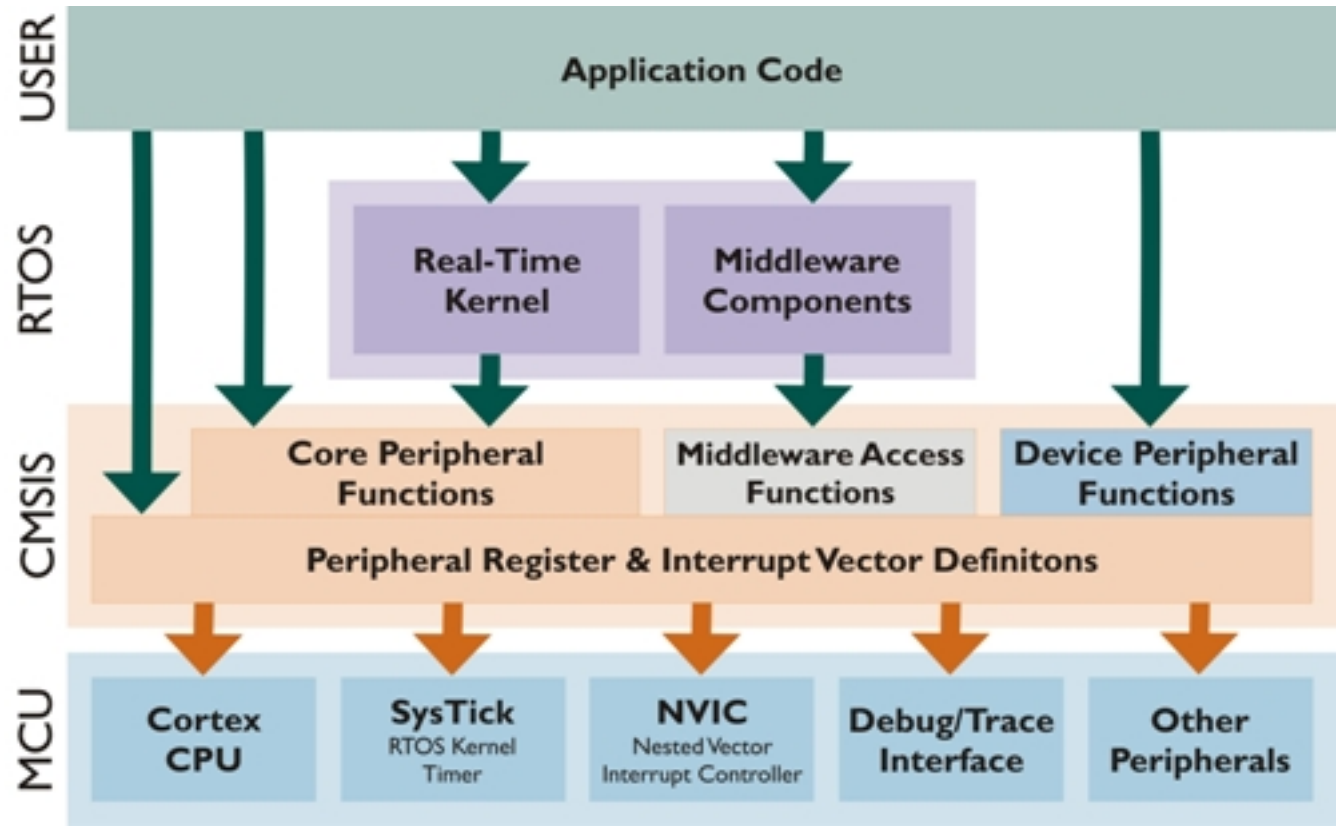
Design Talk: Digital & Analog

Standards for Embedded Microcontrollers

By Reinhard Keil, ARM, www.arm.com [1]

Typically, industries use standards to improve product quality and enable component sharing across projects. In practice, such standards achieve wide acceptance since the synergistic effects provide significant benefits to the user community. The hardware and software industry is full of such standards, but there is an exception: the deeply embedded microcontroller market is still using many proprietary CPU architectures which prevent the introduction of efficient software standards.

Today, the ARM Cortex microcontroller architecture is available as the ARM Cortex-M1 processor (for FPGA) and the ARM Cortex-M3 processor (for silicon implementation). These processors have been widely licensed by more than 25 semiconductor manufacturers and three vendors are currently shipping devices. The Cortex microcontrollers have the attributes of a modern MCU including 32-bit performance, large linear address range, fast interrupt handling, low-power consumption and extensive debug capabilities. Standardizing the software interfaces across all Cortex silicon vendor products has the potential to reduce this cost significantly, especially when creating projects for new devices or migrating existing software to a Cortex processor-based microcontroller from other vendors.



The Cortex Microcontroller Software Interface Standard (CMSIS) enables silicon vendors and middleware providers to create software that can be easily integrated. CMSIS is a vendor-independent hardware abstraction layer (figure 1) for the Cortex-M series of processors that provides a common approach to interface peripherals, real-time operating systems, and middleware components. The standard is scalable to ensure that it is suitable for all Cortex-M series processor microcontrollers from the smallest 8KB device up to devices with sophisticated communication peripherals such as Ethernet or USB-OTG.

Delivering Advanced Residential Gateways

How a Distributed Architecture Supports Efficient Delivery of Advanced Services Residential Gateways

By Ravindra Bhilave, Ikanos Communications, www.ikanos.com [2]

Designing residential gateways (RG) to support multiple services, security and manageability can be a challenge. The various services

require different throughput, latencies and processing power, which puts extreme pressure on the CPU, which can become a bottleneck as the number of service being delivered by an RG increases.

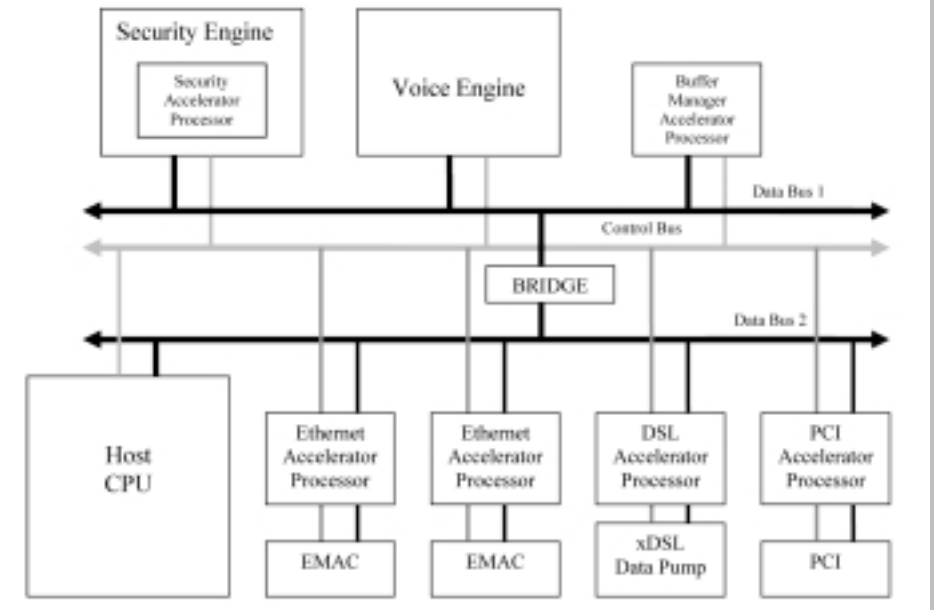


Figure 1: Partial Block Diagram of Fusiv® Architecture

One alternative to a conventional centralized CPU architecture involves using a distributed architecture instead. For example, a distributed architecture based on Fusiv gateway processors uses strategically-located microcode execution units called accelerator processors (APs). The APs efficiently manage the data paths of communications across LAN and WAN interfaces like Ethernet, wireless LAN, xDSL and GPON. APs are also used to offload VPN security and memory management functions.

Such a distributed architecture consists of a host CPU, a dedicated voice engine and several APs, and is designed to efficiently implement the complex requirements of multiple service gateways. By separating the data path from control path the Fusiv architecture eliminates real time inefficiencies associated operating system and applications running on a host processor. This approach also saves a significant amount of host CPU processing power that can be used for providing value-added services. The strategic location of the APs combined with the fact that APs are programmable engines enables the Fusiv distributed architecture to provide hardware-like performance and yet have a software-like flexibility, thus giving it a capacity beyond that of conventional communications processors. These features provide advantages in the RGs, giving them additional processing time for more powerful control and management functions.

Circuit Protection without Degradation

By Phillip Havens, Littelfuse, Inc., www.littelfuse.com **[3]**

 In some ways a circuit protective device is like the bouncer at a bar, charged with keeping out the undesirables but welcoming the paying customers. A good bouncer can tell the difference at a glance and do the job unobtrusively. A bad bouncer may let bad guys in, or be too restrictive and turn away good customers. In the same way, a good protective device will keep high-voltage transients and other disturbances out of a sensitive circuit, yet let the desired signals come in with little or no degradation.

Considerations

If the circuit carries wideband or high frequency signals such as ADSL at 1.1 MHz, VDSL2 up to 30MHz, 100 baseT Ethernet at 100 MHz, or television set-top boxes up to 1 GHz, then select protective device that will not negatively affect the impedance of the circuit. Since overvoltage devices are generally put in shunt, this usually means they must have low capacitance that does not vary. If all the signal frequencies are low —power input lines, for example, or POTS (plain old telephone service), which tops out at 4 KHz— then capacitance is less of a consideration.

DSL and POTS lines, for example, may carry telephone line power up to 56.6 VDC and ringing voltage up to 150 Vrms at 16 to 40 Hz, while also carrying data signals out to 30 MHz. The maximum expected voltage under normal conditions is the peak value of the maximum operating ringing voltage (150 Vrms) plus the maximum DC power (56.6 V) plus the data signal (20 Vpeak), for a worst-case total of about 289 V (considering VDSL signaling levels). Capacitance can also be an issue, because the capacitance of many solid-state protective devices is not only substantial, but varies with varying voltage levels, which can generate all sorts of distortion.

It’s important to understand the nature of the offending event and its current and voltage waveshape. The correct choice for the overvoltage device then depends on the magnitude and duration of the offending event and the interface being protected. If the interface being protected is a power supply line, then a clamping type device should be selected, and for sensitive non-power interfaces, a crowbarring device is usually the better choice. Clamping devices are rated in terms of energy dissipation while crowbarring devices are rated in terms of surge current.

After the event

Some protective devices clamp at a known voltage, while others crowbar: they go to a low-impedance state and stay that way until the available current falls below the holding value of the crowbar device. Clamping and crowbarring devices must be used with overcurrent protection such as fuses or PTCs for the case of power fault

events (contact with a 50 or 60 Hz power line). At the very least, circuit protection shouldn't adversely affect the operation of the circuit. Yet circuit protection goes beyond simply protecting a circuit; it protects your brand, reduces warranty costs, and – when properly applied – it can even improve the function of a circuit.

On Design

By Henry Sommerstorfer

 Think of the worst possible scenario and then make the design. That fore-mostly means: Get rid of all superfluous. Cut to the basic need. Eliminate any fancy ideas. Reduce the weight. Share common fasteners if possible. Reinforce any attachments. Consider heat and cold extremes. Calculate stress and strains. Make the package as small as possible. Try not to cantilever anything. Hide it in small units. However make it easily accessible for replacement or maintenance. Use standard parts as much as possible.

Use the cheapest but effective materials. Securely fasten with standard fasteners. Contain it in a small space and make sure attaching parts, wires, hoses, etc. are accessible and can be secured easily. Make the unit do what it is meant to do. Do not add extraneous options or miscellaneous functions. It needs to be as trouble free as possible. If it is to be enclosed then make the enclosure also easily accessible for maintenance . Again consider the need. Do not complicate it by adding items that could confuse or be installed incorrectly. It may take a little more time, but the results will be worth the effort.

Links:
[1] http://www.arm.com/
[2] http://www.ikanos.com/
[3] http://www.littelfuse.com/