Fundamentals of FPGAs – Part 4: Getting Started with Xilinx FPGAs

Clive "Max" Maxfield ⁝⁝ 1/4/2020

By Clive "Max" Maxfield

Contributed By DigiKey's North American Editors

2020-04-01

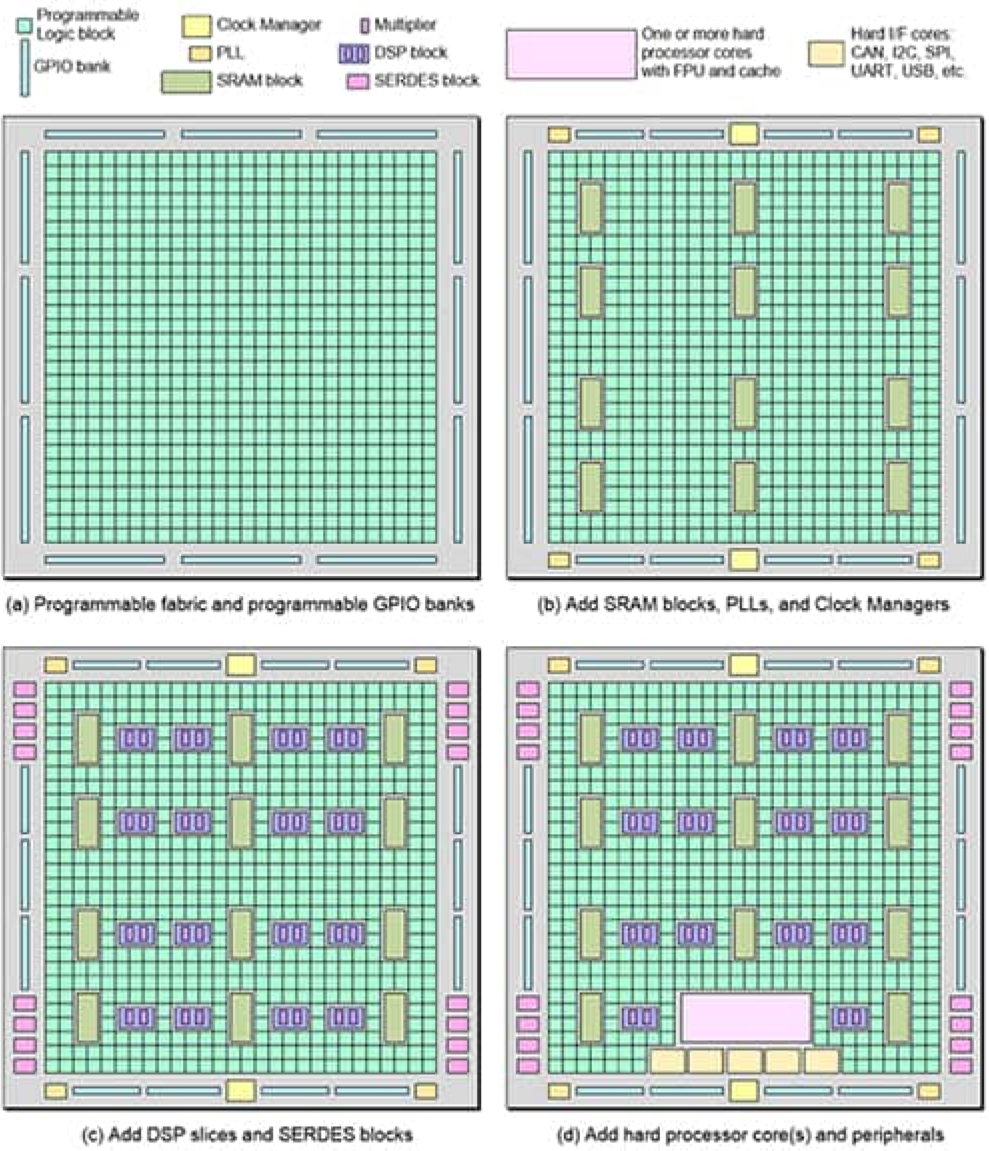
Editor's Note: Optimal processing solutions are often provided by combinations of RISC, CISC, graphics processors, and FPGAs; FPGAs on their own; or by FPGAs that boast hard processor cores as part of their fabric. However, many designers are unfamiliar with the capabilities of FPGAs, how they've evolved, 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*). Here, in Part 4, the components and tools from* Xilinx are discussed*. In Part 5 the offerings from* Altera will be considered*.*

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; this article looks at the lineup from Xilinx.

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 1a). 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 1b). Digital signal processing (DSP) blocks (called DSP slices) and high-speed serializer/deserializer (SERDES) blocks can also be added (Figure 1c).


Programmable Logic block
GPIO bank
Clock Manager
PLL
SRAM block
Multiplier
DSP block
SERDES block
One or more hard processor cores with FPU and cache
Hard I/F cores: CAN, I2C, SPI, UART, USB, etc

(a) Programmable fabric and programmable GPIO banks

(b) Add SRAM blocks, PLLs, and Clock Managers

(c) Add DSP slices and SERDES blocks

(d) Add hard processor core(s) and peripherals

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^{2}C, 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 1d). 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 Altera (which was acquired by Intel), Atmel (which was acquired by Microchip Technology), Efinix, Lattice Semiconductor, Microsemi (which was also acquired by Microchip Technology), 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 that are targeted at high radiation environments like space. Choosing the best device for the task at hand can be tricky because there are so many families, each offering different resources.

Introducing Xilinx FPGAs, SoCs, MPSoCs, RFSoCs, and ACAPs

The performance and capabilities of the programmable device offerings from Xilinx span from modest to extremely high. From traditional FPGAs, to SoCs (FPGA programmable fabric with a single hard core processor), MPSoCs (FPGA programmable fabric with a multiple hard core processors), RFSoCs (MPSoCs with RF capability), and ACAPs (Adaptive Compute Acceleration Platforms) (Figure 2).

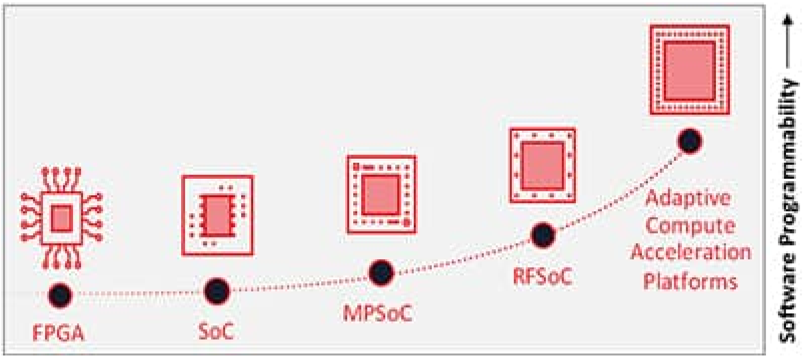


Figure 2: Over time, the Xilinx architectural portfolio has evolved from simple FPGAs containing only programmable fabric, to SoC devices in which the programmable fabric is augmented with a hard core processor, to MPSoCs with multiple processors, to RFSoC with RF capabilities, to the latest generation of ACAPs, which are targeted toward applications like AI. (Image source: Max Maxfield)

Xilinx has such an extensive product portfolio, addresses so many market segments, and provides such a wide range of deployment methods that it can be challenging for someone who is new to FPGAs to understand the "big picture."

In the case of the markets Xilinx addresses, but is not limited to, data centers (compute, network, storage); communications (wired, wireless); aerospace and defense; industrial, scientific, and medical (ISM); test, measurement, and emulation (TME); and automotive, broadcast, and consumer.

In the case of deployment methods, these include what Xilinx calls Hardware Adaptable Devices, which include chips, evaluation boards, and development kits; Deployable End-Systems, which include system-on-modules (SoMs) and PCIe accelerator cards; and FPGA as a service (FAAS), which includes evaluating and leveraging Xilinx technologies via leading cloud providers, including Amazon Web Services (AWS), Alibaba.com, and Nimbix.net

With regard to the Xilinx FPGA offerings, one way to categorize these is by the process technology node (Figure 3).

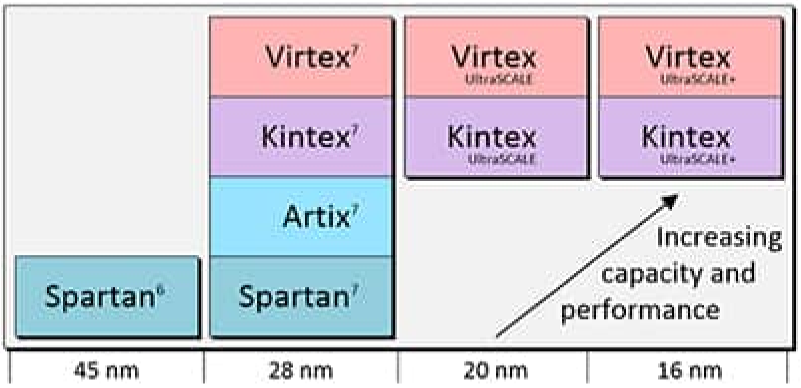


Figure 3: Xilinx FPGA offerings provide a comprehensive multi-node portfolio to address requirements across a wide set of applications. (Image source: Max Maxfield)

Depending on the target application, designers may opt for a low-cost, small footprint FPGA implemented at an earlier technology node, or in the case of a state-of-the art networking application, for example, choose a high capacity, high bandwidth, high performance device implemented at a more recent technology node.

In the case of designs that require one or more hard processor cores (and other hardened functions like GPUs, codecs, and soft decision, forward error correction (SD-FEC) cores), Xilinx offers a portfolio of devices under the umbrella name of Zynq. A summary of the Zynq SoC, MPSoC, and RFSoC offerings is shown in Figure 4. This suite of solutions provides designers with a wide range of capabilities to facilitate optimization for power, performance, cost, and time to market.

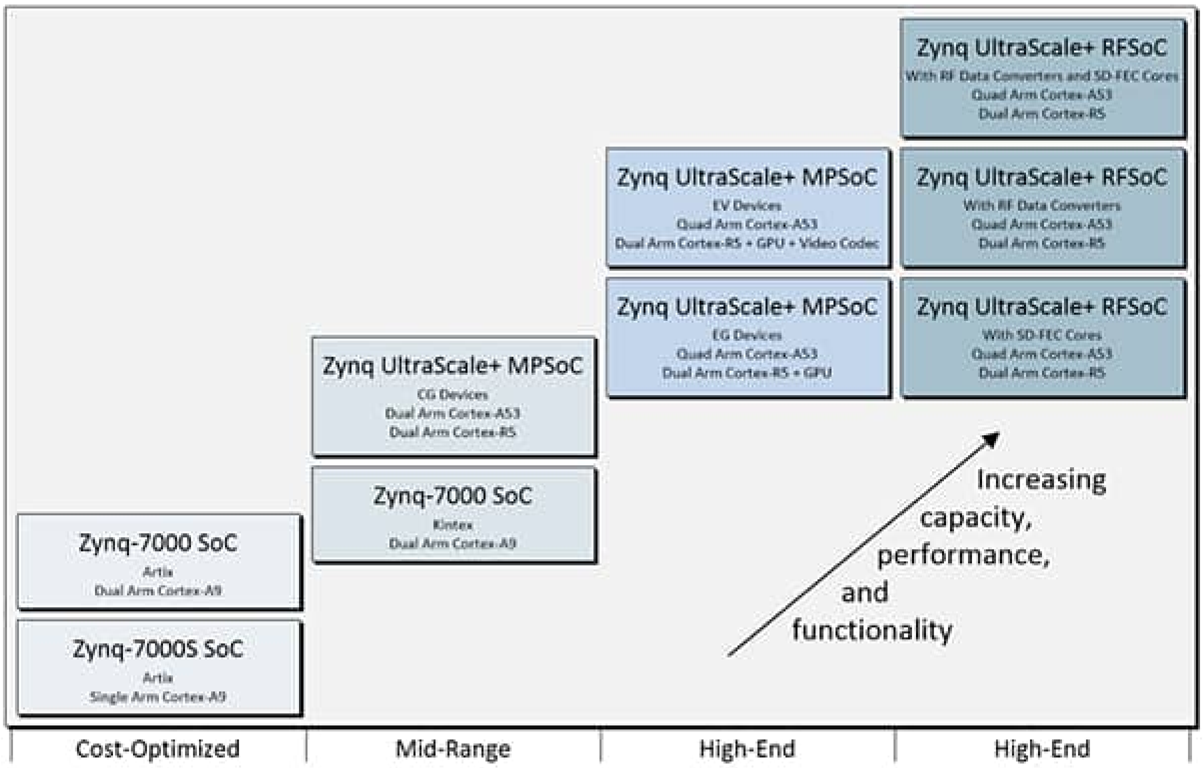


Figure 4: Xilinx SoC, MPSoC, and RFSoC offerings integrate the software programmability of a processor with the hardware programmability of an FPGA, providing designers with system performance, flexibility, and scalability. (Image source: Max Maxfield)

The most recent Xilinx offerings are the Versal Adaptive Compute Acceleration Platform (ACAP) devices, all of which are implemented at the 7 nanometer (nm) process technology node. ACAPs are highly integrated, multicore compute platforms that can adapt to evolving and diverse algorithms. They are dynamically customizable at the hardware and software levels to fit a wide range of applications and workloads. Architected around a programmable network-on-chip (NoC), ACAPs are easily programmed by hardware designers and software developers alike.

New features in the Versal devices include intelligent engines (a massive array of vector processors for ML and DSP workloads); a high bandwidth, low-latency, and low-power programmable NoC that can move terabytes of data; and an integrated shell that provides improved performance, utilization, and productivity with a pre-built core infrastructure and system connectivity.

An overview of the Versal ACAP portfolio is shown in Figure 5.

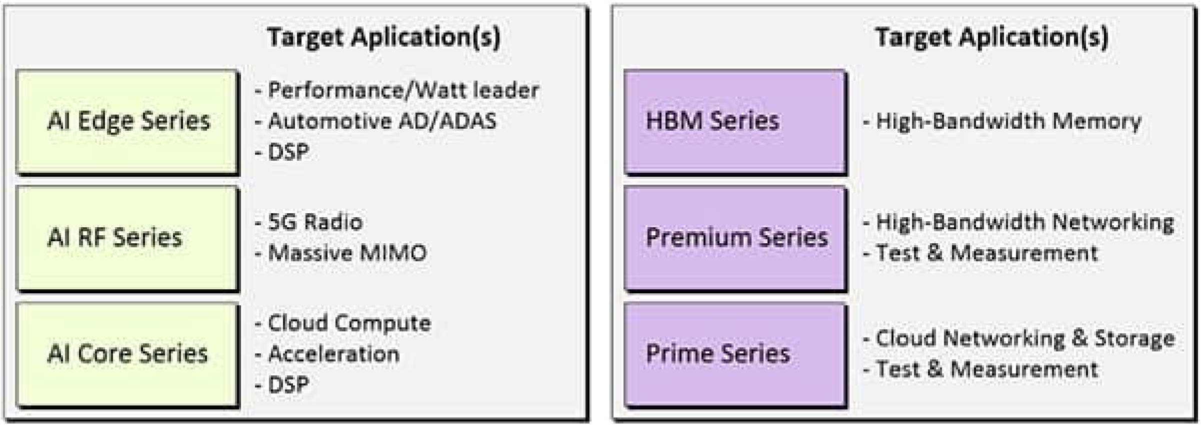


Figure 5: Xilinx Versal ACAPs are highly integrated, multicore compute platforms that can adapt to evolving and diverse algorithms. ACAPs are dynamically customizable at the hardware and software levels to fit a wide range of applications and workloads. (Image source: Max Maxfield)

As will be discussed in the design tools section, a key differentiator associated with the Versal devices is a new software stack that is targeted at data scientists and software engineers, as well as traditional hardware design engineers.

There are myriad Xilinx devices now available. A few representative offerings would be an Artix-7 FPGA, a Kintex UltraScale FPGA, a Kintex UltraScale+ FPGA, a Zynq-7000 SoC Module from Trenz Electronic GmbH, and a Zynq UltraScale+ MPSoC.

Similarly, there is a wide variety of evaluation and development boards available. Some representative offerings would be an Artix-7 FPGA Evaluation Board from Digilent, a Kintex UltraScale FPGA Evaluation Board from Analog Devices, a Kintex UltraScale+ FPGA Evaluation Board from Xilinx, a Zynq-7000 SoC FPGA Evaluation Board from Digilent, and a Zynq UltraScale+ MPSoC FPGA Evaluation Board, also from Xilinx.

Design and development with Xilinx FPGAs, SoCs, and ACAPs

One area where Xilinx really differentiates itself from the competition is in the breadth and depth of its design tools and flows.

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 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.

All of this functionality is embraced by the Vivado Design Suite HLx Editions, the output of which is the configuration bitstream that will be loaded into the targeted FPGA, SoC, MPSoC, RFSoC, or ACAP devices. In addition to allowing hardware developers leverage C-based design and optimized reuse, Vivado also provides IP sub-system reuse, integration automation, and accelerated design closure (Figure 6).

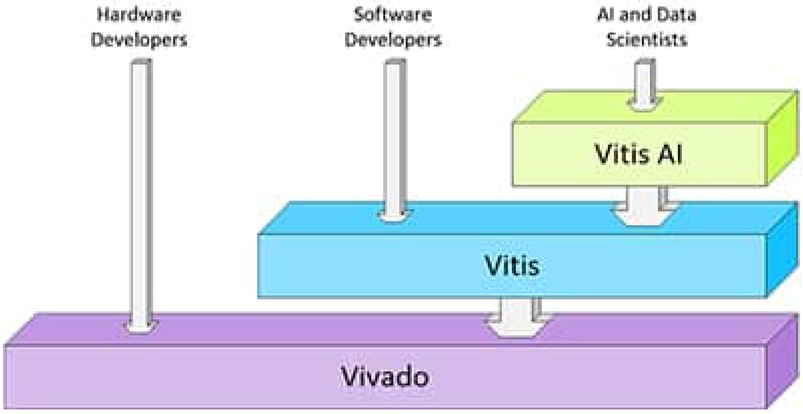


Figure 6: A high-level view of the Xilinx Vivado and Vitis design tool stack reflects how users can work with the tools at the most appropriate levels of abstraction. Hardware designers work with Vivado, software developers work with Vitis, and AI and data scientists work with Vitis AI. (Image source: Max Maxfield)

The next level of abstraction is supported by the Vitis Unified Software Platform, which provides software developers with the ability to seamlessly build accelerated applications. Conceptually "sitting on top" of Vitis is Vitis AI, which allows AI and data scientists to work at the TensorFlow level of abstraction. Vitis AI is a development platform for AI inference on Xilinx hardware platforms, including both edge devices and Alveo PCIe cards. It consists of optimized IP, tools, libraries, models, and example designs, and is designed to exploit the full potential for AI acceleration on Xilinx FPGA and ACAP devices.

Vitis AI feeds into Vitis, which itself feeds into Vivado. The key takeaway from Figure 6 is that users only "see" what they need to "see." That is, hardware developers will only "see" Vivado, software developers will only "see" Vitis, and AI and data scientists will only "see" Vitis AI. In this way, users can work with the tools at the most appropriate levels of abstraction.

Providing software developers with a tool suite like Vitis, which isolates them from the underlying hardware, opens FPGAs up to a much larger pool of developers. Similarly, providing AI and data scientists with a tool suite like Vitis AI, which allows them to focus on their own level of abstraction and isolates them from the underlying software, once again opens up FPGAs to a new class of developers.

In providing these capabilities, Xilinx is at the forefront of an industry-wide push to raise FPGA tools to ever higher levels of design abstraction that will allow developers to more easily take advantage of the capabilities of these devices and integrate them into their next designs.

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.

As shown, the programmable device offerings from Xilinx span from modest to extremely high in terms of performance and capabilities. They range from traditional FPGAs, to SoCs (FPGA programmable fabric with a single hard core processor), MPSoCs (FPGA programmable fabric with a multiple hard core processors), RFSoCs (MPSoCs with RF capability), and ACAPs (Adaptive Compute Acceleration Platforms).

To help designers create designs with these devices, Xilinx offers a suite of tools to address the needs of hardware developers (Vivado), software developers (Vitis), and AI and data scientists (Vitis AI).

DigiKey

Disclaimer: The opinions, beliefs, and viewpoints expressed by the various authors and/or forum participants on this website do not necessarily reflect the opinions, beliefs, and viewpoints of DigiKey or official policies of DigiKey.

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