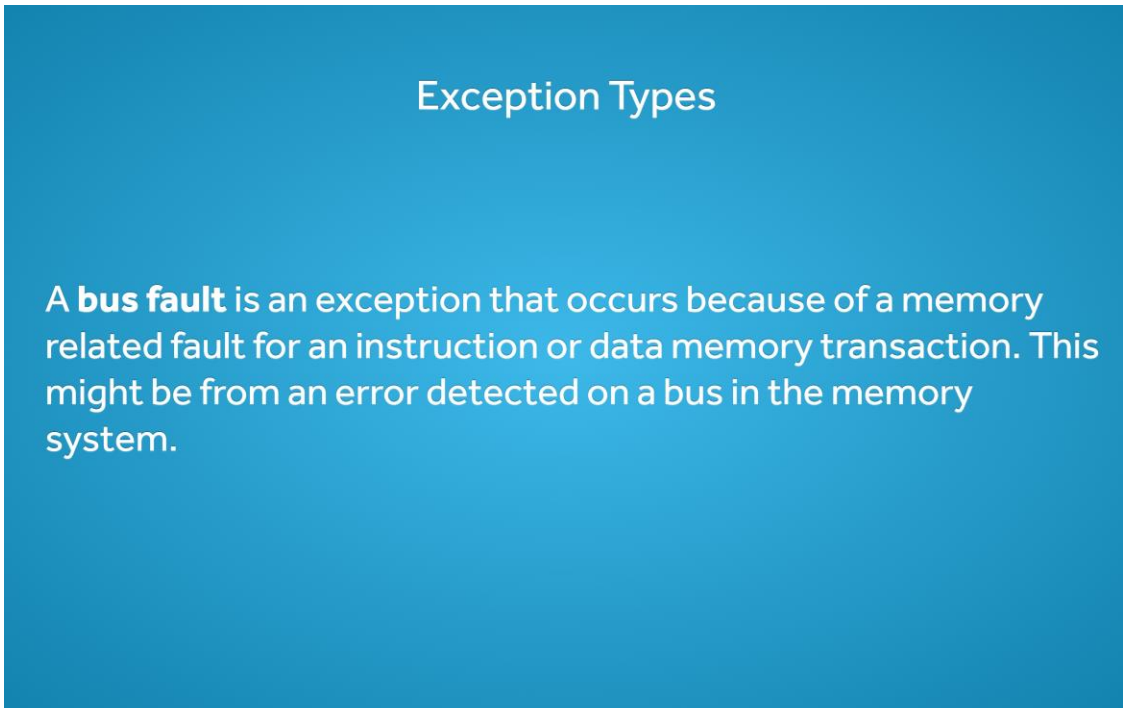
Exception Types

A **hard fault** is an exception that occurs because of an error during exception processing, or because an exception cannot be managed by any other exception mechanism.

Hard faults have a fixed priority of -1 meaning they have higher priority than any exception with configurable priority.

Exception Types

A **memory management fault** is an exception that occurs because of a memory protection related fault. The MPU or the fixed memory protection constraints determines this fault, for both instruction and data memory transactions.



Exception Types

A **usage fault** is an exception that occurs in case of an instruction execution fault. This includes:

- An undefined instruction
- An illegal unaligned access
- Invalid state on instruction execution
- An error on exception return.

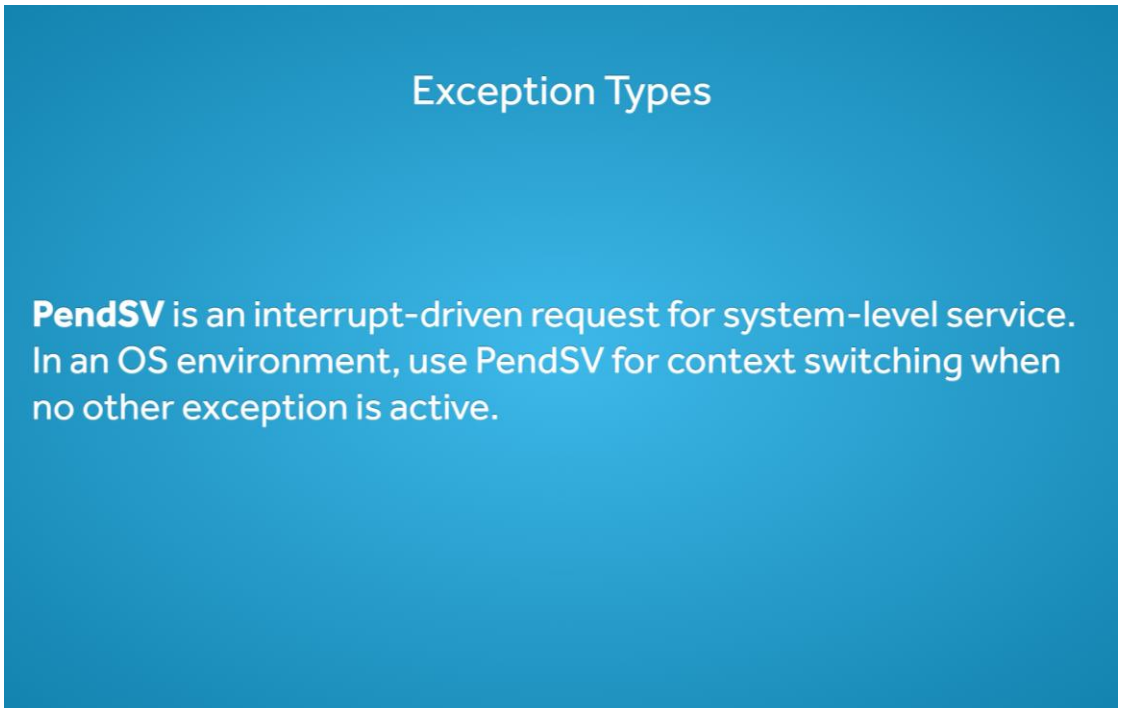
The following can cause a usage fault when the core is configured to report it:

- An unaligned address on word and halfword memory access
- Division by zero

Exception Types

A **supervisor call (SVC)** is an exception that is triggered by the SVC instruction. In an OS environment, applications can use SVC instructions to access OS kernel functions and device drivers.

An example of a SVC call is an unprivileged thread using a SVC call to return the program execution to a privileged thread

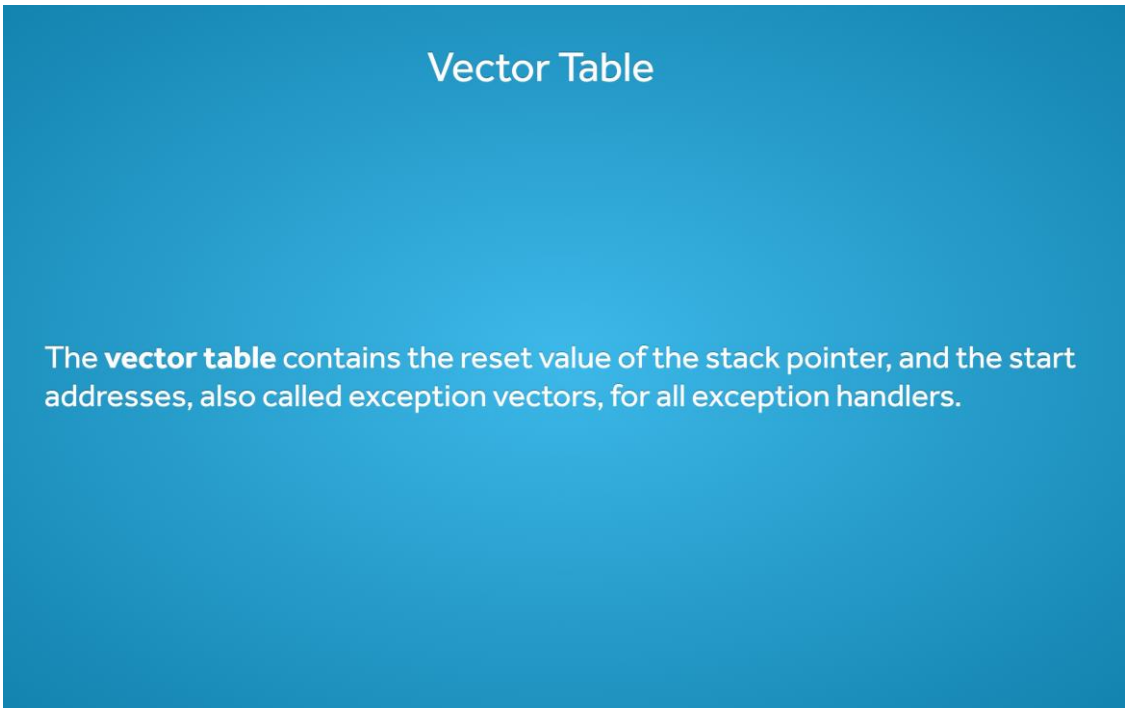


Exception Types

A **SysTick exception** is an exception the system timer generates when it reaches zero. Software can also generate a SysTick exception. In an OS environment, the processor can use this exception as system tick.

Exception Types

An **interrupt, or IRQ**, is an exception signaled by a peripheral, or generated by a software request. All interrupts are asynchronous to instruction execution. In the system, peripherals use interrupts to communicate with the processor.



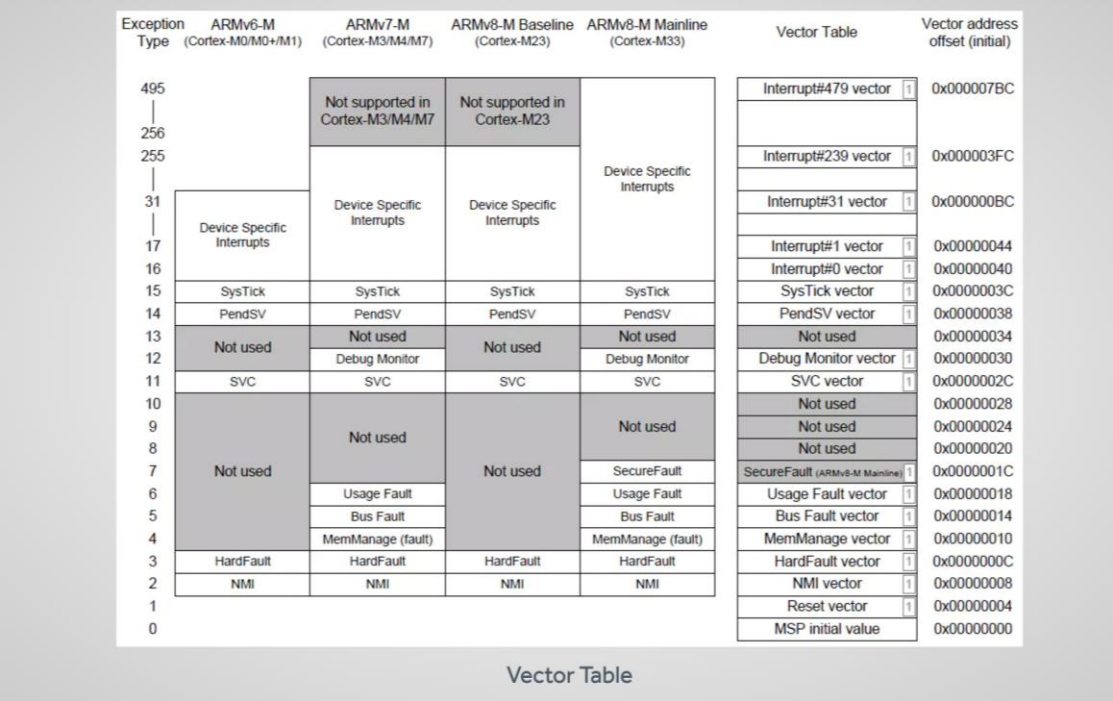
Vector Table

Cortex - M4

Figure 11. Vector table

Exception number	IRQ number	Offset	Vector
255	239	0x03FC	IRQ239
. . .		. . .	. . .
		0x004C	
18	2	0x0048	IRQ2
17	1	0x0044	IRQ1
16	0	0x0040	IRQ0
15	-1	0x003C	Systick
14	-2	0x0038	PendSV
13			Reserved
12			Reserved for Debug
11	-5	0x002C	SVCall
10			
9			Reserved
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

MS30018V1



Exception Type	ARMv6-M (Cortex-M0/M0+/M1)	ARMv7-M (Cortex-M3/M4/M7)	ARMv8-M Baseline (Cortex-M23)	ARMv8-M Mainline (Cortex-M33)	Vector Table	Vector address offset (initial)
495 | 256		Not supported in Cortex-M3/M4/M7	Not supported in Cortex-M23	Device Specific Interrupts	Interrupt#479 vector [1]	0x000007BC
255		Device Specific Interrupts	Device Specific Interrupts		Interrupt#239 vector [1]	0x000003FC
31 | 17	Device Specific Interrupts				Interrupt#31 vector [1]	0x000000BC
16					Interrupt#1 vector [1]	0x00000044
15	SysTick	SysTick	SysTick	SysTick	Interrupt#0 vector [1]	0x00000040
14	PendSV	PendSV	PendSV	PendSV	SysTick vector [1]	0x0000003C
13	Not used	Not used	Not used	Not used	PendSV vector [1]	0x00000038
12		Debug Monitor		Debug Monitor	Not used	0x00000034
11	SVC	SVC	SVC	SVC	Debug Monitor vector [1]	0x00000030
10	Not used	Not used	Not used	Not used	SVC vector [1]	0x0000002C
9					Not used	0x00000028
8					Not used	0x00000024
7				SecureFault	Not used	0x00000020
6		Usage Fault		Usage Fault	SecureFault (ARMv8-M Mainline) [1]	0x0000001C
5		Bus Fault		Bus Fault	Usage Fault vector [1]	0x00000018
4		MemManage (fault)		MemManage (fault)	Bus Fault vector [1]	0x00000014
3	HardFault	HardFault	HardFault	HardFault	MemManage vector [1]	0x00000010
2	NMI	NMI	NMI	NMI	HardFault vector [1]	0x0000000C
1					NMI vector [1]	0x00000008
0					Reset vector [1]	0x00000004
					MSP initial value	0x00000000

Vector Table

