Leveraging FPGA coprocessors to optimize automotive infotainment and telematics systems

By Paul Ekas

Entertainment electronics are becoming a primary source of differentiation between luxury automobiles, thus driving a rapid evolution of features and capabilities that challenge designers with performance, cost, and flexibility trade-offs. High-end applications can include satellite radio, rear-seat entertainment, navigation, all types of audio playback, voice synthesis and recognition, as well as other new applications.

The core technology drivers for automotive entertainment systems have significant differences from historical automotive applications. Unlike any other area of automotive electronics, these entertainment applications are highly visible and have rapidly changing requirements. In addition, out-of-date entertainment systems will be a major weakness in selling new cars and may become a key factor in resale value and the cost of leasing.

The baseline architectures for in-vehicle entertainment systems are now designed to be capable of supporting a flat screen monitor with a graphical human interface capable of displaying dynamic maps and automobile information. These architectures are centered around a highly standardized micro-controller, a variety of standard interfaces, and simple hardware acceleration to support the required low-end graphics processing. This architecture addresses the requirements for the mid-range entertainment systems of the automotive market at a very low cost, while enabling expansion to high-end applications for the top-tier luxury automotive market. The top-tier applications include video imaging and communications. The various standards supporting these applications (Video: MPEG2, MPEG4, H.264; Communications: GSM/EDGE, WCDMA, 1xEVDO, satellite radio, satellite TV, digital video broadcast, Wi-Fi) are based on multiple, evolving signal processing algorithms. These algorithms demand extremely high-performance processing requirements, flexibility requirements, and cost reduction objectives.

High-end automotive infotainment systems integrating data communications, location services, and video entertainment require high-performance programmable processing that can be optimally delivered by integrating FPGA coprocessors into mainstream automotive telematic system architectures. This article describes the requirements for an automotive entertainment system, discusses a mainstream system architecture, and will identify how an FPGA coprocessor can be integrated into both the hardware and software architecture to address high-performance processing requirements, flexibility requirements, and cost reduction objectives.
high programmable processing performance. There are three semiconductor technologies that can be utilized to implement these highly complex algorithms.

These three technologies include programmable Digital Signal Processors (DSPs), Application Specific Standard Products (ASSPs), and Field Programmable Gate Arrays (FPGAs). The DSP provides a high-performance programmable processor specifically designed for signal processing applications. DSP processors are extremely flexible, low-power, and cost efficient, but lack hardware acceleration capabilities and do not provide the necessary compute power for today’s leading image processing and wireless communications algorithms. ASSPs, which often include DSP processors, provide optimized solutions for single video or communications standards, but cannot be programmed to adapt to different standards. FPGAs, on the other hand, provide both very high performance processing and are programmable for adaptation to many applications and standards. Unlike the other two technologies, FPGAs deliver both the flexibility and performance required to cover all the potential algorithms.

The baseline telematics architecture previously described requires additional processing chips to handle the high-end applications. These additional chips, generally ASICs and ASSPs integrate with the processor through a memory or video processing bus and thus become application-specific coprocessors. One of the powerful ways to use an FPGA is as a replacement for this application-specific hardware. Coupling the FPGA to a processor is known as FPGA coprocessing. Using the FPGA in this way enables new application-specific accelerators to be downloaded on demand into the FPGA to assist in any high performance application. This concept is being widely used for advanced military multi-standard radios and is called Software Defined Radio (SDR). In SDR, a single radio unit can automatically adapt to different radio standards with a push of a button. This not only helps future-proof equipment, it also reduces the number of custom processors that sit idle by while a different task is being performed. These soft radio techniques can be utilized in the automotive market for both communications and video applications.

The flexibility an FPGA provides for video processing and wireless connectivity can also save costs and increase system value. Today’s baseline architectures require add-on ASSPs for each new video codec or wireless standard supported. Substituting one FPGA to replace multiple ASSPs reduces the number of architecture permutations that must be deployed and maintained over the life of a vehicle. Extending the baseline in-vehicle entertainment system architecture to include an FPGA enables a single high-end platform that can be programmed across a wide range of video and wireless standards and features. This approach fits in well with advanced automotive entertainment system architectures.

An example of a leading-edge automotive entertainment system architecture1 has been published by Delphi Delco Electronics Systems corporation. The platform leverages a standard SH-4 microprocessor and a companion ASIC, the Hitachi HD64404 Amanda peripheral, to deliver the baseline functionality required for the middle 80 percent of the automotive market. This system provides a common control processor with a standard API layer that enables the abstraction of hardware peripherals and coprocessors. The companion ASIC provides a baseline set of peripherals and an integrated graphics processor. The graphics processor is capable of supporting interactive graphics and scaling functions, but does not provide video codec functionality or other DSP applications. This system provides baseline functionality for all entertainment applications, but still requires additional ASICs or ASSPs for video codec and wireless communications functionality.
The Amanda companion chip in the Delphi architecture (see Figures 1, 2) uses two processing buses, the Pixel Bus for high-performance dataflows such as video processing, and the Register Bus for control applications. Each bus connects to the SH-4 MPX bus and an external memory interface. This combination of buses and memory interfaces provide the perfect interface to support a flexible video codec and wireless communications platform based on an FPGA coprocessor.

In addition, FPGA coprocessing, applied to image processing, can enable support of multiple video codecs including MPEG2, MPEG4, and H.264 with a single FPGA. In fact, it can utilize the same FPGA as that used for the wireless communications.

An FPGA coprocessor integrates with a processor based system through Direct Memory Access (DMA)-based interfaces. The software layer on the embedded processor includes an application interface for each coprocessor that includes an initialization routine that loads the FPGA with the appropriate application coprocessor. Once the application is initialized, software calls to the coprocessor control parameters, timing, and the flow of data into and out of the coprocessor. Depending on the standard being implemented, there may be a high level of interaction between the FPGA coprocessor and the control processor, or the FPGA coprocessor may be completely self-contained. In such cases the control processor simply loads the algorithm and gets out of the way.

Each program image loaded onto the FPGA needs to integrate into the surrounding system. Using an FPGA for programmable functions requires a well defined system interface that each FPGA-based accelerator relies on for communication. In general, the FPGA will have multiple interfaces that connect to a control processor, memory, and other external peripherals or connectors. The FPGA may also contain several coprocessors simultaneously that all share a single interface to the control processor. Each peripheral or coprocessor can have additional buses for high-performance dataflow processing.

In the case of a video codec, there will be an input source and an output destination. The video input interface in the Delphi system architecture is part of the Amanda companion ASIC and utilizes the ITU-R BT.656 interface for streaming video. This can be post-scaled and manipulated by the ASIC to fit a variety of different display panels. The FPGA will likely need to connect to two other buses, the memory bus on the companion chip and the PCI/MPX bus of the host control processor, which also connects to the companion chip. Through these three connections, the FPGA can support video and communications applications with high-bandwidth communication through the memory interface and control communication through the PCI/MPX bus.

The FPGA provides a reprogrammable platform for application-specific processing architectures that complements the host processor. The FPGA program, however, is fundamentally different from that of a standard processor architecture. An FPGA provides a high-performance hardware fabric with programmable logic elements, routing, DSP processing blocks, memory, and I/O. The system architecture of an FPGA is executed in much the same manner as that for a standard ASSP where the dedicated...
functionality of the system is designed and implemented through hardware and software development tools. The output of these tools is a binary image that when loaded onto the FPGA, defines the functionality of all the programmable logic elements, routing, DSP processing blocks, etc. This binary image can be loaded by the host processor during the run-time of the system. A variety of different program images can be created to support MPEG2, MPEG4, H.264, GSM/EDGE, WCDMA, 1xEVDO, GPS, 3D graphics accelerators, or any other algorithms that may go into an automotive telematics system. Depending on the user’s menu selection in the entertainment system, the specific application program will be downloaded by the host processor into the FPGA and will then be under the host processor control.

"Using an FPGA for programmable functions requires a well defined system interface that each FPGA-based accelerator relies on for communication."

Controlling a dedicated hardware accelerator from a host processor is typically done through a register and memory interface with each register controlling some aspect of the hardware accelerator operation. This is true for the default companion chip in the Delphi system and will be true for each coprocessor architecture downloaded into a companion FPGA. Using the FPGA, it is a straightforward task to standardize on a register and memory interface to control any coprocessor that is programmed into the device. This standard interface may define how to read and write data to the coprocessor, how to start and stop it, how to reset it, and include a set of registers for controlling application specific operation. All of these registers will be part of a linear address map within the FPGA so that it is easy for the software physical device drivers to access the registers.

The software physical device drivers for a coprocessor provide a higher level of abstraction than the register interface implemented in the hardware. The software drivers provide a mapping from algorithmic parameters of the system to the control registers so that the application software is easier to write and maintain. The higher layer model device drivers remain portable across changes in implementation of the underlying hardware. The software architecture within the Delphi system provides a strong framework for supporting algorithm implementation either in software or hardware coprocessors, since it provides a couple of layers of abstraction that separates the algorithmic implementation from its physical implementation in hardware or software. FPGA coprocessors fit very nicely into the Delphi software and hardware architecture.

FPGAs are being designed into many systems whose basic architecture is similar to that of the Delphi system architecture. These systems include one or more control or DSP processors and utilize the FPGA to accelerate tasks that require high-performance processing. The key challenges in implementing FPGA coprocessors include designing the various hardware accelerators for an FPGA, integrating the hardware accelerator with the external control processor, and creating a software layer that controls the hardware accelerator. The hardware accelerators required include mainstream algorithms for video and communications. These applications have a broad market, which has evolved to support specialty companies that focus on designing standard specific Intellectual Property (IP) hardware accelerators. These companies provide off-the-shelf algorithms that can be directly implemented in leading, low-cost FPGAs. It is possible to buy commercially available IP blocks for MPEG2, MPEG4, H.264, Wi-Fi, and many other video and communications standards. An example MPEG4 decoder IP block diagram from Amphion Corporation is shown in Figure 3 that is available for ASIC or FPGA applications.

The next step is integrating the hardware accelerator in the FPGA with the external busses for control, data input, and data output. A new category of development tool is available that enable designers to easily perform this integration. Using SOPC Builder, the system integration tool from Altera, designers select the IP blocks from a list of available IP. Upon selection, a parameterized menu appears showing different architectural options the user has control over prior to implementation. Once the parameters are set, the block is included in a list of other peripherals and processors being integrated by the engineer. Once each individual IP block has been selected and parameterized, they need to be integrated into an processing architecture.

SOPC Builder enables the designer to define a high-performance switch architecture that interconnects the various hardware accelerators and peripherals to the external host processor. This switch architecture is defined through mouse clicking on an intuitive matrix representation of the block interconnection. Once the architecture is defined, SOPC Builder automatically assemblesthe various IPs together and generates a Hardware Description Language description for automatic synthesis to the final FPGA program. This final program is then downloaded onto the FPGA at run-time to implement a specific algorithm coprocessor.

After hardware integration is complete, a software physical device driver is required that separates high-level software control from the detailed register and memory map architecture used to control the hardware accelerator. The register and memory fields required to control a hardware accelerator are standard components of the parameterized IP blocks. The integration of multiple peripherals...
and accelerators, however, requires a register and memory map of all programmable features implemented onto the FPGA. SOPC Builder automatically creates this register and memory map while it assembles the IP into the user defined switch architecture.

Each IP block contains a predefined set of software physical device drivers for use on an external host processor to control the IP block. SOPC Builder automatically assembles the various software physical device drivers and automatically associates each driver with the appropriate register and memory map associated with the IP block it controls. In this way, SOPC Builder automatically creates and integrates the hardware and software architecture of the FPGA coprocessor and the control processor. SOPC Builder was developed to address the rapidly evolving capabilities of FPGAs and their increasing ability to absorb complex system implementation.

Since their introduction 20 years ago, programmable logic devices have been rapidly evolving from low-level glue logic to the lowest cost, highest performance programmable processing. Two key factors have driven this advance in FPGA performance and reduction in cost: FPGA architecture evolution and the ways in which FPGAs leverage semiconductor technology. The architecture of FPGAs provide arrays of programmable logic elements that are grouped together with programmable routing. In early low-density FPGAs, this enabled the interconnection of simple processing elements. As the density of FPGAs increased, the arrayed architecture enabled massive parallel processing. FPGA architectures have evolved to include memory blocks, DSP blocks, and programmable I/O throughout the processing array. The resulting processing architectures can easily meet the performance requirements of automotive telematics.

The other key driver of FPGA evolution is process technology and its impact on performance and cost. FPGAs utilize the latest generations of process technology to increase density, performance, and lower costs. At the same time, FPGAs are used to accelerate the development of process technologies. FPGAs are valuable in the development of semiconductor process technology because they utilize a regular structure that goes to high volume production early in its lifecycle. The regular structure of FPGAs enables the collection of good statistics for measuring production defects, which is critical for fine-tuning process technology to achieve ultra-high manufacturing yields. The symbiotic relationship between FPGAs and process technology has enabled a huge increase in FPGA density and a related decrease in component price. As a result, today’s low-cost FPGAs such as Altera’s Cyclone series of devices are cost competitive with dedicated ASICs and ASSPs.

Automotive entertainment systems are rapidly evolving, both technologically and as a point of differentiation among automobiles. Leading-edge system architectures are designed to serve the majority of the mainstream automotive market, while enabling high-end differentiation through additional supporting ASSPs and software. FPGAs provide a complementary high-performance and flexible coprocessing platform that consolidates the functionality of many companion ASSPs into one reprogrammable platform. FPGA coprocessors fit nicely into mainstream automotive entertainment architectures like the Delphi architecture described. By using FPGA coprocessors as part of high-end automotive entertainment system architectures, automobile companies can provide a multitude of high-end video

and communication features through software programming that cannot be enabled by ASSPs alone. Using FPGAs to facilitate high-end flexible automotive entertainment architectures can enable the ability to upsell new features at the time of the vehicle sale and throughout the lifetime of the vehicle. The ability to enhance an automotive entertainment system, at the time of sale and thereafter, can increase the value of the car when it is initially sold, as well as later, when the resale value of a previously leased vehicle remains an important component in auto manufacturer’s profitability.

Reference:
1 “Mobile Multi Media Open Computing Platform” by Phil Motz, Arch Wills, Jill Hersberger, Mike Laur, Delphi Delco Electronics Systems Corp. – SAE 2001 World Congress March 5-8, 2001

Paul Ekas joined Altera in August 2002 as the Senior DSP Marketing Manager in charge of the company’s Code:DSP corporate marketing initiative. Ekas has more than 17 years of business experience in electronic design automation and complex semiconductor systems. He holds a BSEE and MSEE from the University of Washington.

For further information, contact Paul at:

Altera Corporation
101 Innovation Drive • San Jose, CA 95134
Tel: 408-544-8388 • Fax: 408-544-7280
E-mail: pekas@altera.com • Website: www.altera.com

Reprinted from Embedded Computing Design / Spring 2004