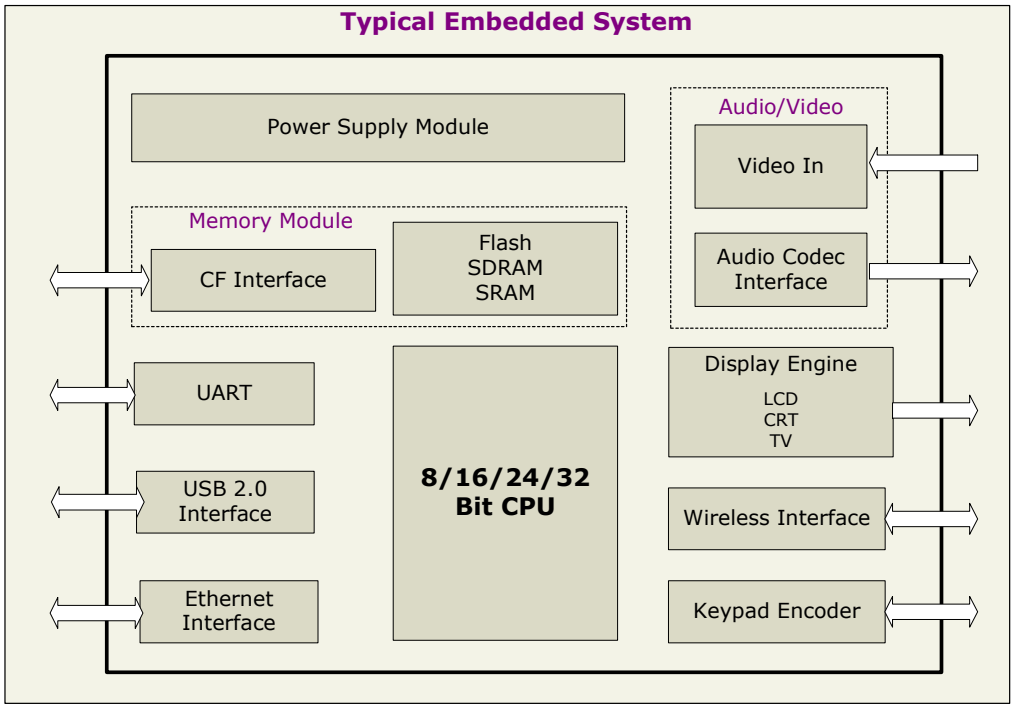
CPU's in Embedded Systems

Considering just performance and power can be pretty misleading in CPU selection. There is always an optimal solution for the desired functionality and performance. The same holds true for deciding upon a CPU in an Embedded system. Generally, designers chose the processor solely based on the past experience with a particular CPU.

To decide upon which CPU to use, the designer should consider overall system features such as: complexity of overall design, design reuse, performance, power, size, cost, tools and OS support and availability.



The design's complexity helps determine the CPU to use. If the design calls for the deployment of a single state machine with interrupts from a small set of peripherals, then a small CPU and/or micro controller such as the MCS51 or the Z80 could be the best choice. Many systems such as industrial timer may fit this category, as the memory footprint is small, the signal is slow and battery consumption must be extremely low.

The application and its interaction will dictate the design's complexity and may also determine whether it requires a real-time operating system (RTOS). Typically, as the application complexity increases, the need for a greater bit-width processor increases.
The selection of the CPU will greatly impact performance of the overall system. Specifically, features like 8 / 16 / 24 / 32 bit architecture, RISC / CISC / DSP architecture, cache, MMU, pipelining, branch prediction and super-scalar architecture, all affect the speed of a system. Depending on system needs, these features may be necessary to achieve peak performance of the system. There are various benchmarking threshold data available for various 8/16/32 bit processors like MIPS (Million Instructions per Second), EEMBC, Dhrystone, MIPS/MHz etc, which can be taken as reference or comparison base.

Utility of the embedded system will determine the power the design can consume. If the design is battery operated, the CPU will need to be as power-conscious as possible. For instance, some CPUs have the ability to sleep / snooze. These modes allow the CPU, when idle, to suspend operation and consume less power. The ARM processor family is a good example.

The availability of an RTOS and middleware may dictate the choice. In designing a PDA, one may require middleware that is available for specifically for Linux OS. Further, if the design calls for a graphics system or a file system, then the choice of RTOS will dictate the type of CPU. Many RTOS vendors target specific families, leaving others untouched. Most 8-bit CPUs have simple schedulers that are adequate for small designs, but not suitable for complex system designs. The kind of tools needed for the design, their availability for a standard ANSI C/C++ compiler and design debugging are other considerations.

Modern embedded system design has become all the more challenging for the system architect. He can therefore decide keeping in mind factors like overall design complexity, performance, design reuse, power, size, cost, tools and middleware availability.