XIPHERA

PEACE OF MIND IN A DANGEROUS WORLD

XIP4003C: X25519 AND Ed25519

Curve25519 Key Exchange and Digital Signature IP Core

Product Brief
ver. 1.0
September 20, 2023

sales@xiphera.com

Introduction

XIP4003C from Xiphera is a very compact Intellectual Property (IP) core designed for efficient X25519 key exchange and Ed25519-based Edwards-curve Digital Signature Algorithm (EdDSA). XIP4003C implements arithmetic on Curve25519[1] [3], and provides a security level of 128 bits. Curve25519 is used in numerous contemporary security protocols and applications, including TLS 1.3.

XIP4003C has been designed for easy integration with FPGA- and ASIC-based designs in a vendor-agnostic design methodology, and the functionality of XIP4003C does not rely on any FPGA manufacturer-specific features.

Key Features

- **Minimal Resource Requirements:** The entire XIP4003C requires 43213 logic elements (MAX® 10) and uses only 1-2 multipliers/DSP Blocks[2] and 1-2 internal memory block in a typical Intel® FPGA implementation.
- **Constant Latency:** The execution time of XIP4003C is independent of the key value, and consequently provides protection against timing-based side-channel attacks.
- **Performance:** Despite its small size, XIP4003C can support more than 100 key exchange or digital signature operations per second.
- **Standard Compliance:** XIP4003C is compliant with RFC7748 [3], RFC8032 [2], and the draft vesion of FIPS 186-5 [1]. XIP4003C can be used as a part of many public-key protocols including IKEv2 (RFC 8031) [4] and TLS 1.3 (RFC 8446) [5].

[1] Curve25519 is formally defined as $y^2 = x^3 + 486662x^2 + x$ over the finite field defined by the prime number $2^{255} - 19$.

[2] The exact number depends on the targeted Intel® FPGA architecture

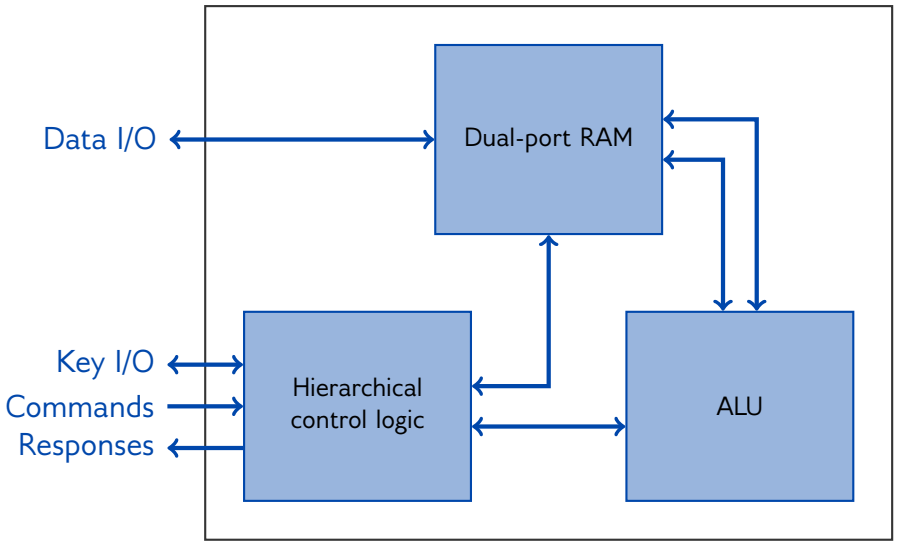

Figure 1: Internal high-level block diagram of XIP4003C

Functionality

XIP4003C supports the following operations:

- Constant-time x-coordinate-only scalar multiplication (for X25519) with the Montgomery ladder algorithm
- Constant-time fixed-base scalar multiplication (for Ed25519 signature generation)
- Double-base scalar multiplication (for Ed25519 signature verification)
- Point compression/decompression
- Other modular arithmetic for generating and verifying Ed25519 signatures

The internal word width is set to 17 bits, as this leads to an efficient internal implementation[3] of the multiplication algorithm. The external bus widths for `din` and `dout` (See also Figure 2) are set to 16 bits.

Block Diagram

The internal high-level block diagram of XIP4003C is depicted in Figure 1.

Interfaces

The external interfaces of XIP4003C are depicted in Figure 2.

This Product Brief describes a high-level overview of the functionality and capabilities of XIP4003C. Please contact sales@xiphera.com for a complete datasheet with a detailed description of the input and output signals, startup procedure of XIP4003C, example simulation waveforms, and the FPGA resource requirements of your targeted FPGA family.

[3] $15 \cdot 17\ bits = 255\ bits$.

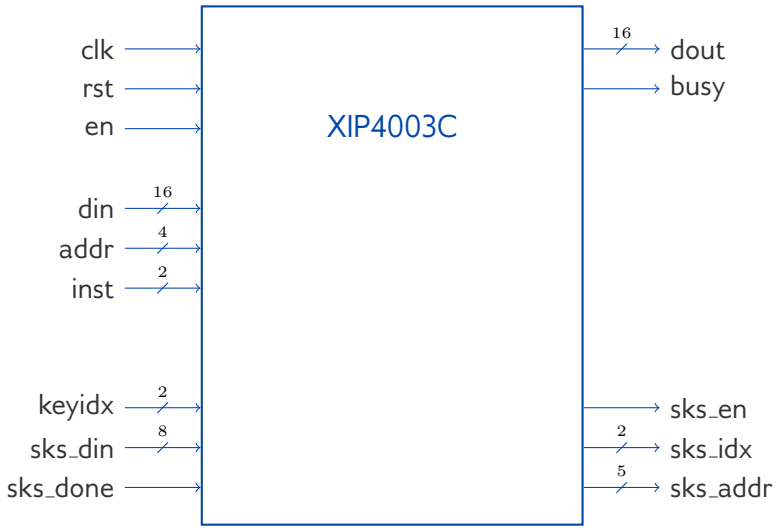

Figure 2: External interfaces of XIP4003C

FPGA Resources and Performance

Table 1 presents the Intel® FPGA resource requirements for representative low-cost Intel® FPGA families. On request, the resource estimates can also be supplied for other Intel® FPGA families.

Device	Resources	f_{MAX}
Intel® MAX® 10*	43213 LE , 2 Mult. (9bit)	83.58 MHz
Intel® Cyclone® 10 LP*	2957 LE, 2 M9K, 2 Mult. (9bit)	133.76 MHz
Intel® Cyclone® V SX SoC*	1065 ALM, 2 M10K, 1 DSP	188.43 MHz

Table 1: Resource usage and performance of XIP4003C on representative Intel® FPGA families.

Example Use Cases

XIP4003C can be used in combination with other Xiphera IP cores to design an FPGA-based security solution. Possible use cases include:

- Using the TRNG IP core XIP8001B to supply the required number of random bits for secret key derivation by the HKDF IP core XIP3322B, whose result will be used by XIP4003C.
- Using XIP4003C to exchange the 128-bit secret key required for the AES-GCM -based communication with Xiphera IP cores XIP111B or XIP1111H.
- Using XIP4003C with XIP3027C in digital signature generation and verification

The setup described above is also depicted in Figure 3.

XIP4003C can also be used to offload microcontroller / -processor based designs, if a software-based implementation of Curve25519 arithmetic is too slow.

*Quartus® Prime Lite 20.1.1, default compilation settings, industrial speedgrade.

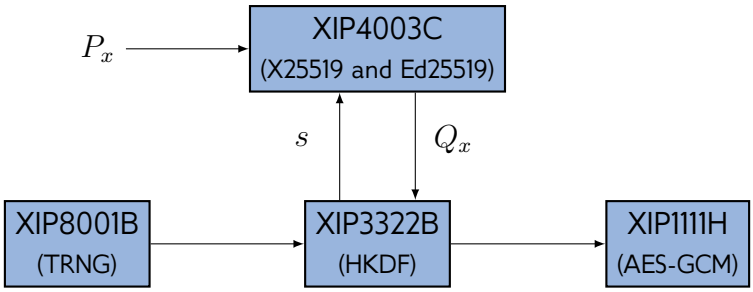

Figure 3: Using XIP4003C with XIP8001B, XIP3322B, and XIP1111H

Ordering and Deliverables

Please contact sales@xiphera.com for pricing and your preferred delivery method. XIP4003C can be shipped in a number of formats, including netlist, source code, or encrypted source code. Additionally, synthesis scripts, a comprehensive testbench, and a detailed datasheet including an integration guide are included.

Export Control

Xiphera specializes in secure and efficient implementations of standardized cryptographic algorithms on Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs). Our fully in-house designed product portfolio includes individual cryptographic Intellectual Property (IP) cores, as well as comprehensive security solutions built from a combination of individual IP cores.

Xiphera is a Finnish company operating under the laws of the Republic of Finland, and is fully owned by Finnish citizens and institutional investors.

Contact

Xiphera Oy
Tekniikantie 12
FIN-02150 Espoo
Finland
sales@xiphera.com
+358 20 730 5252

References

[1] NIST Computer Security Division. FIPS PUB 186-5 (Draft) Digital Signature Standard (DSS). Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States, 2019.

[2] Simon Josefsson and Ilari Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017.

[3] Adam Langley, Mike Hamburg, and Sean Turner. Elliptic Curves for Security. RFC 7748, January 2016.

[4] Yoav Nir and Simon Josefsson. Curve25519 and Curve448 for the Internet Key Exchange Protocol Version 2 (IKEv2) Key Agreement. RFC 8031, December 2016.

[5] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018.