SLX Release Notes 2020.4-sp1

SILEXICA

Destination Control Statement
All technical data contained in this publication is subject to the export control laws of the German Federal Republic. Disclosure to nationals of other countries contrary to Germany law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer
SILEXICA GMBH, MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks
Silexica and certain Silexica product names are trademarks of Silexica GmbH. All other product or company names may be trademarks of their respective owners.

Third-Party Links
Any links to third-party websites included in this document are for your convenience only. Silexica does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Third-Party Open-source Software
SLX contains third-party open-source software components. The full list of third-party software components and their licenes can be found in the `data/third-party-licenses` sub-directory of the **SLX** installation and online at https://www.silexica.com/tps/.

Silexica GmbH
Lichtstr. 25
50825 Cologne, Germany
www.silexica.com

Contents

Preface

Typographic Conventions

This manual uses specific typographic conventions. The following table summarizes how font styles are used to emphasize important elements throughout the text.

Style	Usage
Bold	Names of Silexica products, such as **SLX for FPGA**
`Typewriter`	Literal input, e.g., user input from the command line and actual code listings
Slanted	To introduce terminology used in the Silexica (SLX) series of products

Special symbols are used to denote useful information, a remark, a warning, an expected failure or a requirement for user interaction. They are used within a shaded box with rounded corners.

Symbol	Meaning
	Useful tip
	A noteworthy point or remark
	A warning on tool-specific behavior
	An expected error or failure
	Required input
	Required selection of choices

1

New Features

This document gives an overview of the new features of **SLX 2020.4** and **2020.4-sp1**, as well as information on migrating from previous releases. For in-depth details on the new features mentioned below, please refer to our User Guides (**SLX FPGA User Guide** and **SLX C/C++ User Guide**). For information on the previous release of **SLX**, click here for the release notes for the **Previous Release of SLX**.

1.1 New Features of SLX 2020.4-sp1

- **SLX FPGA** now supports the Versal ACAP platform, the Adaptive Compute Acceleration Platform from Xilinx.
- A new project importer is available for importing **Xilinx Vitis HLS** projects into **SLX FPGA**.
- Two examples are included with **SLX 2020.4-sp1** that demonstrate the loop interchange capabilities of **SLX FPGA** and the **SLX Plugin**: a Dimension Reduction example and a BOTDA (Brillouin Optical Time Domain Analyzer) example. Visit the chapter on **Provided Examples** in the **SLX FPGA User Guide** for more details.
- Support has been incorporated for array reshape on any array dimension; this is only enabled when no interface constraints are set.

1.2 New Features of SLX 2020.4

- Support has been added for the Vivado flow of **Vitis HLS**. The new Vitis HLS compiler becomes the new HLS compiler in **SLX**, replacing Vivado HLS when Vitis HLS 2020.2 is selected in **SLX** Xilinx Tools configuration.

- **SLX 2020.4** adds support for the **SLX Plugin**, which enables the addition of new pragmas and compiler optimizations when designing for FPGAs using high-level synthesis. Key features of the plugin include:
  - Loop Interchange pragmas, which enables reordering of nested loops to remove dependencies that prevent improved parallelism, pipelining, and memory access regularity.
  - IR Design Optimizations that improve an HLS design's internal program representation before the RTL code is generated, helping to boost Vitis HLS' capabilities for automatic relocation of loop invariant memory accesses.
  - Transformation checks that are enabled for new pragmas and IR design optimizations preventing compiler transformation from being applied when not safe or legal.

  The Loop Interchange capability is only available when Vitis HLS 2020.2 has been installed alongside **SLX 2020.4** or above on Ubuntu 18.04.

- **SLX 2020.4** introduces the ability to understand the execution costs from an FPGA implementation point of view using estimation models to reveal how each function and loop contributes to overall latency of the design. The **Call Graph (formerly SW Call Graph)** now display the total cost for each function for the FPGA implementation. The display can be toggled between the FPGA and the CPU values. The SLX Hints View reveals total costs down to the loop level.

- A new interactive **SLX FPGA** flow guidance walkthrough is provided in the *Welcome Page*. The walkthrough can be used for any imported or newly created **SLX FPGA** project and guides the user through the different steps of a typical **SLX FPGA** design flow.

- Support has been incorporated for multidimensional array partitioning; **SLX** can now identify which array dimension would lead to the best performance when partitioned.

2

Compatibility Information

This release of **SLX** has file formats containing profiling and analysis results that may be incompatible with previous releases of **SLX**. If you are experiencing difficulties migrating an existing project that was created with an earlier version of **SLX**, perform a *Clean* of the project and try again.

☞ To view a full list of limitations of **SLX**, please visit the *Known Issues and Limitations* chapter of the **SLX FPGA User Guide**.