Design of RLWE Cryptoprocessor Based on Vector-Instruction Extension with RISC-V Architecture

Quan Zhang, Yujie Huang, Yujie Cai, Yalong Pang, Jun Han

State Key Laboratory of ASIC & System, Fudan University, Shanghai 200433, China
*Email: 17212020059@fudan.edu.cn

Abstract

A ring learning with errors(RLWE) cryptoprocessor based on the RISC-V instruction set architecture is proposed in this work. The cryptoprocessor is the integration of RISC-V core and co-processor. The co-processor is designed to complete complex polynomial operation such as addition, subtraction and multiplication. And RISC-V core is responsible for sending simple signals to control the operation of co-processor. To support parallel data processing and increase the bandwidth of accessing memory, this work extends vector channels and uses vector paths in internal data bus to transfer data. Besides, Operands adopt a memory-memory approach to reduce the latency of accessing data. The polynomial multiplication chooses the algorithm based on number theoretic transform(NTT). In the cryptosystem, arithmetic operations are performed on the NTT domain, which avoids the frequent operations of conversion to the finite-loop domain. And polynomial processing unit adopts the architecture of 8-lanes commutator to improve the degree of data parallelism. Barrett algorithm is chosen as module reduction operation in finite-loop domain. Simulation results show that RLWE cryptoprocessor operates properly and requires 60.5/22.0us to complete encryption/decryption. Results depict time-taken in encryption and decryption are both reduced comparing to designs based on FPGA Virtrex.

1. Introduction

RLWE encryption algorithm is considered as a potential post-quantum cipher. And it has a linear characteristic and a good degree of data parallelism. Even though custom circuit has been deigned, special co-processor has not been realized for RLWE algorithm. Aiming at above problems, this work designs special co-processor based on RISC-V architecture and integrates it to RISC-V core to implement RLWE algorithm. RISC-V core sends control signals to co-processor, and the latter is responsible for frequent data processing.

This work was supported by the National Science Foundation of China (61574040, 61525401, 61751401).

The remainder of the paper is organized as follows. In Section 2, this work describes the architecture of the RLWE cryptoprocessor and illuminate how it implements in detail. Section 3 shows the simulation results. Finally, we make a conclusion in Section 4.

2. RLWE processor Architecture

As shown in Figure.1, RISC-V core and extended co-processor work in collaboration. RISC-V core is responsible for starting and stopping encryption/ decryption. And co-processor mainly takes charge of NTT polynomial operation. RISC-V core sends macro instructions to co-processor through asynchronous FIFO. When detecting FIFO is not empty, co-processor starts to work and stores subsequent instructions to instruction memory. They share the tightly coupled memory(TCM), and the bandwidth of accessing memory is 8-lanes (each lane is 32 bits) which is consistent with the architecture of radix-8 multiple path delay commutator (R8MDC). R8MDC-NTT unit will be described in detail later.

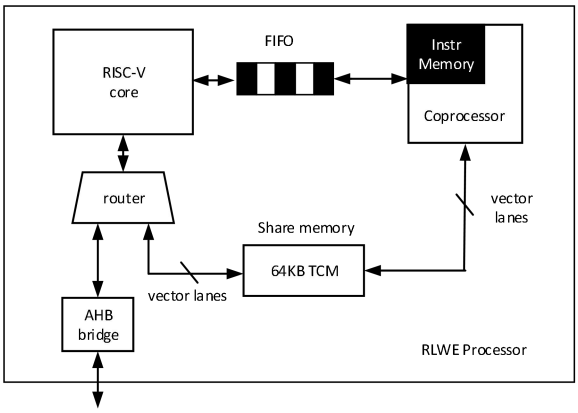

Figure.1 The architecture of RLWE cryptographic processor

2.1 Extended vector-instruction based on RISC-V

In this work, based on standard RV32IMC instruction set, we propose and implement a vector extension for RISC-V architecture. The proposed architecture provides ability of parallel computation. Moreover，the bandwidth of accessing memory is extended and the delay of

978-1-5386-4441-6/18/$31.00 ©2018 IEEE

accessing memory is hided. The basic architecture of extended processor is described in Figure.2. The main units include scalar registers, vector registers, operation unit and vector load/store unit.

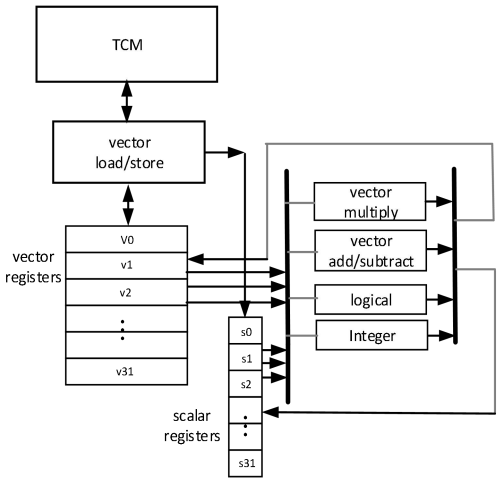

Figure.2 Basic architecture of RISC-V vector processor

2.2 Architecture of extended co-processor

Co-processor completes complex operation such as NTT/INTT and point-wise operation to increase the degree of parallelism. The architecture of co-processor is displayed in Figure.3. When detecting that the FIFO is not empty, co-processor starts to work and stores macro instructions into instruction memory. The co-processor consists of four stage pipelines. Instruction fetch(IF) unit loads instruction from memory and pass it to instruction decode(ID) unit. ID unit decodes instruction signals and get the source operands to execute(EXE) unit to execute corresponding operation. RISC-V previously load processed data to TCM and is not allowed to access TCM until co-processor stops working.

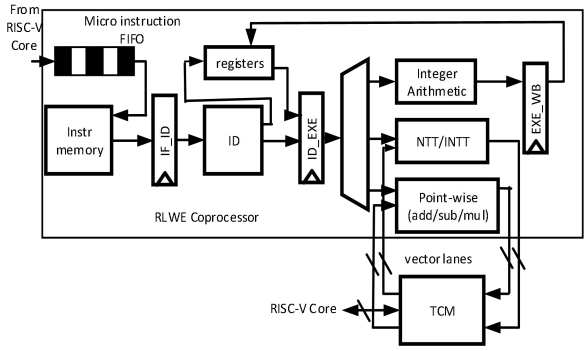

Figure.3 the architecture of co-processor

2.3 Modular reduction unit

NTT and the whole encryption algorithm need the operation of modular reduction in finite-loop domain. And Barrett reduction algorithm has better performance under smaller field. Circuit architecture is depicted in Figure.4. Every modular reduction needs multiplication and subtraction for twice, as well as shift and comparison for once. Registers are inserted after each multiple operation and output, so delay time is three clock cycles.

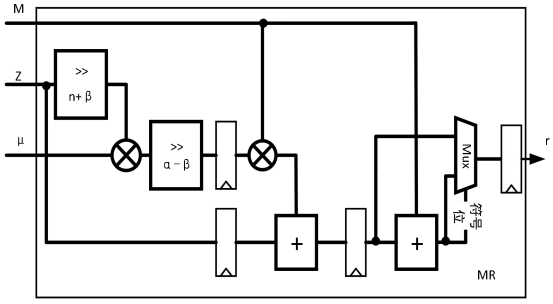

Figure.4 The architecture of Barrett modular reduction

2.4 R8MDC-NTT unit

To increase parallelism of data process as well as the consideration of compromise between area and complexity, this design chooses the architecture of R8MDC-NTT. The architecture is displayed in Figure.5, and mainly includes BF (Butterfly processing Element)-8 unit, commutator unit, delay unit, reverse unit, twiddle factor unit. BF8 unit realizes the multiplication of vector and matrix. Delay unit adjusts the timing of input data for commutator unit, and the commutator unit changes the location for BF8 unit. Twiddle factors are previously stored in ROM.

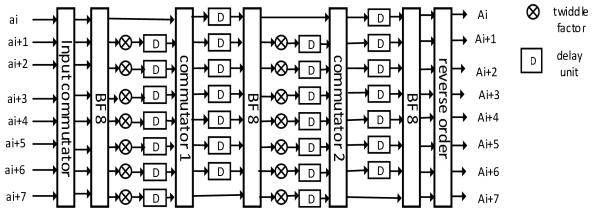

Figure.5 The architecture of 512 points R8MDC-NTT

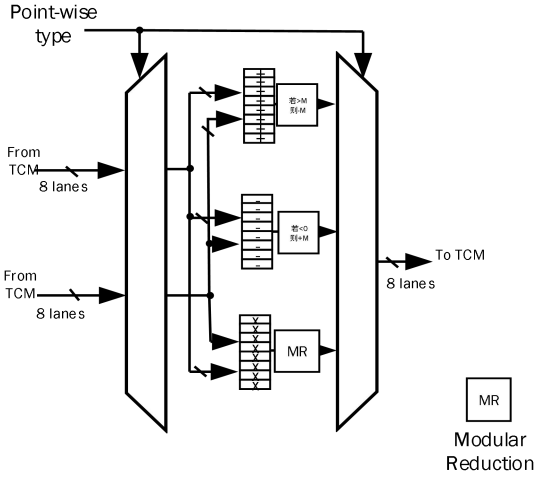

Figure.6 The architecture of point-wise operation unit

Table 1: The comparison of Ring-LEW performance (α FFs, β Regs)

Reference	Device	FFs/Regs	Mem /DSPs	Fre (MHZ)	Cycles Enc/Dec	Times(us) Enc/Dec	Thr.(Mbps) Enc/Dec
Clercq[5]	Cortex-M4F	-	-/-	168	261939/ 9620	1559/574.5	4.60/0,89
Göttert[6]	Virtrex-7	430243$^{\beta}$	0/0	-	-/-	-/-	7/0.29
RLWE-Enc[7]	Spartan-6	238$^{\alpha}$	2/95	144	136000/-	946/-	3.24/-
RLWE-DEC		87$^{\alpha}$	2/32	189	-/66000	-/351	-/0.73
Pöppekmann[8]	Virtrex-6	4760$^{\alpha}$	28/1	250	13769/8883	54.86/35.39	130.66/14.47
Roy[9]	Virtrex-6	953$^{\alpha}$	6/0	277	13300/5800	47.9/21.00	149.64/24.38
This work	Virtrex-7	27383$^{\alpha}$	64/390	125	7567/2757	60.5/22.0	118.81/23.4

2.5 Point-wise operations unit

Point-wise operations unit adopts 8-lanes vectors and each operation decides by decoded signals form instruction pipelines. The operations include addition, subtraction and multiplication. And results should be reduced to finite-loop domain by modular reduction module. The architecture is shown in Figure.6.

3. Results

This work chooses P_2 = (512, 12289, 12.18) as parameter, so comparison is obtained among the peers that also choose P_2. The results are shown in table 1. Simulation results indicate that this work operates properly and requires only 60.5/22.0us to complete encryption/ decryption. Time-taken is reduced a lot compared to software design based on Cortex-M4F. Besides, clock cycles taken in encryption and decryption are both reduced compared to designs based on FPGA Virtrex.

4. Conclusion

The design of RLWE cryptoprocessor based on RISC-V architecture is the focus of this work. In this design, extension of RISC-V vector channels improves the degree of data parallelism and scalar operation also reuses vector channels as to save area overhead. The integration of RISC-V core and co-processor simplify the design process and can also accelerate any algorithm based on polynomial operation. Modular reduction adopts Barrett algorithm in pipeline to short critical path and obtain higher frequency. And arithmetic operations are performed on the NTT domain to avoid frequent conversion. NTT and point-wise operation utilize architecture of 8-channel to increase throughout rate and data parallelism. Besides the latency of access is reduced by using memory-memory approach. Results prove the validity of the realization of RLWE algorithm and the improvement of performance. Further work should be focused on enriching instruction set to support the realization of more kinds of encryption algorithm.

Reference

[1] RENTERIA-MEJIA C P, VELASCO-MEDINA J. High-Throughout Ring-LWE Cryptoprocessors[J]. IEEE Transactions on Very Large Scale Integration Systems. Page.2332-2345(2017).

[2] BATTEN C F. Simplified vectir-thread architectures for flexible and efficient data-parallel accelerators[J] (2010).

[3] DU C, BAI G, WU X. High-speed polynomial multiplier architecture for ring-LWE based public key cryptosystems[C]//Great Lakes Symposium on VLSI, 2016 International. Page.9–14(2016).

[4] YALONG PANG J H Ying Zhang, ZENG X. Fp2 Arithmetic Acceleration Based on Modified Barrett Modular Multiplication Algorithm[C]// International Conference on ASIC. IEEE. Page.561-564(2017).

[5] CLERC R D, ROY S S, VERCAUTEREN F, et al. Efficient software implementation of ring-LWE encryption [C]//Design, Automation & Test in Europe Conference & Exhibition. Page.3339-344(2015).

[6] GÖTTERT N, FELLER T, SCHNEIDER M, et al. On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes[C]// PROUFF E, SCHAUMONT P. Cryptographic Hardware and Embedded Systems –CHES 2012. Ed. by PROUFF E, SCHAUMONT P. Berlin, Heidelberg:Springer Berlin Heidelberg, page. 512–529(2012).

[7] POPPELMANN T, GUNEYSU T. Area optimization of lightweight lattice-based encryption on reconfigurable hardware[C]// IEEE International Symposium on Circuits and Systems. Page.2796–2799(2014).

[8] PÖPPELMANN T, GÜNEYSU T. Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware[C]// Selected Areas in Cryptography. Page. 68–85(2013).

[9] ROY S S, VERCAUTEREN F, MENTENS N, et al. Compact Ring-LWE Cryptoprocessor[C]// International Workshop on Cryptographic Hardware and Embedded Systems. Page.371–391(2014).