Fundamentals of FPGAs—Part 5: Getting Started with Intel (Altera) FPGAs

Clive "Max" Maxfield ⋮⋮ 7/5/2020

By Clive "Max" Maxfield

Contributed By DigiKey's North American Editors

2020-05-07

Editor's Note: Optimal processing solutions are often provided by combinations of RISC, CISC, graphics processors, and FPGAs; FPGAs on their own; or by FPGAs that boast hard processor cores as part of their fabric. However, many designers are unfamiliar with the capabilities of FPGAs, how they've evolved, and how to use them. Part 1 of this multi-part series provides a high-level introduction to FPGAs; Part 2 focused on FPGA offerings from Lattice Semiconductor; and Part 3 concentrated on the FPGA device families and design tools from Microchip Technology (from its subsidiary Microsemi Corporation). Part 4 took a deep dive into the components and tools from Xilinx. Here, Part 5 considers the offerings from Altera, which was acquired by Intel and now forms the Programmable Solutions Group at Intel (for the remainder of this paper, the name Intel is used).

As discussed in Part 1, field-programmable gate arrays (FPGAs) have many characteristics that make them an invaluable computing asset, either used standalone or in a heterogeneous architecture; but many designers are unfamiliar with FPGAs and how to go about incorporating these devices into their designs. One way to overcome this impediment is to look more deeply at FPGA architectures and associated tools from major vendors.

High-level FPGA options overview

There are many different types of FPGAs on the market, each with different combinations of capabilities and functions. At the heart of any FPGA is its programmable fabric. This is presented as an array of programmable logic blocks, also known as logic elements (LEs) (Figure 1(a)). The next step up in the FPGA fabric is to include things like blocks of SRAM, called block RAM (BRAM), phase lock loops (PLLs), and clock managers (Figure 1(b)). Digital signal processing (DSP) blocks (called DSP slices) and high-speed serializer/deserializer (SERDES) blocks can also be added (Figure 1(c)).

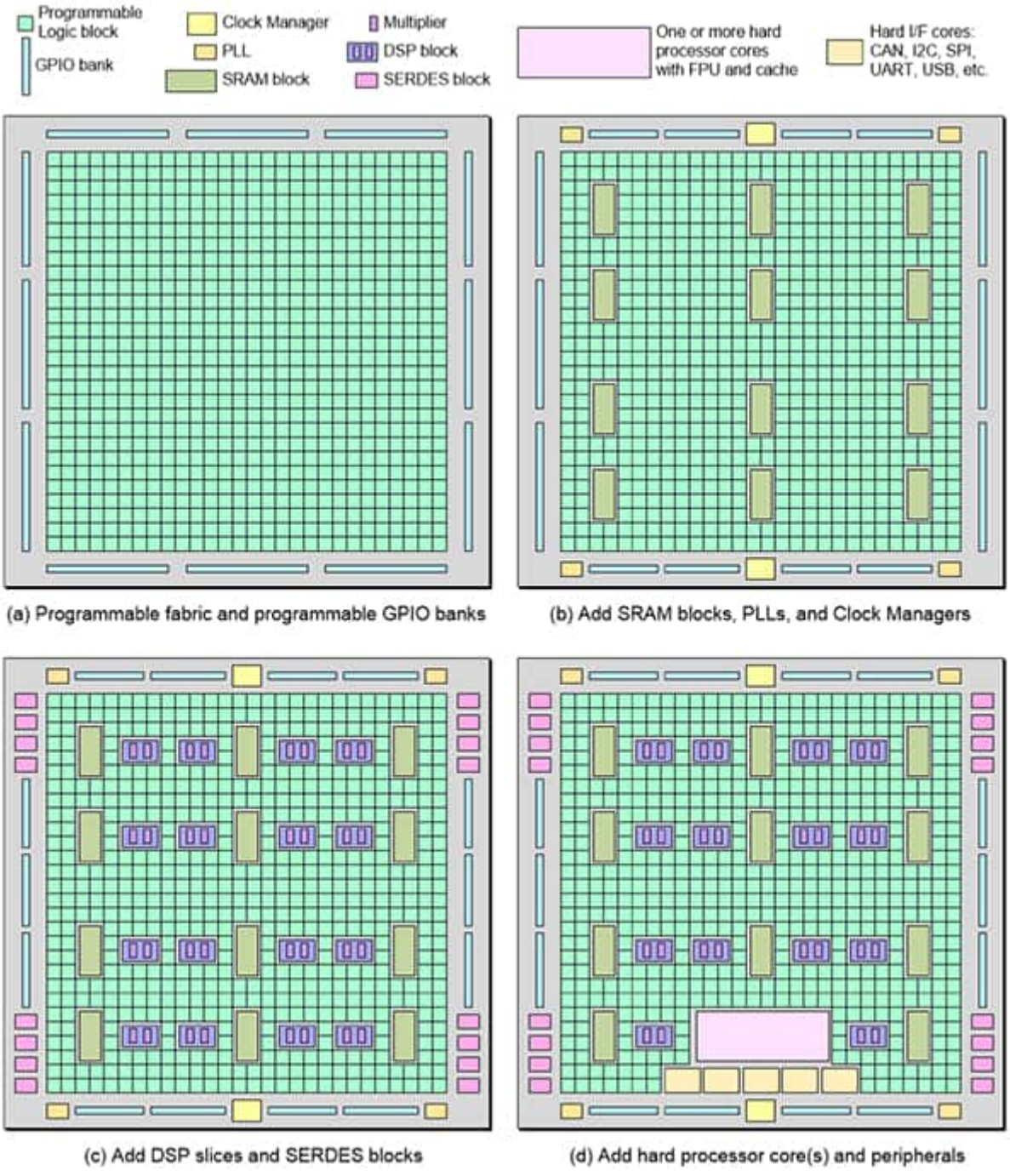

Figure 1: The simplest FPGAs contain only programmable fabric and configurable general purpose IO (GPIO) (a); different architectures augment this fundamental fabric with SRAM blocks, PLLs, and clock managers (b); DSP blocks and SERDES interfaces (c); and hard processor cores and peripherals (d). (Image source: Max Maxfield)

Peripheral interface functions like CAN, I^2C, SPI, UART, and USB can be implemented as soft cores in the programmable fabric, but many FPGAs include them as hard cores in the silicon. Similarly, microprocessors can be implemented as soft cores in the programmable fabric or as hard cores in the silicon (Figure 1(d)). FPGAs with hard processor cores are referred to as system-on-chip (SoC) FPGAs. Different FPGAs offer different collections of functions, features, capabilities, and capacities targeted at different markets and applications.

There are a number of FPGA vendors, including Intel (which acquired Altera), Efinix, Lattice Semiconductor, Microchip Technology (which acquired Atmel and Microsemi), and Xilinx.

All of these vendors field multiple families of FPGAs: some offer SoC FPGAs; some offer devices targeted at artificial intelligence (AI) and machine learning (ML) applications; and some offer radiation tolerant devices for space applications. Choosing the best device for the task at hand can be tricky because there are so many families, each offering different resources.

Introducing Intel FPGAs and SoC FPGAs

The performance and capabilities of the programmable device offerings from Intel span from modest to extremely high, from traditional FPGAs to SoC FPGAs (i.e., devices that combine FPGA programmable fabric with one or more hardened processor cores).

Intel has five active FPGA families in its current product portfolio, including the Max 10, Cyclone 10, and Arria 10 (Figure 2). Intel also has the extremely high-end Stratix 10 and its newer, state-of-the-art Agilex devices.

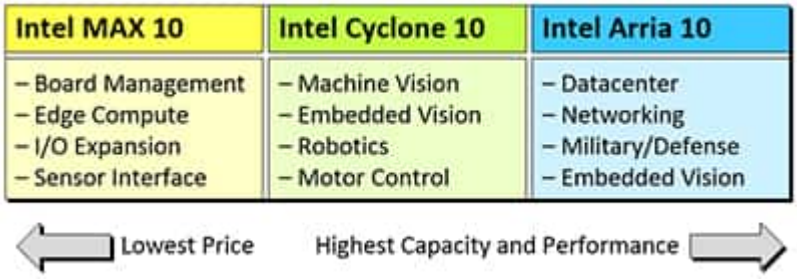

Figure 2: Target applications and markets for three of Intel's five FPGA families—from lowest cost (Max 10) to highest capacity and performance (Arria 10)—range from board management and I/O expansion to embedded vision and military/defense. The Stratix 10 and newer Agilex devices further extend performance. (Image source: Max Maxfield)

The Intel Max 10 FPGAs incorporate on-chip NOR flash memory to hold the FPGA configuration. The other product families have SRAM-based configuration cells and load their configurations on power-up from an on-board external memory device or from elsewhere in the system (e.g., under control of an external processor). Except for the MAX 10 family, all of the other Intel FPGA families are available as FPGAs or as SoC FPGAs with integrated hardened microprocessor systems.

The MAX 10 FPGAs' integrated features include analog-to-digital converters (ADCs) and dual configuration flash, which provides the ability to store and dynamically switch between two images on a single chip. MAX 10 FPGAs also feature Nios II soft core embedded processor support, DSP blocks, and soft double data rate 3 (DDR3) memory controllers.

A representative device from the MAX 10 portfolio would be the 10M16SCU169I7P with 16,000 logic elements/cells, 562,176 total RAM bits, and 130 input/outputs (I/O). A suitable evaluation board for the MAX 10 is Intel's EK-10M08E144. Another is the HINJKIT, an FPGA IoT sensor hub dev kit from Alorium Technology, LLC (Figure 3). The HINJKIT is of particular interest because its MAX 10 FPGA includes Alorium Technology's AVR compatible 8-bit microcontroller, thereby allowing the HINJKIT to be used with the Arduino IDE. The board also provides a high level of interface flexibility for third-party IoT carrier boards, modules and accessories.

Figure 3: The HINJKIT is a MAX 10 FPGA development platform from Alorium that provides a high level of interface flexibility for third-party IoT carrier boards, modules and accessories. (Image source: Alorium Technology)

Intel's Cyclone 10 GX FPGAs provide 12.5 gigabits/sec (Gbps) transceiver-based functions, 1.4 gigabit per second (Gbit/s) low-voltage differential signaling (LVDS), and an up to 72-bit wide DDR3 SDRAM interface for data rates of up to 1,866 megabits per second (Mbits/s). Cyclone 10 GX FPGAs are optimized for high-bandwidth performance applications such as machine vision, video connectivity, and smart vision cameras.

By comparison, the Cyclone 10 LP FPGAs are optimized for low static power, low-cost applications such as I/O expansion, sensor fusion, motor/motion control, chip-to-chip bridging, and control. Both the GX and LP device families support vertical migration, allowing designers to commence designs with one device and migrate to adjacent densities at a later date.

A representative member of the Cyclone 10 portfolio would be the 10CL006YU256A7G with 6,272 logic elements/cells, 276,480 total RAM bits, and 176 I/O. There are a variety of related development and evaluation boards and modules available, including the P0496 DE10-NANO Cyclone V SE SoC Kit (Figure 4), the P0553 TSoM Cyclone V SE evaluation board (again based on the Cyclone V SoC FPGA), and the P0581 TSoM Cyclone V SE FPGA evaluation board, all from Terasic Inc. (Note that the Cyclone V FPGA featured on these boards is an earlier generation of the Cyclone 10 family.)

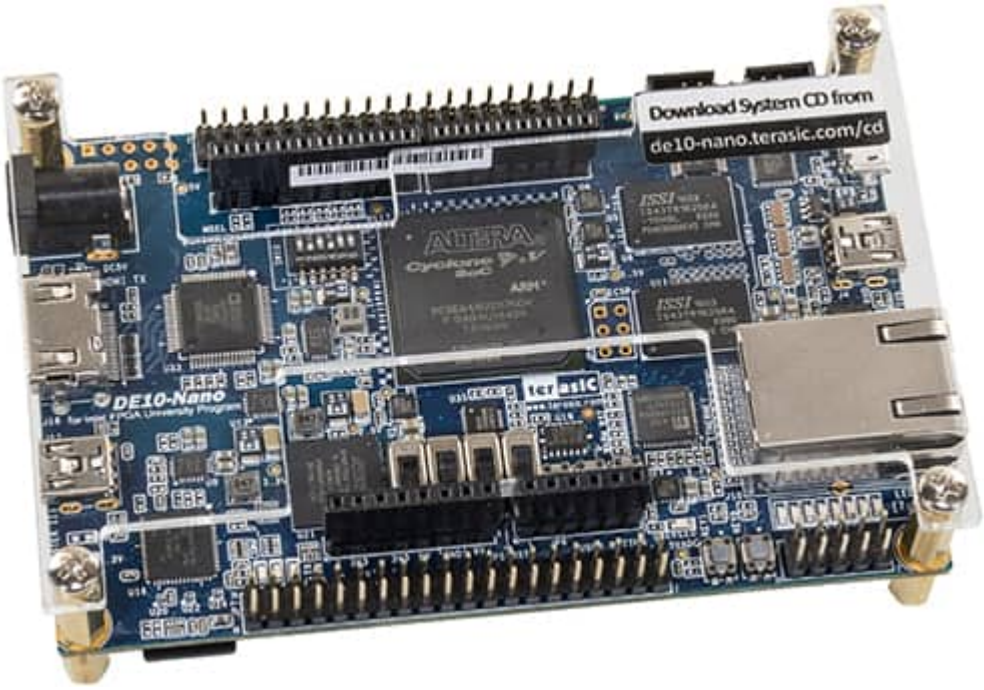

Figure 4: The low-cost P0496 DE10-NANO development kit presents a robust hardware design platform allowing users to experiment with and evaluate Cyclone FPGAs. (Image source: Terasic Inc.)

Based on TSMC's 20 nanometer (nm) process technology, Arria 10 devices combine a dual-core Arm Cortex-A9 MPCore Hard Processor System (HPS) with programmable fabric that includes hardened floating-point DSP blocks. Arria 10 devices also feature embedded high-speed transceivers, hard memory controllers, and protocol intellectual property (IP) controllers, all in a single highly integrated package.

A representative device from the Arria 10 portfolio would be the 10AX022C4U19E3SG, with 220,000 logic elements/cells, 13,752,320 total RAM bits, and 240 I/O. To develop with this FPGA, a suitable platform is Intel's DK-DEV-10AX115S-A Arria 10 GX FPGA evaluation board (Figure 5). This board allows designers to develop and test Arria 10-based PCI Express (PCIe) 3.0 designs; develop and test memory subsystems consisting of DDR4, DDR3, QDR IV, and RLDRAM III memories; and develop modular and scalable designs by using the FPGA mezzanine card (FMC) connectors to interface to an FMC mezzanine card provided by Intel partners.

Figure 5: The DK-DEV-10AX115S-A Arria 10 GX FPGA development kit delivers a complete design environment that

includes all hardware and software required to develop and test PCIe-based designs, memory subsystems, and systems based on FMC cards. (Image source: Intel)

Intel's Stratix 10 FPGAs and SoC FPGAs feature the Intel Hyperflex FPGA Architecture. FPGAs in this line of devices address the needs of high-throughput systems with up to 10 TFLOPS of floating-point performance and transceiver support from 28.3 Gbits/s (GX) to 57.8 Gbits/s, the latter in up to 144 transceiver lanes to meet the bandwidth demands of 5G communications, cloud computing, network virtualization, and optical transport networks. A variant includes a PCIe Gen4 interface with up to x16 configuration at 16 giga transfers per second (GT/s) and a new memory controller to support select Intel Optane DC persistent memory.

Supporting development with the Stratix 10 line is Intel's DK-DEV-1SGX-L-A Stratix 10 GX FPGA evaluation board (Figure 6). This board allows designers to develop and test Stratix 10-based PCI Express (PCIe) 3.0 designs; develop and test memory subsystems consisting of DDR4, DDR3, QDR IV, and RLDRAM III memories; and develop modular and scalable designs by using the FPGA mezzanine card (FMC) connectors to interface to a FMC mezzanine card, again provided by Intel partners.

Figure 6: The DK-DEV-1SGX-L-A Stratix 10 GX FPGA evaluation board delivers a complete design environment that includes all hardware and software required to develop and test PCIe-based designs, memory subsystems, and systems based on FMC cards. (Image source: Intel)

Design and development with Intel's FPGAs and SoCs FPGAs

In Part 1 of this series on FPGAs, we noted that the traditional way to design these devices is for engineers to use a hardware description language (HDL) like Verilog or VHDL to capture the design's intent at a level of abstraction known as the register transfer level (RTL). These RTL descriptions may first be simulated to verify they perform as required, after which they are passed to a synthesis tool that generates the configuration file used to configure (program) the FPGA.

The next step up in abstraction is to capture the design's intent using a programming language like C/C++, or a special implementation like SystemC, which is a set of C++ classes and macros that provide an event-driven simulation interface. These facilitate simulation of concurrent processes, each described using plain C++ syntax. Such descriptions can be analyzed and profiled by running them like regular

programs, after which they are passed to a high-level synthesis (HLS) engine, which outputs RTL that is fed into the regular synthesis engine.

At a conceptually higher level of abstraction is Open CL (Open Computing Language), which is a framework for writing programs that execute across heterogeneous platforms. In addition to FPGAs, these platforms include central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs), and other processors or hardware accelerators. OpenCL specifies programming languages (based on C99 and C++11) for programming these devices, along with application programming interfaces (APIs) to control the platform and execute programs on the compute devices.

In the case of Intel FPGAs and SoC FPGAs, users develop designs for all product families using Intel Quartus Prime Design Software. This software is available in three versions: Quartus Prime Pro Edition, Quartus Prime Standard Edition, and Quartus Prime Lite Edition. The Pro and Standard Editions require paid licenses, while the Lite Edition can be downloaded for free.

The Pro Edition is optimized to support the advanced features in next-generation FPGAs and SoC FPGAs with the Agilex, Stratix 10, Arria 10, and Cyclone 10 GX device families; the Standard Edition includes extensive support for earlier device families, in addition to the Cyclone 10 LP device family; and the Lite Edition, which is available as a free download with no license file required, provides an ideal entry point to high-volume device families.

Intel offers a tremendous range of additional design tools and resources, including Intel DSP Builder, Intel High Level Synthesis (HLS) Compiler, and Intel FPGA SDK for OpenCL.

Also useful is the Intel Distribution of the OpenVINO Toolkit, which allows developers to create artificial intelligence (AI), computer vision, audio, speech, and language applications, enabling deep learning inference from edge to cloud. In addition to FPGAs, this toolkit supports heterogeneous execution across Intel architecture and AI accelerators—CPUs, iGPUs, Intel Movidius Vision Processing Unit (VPU), and Intel Gaussian & Neural Accelerator (GNA)—using a common API. This toolkit is supported by the OpenVINO Starter Kit based on Cyclone V GT FPGA.

Conclusion

Optimal processing design solutions are often provided by combinations of processors and FPGAs, by FPGAs on their own, or by FPGAs that boast hard processor cores as part of their fabric. As a technology, FPGAs have evolved rapidly over the years and are able to address many design requirements in terms of flexibility, processing speed, and power, making them very useful for a wide range of applications from intelligent interfaces to machine vision and artificial intelligence.

The programmable device offerings from Intel span the low to extremely high range, from traditional FPGAs to SoC FPGAs. As shown, to create designs with these devices, Intel and other third parties offer a powerful suite of development kits and tools to address the needs of hardware developers, software developers, and the creators of artificial intelligence and computer vision systems.

Further reading

1. Fundamentals of FPGAs: What Are FPGAs and Why Are They Needed?

2. Fundamentals of FPGAs — Part 2: Getting Started With Lattice Semiconductor FPGAs
3. Fundamentals of FPGAs — Part 3: Getting Started With Microchip Technology's FPGAs
4. Fundamentals of FPGAs — Part 4: Getting Started With Xilinx FPGAs
5. FPGAs 101: A Beginners' Guide
6. Implement Low-Power, High-Performance Edge Computing Using Efinix's Quantum-Enabled FPGAs

About this author

Clive "Max" Maxfield

Clive "Max" Maxfield received his BSc in Control Engineering in 1980 from Sheffield Hallam University, England and began his career as a designer of central processing units (CPUs) for mainframe computers. Over the years, Max has designed everything from silicon chips to circuit boards and from brainwave amplifiers to steampunk Prognostication Engines (don't ask). He has also been at the forefront of Electronic Design Automation (EDA) for more than 30 years.

Max is the author and/or co-author of a number of books, including Designus Maximus Unleashed (banned in Alabama), Bebop to the Boolean Boogie (An Unconventional Guide to Electronics), EDA: Where Electronics Begins, FPGAs: Instant Access, and How Computers Do Math. Check out his "Max's Cool Beans" blog.

About this publisher

DigiKey's North American Editors