International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1032-1041

Design and Analysis of Digital Signature Accelerators Using Cryptography Based on Ed25519 on FPGA

Mrs.N.Swarupa Rani [(1)], Dr.P.Prasanna Murali Krishna [(2)], D Khadar Vali [(3),] Makkela Veeranjaneyulu [(4),] Vangepurapu Sandeep [(5),] Thirumala Reddy Subhash Reddy [(6)]
[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*
[3,4,5,6,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract. In this paper,The Ed25519 digital signature algorithm (Edwards curve digital signature algorithm, or EdDSA) is presented in extremely optimized versions. Comparing this approach to exiting digital signature algorithms, it greatly reduces execution time without compromising security. While EdDSA is used in many popular protocols, like SSH and TLS, there don't seem to be many hardware implementations that specialize in EdDSA. Considering this, we suggest two distinct field-programmable gate array (FPGA)-based EdDSA implementations, namely, effective, and powerful Ed25519 designs suitable for a security level equivalent to AES-128. By lowering the needed space, our efficient Ed25519 system achieves an improvement over 84% over the best recent work. Additionally, it has a speedup of more than 8×. Additionally, our suggested high-performance architecture demonstrates a 21× speedup on a Xilinx Vivado FPGA, processing over 6200 digital signature algorithms per second, indicating a noteworthy gain in terms of utilized space × time. Ultimately, our suggested solutions incorporate side-channel countermeasures that are both effective and better than those of the past..

Keywords: Digital Signature Analaysis, FPGA, Verilog HDL, Eds25519,AES-128.

I.INTRODUCTION

EDWARDS curve digital signature algorithm (EdDSA) developed by Bernstein et al. [1] has gained prominent attention among the existing digital signature algorithms due to its fast operations without affecting the required security. The Ed25519, as the most popular instance of EdDSA, is widely used as a digital signature method to guarantee the validity of the communications. On the other hand, the elliptic curve digital signature algorithm (ECDSA) is no longer suitable for embedded devices due to its vulnerability against side-channel analysis (SCA) attacks [2], [3]. Hence, most HTTPS websites are switching to Ed25519, suitable for higher level security requirements, which address some backdoor issues [4] in other ECDSA constructions at the same time.

Although most current cryptosystems will be broken by quantum computing based on Shor's algorithm [5], the transition to postquantum cryptography (PQC) includes an emerging field called hybrid systems [6], requiring both classic and PQC [7]. Hence, designing high security ECC-based digital signature for different applications is crucial. EdDSA is notable for high speed and constant-time implementations and was quickly implemented as a part of the TLS and OpenSSH protocols [8]. Hence, it has to be implemented in various platforms subject

to the performance requirement of the target application, such as constrained IoT devices. However, EdDSA has not got sufficient study, especially in the field of hardware implementation based on field-programmable gate arrays (FPGAs). Therefore, investigation of the hardware implementation of this algorithm is required considering the advantages of FPGA-based designs to exploit parallelism, which leads to improvements in the efficiency of the overall system.

There are two main solutions to enable the hardware-based digital signature algorithm in the constrained IoT, including: 1) HW/SW approach to cope with embedded constraints and 2) pure HW method that includes all in hardware instructions. The HW/SW method makes the design smaller, slower, and more controllable/programmable compared to pure HW schemes. Although the pure HW approach leads to better performance, HW/SW can be a better choice for IoTs since it provides flexibility to switch security levels based on performance targets. In [9], the comparison of the CryptoCell API over nRF52840 as an internal HW/SW solution and the external cryptochip ATECC608A as a pure HW is thoroughly studied. Furthermore, to address higher security needs, new NIST and IETF recommendations make Curve448 suitable for higher level security requirements [10], [11]. Hence, implementing HW/SW architecture brings the required flexibility among different security levels, while a general architecture can be implemented in HW and controlled by instruction set processors such that the hardware remains largely flexible, which is beyond the scope of this work.

2.LITERATURE

As one of the first FPGA-based works in ECC-based digital signature, Glas et al. [12] proposed architecture for 128-bit security to integrate into a vehicle-to-vehicle communication system. Furthermore, Panjwani [13] presented a scalable hardware implementation in prime fields over NIST recommended field sizes up to 521 bit, employing hardware– software codesign approach. The work of Vliegen et al. [14] introduced a compact core over the NIST P-256 curve resistant against simple power analysis (SPA) attacks. Moreover, Zhang and Bai [15] proposed a core with a security level 128 bit over the SM2 curve.

Recently, a number of hardware implementations have been introduced to implement an elliptic curve point multiplication (ECPM) core over Curve25519. Sasdrich and Güneysu [16] proposed the first Curve25519 implementation using a DSP-based single-core architecture. This work has been extended by adding side-channel countermeasures in [17] and [18] to provide an evaluation against common physical attacks. In [19], fast and compact implementations of ECPM were proposed. This architecture employs a semisystolic bit-serial multiplier and carry-compact addition to provide a high-performance architecture. The work of Koppermann et al. [20], [21] presented a high-speed prime field multiplier with a latency of 92 μs for a point multiplication. In addition, in [22], a low-latency ECPM was proposed employing a pipelined arithmetic architecture on FPGA and ASIC platforms. It should be noted that FPGA implementations of Curve25519 in the literature cannot be directly compared to ours because the ECPM core in EdDSA occupies more resources for implementing hash core and module L reduction.

https://ijgst.com.2024.v13.i2.pp1032-1041

Furthermore, it requires more time for a point multiplication since this architecture is reused for nonmodular multiplication and module L reduction.

A non-DSP-based Ed25519 point multiplication core was presented by Mehrabi and Doche [23] using the double-andadd algorithm. Hence, this architecture is a nonconstant-time core vulnerable to SPA attacks. Notably, the reported area does not include all the required modules for providing a digital signature, such as hash function and modulus L reduction. We explore that SHA-512 increases almost 25% utilized area in Ed25519. Moreover, Turan and Verbauwhede [24] proposed an Ed25519 architecture combined with the X25519 key exchange. This design targets resource-constrained devices on a Zynq SoC. Turan and Verbauwhede [24, Sec. 3.3] claimed that the cost of computing using restricted-X coordinates of a point on the Montgomery curve is more than extended coordinates on the twisted Edwards curve due to the complexity of coordinate conversion. Therefore, the core works over the twisted Edwards curve. Besides, although side channel countermeasures are considered for the ECPM core, the authors do not include a resistant SHA-512 core, allowing vulnerability against SCA, as shown in [25].

Based on the discussions, the tradeoff explorations between resource utilization and performance to implement an efficient Ed25519 implementation from different optimization perspectives have not been thoroughly studied. Particularly, designing a unified architecture consisting of physical protection against SCA in all submodules to perform secure key generation, signature generation, and signature verification is required. Besides, employing the fast and efficient Karatsuba-based multiplier for designing a highperformance Ed25519 architecture should be investigated. Eventually, the signature computation cost over the Edwards domain compared to the Montgomery domain for a highly parallel design should be investigated.

II. PROPOSED METHOD AND ITS METHODOLOGY

1.EXISTING SYSTEM

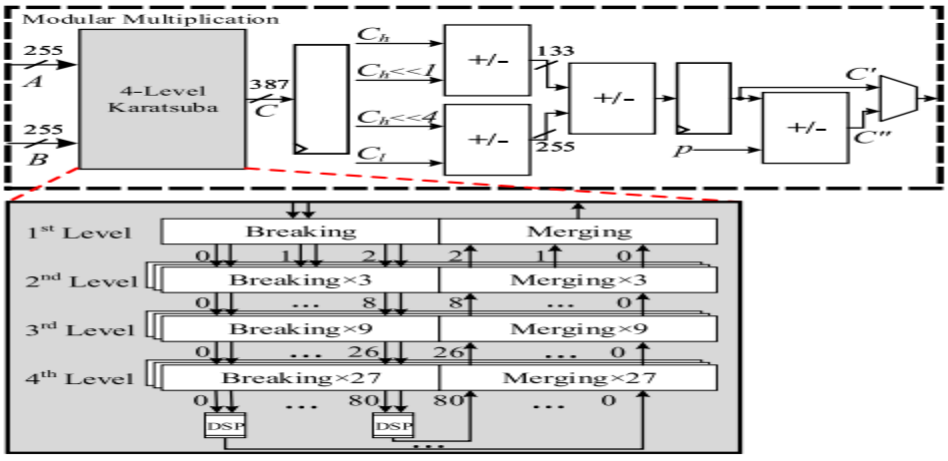

Fig. 1. Highly parallel modular multiplier in the high-performance scheme.

https://ijgst.com.2024.v13.i2.pp1032-1041

To design a high-performance Ed25519 scheme, we need to accelerate the scalar multiplication procedure as the more time-consuming part of the signature algorithm, particularly its modular multiplication unit. Hence, we design a low latency modular multiplier followed by an interleaved reduction. In this scheme, the full width of 255-bit is implemented to minimize data transition latency and maximize parallelization within the arithmetic logic unit (ALU). Therefore, loading and storing data take only one cycle to accelerate ALU throughput. Addition/subtraction between two operands is performed in 255-bit data width in one clock Moreover, the interleaved reduction is performed at the cost of one additional cycle in a pipeline fashion.

Our modular arithmetic units for the proposed high performance design are illustrated in Fig. 1. In this scheme, a 255×255-bit multiplication is decomposed to 81 16×16-bit multipliers in four consecutive levels. All partial products work in one cycle simultaneously. An addition tree is designed in a backward direction to merge the products and build the result. The pipelined multiplier has five stages, of which three are required for the multiplication and the remaining ones for the interleaved reduction in a pipeline fashion. Hence, the full five cycles are taken only for the first multiplication, and then, a 255×255-bit multiplication computation is becoming available with a latency of only one cycle.

2.PROPOSED SYSTEM

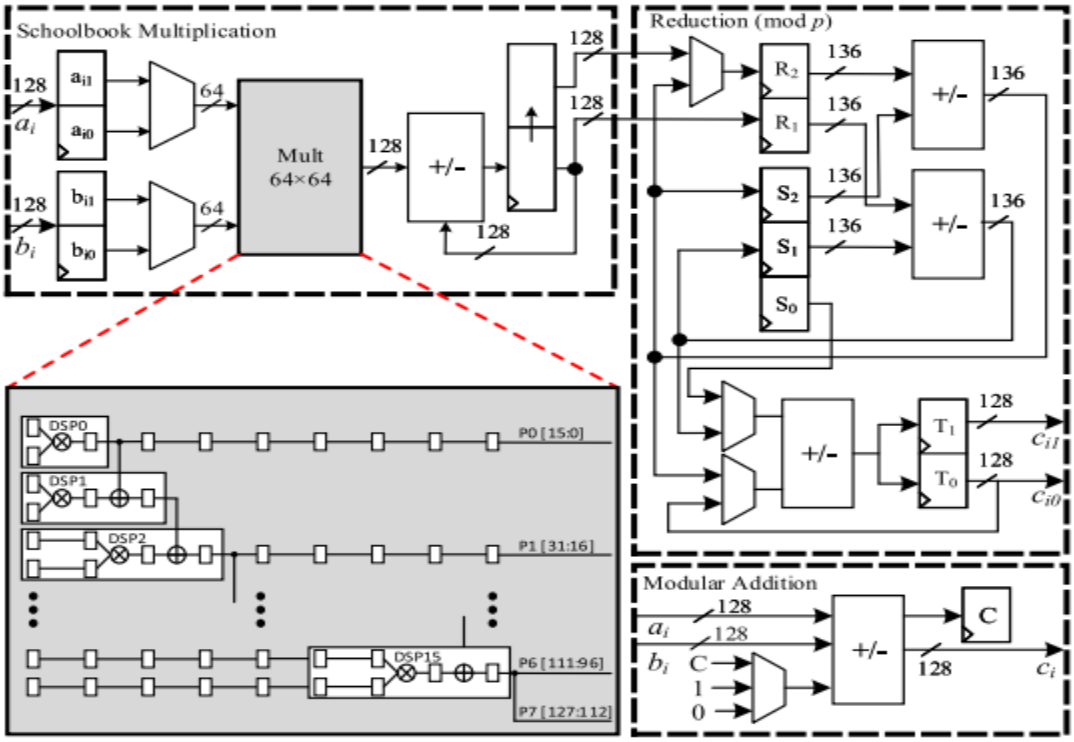

Fig2. Lower-level arithmetic operations in the proposed fully pipelined efficient Ed25519 scheme. ai and bi are read from memory unit, and ci or (ci1, ci0) is stored to memory unit.

https://ijgst.com.2024.v13.i2.pp1032-1041

Employing the Karatsuba multiplication in the first level can be also used for implementing the fast modular reduction to optimize computations. This multiplication includes two main stages: breaking inputs and merging the results. Breaking stage decomposes A to a1, a0, and a1 + a0 (B is decomposed similar to A), and the merging stage computes addition between C2 = a1b1, C0 = a0b0, and C1 = (a1+a0)·(b1+b0), where φ = 2(256/2) = 2128 in the first level. Due to the fact that 2 p ≡ 2256 −38 mod p, the merging stage in the first level of the Karatsuba multiplication can be used for the fast reduction such that C = A · B = (a12128 + a0) · (b12128 + b0) = a1b12256 + a0b0 + ((a1 + a0) · (b1+b0)−a1b1−a0b0)2128 = 38C2+C0+(C1−C2−C0)2128.

Hence, the computed C can be presented in 387 bit. Thus, the first reduction stage optimizes the obtained result width from 512 to 387 bit, which increases our expected performance. Suppose that C is presented in two parts: Cl and Ch , which are its first 255-bit and rest 132 bit such that C = Ch 2255 + Cl . Therefore, the subsequent reduction stage is applied to C such that C = 19Ch + Cl. In addition, C = C − p is computed in the case of C > p, and the output is chosen between C and C considering subtraction borrow flag.

Fig. 2 shows the lower level arithmetic operations for our proposed efficient architecture. In this architecture, we consider decreasing the required resources as the main optimization objective, while the area–time factor is simultaneously improved. Furthermore, DSP components as the critical resource in FPGA significantly affect architecture performance. Therefore, improvement of Ad × T metrics should be considered as another vital factor to describe efficiency, where Ad is the number of employed DSPs. In this scheme, the data width of 128-bit is implemented within ALU to decrease the CPD. However, in the modular reduction unit, redundant representation providing more 8 bits, i.e., 136-bit is implemented to avoid the cost of carry propagation between digits. Moreover, addition/subtraction between two operands, i.e., C = A ± B, is performed in 128-bit data width, which takes two clock cycles. Hence, the carry is propagated between digits employing a register. Furthermore, the reduction stage performs $C\overline{\overline{}} = C \mp p$ at the cost of two additional cycles. Both C and $C\overline{\overline{}}$ are stored in the memory unit, and the correct result is determined by a flag obtained from the previous carry/borrow.

1) Modular Multiplication: Modular multiplication can be computed by four 128 × 128-bit partial products, i.e., a0b0, a0b1, a1b0, and a1b1. Operands can be read from memory unit in a cycle to feed two input registers. Then, four multiplications are consecutively performed for these required products. For example, a0b0 is computed by a00b00, a00b01, a01b00, and a01b01, where a0 = a01264 + a00 and b0 = b01264 + b00.
2) The centerpiece of the modular multiplication unit is a 64×64 pipelined schoolbook multiplier implemented by 16 DSPs. The architecture of our proposed multiplication core is illustrated in Fig. 3. In order to accumulate the partial products, a 256-bit register and a 128-bit adder are designed. Thus, the partial product is accumulated with the upper half of the register. Furthermore, according to the sequence of multiplications, i.e., start from a00b00, then a00b01,

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1032-1041

a01b00, and eventually a01b01, the register is shifted downward by 64 bit before accumulating the second and fourth partial products. Thus, when the pipeline stages are full, 64 × 64-, 128 × 128-, and 255×255-bit multiplications are becoming available with a throughput of one, four, and 16 cycles, respectively.

III. RESULTS AND ANALYSIS DISCUSSION

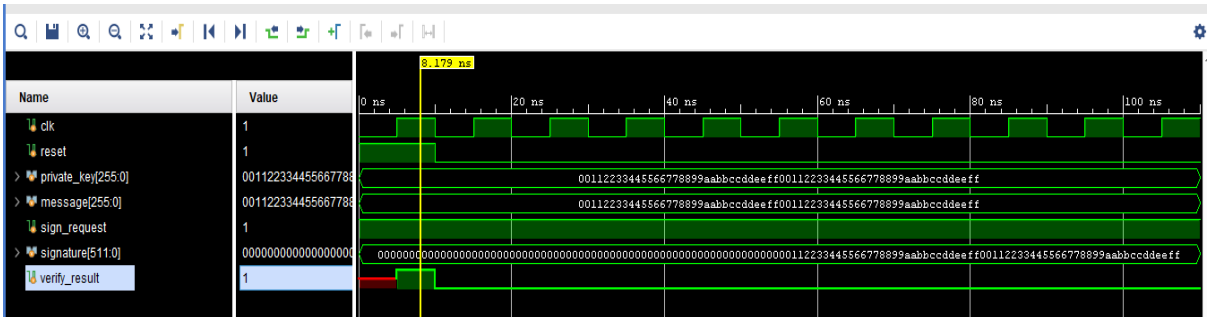

FIG1.SIMULATION OUTPUT

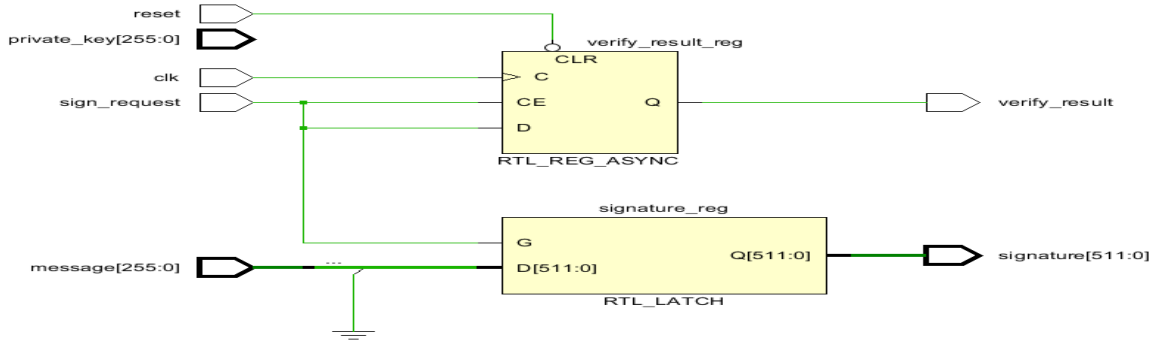

FIG2. RTL SCHEMATIC DIAGRAM

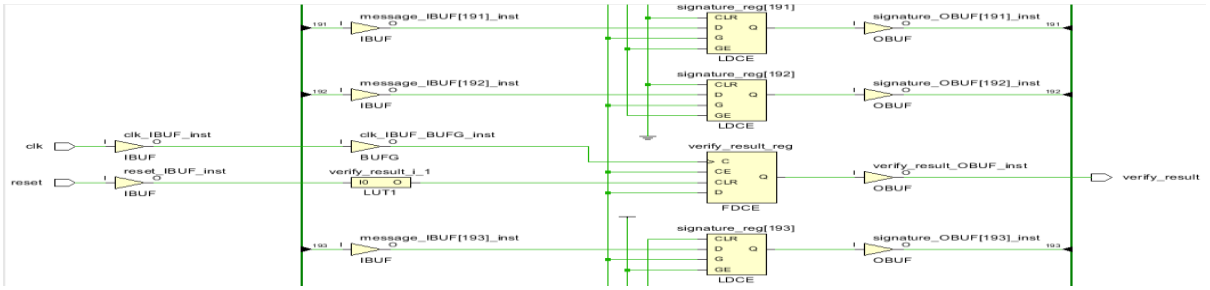

FIG3.SYNTHESIS DESIGN

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1032-1041

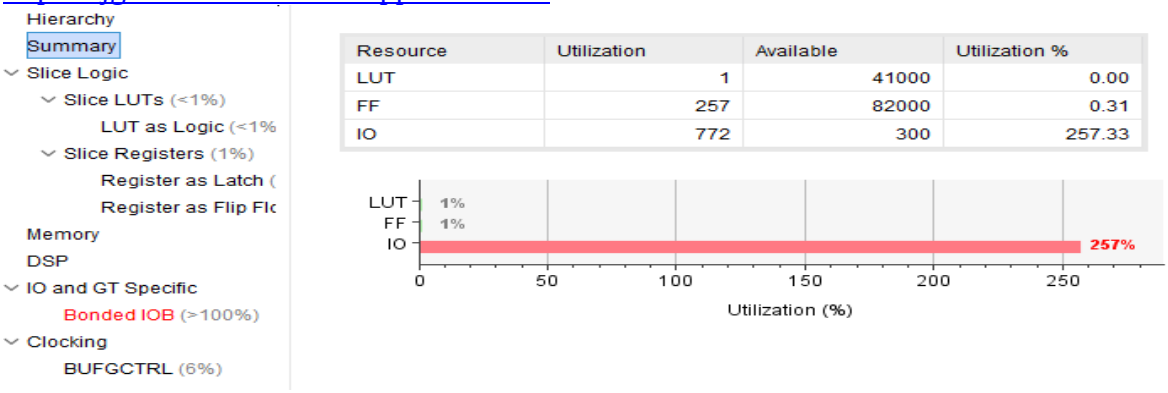

FIG4.UTILIZATION

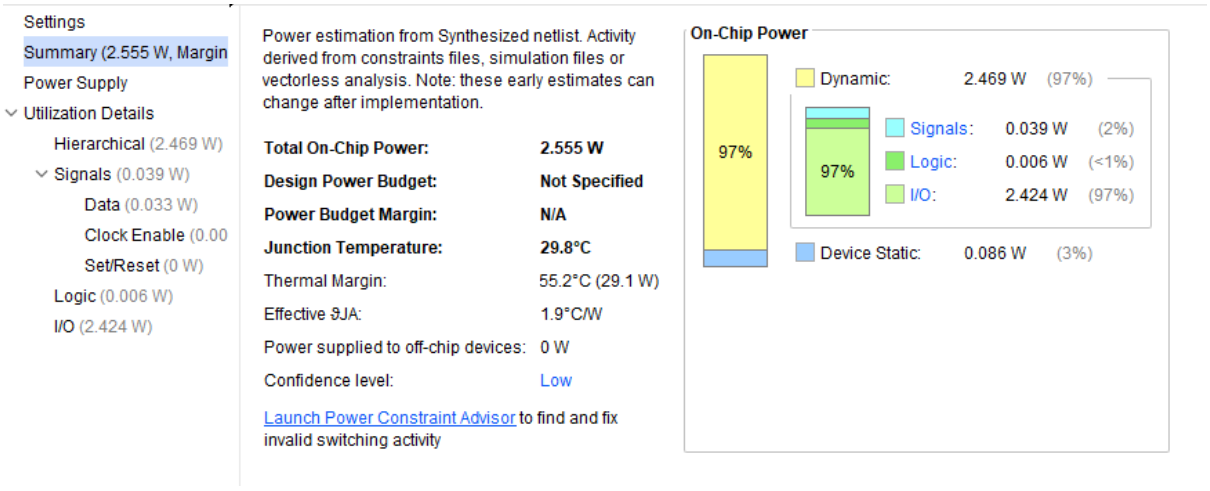

FIG5.POWER REPORT

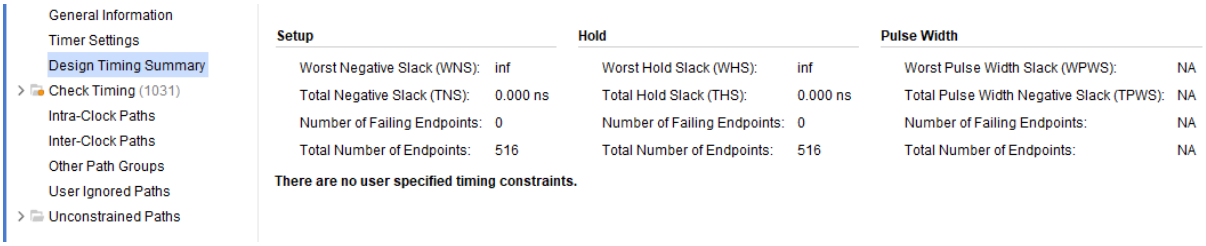

FIG6.TIMIMG REPORT

CONCLUSION

This paper, we have proposed hardware design strategies for recently proposed Edwards curve digital signatures Ed25519 on Xilinx Zynq-7020 FPGA, including advanced protection against side-channel attacks. The proposed architectures achieve above 84% efficiency improvement of the area–time product using pipelined architecture and interleaved multiplication. Our high-performance and efficient architectures compute more than 6200 and 2200 signings and 5100 and 1500 verifications per second, respectively. We also show the design can outperform

recently presented works using only moderate resource requirements.

3.REFERENCES

[1] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B. Yang, "Highspeed high-security signatures," in Proc. 13th Int. Workshop, Nara, Japan, Sep./Oct. 2011, pp. 124–142.

[2] A. C. Aldaya, C. P. García, and B. B. Brumley, "From A to Z: Projective coordinates leakage in the wild," Cryptol. ePrint Arch., Tech. Rep. 2020/432, 2020.

[3] K. Ryan, "Return of the hidden number Problem: A widespread and novel key extraction attack on ECDSA and DSA," Trans. Cryptograph. Hardw. Embedded Syst., vol. 2019, no. 1, pp. 146–168, Nov. 2018.

[4] D. J. Bernstein and T. Lange. (2011). Security Dangers of the Nist Curves. [Online]. Available: https://www.hyperelliptic.org/tanja/vortraege/20130531.pdf

[5] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proc. 35th Annu. Symp. Found. Comput. Sci., Santa Fe, NM, USA, Nov. 1994, pp. 124–134.

[6] N. Bindel, U. Herath, M. McKague, and D. Stebila, "Transitioning to a quantum-resistant public key infrastructure," in Proc. IACR, 2017, p. 460.

[7] M. Bisheh-Niasar, R. Azarderakhsh, and M. Mozaffari-Kermani, "High-speed NTT-based polynomial multiplication accelerator for CRYSTALS-Kyber post-quantum cryptograsphy," Cryptol. ePrint Arch., Tech. Rep. 2021/563, 2021.

[8] (2020). Things That Use Ed25519. [Online]. Available: https://ianix.com/pub/ed25519-deployment.html

[9] P. Kietzmann, L. Boeckmann, L. Lanzieri, T. C. Schmidt, and M. Wählisch, "A performance study of crypto-hardware in the low-end IoT," in Proc. IACR, 2021, p. 58.

[10] M. Bisheh Niasar, R. Azarderakhsh, and M. Mozaffari Kermani, "Efficient hardware implementations for elliptic curve cryptography over Curve448," in Proc. 21st Int. Conf. Cryptol. India, Bangalore, India, Dec. 2020, pp. 228–247.

[11] M. Bisheh-Niasar, R. Azarderakhsh, and M. Mozaffari-Kermani, "Areatime efficient hardware architecture for signature based on Ed448," IEEE Trans. Circuits Syst. II, Exp. Briefs, early access, Mar. 23, 2021, doi: 10.1109/TCSII.2021.3068136.

[12] B. Glas, O. Sander, V. Stuckert, K. D. Müller-Glaser, and J. Becker, "Prime field ECDSA signature processing for reconfigurable embedded systems," Int. J. Reconfigurable Comput., vol. 2011, Oct. 2011, Art. no. 836460.

[13] B. Panjwani, "Scalable and parameterized hardware implementation of elliptic curve digital signature algorithm over prime fields," in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Sep. 2017, pp. 211–218.

[14] J. Vliegen et al., "A compact FPGA-based architecture for elliptic curve cryptography over prime fields," in Proc. 21st IEEE Int. Conf. Appl.- Specific Syst., Architectures Processors, 2010, pp. 313–316.

[15] D. Zhang and G. Bai, "High-performance implementation of SM2 based on FPGA," in Proc. 8th IEEE Int. Conf. Commun. Softw. Netw. (ICCSN), Jun. 2016, pp. 718–722.

[16] P. Sasdrich and T. Güneysu, "Efficient elliptic-curve cryptography using curve25519 on reconfigurable devices," in Proc. 10th Int. Symp., D. Goehringer, M. D. Santambrogio, J. M. P. Cardoso, and K.

Bertels, Eds., Vilamoura, Portugal, 2014, pp. 25–36.
[17] P. Sasdrich and T. Güneysu, "Implementing Curve25519 for sidechannel-protected elliptic curve cryptography," ACM Trans. Reconfigurable Technol. Syst., vol. 9, no. 1, pp. 1–15, Nov. 2015.
[18] P. Sasdrich and T. Gäneysu, "Exploring RFC 7748 for hardware implementation: Curve25519 and Curve448 with side-channel protection," J. Hardw. Syst. Secur., vol. 2, no. 4, pp. 297–313, Dec. 2018.
[19] M. Bisheh Niasar, R. El Khatib, R. Azarderakhsh, and M. Mozaffari-Kermani, "Fast, small, and area-time efficient architectures for key-exchange on Curve25519," in Proc. IEEE 27th Symp. Comput. Arithmetic (ARITH), Jun. 2020, pp. 72–79.
[20] P. Koppermann, F. DeSantis, J. Heyszl, and G. Sigl, "X25519 hardware implementation for low-latency applications," in Proc. Euromicro Conf. Digit. Syst. Design, P. Kitsos, Ed., Limassol, Cyprus, 2016, pp. 99–106.
[21] P. Koppermann, F. De Santis, J. Heyszl, and G. Sigl, "Lowlatency X25519 hardware implementation: Breaking the 100 microseconds barrier," Microprocessors Microsyst., vol. 52, pp. 491–497, Jul. 2017.
[22] R. Salarifard and S. Bayat-Sarmadi, "An efficient low-latency pointmultiplication over Curve25519," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 10, pp. 3854–3862, Oct. 2019.
[23] M. A. Mehrabi and C. Doche, "Low-cost, low-power FPGA implementation of ED25519 and CURVE25519 point multiplication," Information, vol. 10, no. 9, p. 285, Sep. 2019.
[24] F. Turan and I. Verbauwhede, "Compact and flexible FPGA implementation of Ed25519 and X25519," ACM Trans. Embedded Comput. Syst., vol. 18, no. 3, pp. 1–21, 2019.
[25] N. Samwel, L. Batina, G. Bertoni, J. Daemen, and R. Susella, "Breaking Ed25519 in WolfSSL," Cryptol. ePrint Arch., Tech. Rep. 2017/985, 2017.
[26] S. Josefsson and I. Liusvaara, Edwards-Curve Digital Signature Algorithm (EdDSA), document RFC 8032, 2017, pp. 1–60.
[27] D. J. Bernstein, "Curve25519: New Diffie-Hellman speed records," in Proc. 9th Int. Conf. Theory Pract. Public-Key Cryptogr., M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds., New York, NY, USA, 2006, pp. 207–228.
[28] H. Hisil, K. K.-H. Wong, G. Carter, and E. Dawson, "Twisted edwards curves revisited," Cryptol. ePrint Arch., Tech. Rep. 2008/522, 2008.
[29] M. Hamburg, "Fast and compact elliptic-curve cryptography," in Proc. IACR, 2012, p. 309.
[30] K. Okeya and K. Sakurai, "Efficient elliptic curve cryptosystems from a scalar multiplication algorithm with recovery of the Y-coordinate on a montgomery-form elliptic curve," in Proc. Int. Workshop, Paris, France, May 2001, pp. 126–141.
[31] D. F. Aranha, F. R. Novaes, A. Takahashi, M. Tibouchi, and Y. Yarom, "Ladderleak: Breaking ECDSA with less than one bit of nonce leakage," Cryptol. ePrint Arch., Tech. Rep. 2020/615, 2020.
[32] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," in Proc. Cryptograph. Hardw. Embedded Syst., Ç. K. Koç and C. Paar, Eds., Worcester, MA, USA, 1999, pp. 292–302.
[33] P. Schwabe. (Sep. 2013). Scalar-Multiplication Algorithms. [Online]. Available:

https://ijgst.com.2024.v13.i2.pp1032-1041

https://cryptojedi.org/peter/data/eccss-20130911b.pdf

[34] M. Bisheh Niasar, R. Azarderakhsh, and M. Mozaffari Kermani, "Optimized architectures for elliptic curve cryptography over Curve448," in Proc. IACR, 2020, p. 1338.

[35] M. Scott, "On the deployment of curve-based cryptography for the Internet of Things," in Proc. IACR, 2020, p. 514.

[36] H. Fujii and D. F. Aranha, "Curve25519 for the cortex-M4 and beyond," in Proc. 5th Int. Conf. Cryptol. Inf. Secur. Latin Amer., Havana, Cuba, Sep. 2017, pp. 109–127.

[37] D. Bernstein and T. Lange. EBACS: ECRYPT Benchmarking of Cryptographic Systems. Accessed: Mar. 22, 2021. [Online]. Available: https://bench.cr.yp.to