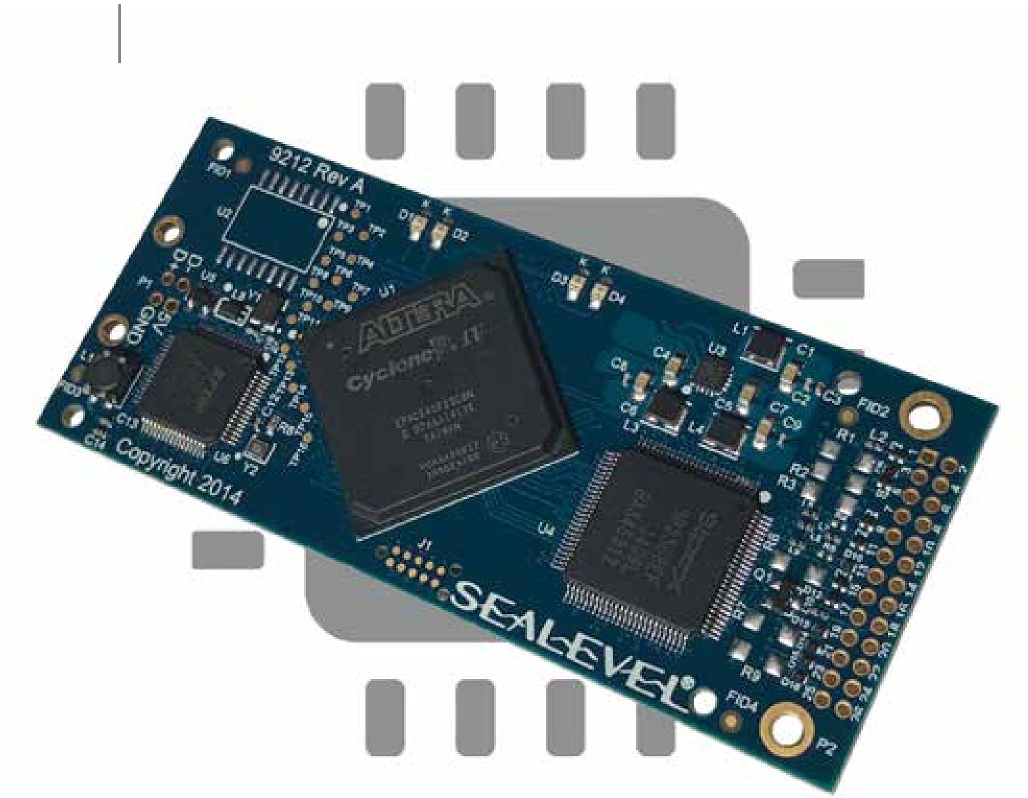
Field-Programmable Gate Arrays:
A Fresh Look at a Classic Embedded Design Tool

Leah VanSyckel and Greg Harrison

Introduction



In “Using FPGAs to Improve Embedded Design,” Sealevel Engineer Greg Harrison explains FPGAs on a technical level and illustrates their use cases with Sealevel products. He shows FPGAs to be useful in interface conversions, COM Express simplification and optimizing embedded systems. In this white paper, we will discuss the basic definitions and the benefits of using FPGAs as well as their future in smart technology.

What is an FPGA?

A Field Programmable Gate Array (FPGA) is an integrated circuit (IC), or chip, used in embedded system design. There are four main FPGA producers: Xilinx, Intel Programmable Solutions Group (Intel PSG, formerly Altera), Lattice Semiconductor and Microsemi, a Microchip company (formerly Actel). The primary characteristic of an FPGA is that it is programmable after manufacturing, which gives it versatility in application use and customization. These architectures are built and programmed by embedded hardware engineers, such as Sealevel's Greg Harrison. They require a fundamental understanding of computing hardware. Engineers design with FPGAs using hardware description languages (HDL), including VHDL and Verilog.

These devices contain architectures of interconnects and logical elements that are highly configurable. These elements, when in a completed, programmed device, can accept an input and produce a logical output based on the programmed configuration. These activities happen at high speeds due to parallel, independent logic. Their programmable, flexible routing and low-power demands make them ideal for telecommunications, networking, and signal/data processing.

FPGAs can be an alternative to Application Specific Integrated Circuit (ASIC) chip designs. ASICs have a permanent hardware configuration. Whereas ASIC designs have only one identity and function, an FPGA has resources that can change functionality for different applications. ASICs are found in almost every mainstream electronic computing device, such as televisions, cell phones and other consumer goods.

ASICs are extremely efficient, but they are expensive to design and build; moreover, their construction process is lengthy. They are also mostly profitable when dealing with units in the hundreds of thousands — like mainstream goods — rather than the ones, tens and hundreds of units. However, until FPGAs emerged as a feasible alternative solution, ASICs were the default chips on electronic devices.

When FPGAs advanced in their development with lowered power and increased density, they became popular for use in small-unit production. More density on an FPGA means that a higher number of ASIC-type functions can occur. Apart from being suited to different tasks and volumes, there are few benefits between choosing one or the other. From a design perspective, both require extensive knowledge of hardware and have similar levels of reliability and durability.

FPGAs are not the same thing as microprocessors or microcontrollers. A microprocessor is like a small CPU that executes specific instructions and has a fixed set of interfaces/peripherals made possible by the logic blocks designed into it. A microprocessor can

What is an FPGA?

be partnered with an FPGA, or even hardwired into its chip architecture, but it is not the FPGA. The FPGA would carry out certain, unique tasks, and the microprocessor would do general tasks. A microprocessor hardwired into the chip is called a hardened processor. For example, Intel PSG and Xilinx sell FPGAs with ARM processors hardened into the device. An FPGA with a processor is called a System-on-a-Chip (SOC). However, SOCs are not always this system combination. For example, the popular Raspberry Pi is an SOC comprised of an integrated CPU, GPU, DSP, and SDRAM.

Microcontrollers are chips that have a fixed instruction set and interfaces/peripherals. Microcontrollers are typically coded in a high-level language like C, C++. Microcontrollers require less power than FPGAs and can be purchased off the shelf. Arduino Kits are microcontroller kits used in the Maker movement.

When do you use an FPGA?

FPGAs can be useful in most electrical engineering designs that require custom interfaces, custom logic, or high signal/data processing speeds. They have a faster time to market than custom ASICs; moreover, they avoid upfront non-recurring expenses (NREs). FPGAs have a predictable project cycle and field reprogramability due to remote bitstream upload capabilities, for changes even after product launch if needed. However, they are not for everyone and everything, as they do have limitations.

Generically, FPGAs are useful in many kinds of projects. Designs that require customization and result in smaller volume are prime opportunities to use this technology. FPGA use gives customers more freedom to choose specific interfaces and build products that suit their application perfectly. Due to FPGA intelligence, they also run hardware-level tasks more efficiently and at higher speeds, which lets custom software programs control higher level functions. Ultimately, they also allow custom and standard products alike to guard against part obsolescence.

Sealevel uses FPGAs in some applications. One of them is a 1U, 19" rackmount COM Express system. It has 256 I/O points and 40 channels of A/D. This device often needs to rapidly poll the I/O to receive updated information. Without an FPGA, this process requires processor overhead, which slows down the machine's tasks and interface. However, with the FPGA, the COM Express processor simply reads the FPGA's memory buffer, which stores the continuously polled I/O, speeding up the task. The benefit of this on-board intelligence is diminished risk — by avoiding slow computation — and rapid operation, which leads to fewer errors.

Perhaps one of the most famous examples of FPGA use is the Mars Exploration Rover project, which began mission objectives in 2004. Actel used 28 radiation-tolerant and hardened FPGAs in various systems on the two Rovers for several functions. Their primary benefits were reliability at extreme temperatures, reduced component cost, minimized power consumption and diminished board space and weight. These FPGAs could also be reprogrammed remotely as needed to conduct different activities.

What's the future of FPGA?

One of the big issues with FPGAs is the depth and breadth of hardware expertise required for effective design and use. This knowledge is unlike some design tools where the coding can be built into easy abstracted tasks and commands. An engineer needs to know the HDL, know the components and get into the hardware to make the design decisions. The complicating factor to this knowledge is how dated it is. Unlike other innovative forms of processing units, like Graphical Processing Units (GPUs), FPGAs have remained mostly static in their low-level development methodology. The tools and processes to develop and utilize FPGAs are not as high-level and abstracted as those of other processing platforms. These concerns can present boundaries for expanded, mainstream use.

However, they are attractive tools for data processing services, such as machine learning and IoT. This increasing appeal is directly related to the developments being made to put FPGA design on the cloud. When it comes to machine learning and other Artificial Intelligence, FPGAs are attractive for simplifying calculations, while maintaining sufficient accuracy, and greatly increasing data processing speeds. This does require a low-precision data type though. In instances where high-precision accuracy is needed, the alternative GPU technology is a worthwhile alternative. Nevertheless, FPGAs do provide a better ratio of performance to consumption, regarding computation speed and energy use.

An example of this initiative is the Amazon AWS EC2 F1 FPGA instances of cloud computing. According to Electronic Engineering Journal's editor's blog on the EC2 F1 by Steven Leibsen, "people have been using this cloud-based hardware acceleration capability to speed up the execution of diverse tasks."

These diverse tasks have included convolutional neural networks (CNNs), which are essential to the inference stage of AI machine deep learning. Inference is when the machine takes what it has been trained to do and practices it. Other projects using this FPGA instance are genome sequencing and video transcoding. Experts believe high-frequency equity trading will eventually be used on these types of cloud hardware acceleration applications.

The biggest point going forward with these cloud-based FPGA services is their supposed ease of use for non-specialist engineers. The creators of these tools are trying to remove

What's the future of FPGA?

the middleman, embedded engineers, from the FPGA design and implementation process. However, the process is almost more difficult because it still requires extensive hardware knowledge without access to a physical board. Nonetheless, these cloud-based FPGA platforms may become more readily-accessible for a wider engineering base as FPGA design tools and methodologies continue to strive for higher levels of abstraction.