The Memory Management Unit

An important function of the *Memory Management Unit* (MMU) is to enable the system to run multiple tasks, as independent programs running in their own private virtual memory space. They do not need any knowledge of the physical memory map of the system, that is, the addresses that are actually used by the hardware, or about other programs that might execute at the same time.

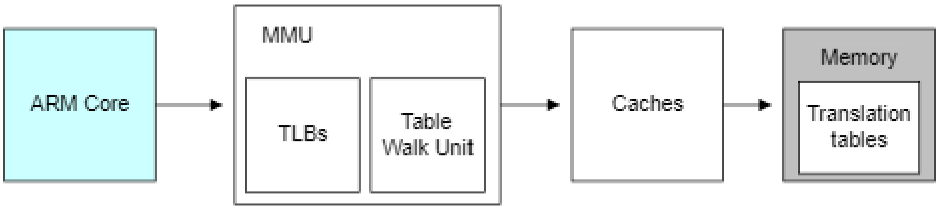

Figure 12.1. The Memory Management Unit

You can use the same virtual memory address space for each program. You can also work with a contiguous virtual memory map, even if the physical memory is fragmented. This Virtual Address space is separate from the actual physical map of memory in the system. You can write, compile, and link applications to run in the virtual memory space.

An example system, illustrating the virtual and physical views of memory, is shown in Figure 12.2. Different processors and devices in a single system might have different virtual and Physical Address maps. The OS programs the MMU to translate between these two views of memory.

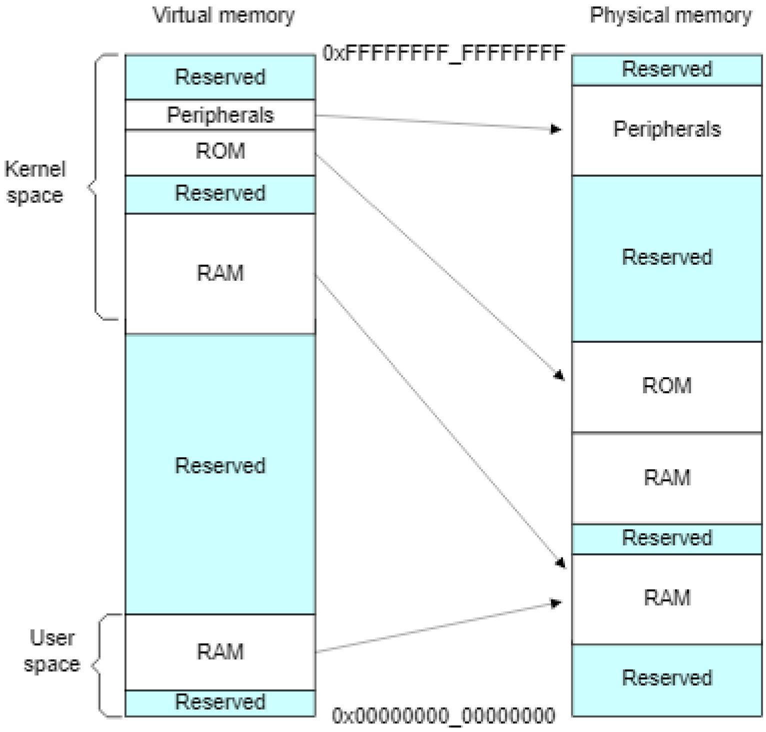

Figure 12.2. Virtual and physical memory

To do this, the hardware in a virtual memory system must provide address translation, which is the translation of the Virtual Address issued by the processor to a Physical Address in the main memory.

Virtual Addresses are those used by you, and the compiler and linker, when placing code in memory. *Physical Addresses* are those used by the actual hardware system.

The MMU uses the most significant bits of the Virtual Address to index entries in a *translation table* and establish which block is being accessed. The MMU translates the Virtual Addresses of code and data to the Physical Addresses in the actual system. The translation is carried out automatically in hardware and is transparent to the application. In addition to address translation, the MMU controls memory access permissions, memory ordering, and cache policies for each region of memory.

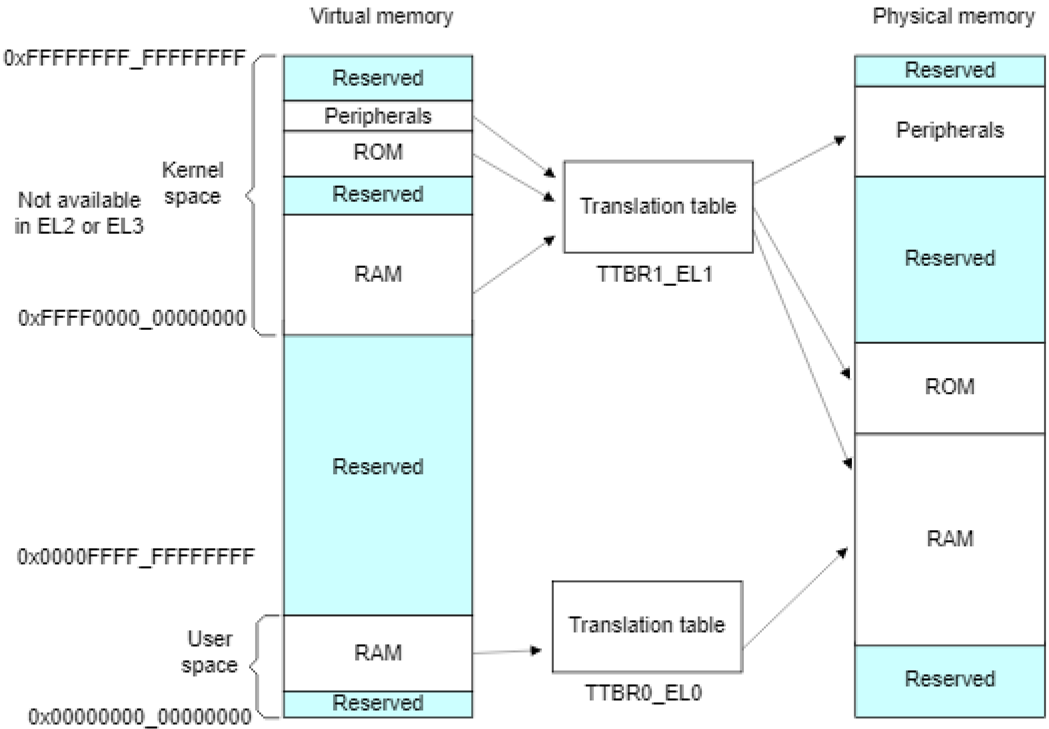

Figure 12.3. Address translation using translation tables

The MMU enables tasks or applications to be written in a way that requires them to have no knowledge of the physical memory map of the system, or about other programs that might be running simultaneously. This allows you to use the same virtual memory address space for each program.

It also lets you work with a contiguous virtual memory map, even if the physical memory is fragmented. This Virtual Address space is separate from the actual physical map of memory in the system. Applications are written, compiled and linked to run in the virtual memory space.