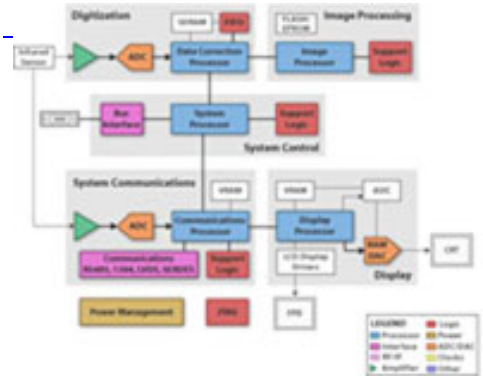
Developing Mission Critical Products with Embedded Processors and Software

Mukesh Kumar, Texas Instruments

The military electronics industry, which includes mission critical products, has always had quality as its ultimate objective. It was a widely held belief that the military's computers, communications systems and all things electronic had to be a step or two ahead of everyone else's systems. It was a matter of national security. As a result, quality was and still remains at the forefront of design considerations.

Quality is very important for military/defense related products, which is why there are a multitude of specifications and guidelines, like the MIL-SPEC, MIL-STD and National Aerospace Standards, for developing high quality products. Today's embedded processors and software can facilitate the creation of high quality and high performance military/defense electronics.



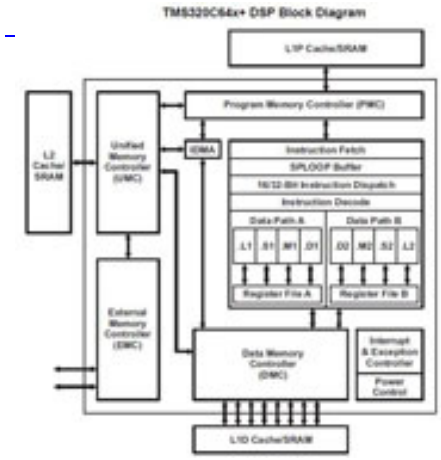
[1]

Over the years, the military electronics and commercial mission critical electronics industries have shifted from designing analog or ASIC based fixed function equipment to integrating software based programmable systems for a better end result. Even though radar systems have been designed since World War II, the modern systems used today do significant processing in software and programmable hardware. Many of the wireless communication equipment are based on implementation of various software defined radios. Other examples of software based products include signal intelligence equipment and control and imaging functions in smart weaponry. Figure 1 highlights the different functions performed on one or more processors.

Functionality

Embedded processors need to provide sufficient performance and features in military/defense electronics. Implementing the desired system functionality is the most important part of achieving quality. In general, this translates into requiring a faster general purpose processor (GPP), digital signal processor (DSP) and/or FPGA for raw processing capability. Additionally, it also determines other aspects, such as availability of on and off chip memory, availability of high bandwidth I/O and connectivity options, as well as hardware acceleration of specific functions.

Even greater performance requires the use of multicore DSPs, the ideal way to implement future products. Many algorithms require high speed and high accuracy computation. Therefore these systems utilize 32 bit and 64 bit processors, are capable of both fixed and floating point math, are based on parallel processing pipelined architectures, have advanced direct memory access (DMA) architectures and possibly contain hardware based inter process/processor communication mechanisms. Figure 2 shown below is a block diagram of Texas Instruments (TI) latest C64x+ fixed Point VLIW (Very Long Instruction Word) DSP architecture.



[2]

Handheld and portable systems, such as a JTRS radio and radar gun, are ideal for military combat use since they are battery operated and easily portable. Designing such systems require the use of ultra low power processors with advanced power islanding and power management capabilities. System on Chip (SoC) solutions for portable devices frequently implements all the necessary peripherals, accelerators and possibly a GPP and a DSP on the same chip.

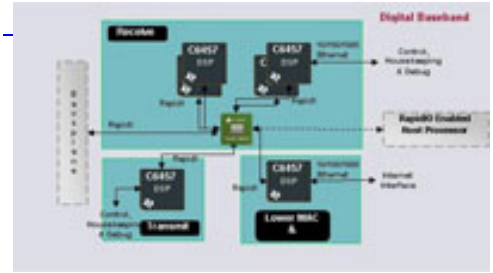
Software enables the development of features which would otherwise have been impossible to design using analog systems. It enables fast and low cost development (agility), ease of modification (flexibility) and reuse for products with varying levels of functionality (scalability and upgradeability). However, software is generally not as stable as hardware and requires greater resources during design, development and testing phases. Maintaining and testing are among other key requirements for software based designs. Features enabling upgrading software in the field, remote monitoring and debugging capability make the software easy to maintain while software with descriptive error messages providing access to the internal data structures during run time and other debugging features makes the

system testable.

Processors have various hardware features like JTAG interface, hardware emulation logic and internal trace buffers to help in writing testable software. There are multiple standards which define software development processes, including the CMM, TQM and ISO. Additionally, other quality standards for specific aspects of software development are set. For example, the “MISRA C” is a C language coding guideline for the automotive industry.

Reliability

Processors utilized in reliable mission critical applications require working under extreme physical conditions, such as operating in temperatures ranging from -55°C to 120°C and bearing electromagnetic radiation. However that is not all, software should also be reliable and maintainable. Various software fault-tolerance and redundancy schemes are used in these products ranging from simple automatic restart of the system upon failure to elaborate software plus hardware redundancy schemes in multi-card systems.



[3]

These systems entail one or more redundant cards to act as a hot backup for other cards and the associated software, which switches to the backup card when a failure is detected. The systems also monitor and react to changes in system condition such as enclosure temperature, functioning of cooling fans and software failures. One common failure point in a software based system is the failure of on and off chip memory like SRAM and FLASH. Modern processors and memory chips have hardware features to detect and correct such error at the same time while the operating system (OS) software has features to overcome failures, like copying data over to a different part of the memory.

Security

Security is the capability of a system to protect itself from unintended tampering, usage or to be rendered out of use. Modern processors have hardware features such as a secure ROM to store bootloader, which will prevent unwanted software to be loaded onto the system, and logic to determine memory tampering. Additionally, some chips have hardware accelerators for implement cryptological algorithms to support secure communications.

Operating systems incorporate various security features to prevent viruses and other malware from interrupting the execution of core software . Software engineers design and implement secure software by using defensive programming skills and the OS features.

Portability

Software portability is a highly desirable feature,. Essentially, portability means the ability to use the same software as the underlying hardware while it goes through revisions -- or using the same application software when the OS is changed. Software portability is often achieved by architecting the software in a layered manner where the main function is abstracted from hardware through hardware abstraction layers (HAL), from OSs through OS abstraction layer (OSAL) and writing OS independent code, such as device drivers and algorithms. Semiconductor companies try to develop successive generations of processor architectures with the aim of maintaining forward and backward compatibility of code. This ensures that high quality software implemented on a processor can be reused on newer platforms.

Reusability of hardware is another desirable characteristic. Since most of the functionality is implemented in software, it is common to expect the same platform to be used for different end products just by changing the software load, like utilizing the same platform for supporting different waveforms in software defined radio (SDR). Processors with integrated interfaces like RapidIO allow connecting multiple chips together while PCIe enables inter card communication. This enables the design of scalable hardware. Figure 3 shows a high level block diagram of a SDR baseband processing card utilizing multiple DSPs.

Usability

“Ease of use“is another important characteristic. The user interface is expected to be simple and self-explanatory. Simple menu-based graphical user interface (GUI) is frequently preferred. Today’s SoCs often have on chip peripherals to support touch screen controllers, audio, video and other interfaces. The OS provides advanced GUI software which enable developing comprehensive user interfaces.

Enabling high quality software-defined products requires implementing many explicit and implicit features in both software and hardware. The proliferation of multicore DSPs and GPPs has increased the software programming complexity and therefore the degree of difficulty in achieving quality as well. Semiconductor companies, like TI, continuously invest in developing new and higher performance multicore processors to enable software developers to write higher quality code. At the same time, the software industry is coming out with various standards and frameworks to make the best use of multicore processors. These advancements in technology are creating new opportunities to develop many more high quality software-based mission critical products for military and commercial markets.

Mukesh Kumar, Multicore DSP Marketing Manager, Texas Instruments.

Source URL (retrieved on *05/21/2013 - 12:59am*):
http://www.wirelessdesignmag.com/articles/2010/08/developing-mission-critical-products-embedded-processors-and-software?qt-most_popular=0

Links:
[1] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/1008/wd1008_TI_Figure1._lrg.jpg
[2] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/1008/wd1008_TI_Figure2_lrg.jpg
[3] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/1008/wd1008_TI_Figure3._lrg.jpg