XIPHERA

PEACE OF MIND IN A DANGEROUS WORLD

XIP4001C: X25519

Curve25519 Key Exchange IP Core

Product Brief
ver. 1.0
September 20, 2023

sales@xiphera.com

Introduction

XIP4001C from Xiphera is a very compact Intellectual Property (IP) core designed for efficient key exchange using the X25519 protocol. XIP4001C implements arithmetic on Curve25519[1] [1], and provides a security level of 128 bits. Curve25519 is used in numerous contemporary security protocols and applications, including TLS 1.3.

XIP4001C has been designed for easy integration with FPGA- and ASIC-based designs in a vendor-agnostic design methodology, and the functionality of XIP4001C does not rely on any FPGA manufacturer-specific features.

Key Features

- **Minimal Resource Requirements:** The entire XIP4001C requires less than 1k Logic Elements and uses only 1-2 multipliers/DSP Blocks[2] and one internal memory block in a typical FPGA implementation.
- **Constant Latency:** The execution time of XIP4001C is independent of the key value, and consequently provides protection against timing-based side-channel attacks.
- **Performance:** Despite its small size, XIP4001C can support more than 100 key exchange operations per second.
- **Standard Compliance:** XIP4001C is compliant with RFC7748 [1], and can be used as a part of many public-key protocols including IKEv2 (RFC 8031) [3] and TLS 1.3 (RFC 8446) [4].

[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 FPGA architecture

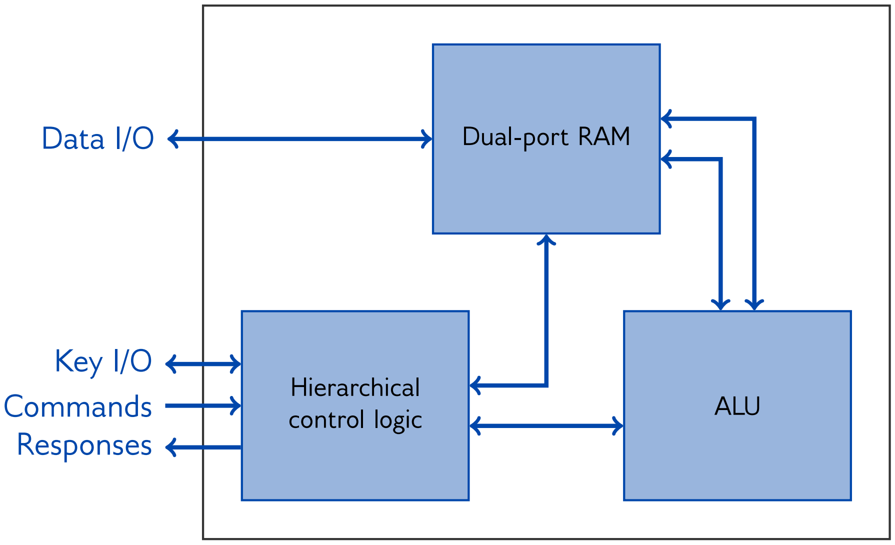

Figure 1: Internal high-level block diagram of XIP4001C

Functionality

XIP4001C calculates the operation $Q_x = sP_x$ using the Montgomery Ladder Algorithm [2], where

- P_x is the 255 bits long input argument
- s is the secret key (32 bytes long)
- Q_x is the 255 bits long point multiplication result

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

Block Diagram

The internal high-level block diagram of XIP4001C is depicted in Figure 1.

Interfaces

The external interfaces of XIP4001C are depicted in Figure 2.

This Product Brief describes a high-level overview of the functionality and capabilities of XIP4001C. Please contact sales@xiphera.com for a complete datasheet with a detailed description of the input and output signals, startup procedure of XIP4001C, example simulation waveforms, and the FPGA resource requirements of your targeted FPGA family.

FPGA Resources and Performance

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

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

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

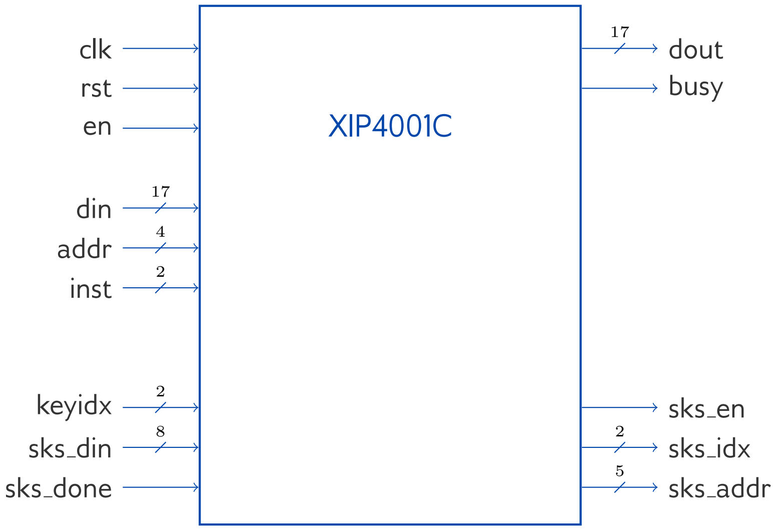

Figure 2: External interfaces of XIP4001C

Device	Resources	f_{MAX}
Intel® Cyclone® V SX SoC*	324 ALM, 1 M10K, 1 DSP	186.50 MHz
Intel® Cyclone® 10 LP*	720 LE, 1 M9K, 2 Mult. (9bit)	155.33 MHz
Intel® MAX® 10*	31272 LE , 2 Mult. (9bit)	69.11 MHz
Xilinx® Zynq-7000® †	475 LUT, 1 RAMB18, 1 DSP	191.53 MHz
Xilinx® Artix-7® †	474 LUT, 1 RAMB18, 1 DSP	220.17 MHz
Xilinx® Spartan-7® †	474 LUT, 1 RAMB18, 1 DSP	213.63 MHz
Lattice® ECP5® ‡	1003 LUT4, 1 EBR, 1 MULT18	163.43 MHz
Microchip® PolarFire® §	1389 4LUT, 1 LSRAM, 1 Math	127.55 MHz

Table 1: Resource usage and performance of XIP4001C on representative FPGA families.

Example Use Cases

XIP4001C 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 XIP4001C.
- Using XIP4001C to exchange the 128 bits long secret key required for the AES-GCM -based communication with Xiphera IP cores XIP111B or XIP1111H.

The setup described above is also depicted in Figure 3.

If EdDSA (Edwards-curve Digital Signature Algorithm) digital signature verification is also required, the extended functionality offered by Xiphera IP Core XIP4003C is recommended.

XIP4001C can also be used to offload microcontroller / -processor based designs, if a software-based implementation of Curve25519 arithmetic is too slow.

† Vivado 2020.2, default compilation settings, industrial speedgrade.
‡ Diamond 3.12.0, default compilation settings, synthesised with Synplify.
§ Libero 2022.1.0.10, default compilation settings, industrial speedgrade.

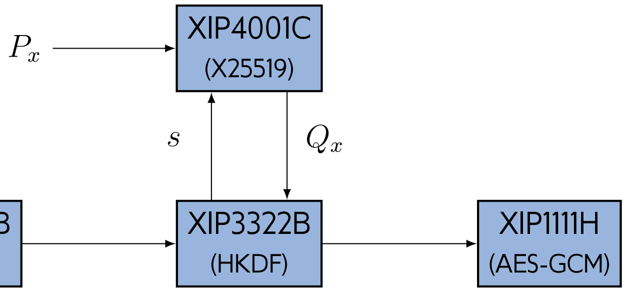

Figure 3: Using XIP4001C with XIP8001B, XIP3322B, and XIP1111H

Ordering and Deliverables

Please contact sales@xiphera.com for pricing and your preferred delivery method. XIP4001C 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.

About Xiphera

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

[2] Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48:243–264, 1987.

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

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