Memory protection units (MPU). A look at ARM MPUs.

One typical requirement of multitasking operating systems is to run tasks encapsulated so that one task does not influence another task. There are two main approaches how this isolation of tasks can be done:
- In software; there is not hardware to enforce isolation, tasks must cooperate. As you can imagine in complex systems this is not so easy to do.
- In hardware and software; there exist special hardware blocks that can prohibit the access to different resources.

For memory protection the two main hardware blocks employed by hardware manufacturers are the memory management unit (MMU) and the memory protection unit (MPU). The main difference between this two hardware blocks is that the MMU also provides virtual memory capabilities. There are many platforms out there where an MMU is not really needed because all the tasks are allocated static (for example in avionics or automotive domain, and not only there).

What does the ARM MPU...

The ARM MPU protection mechanism is based (as also other vendors MPUs) on defining regions. A region is basically a chunk of memory and a set of attributes that define the access rights and modes to that memory space. One can configure a region by specifying the start location and the size of the location (the size of the location varies from 4KB to 4GB). The OS sets up the access rights (read and write, read only, no access) for different masters on the bus, cache and write buffer policies. Additional conditions can be set based on the processor mode, for instance a user space application could be prohibited to access a
certain memory space while a kernel mode application could access it. While running, the MPU will check if a certain access satisfies the conditions set and if this is not the case a signal is generated. This signal is sent then to the core that reacts by taking an exception that is sent then to the abort handler. This handler determines the abort type and then calls an appropriate service routine.

The ARM MPU regions have the following attributes:
- Region start address is a multiple of its size
- Region size is a power of two between 4KB and 4GB
- Regions can overlap other regions
- Regions have assigned priorities
- Attributes of overlapping regions will be set to the attributes of the higher priority region (if overlapping regions exist)