

The selection of a microcontroller from the plethora of available controllers is fundamentally a balancing act between several equally important considerations. Peripheral set, cost, performance, and availability are all critical considerations in the selection of a processor, while instruction set can be a secondary consideration. Instruction sets for modern Reduced Instruction Set Computer (RISC) processors are generally within a factor of two performance of each other if not within a few percent on most tasks measured on a per cycle basis. In addition, modern engineering does not use assembly language to any significant extent or on anything but simple projects today. Higher-level languages that are portable across processors, such as C, make the instruction set less and less relevant as a point of engineering choice or attention. The only impediments to switching instruction sets is the fact that tools for a new processor type might cost dollars or engineering and computer time to install and maintain, or might be totally unavailable in the case of a specific brand of Operating System (OS). Fixed function hardware can sometimes compete with processors when a specific solution provides exactly the solution needed. In some cases, hardware might exist to perform the required function without user provided design files or software. This can be in the form of a chip designed to perform that specific function, or Intellectual Property (IP) provided for use in ASIC or FPGA solutions. In some cases, this is just a processor with software provided by the IP or chip vendor; in other cases, hardware is used, and mixtures of both are becoming more and more common. Performance is always an issue when selecting a processor. First and foremost, the processor must execute the code it is required to execute within the design limits required on the module. This can be a bulk measurement of millions of instructions per second, or the time taken in the worst case to handle a specific event. In both cases, faster higher performance processors generally work better but cost more; however, there are limits to the ability of a single processor to respond to events that occur at the same time no matter how high their performance is. Deterministic prioritized operating systems such as OSEK (Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug or Open Systems and the Corresponding Interfaces for Automotive Electronics) help to order software development to guarantee deterministic timing; however, there is really no magic here. If too many things are happening at one time, it is difficult or impossible to guarantee that timing will be met in all cases without failure. One way to eliminate this problem is to use more than one processor. This can be very costly when discrete processors are used; however, in system on a chip solutions in ASIC or FPGA, the cost of additional processors running small tasks from on-chip memory is minimal. Availability and cost are always issues that must be dealt with. In general, the cheapest solution that meets engineering requirements is the one that is chosen, but if engineering requirements are not met, then low price is no advantage. In many cases, solutions that are economical in one set of requirements become impractical when engineering requirements expand. The ability of a solution to grow with time and further requirements is referred to as "headroom" and is of value to engineers who must often add new features. Examples of providing headroom are the following: ● The availability of higher-performance parts that are compatible with lower-cost, lower-performance parts in the same product line. ● For long-lived products, the availability and price of silicon over time. A detailed discussion of standard parts, custom microprocessors, and FPGA solutions from this perspective can be found in the "Examining Processor Obsolescence" section. Matching the input and output required by the

module specification to the peripherals offered in a specific module drives processor selection in many cases. Types of input and output required include some number of general purpose IO (input/outputs), UARTs (Universal Asynchronous Receiver–Transmitter), SPI (Serial Peripheral Interface), I2C (Intelligent Interface Controller), CAN (Controller Area Network), PCI (Peripheral Component Interconnect), memory controllers, event timers, and a wide range of other products. If a processor does not exist with the required peripherals, the customer has a number of options that include going to the manufacturer and requesting a new custom part, using an ASIC, or using peripheral chipsets driven by a generic bus like PCI.