

# Using Model-Based Design to Accelerate FPGA Development for Automotive Applications

**Sudhir Sharma and Wang Chen**

The MathWorks

Copyright © 2009 The MathWorks, Inc. Published by SAE International with permission.

## ABSTRACT

A recent Gartner Dataquest study predicts that the total worldwide automotive semiconductor market will grow from \$20.1 billion in 2007 to \$25.9 billion by 2010. The study also predicts that revenue from automotive usage of FPGAs will triple to approximately \$312 million during that same period<sup>[1]</sup>.

Many of these FPGAs will be deployed in safety applications such as back-up cameras, lane departure warning systems, blind-spot warning system, and adaptive cruise control. FPGAs will also be deployed in next-generation engine electronics, emissions control, navigation, and entertainment applications.

Automotive systems engineers are adept at using Model-Based Design for implementing some of these embedded applications on DSPs and microcontrollers. Many of these engineers are new to FPGA design and waking up to a fragmented workflow that is making it harder to meet time-to-market and cost objectives.

For example, engineers who are migrating their systems designs from DSPs to FPGAs are discovering that additional verification steps such as bit-true, cycle-accurate simulations are required to ensure that the FPGA functions the same as the system specification. This is a time-consuming and error-prone activity involving file exchanges between the system designer and the FPGA designer. Geographically distributed teams face an even bigger challenge since the system

engineer and FPGA designer may be many miles away from each other.

Common applications for FPGAs in automotive industry include:

- Prototyping in high-volume applications such as engine and steering control, where the final production deployment will be an ASIC. In this workflow, the proof of concept work is done using FPGAs.
- Production deployment to low-volume high-processing power applications such as driver-assistance and infotainment systems.

This paper illustrates how Model-Based Design integrates the world of system designers, FPGA designers, and verification engineers to increase productivity and produce correct-by-construction designs that match the system specification. Using the concept of executable design specification, this paper discusses how Model-Based Design streamlines both design and verification of FPGAs for an engine control application.

## INTRODUCTION

Many new cars now include electronic safety systems such as collision avoidance, adaptive cruise control, and lane-departure system. These new features often require fast processing power and large memory to handle video data streaming, image recognition, or other signal processing, which greatly increase the hardware and software complexity in automotive electronic systems.

Many of these complex functions can not be implemented in software running on traditional DSP and microcontrollers. Instead, automotive hardware designers are now focusing on FPGAs and ASICs.

Compared to traditional DSPs and microcontrollers (MCUs), FPGAs and ASICs offer faster processing speed and more functionality to support more advanced features. Choosing between an ASIC or an FPGA implementation depends on the application and is beyond the scope of this paper, but, broadly speaking, an FPGA implementation can be a faster time-to-market and lower-cost solution than an ASIC design. FPGAs also offer the added benefit of reconfigurability when the design specification changes. On the other hand, an ASIC may be the right solution for a large volume, very high-speed, or power-sensitive application.

FPGA usage is growing rapidly because it satisfies the automotive industry's demands for faster processing speed, higher logic density, shorter time-to-market cycle, and reconfigurability<sup>[2]</sup>.

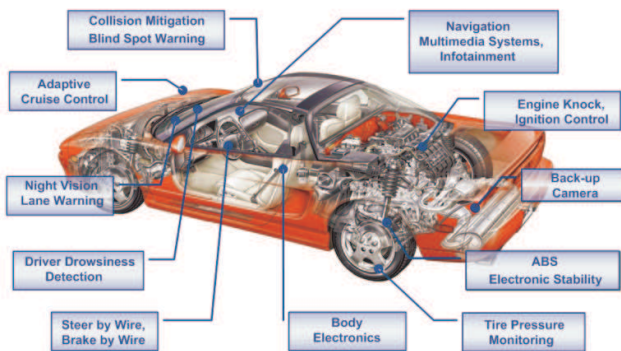


**Figure 1. FPGA applications in the automotive industry.**

Figure 1 lists a few of the applications where FPGAs may be deployed. These include active safety system, powertrain, chassis, body electronics, and infotainment systems.

In Table 1, we capture five major automotive applications and their suitability for FPGA-based implementation.

Active-safety systems and infotainment systems are the most popular areas for FPGA applications. These applications need to process large amounts of streaming input data and provide responses in real time. Their image processing and radar signal processing algorithms may easily consume the processing power of an entire DSP processor. In such applications, instead of using a fast, stand-alone DSP, the adoption of an FPGA as a hardware coprocessor can offer a more compelling solution that allows the designer to use a smaller DSP. This can result in greater system performance at a lower cost<sup>[3,4]</sup>.

FPGAs are also found in powertrain systems as a coprocessor for engine controller tasks such as knock detection and injector control. While ASICs are often used in high speed applications such as gasoline direct injection, diesel multiple injection, and electronic valve lifting, newly introduced flash-based FPGAs may be an attractive option for these applications<sup>[1,5]</sup>.

As of two years ago, premium vehicles such as BMW 5 Series employed well over 100 MCUs to control the various systems in the car. Automotive engineers have been working to consolidate the functions of multiple MCUs into one FPGA to reduce the system complexity<sup>[6]</sup>. Even as the unit costs of both DSP and FPGA devices are falling, the growing system complexity and increasing need for raw processing power make FPGAs a compelling solution for many automotive applications.

For example, Saab's trionic engine combustion control processing is straining the abilities of MCUs<sup>[7]</sup>. Conceivably, the engine can burn hydrocarbons and CO in polluted air that is sucked into the intake<sup>[8]</sup>. The ECU uses a Motorola 683xx 32-bit processor, 4Mbit of memory, an 8-bit coprocessor, barometric sensor, and other components<sup>[9]</sup>.

By using the spark plug as an ionization probe during each idle cycle in the four-cycle process and sensing temperature, pressure, and unburned exhaust, everything can be adjusted to its optimum to wring out every bit of power available in the gasoline, based on real-time combustion results and driving conditions.

Taking this technology farther will require more than a faster embedded processor if it is to be done efficiently for economy autos as well as high-end luxury vehicles.

| Applications                 | Examples                                                                                                                                                                | Key Requirements                                        | Present Technology                                                                                      | FPGA Viability                                                                                                                                    |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Active Safety Systems</b> | Adaptive cruise control<br>Collision mitigation<br>Lane departure warning<br>Back-up camera<br>Blind spot warning.                                                      | Throughput<br>Reconfigurability                         | DSP and ASIC/FPGA<br><br>New systems have more FPGAs in them.                                           | Yes.<br>As a coprocessor for video and image processing.<br>For high processing speed and reconfigurability.                                      |
| <b>Powertrain</b>            | Engine control module:<br>– Fuel injection<br>– Knock detection<br>– Ignition timing<br>Power electronics control<br>Transmission control<br>Vehicle energy management. | Cost/Memory<br>Speed<br>Complexity<br>Reconfigurability | MCU (high end)<br><br>ASICs are used today extensively in engine control and power electronics control. | Yes.<br>As a co-processor for tasks such as knock detection, fuel injector control, and power electronics control.<br><br>For HW and SW codesign. |
| <b>Chassis</b>               | Steering control<br>ABS<br>Electronic stability (ESP)<br>Ride control<br>Brake-by-wire                                                                                  | Cost/Memory<br>Speed<br>Redundancy                      | MCU (mid end)<br><br>DSPs are sometimes used for steering control.                                      | Yes.<br>Possible for power steering control, which requires high processing speed.                                                                |
| <b>Body Electronics</b>      | Roof, window, mirror, seat<br>Climate control<br>Instrument cluster<br>Central body control<br>Network gateway.                                                         | Cost/Memory<br>Low Power<br>Reconfigurability           | MCU (low end)                                                                                           | Yes.<br>To consolidate the functions of multiple MCUs that are distributed around the vehicle into one FPGA [3].                                  |
| <b>Infotainment Systems</b>  | Navigation<br>Voice recognition<br>Digital audio/TV<br><br>Rear-seat entertainment.                                                                                     | Throughput<br>Reconfigurability                         | MCU and DSP                                                                                             | Yes.<br>For high processing speed and reconfigurability.                                                                                          |

**Table 1. Factors to consider when choosing to deploy FPGAs in automotive applications.**

## MODEL-BASED DESIGN

Model-Based Design improves design quality and accelerates design and verification tasks by employing an executable specification. This executable specification is elaborated to create hardware and software partitioning, automatically create hardware and software implementation code, and verify the hardware and software implementations in the context of the complete system (as shown in Figure 2). Significant advantages of Model-Based Design include the fact that it facilitates rapid design iterations and it moves the verification process all the way to the beginning of the design cycle. This helps detect system specification related errors, design errors, and implementation errors early.

## FPGA SYSTEM DESIGN CHALLENGE

As stated in the introduction, automotive systems engineers are adept at using Model-Based Design for implementing embedded applications on microcontrollers and DSPs. They are typically new to FPGA design. Hardware design workflow is significantly different from software design workflow and the transition can be difficult. The following sections outline

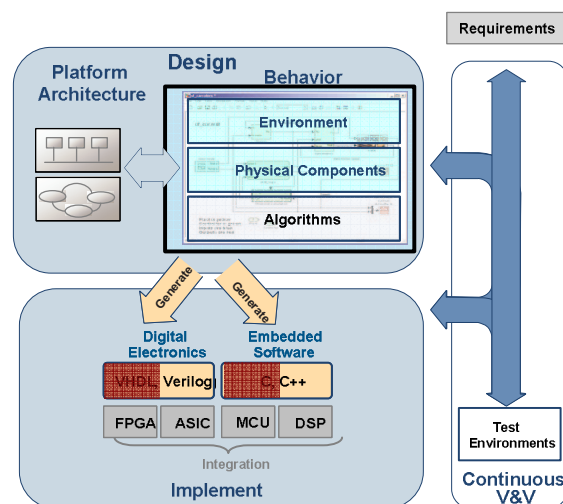


**Figure 2. Model-Based Design that uses an executable specification and allows continuous system-level verification.**

some of the typical problems with the manual FPGA design workflow and propose an integrated system-level design approach to FPGA development.