

Fundamentals of FPGAs – Part 2: Getting Started with Lattice Semiconductor's FPGAs

Clive "Max" Maxfield :: 18/12/2019

By Clive "Max" Maxfield

Contributed By DigiKey's North American Editors

2019-12-18

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 and why they are needed. Here, [Part 2](#) focuses on the FPGA device families and design tools offered by FPGA vendor, [Lattice Semiconductor](#). [Part 3](#), [Part 4](#), and [Part 5](#) will look at FPGAs from [Altera](#), [Microchip](#), and [Xilinx](#).

As discussed in [Part 1](#), field-programmable gate arrays (FPGAs) have many characteristics that make them an invaluable computing asset, either 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, starting here with Lattice Semiconductor's lineup.

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, which is presented as an array of programmable logic blocks (Figure 1a). The next step up the FPGA fabric is to include things such as blocks of SRAM, called block RAM (BRAM), phase lock loops (PLLs), and clock managers (Figure 1b). Digital signal processing (DSP) blocks (DSP slices) and high-speed serializer/deserializer (SERDES) blocks can also be added (Figure 1c).

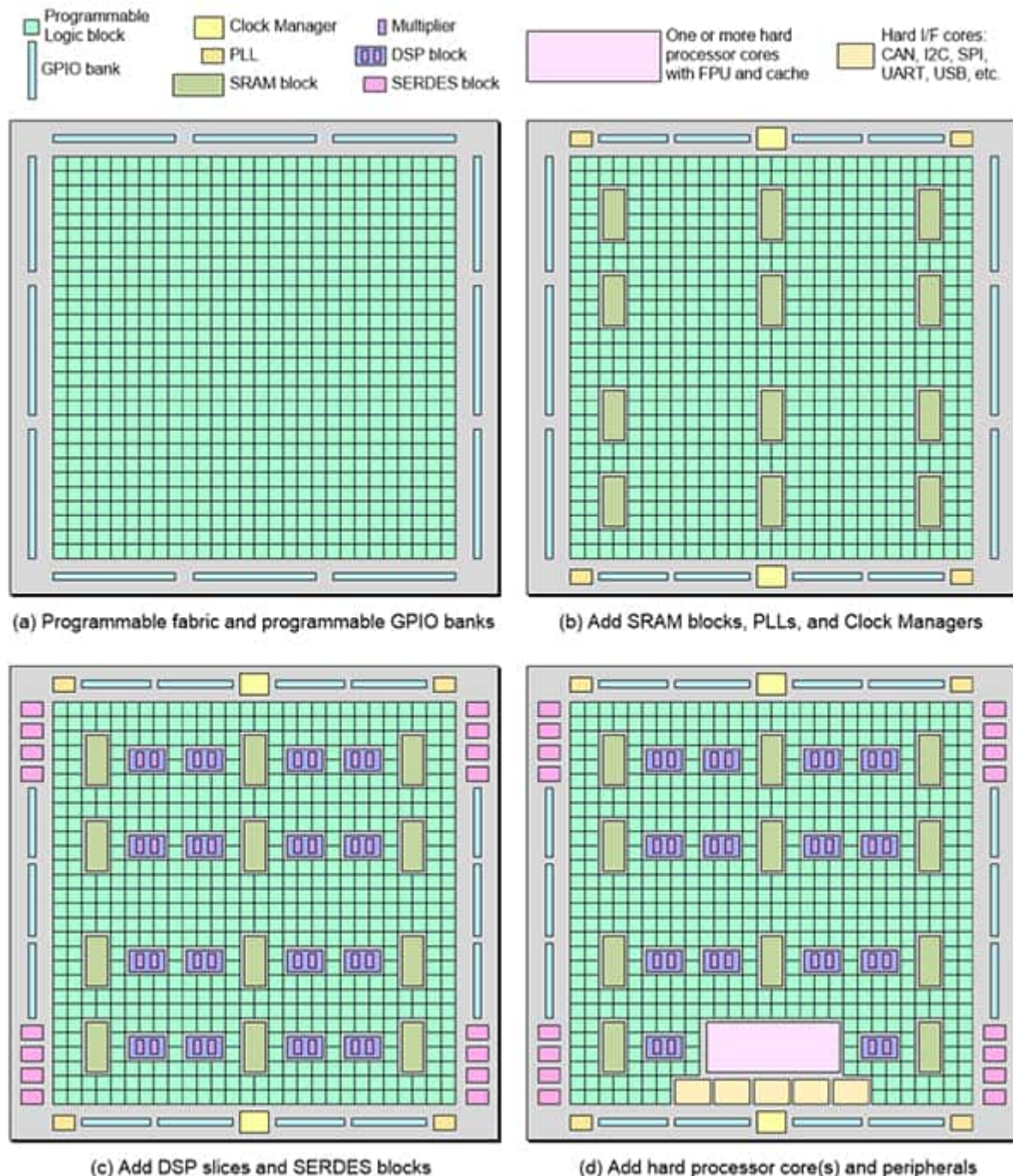


Figure 1: The

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, I²C, 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). 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), Lattice Semiconductor, [Microsemi](#) (which was also acquired by Microchip Technology), and Xilinx.

All of these vendors field multiple families of FPGAs; some offer system-on-chip (SoC) FPGAs (containing hard processor cores), 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. This article focuses on the device families and design tools offered by Lattice Semiconductor.

Introducing Lattice Semiconductor FPGAs

Lattice Semiconductor's FPGA offerings span the low to mid-range, with a focus on low-power devices that address customer problems across the network, from the edge to the cloud, in the rapidly growing communications, computing, industrial, automotive, and consumer markets.

Lattice offers four main FPGA families:

- iCE (which it cites as the world's smallest, ultra-low-power FPGAs)
- CrossLink and CrossLinkPlus (optimized for high-speed video and sensor applications)
- MachXO (optimized for bridging, expansion, and platform management and security applications)
- ECP (general purpose devices targeted at connectivity and acceleration applications)

Lattice also offers a number of design and verification tool suites, including [Lattice Diamond Software](#) (for CrossLink/CrossLinkPlus, MachXO, and ECP devices) and [Lattice Radiant Software](#) (for iCE FPGAs and future architectures), along with [LatticeMico](#) (a graphical tool to create soft microprocessor-based designs) and the [Lattice sensAI Stack](#) and the [Neural Network Compiler](#) (for artificial intelligence (AI) and machine learning (ML) designs).

Traditional: ECP FPGAs

Lattice's ECP devices are what many designers would consider to be "traditional" FPGAs. They can contain up to 85,000 (k) four-input look-up tables (LUTs) in a 10 millimeter (mm) x 10 mm, 0.5 mm pitch package. They have low static and dynamic power consumption with single-channel protocol agnostic SERDES functions below 0.25 watts and quad-channel SERDES functions below 0.5 watts.

In addition to SRAM blocks, digital signal processing (DSP) blocks, phase lock loops (PLLs), and clock managers, ECP FPGAs boast programmable I/O support for LVCMOS 33/25/18/15/12, XGMII, LVTTTL, LVDS, Bus-LVDS, 7:1 LVDS, LVPECL and MIPI D-PHY input/output interfaces.

The configuration cells in ECP FPGAs are SRAM-based, which means that—like all other SRAM-based FPGAs—their configuration has to be loaded from an external source (e.g., flash memory device, microprocessor, microcontroller) when power is applied to the system.

A good example of an ECP device is the [LFE5UM5G-25F-8BG381C](#) ECP5 5G SERDES-capable FPGA in a 10 mm x 10 mm package. To allow designers to better investigate and experiment with the features of the ECP5 FPGA family, Lattice also provides a corresponding ECP5-5G development board, the [LFE5UM5G-45F-VERSA-EVN](#) (Figure 2).

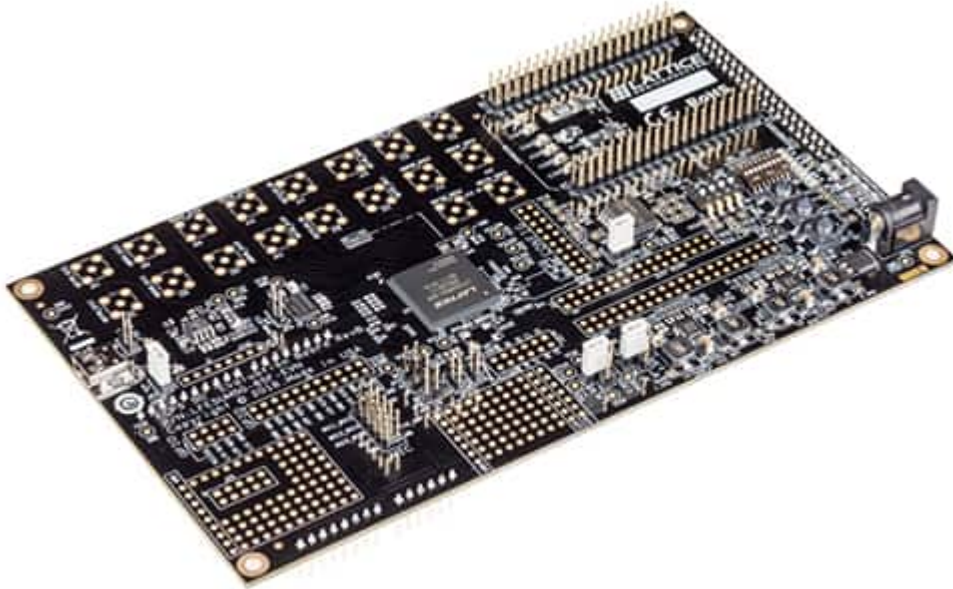


Figure 2: The ECP5

Evaluation Board is a prototyping board with abundant logic, I/O, 5G SERDES, and expansion headers. (Image source: Lattice Semiconductor)

Small in size, but powerful: Lattice iCE FPGAs

The iCE devices are the smallest FPGAs available, with the smallest member of the family offering 18 I/Os in a 1.4 mm x 1.4 mm package. iCE FPGAs offer a flexible logic architecture with up to 5k four-input LUTs, up to 128 kilobits (Kbits) of Lattice's embedded sysMEM BRAM, and 1 megabit (Mbit) of single-port RAM (SPRAM), high performance DSP blocks, and customizable I/Os.

Although they are small and low power with sleep current as low as 75 microamps (μA) and active current from 1 to 10 milliamps (mA) for most applications, iCE FPGAs are also powerful. For example, they are able to implement artificial neural networks (ANNs) for the pattern matching necessary to bring always-on intelligence to the edge.

The configuration data in iCE FPGAs is stored in non-volatile memory (NVM), which means these devices are one-time programmable (OTP). Having said this, they also contain SRAM-based configuration cells. During development, the design can be tested by directly loading the SRAM-based configuration cells from the outside world. Once the design is committed, it can be loaded into the NVM. When the device is powered up, the configuration stored in the NVM is automatically copied into the SRAM-based configuration cells in a massively parallel fashion.

An example iCE device would be the [ICE40UL1K-SWG16ITR1K](#) iCE40 UltraLite, which is the world's smallest form factor FPGA (at the time of this article's posting), providing 42 microwatt (μW) static power in a 1.4 mm x 1.4 mm package. A representative development board would be the [HM01B0-UPD-EVN](#) Himax HM01B0 UPduino shield and sensAI modular demonstration board (Figure 3).

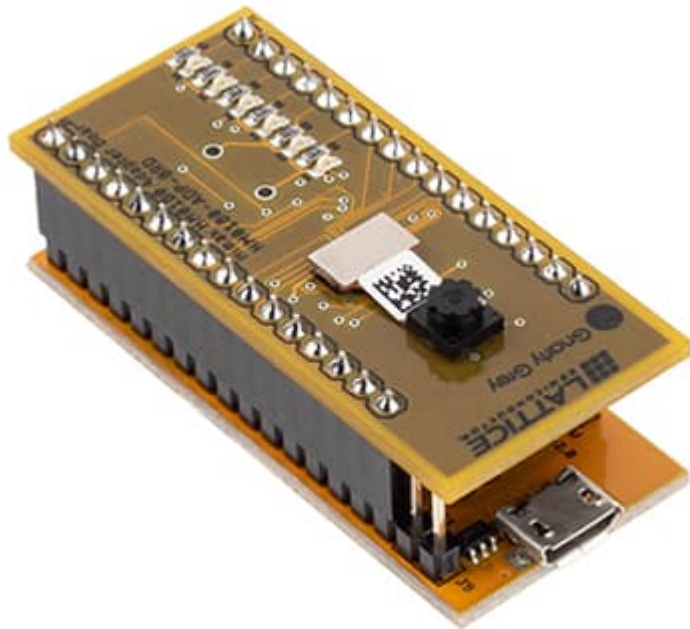


Figure 3: The Himax HM01B0 UPduino shield is a complete development kit for implementing artificial intelligence (AI) using vision and sound as sensory inputs. (Image source: Lattice Semiconductor)

The kit is based on the UPduino 2.0 board, a rapid prototyping development board in the Arduino form factor with the performance and I/O capabilities of the iCE40 UltraPlus FPGA. It also includes the Himax HM01B0 low-power image sensor module and two I²S microphones.

Specialized: CrossLink and CrossLinkPlus FPGAs

The CrossLink and CrossLinkPlus families are specialized FPGAs that in addition to programmable logic and substantial I/O capabilities, contain hardened versions of specifications that are widely used in industrial and automotive applications. These include the Mobile Industry Processor Interface (MIPI) D-PHY high speed data communications physical layer standard, Camera Serial Interface 2 (CSI2), and Display Serial Interface 2 (DSI2) cores, all in packages measuring 6 mm x 6 mm (CrossLink) and 3.5 mm x 3.5 mm (CrossLinkPlus).

Like iCE FPGAs, the configuration data in CrossLink devices is stored in OTP NVM, and they too contain SRAM-based configuration cells that can be directly loaded during development for testing. Once the design is committed, it is loaded into the NVM and is automatically copied into the SRAM-based configuration cells in a massively parallel fashion at power up. By comparison, the configuration cells in CrossLinkPlus devices are flash-based, which means these devices can be reprogrammed as required; also, they provide instant-on capability in less than 10 milliseconds (ms).

An example CrossLink device would be the [LIF-MD6000-6JMG80I](#) with 5,936 logic elements/cells, 184,320 total RAM bits, and 37 I/Os. To get started on embedded vision designs, the [LF-EVDK1-EVN](#) Embedded Vision Development Kit allows designers to combine CrossLink-based MIPI inputs with ECP5 FPGA processing to prototype embedded vision designs (Figure 4).

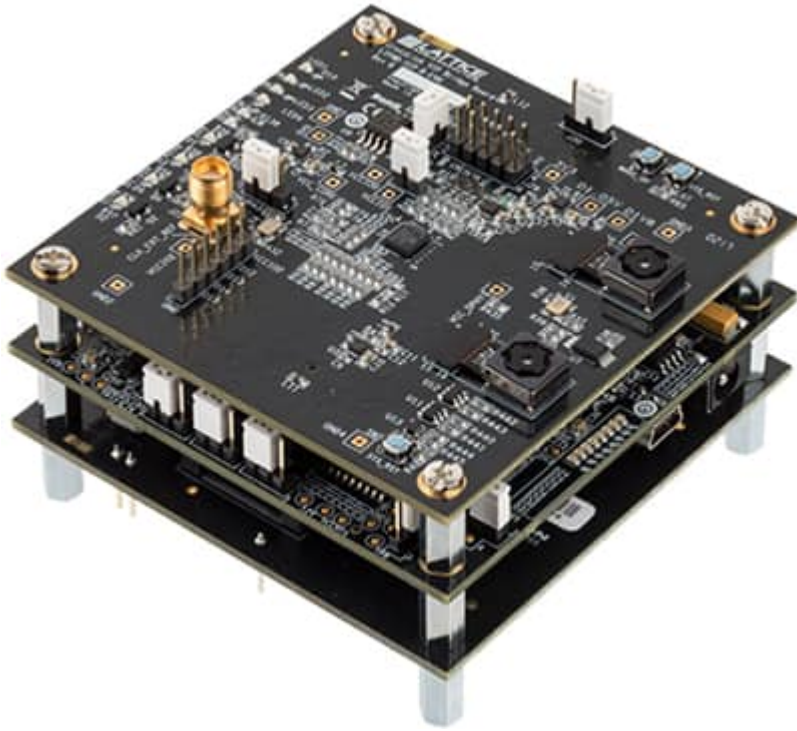


Figure 4: The Lattice LF-EVDK1-EVN

Embedded Vision Development Kit provides embedded system designers with a software and hardware prototyping environment that includes the ability to mix and match input and output boards to interface with a variety of image sensors and displays. (Image source: Lattice Semiconductor)

For I/O and power management features: MachXO FPGAs

With hundreds of I/Os, MachXO FPGAs are perfect for a wide range of applications that require GPIO expansion, interface bridging, and power-up management functions. Recent family members are designed for NIST compliance and include additional security features to ensure that system hardware and firmware is kept secure.

MachXO FPGAs have a robust set of GPIOs that include features like "hot socketing" capability, which means voltage can be applied to the I/Os irrespective of the state of the power supply rails. Furthermore, while most FPGA inputs default to a pull-up state, MachXO inputs default to pull-down, thereby making them well-suited to control function applications. With instant-on in less than 10 ms, MachXO FPGAs provide ideal solutions for "first on, last-off" control devices that manage and sequence other components during system power-up and power-down.

The configuration data in MachXO devices is stored in flash memory. MachXO devices also contain SRAM-based configuration cells. When the device is powered up, the configuration data stored in flash is automatically copied into the SRAM-based configuration cells in a massively parallel fashion. Furthermore, while the device is running, it is possible to load a new configuration into flash, and then cause this new configuration to be copied into the SRAM cells at an appropriate time.

A good example of a MachXO device is the [LCMXO3LF-9400C-6BG256C](#) with 9,400 logic elements/cells, 442,368 total RAM bits, and 206 I/Os. A representative development board would be the [LCMXO3LF-6900C-S-EVN](#) MachXO3 starter kit (MachXO3L version).

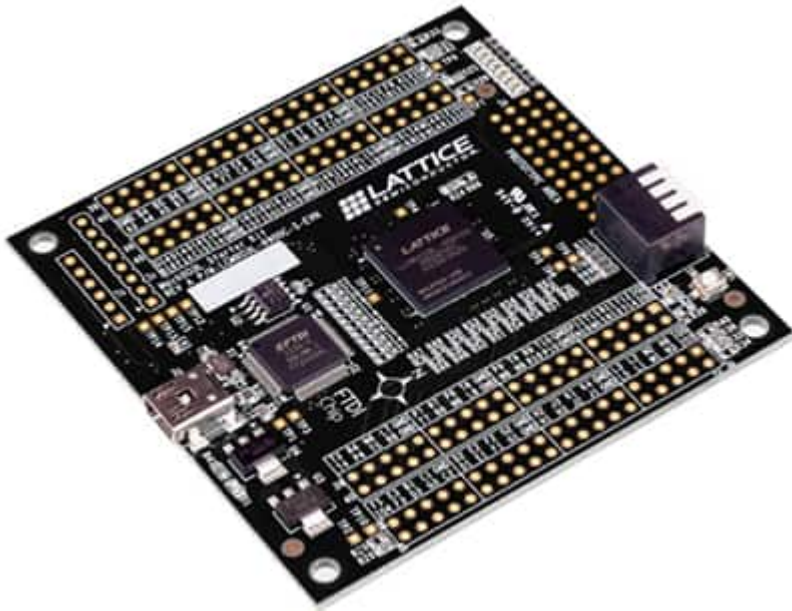


Figure 5: The MachXO3L starter kit is a basic breakout board to allow simple evaluation and development of MachXO3L based designs. (Image source: Lattice Semiconductor)

SPI flash is available on the kit board for evaluating external boot or dual-boot functional capabilities. The [LCMXO3L-DSI-EVN](#) MachXO3L DSI breakout board is recommended for MIPI DSI and CSI2 I/O evaluation, and the [LCMXO3L-SMA-EVN](#) MachXO3L SMA breakout board is recommended for high-speed differential I/O evaluation.

Design and development with Lattice Semiconductor FPGAs

One of the most common techniques used to develop with FPGAs is that of language-driven design (LDD). This involves capturing the design intent at a level of abstraction known as register transfer level (RTL) using a hardware description language (HDL), such as Verilog or VHDL. Following verification via logic simulation, this representation is then fed into a synthesis engine, along with additional information such as the target FPGA type, pin assignments, and timing constraints (e.g., maximum input to output delays). The output from the synthesis engine is a configuration file, which is either loaded directly into the FPGA, or loaded into an external memory device in the case of SRAM-based FPGAs (Figure 6).

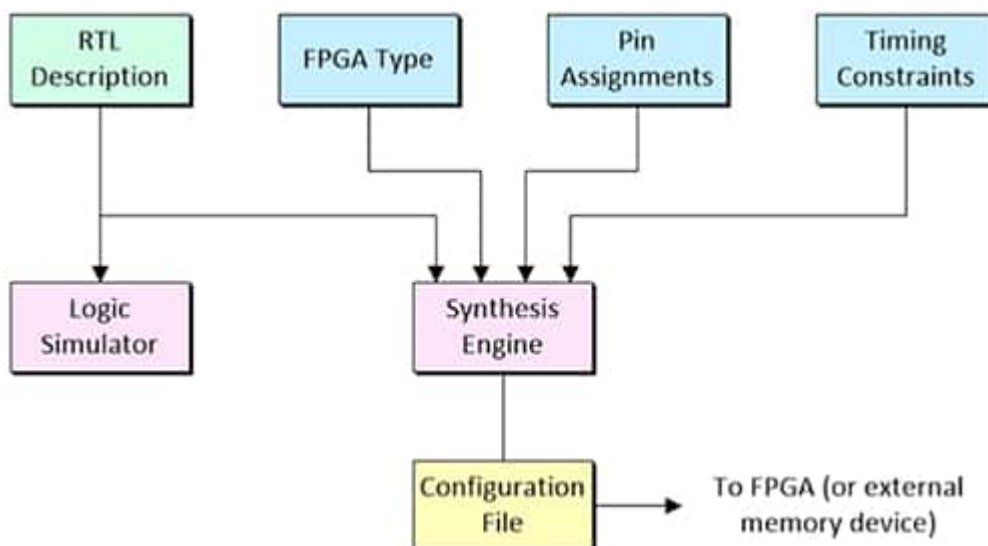


Figure 6: Following verification via logic simulation, the RTL design description is fed into a synthesis engine along with

additional design details like the FPGA type, pin assignments, and timing constraints. The output from the synthesis engine is a configuration file which is loaded directly into the FPGA (in the case of NVM or flash-based devices), or into an external memory device (in the case of SRAM-based devices). (Image source: Max Maxfield).

Lattice Diamond falls into this class of tool and offers a complete GUI-based FPGA design and verification environment for CrossLink, MachXO, and ECP devices.

Like Lattice Diamond, Lattice Radiant also offers a complete GUI-based FPGA design and verification environment, but Lattice Radiant is targeted at iCE FPGAs and future architectures.

Lattice Radiant offers a variety of features:

- Industry standard IEEE 1735 intellectual property (IP) encryption and Synopsys Design Constraints (SDC) for maximum interoperability
- Ease of design navigation and debugging enabled by an integrated tool set environment
- Simple “one click” design implementation execution enabled by a new process toolbar
- Complete closed-loop physical to logical design implementation cross probing
- An IP packaging capability that allows developers and third-party IP providers to package encrypted IP in a form suitable for distribution

LatticeMico system development tools

Lattice offers two soft processor cores—the LatticeMico8 and the LatticeMico32—both of which can be implemented into an FPGA's programmable fabric.

The LatticeMico8 is an 8-bit microcontroller optimized and fully tested for the MachXO2 family of programmable logic devices (PLDs). It can also be used as a reference design for the other families of FPGAs. The microcontroller core combines a full 18-bit wide instruction set with 32 general purpose registers and consumes minimal device resources—less than 200 LUTs in the smallest configuration—while maintaining a broad feature set.

The LatticeMico32 is a 32-bit Harvard, RISC architecture microprocessor. By combining a 32-bit wide instruction set with 32 general purpose registers, the LatticeMico32 provides the performance and flexibility suitable for a wide variety of markets. By using a RISC architecture, the core consumes minimal device resources, while maintaining the performance required for a broad application set. To accelerate development of microprocessor systems, several optional WISHBONE controller compatible peripheral components may be integrated with the LatticeMico32.

The [LatticeMico system development tools](#) provide a graphical user interface that allows users to drag-and-drop LatticeMico processor cores and peripherals, connect them to buses, and define various parameters for each component, such as its location in the processor's address space. Once the system has been defined, the tools can automatically generate the corresponding RTL to be simulated and synthesized. The system also includes the tools that allow users to generate the software that will run on the processor cores.

Machine learning tools: Lattice sensAI Stack and Neural Network Compiler

Machine learning (ML) and artificial intelligence (AI) applications are currently being deployed in a wide range of embedded systems and throughout the Internet of Things (IoT), including the Industrial IoT (IIoT).

The Lattice sensAI Stack includes everything required to evaluate, develop, and deploy FPGA-based ML/AI solutions, including modular hardware platforms, example demonstrations, reference designs, neural network IP cores, software tools for development, and custom design services. This stack is designed to speed time to market for developers of flexible machine learning inferencing in consumer and industrial IoT applications.

The Lattice convolutional neural network (CNN) accelerator IP core is a calculation engine for deep neural networks (DNNs). The engine is optimized for convolutional neural networks, so it can be used for vision-based applications such as classification or object detection and tracking. The CNN IP core does not require an extra processor as it can perform the required calculations.

Meanwhile, the Lattice Neural Network Compiler allows designers to take neural networks created in common development frameworks—such as TensorFlow, Caffe, and Keras—and compile them for implementation in Lattice CNN and compact CNN accelerator IP cores.

Conclusion

Optimal 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 of today's 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 AI.

As shown, Lattice Semiconductor's FPGA offerings span the low to mid-range, with a focus on low-power devices that address network problems from the edge to the cloud in the rapidly growing communications, computing, industrial, automotive, and consumer markets. Lattice also offers several design and verification tool suites that address multiple design flows, ranging from language-based design to graphical processor-based design, to designs focused on machine learning and artificial intelligence applications.

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