Lattice Propel Helps Designers Create Processor-Based Systems in Minutes

A Lattice Semiconductor White Paper.

February 2021

Learn more:

www.latticesemi.com

Contact us online:

www.latticesemi.com/contact
www.latticesemi.com/buy

TABLE OF CONTENTS

Section 1	Introduction	Page 3
Section 2	FPGA-Based Solutions	Page 3
Section 3	RISC-V	Page 5
Section 4	Lattice Propel	Page 5
Section 5	Summary	Page 7

Introduction

Almost every electronics designer and embedded systems developer has heard the term field-programmable gate array (FPGA). In the case of a physical FPGA device, designers and developers are typically aware that it contains a programmable fabric that can be configured to perform any desired function, but this may be the extent of their knowledge. Similarly, when it comes to creating a design that can be implemented in the FPGA, they may have heard terms like hardware description language (HDL) and register transfer level (RTL), but may not fully comprehend what these terms mean.

An HDL, like Verilog or VHDL, allows an FPGA designer to capture the design's intent in the same way that a software developer would use a programming language, like C or C++, to capture their code. One way to think about an HDL is that it can describe things that happen concurrently (at the same time), which is the way things work in hardware in the real world. By comparison, software programming languages are typically used to describe things that happen sequentially (one after the other).

Meanwhile, RTL refers to a level of abstraction that is commonly employed as an input to a logic synthesis engine. This tool takes the RTL and converts it into the network of logical elements and interconnects that will be implemented in the FPGA's programmable fabric. The logic synthesis engine may be contrasted to a software developer's compiler, which takes a high-level program as input and converts it into the machine code that will be executed by the processor.

The programmable fabric inside an FPGA can be used to create hardware accelerators that perform data processing tasks while consuming little power. This programmable fabric can also be configured to create one or more "soft" processors that are better suited to decision-making tasks, and that can be used to control the hardware accelerators, including feeding them data and subsequently taking actions based on the results.

One such processor is the RISC-V, which is an open standard instruction set architecture (ISA) available under open source licenses that do not require use fees. The creators of the RISC-V hardware concept were inspired by the success of Linux open source software. A big advantage of RISC-V is software compatibility across multiple implementations, and there is currently exponential growth in the use of these processors.

However, one problem for non-FPGA designers who wish to use such a processor is lack of expertise with regard to FPGA design languages, tools, and flows. In order to address this, Lattice Semiconductor, a leading provider of low-power programmable FPGAs, has created a tool called Lattice Propel™, a state-of-the-art graphical user interface (GUI)-based design environment that allows any user (with or without FPGA expertise) to employ a drag-and-drop methodology to design RISC-V processor-based systems in minutes.

The output from Propel is an RTL file that can be passed to the synthesis engine, which generates the configuration file to be loaded into the FPGA, after which software developers can run their RISC-V executable files on this FPGA-based RISC-V implementation as they would any other RISC-V processor. Lattice offers royalty-free RISC-V cores to users of its FPGAs.

FPGA-Based Solutions

Data processing requires appropriate computational engines. There are a variety of different options available to developers, including microprocessors (MPUs), microcontrollers (MCUs), graphics processing units (GPUs), FPGAs, and System-on-Chip (SoC) devices.

MPUs and MCUs are efficient when it comes to decision-making tasks, but they can be inefficient in terms of time and power consumption when it comes to implementing raw data processing algorithms. SoCs offer the highest performance with the lowest power consumption, but they are expensive, resource-intensive, time-consuming to develop, and any algorithms that are implemented in the fabric of the chip are essentially "frozen in silicon," which becomes problematical in the case of systems employing evolving protocols and standards.

Certain data processing tasks, including many AI/ML algorithms, can benefit from parallel processing. The programmable fabric in an FPGA (Figure 1a) can be configured to implement hardware accelerator (HA) functions that perform such tasks in a massively parallel fashion (Figure 1b), thereby dramatically increasing performance while -- at the same time -- reducing power consumption.

In many cases, data co-processing is required to augment the hardware accelerators with a central processing unit (CPU) that can perform high-level decision-making and control functions. As opposed to a hard-core CPU that is implemented directly in the silicon, some of the FPGA's programmable fabric can be used to implement a soft-core CPU along with an associated bus structure (address, data, control) and any required peripheral intellectual property (IP) functions (Figure 1c).

Figure 1. A structured bus approach is applicable to a wide variety of applications, including embedded vision, security, and artificial intelligence.

Note that additional programmable logic functions and peripheral communications functions (e.g., USB, MIPI, I2C, SPI, CAN, and UART) may be implemented as hard and/or soft cores depending on the FPGA and the users' requirements, but these aren't shown here for simplicity.

There are multiple advantages to using a soft-core CPU, including the ability to configure the operation of the processor and many of its optional functions -- such as a dynamic memory access (DMA) controller, for example -- thereby tailoring it to efficiently meet the requirements of the target application. Furthermore, if required, some of the programmable fabric can be configured to implement additional peripheral intellectual property (IP) functions. In the case of AI, for example, some of the programmable fabric can be employed to create a simple artificial neural network for use with tasks like inferencing.

RISC-V

As was previously noted, RISC-V is an open standard ISA based on established reduced instruction set computer (RISC) principles that is made available under open source licenses. Furthermore, a number of companies are offering RISC-V hardware or open source operating systems with RISC-V support, and the instruction set is supported in several popular software toolchains.

RISC-V has a modular design consisting of alternative base parts with added optional extensions. The ISA base and its extensions have been developed in a collective effort between industry, the research community, and educational institutions. The base specifies instructions (and their encoding), control flow, registers (and their sizes), memory and addressing, logic (i.e., integer) manipulation, and ancillaries. The base alone can implement a general-purpose computer, with full software support, including a general-purpose compiler.

Additional capabilities are specified as optional extensions, thereby giving designers the flexibility to pick and choose what they require for their applications. RISC-V defines a number of extensions -- including A (Atomic), F (Single-precision floating-point), D (Double-precision floating-point), Q (Quad-precision floating -point), and C (Compressed 16-bit instructions to reduce code size in memory-limited systems) -- all of which are optional.

Figure 2. Lattice is the first flash and SRAM-based FPGA supplier to support RISC-V.

A huge advantage of a soft-core FPGA-based RISC-V implementation -- as compared to a hard-core realization as a dedicated processor -- is that the FPGA has the potential to be reconfigured to accommodate various extensions as required.

The Lattice RISC-V soft IP suite features a 32-bit RISC-V processor core and optional Timer and Programmable Interrupt Controller (PIC) submodules. The CPU core supports the RV32I instruction set, external interrupt, and JTAG IEEE 1149.1-compliant debugging.

The Timer submodule is a 64-bit real time counter, which compares a real-time register to another register to assert the timer interrupt. The PIC submodule aggregates up to eight external interrupt inputs into one interrupt. The submodule registers are accessed by the processor core using an industry-standard 32-bit AHB-L bus interface.

Lattice Propel

Many designers of embedded systems are interested in using FPGAs but are intimidated by the thought of using traditional FPGA design tools and HDLs. To address this, Lattice Propel uses a (GUI)-based design environment that allows any user (with or without FPGA expertise) to capture and configure a RISC-V processor-based design in minutes using a drag-and-drop design methodology.

The output from Propel is an RTL file in Verilog HDL that can be passed to the synthesis engine, which generates the configuration file to be loaded into the FPGA. This configuration file can be targeted at the Lattice CrossLink™-NX (targeting embedded vision applications), Certus™-NX (a general-purpose FPGA), and the MachXO3D™ and Mach™-NX FPGA families (for secure system control). Once the FPGA has been configured, software developers can run their RISC-V executable files on this FPGA-based RISC-V implementation as for any other RISC-V processor.

Lattice Propel has two components. First is the Lattice Propel Builder, which provides the graphical drag-and-drop interface that allows users to select IP blocks and connect them together. These IP blocks include the RISC-V processor (with optional timer and interrupt controller), AMBA bus structures, interfaces, memory, input/output (I/O), etc. Upgrades to existing IP blocks and new IP blocks are easily accessible online.

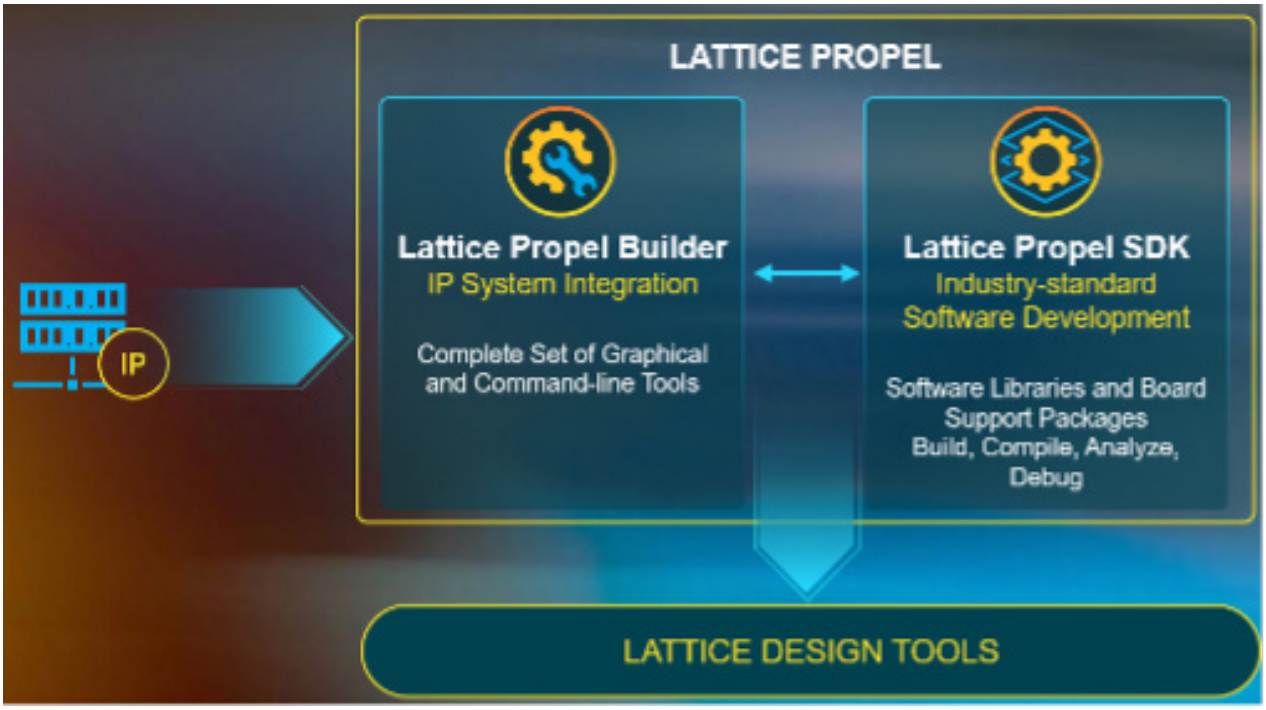

Figure 3. Propel is intuitive, easy to use, and extremely powerful.

In addition to drag-and-drop instantiation, the Propel Builder features automated pin-to-pin connections, wizard-guided configuration and parameterization, and correct-by-construction IP integration.

The second aspect is a seamless software development environment in the form of the Lattice Propel SDK (software development kit). It features an industry-standard integrated development environment (IDE) and toolchain. The SDK also offers software/hardware debugging capabilities along with software libraries and board support packages (BSPs) for Propel Builder-defined systems.

One important point to note is that, although it feeds downstream tools, Propel is a standalone utility -- only 0.5 GB in size -- that can be quickly and easily downloaded and installed. Also of interest is the fact that all of Propel's commands are Tcl scriptable, thereby facilitating automation and tight integration into users' own design environments.

Figure 4. Propel allows users to achieve success in minutes, from simple "Hello World" type applications to complex embedded control and data processing systems

In the case of hardware design, Propel is ideal for teams that require the advantages of FPGAs but don't have FPGA hardware design experience. Having said this, if any members of the team do have FPGA experience, then they can exert finer control if they so desire. In the case of software design, Propel offers an industry standard C/C++ development environment. As far as software developers are concerned, it appears as though they are working with an off-the-shelf microcontroller.

Using Propel, designers can quickly and easily generate soft RISC-V-based processor systems to be implemented in the programmable fabric of CrossLink-NX, Certus-NX, and Mach-NX FPGAs, thereby providing sophisticated video processing, system control, and system security capabilities with much less latency than can be achieved by means of an external stand-alone processor.

Summary

In addition to logic functions and hardware accelerators that perform data processing tasks with high performance while consuming little power, the programmable fabric inside an FPGA can be used to implement one or more "soft" processors that are more suited to decision-making tasks, and can be used to control functions like hardware accelerators, including feeding them data and subsequently taking actions based on the results.

One such processor is the RISC-V, which is an open standard instruction set architecture (ISA) made available under open source licenses that do not require fees to use. Lattice is the first flash and SRAM-based FPGA vendor to support RISC-V and to make royalty-free RISC-V cores available to its users.

One problem for non-FPGA designers who wish to use such a processor is lack of expertise with regard to FPGA design languages, tools, and flows. In order to address this, Lattice provides the Propel design environment to enable a simple design process that developers can use to build a RISC-V processor-based system in minutes.

.

Learn more:

www.latticesemi.com

Contact us online:

www.latticesemi.com/contact
www.latticesemi.com/buy