DESIGN A FFT AND VERIFY USING RAZOR FLIP FLOP RAM SETS

[1]AJAY KUMAR MALLAVARAPU, [2] M.VENKATESWARA RAO
[1]M-tech student, Sai Tirumala NVR Engineering College, Jonnalagadda, Guntur, A.P
[2]Assistant professor, Sai Tirumala NVR Engineering College, Jonnalagadda, Guntur, A.P

Abstract: Basically, cryptographic algorithms consists of large integers like RSA and Diffie-hellman. This algorithm performs mostly time consuming operations like modular multiplication operation. The implementation of modular multiplication function takes more than 75 percent of time within in the RSA for more than 1024-bit moduli. Here fast multiplier architectures are used to minimize the delay and increase the throughput using parallelism and pipelining. But this architectures occupy large area and gives less efficiency. In this paper it consists of integration of fast Fourier transform method into the framework and as well as it produces improved FFT-based Montgomery modular multiplication (MMM) algorithm to obtain high area-time efficiency. In existed system zero padding operation is performed to compute the modular multiplication steps directly using cyclic and nega cyclic convolutions. Now to reduce the convolution length by half in the number weighted theoretic algorithm, FFT algorithm is used which provide fast convolution computation. In proposed algorithm a general method is introduced for efficient parameter selection. In this proposed architecture single and double butterfly structures are designed to get low area-latency solutions and these are implemented on Xilinx Virtex-6 FPGAs. So from this it can be observed that the proposed system gives better area-latency efficiency compared to the existed system.

IndexTerms:**Montgomery modular multiplication, number-theoretic weighted transform, fast Fourier transform (FFT), field-programmable gate array (FPGA).**

I. INTRODUCTION

In this paper it mainly focuses on hardware implementation of RSA algorithm with more than modulus length 1024-bit. The main intent in this is to create the implementations that achieve high area time efficiency. RSA algorithm is the first public key encryption and digital signature Algorithm. RSA algorithm is mainly used from smart cards to cell phones and SSL boxes. The security in RSA algorithm depends upon the difficulty of factoring a modulus n to find its two prime factors p and q. By selecting higher modulus the security in RSA algorithm is increased. In 1980s the first implementation of RSA algorithm is introduced with 512-bit modulus. Later the bit modulus is extended to 1024-bit. Various implementations are introduced but the national institute of standard and technology recommends 3072-bit or 4096-bit modulus to maintain RSA secure.

Now to compute hardware resources, RSA computation requires modular exponentiation (x^m modN) which is computed by repeated modular multiplications. There will be direct impact on efficiency of RSA computation. So high performance modular multiplier supports 3072-bit size. To compute the modular multiplications one of the effective method is Montgomery modular multiplication (MMM). In this algorithm, the time consuming trail division is replaced by multiplication and reductions modulo R. to improve the Montgomery modular multiplication integer multiplications are multiplied. Here the existed system divides the multiplication methods into two groups mainly they are first group and second group. The first group is performed in time domain and second group is performed in both time domain and spectral domain. But here for Fast Fourier Transform (FFT) algorithm second group is applied because it produces lower asymptotic complexity. To implement hardware multiplication there are various methods of multiplication they are the schoolbook method, Karatsuba method and SSA.

To improve the FFT-based Montgomery product reduction (FMPR) algorithm, state of art architecture is proposed. In this the multiplication and addition steps are of MMM are performed in spectral domain and the time-spectral domain transforms are supported by FFT. Here in existed system Zero padding operation produces less efficiency. So it can be avoided by using modified version of MMM. At last we propose an FFT-based MMM algorithm under framework. Here the time spectral domain transform without zero-padding is the repeated basic operation. Depending upon the different parameter sets the cost and cycle of FFT has high area time efficiency.

II. EXISTED SYSTEM

The below figure (1) shows the block diagram of existed system. Existed system mainly consists of four parts they are input and output index vector generator, computation address generator, three different memory bank groups, a PE unit applied to HRSB scheme and an exponent scaling unit. Let us discuss them in detail.

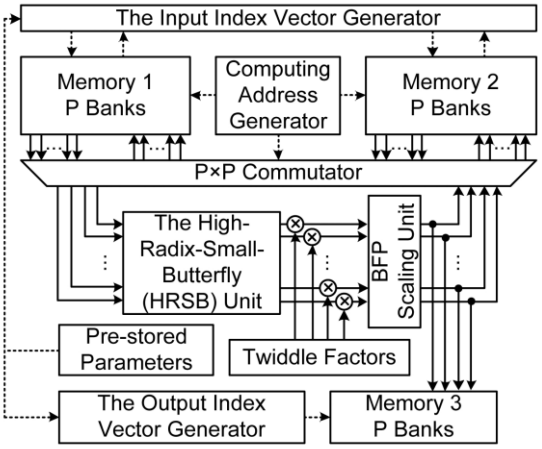

Fig. 1. Existed system

The input index vector generator distributes the input data to different memory banks without data conflicts and the output index vector generator reorders the output data to a natural sequence. Next one is computation address generator, it obtains all the concurrent data of each cycle and stores back the intermediate results. Another one is three memory bank groups, it exist in NSPP FFTs. Memory groups 1 and 2 are in a Ping-Pong mode to holdtwo continuous data symbols in input sampling, and memory group 3 is used to output the computed data in right order. The next one is the HRSB unit, this is the kernel processing engine. The commutators located between the memories and HRSB unit provide efficient data routing mechanism which iscontrolled by the computation address generator. The last one is exponent scaling unit. This unit will increase the signal to quantisation noise ratio and reduce the memory storage.

Here the 2^n-point FFT is operated in continuous flow mode using two parallel radix MDC units. The computation is completed within 128 clock cycles. The both input and output index generators get merged as one in 2^n-pont FFT. The index vector generator can be designed hieratically. The index counter counts in natural order and mutually prime factors are obtained using inverse PFA mapping. At last the trivial factors are used to obtain memory bank and address.Here the binary representation for input is in forward manner and the representation for output is in reverse manner. So from this it can be that due to reverse process the results obtained will be not in effective manner. To overcome this a new system is proposed which is shown in below section.

III. PROPOSED SYSTEM

The below figure (2) shows the architecture of proposed system. This shows the top level architecture of FFT-RAM. The operations involved in FFT-RAM are computed sequentially. In this the pipelined architecture is designed for each unit. The components used in architecture are multiply adder, FFT, ripple carry adder, subtractor, shift module, RAM sets. Let us discuss each of them in detail.

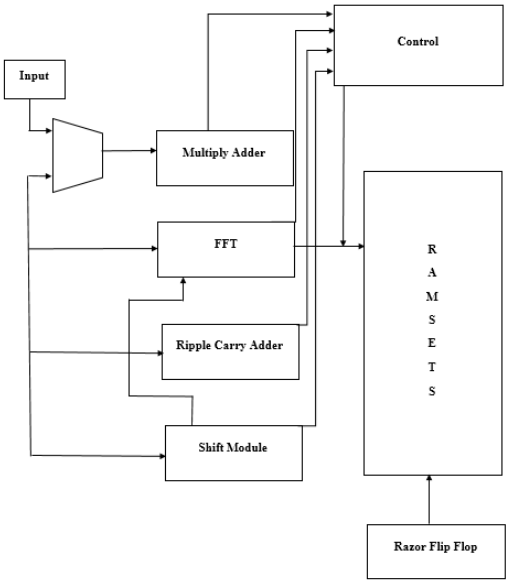

Fig. 2. Proposed system

The first and main important component in the architecture is multiply and adder unit. This unit implements the component wise multiplication and addition of FFT-RAM. To realize the component wise multiplication when operand size is not larger than few hundred bits then karatsuba method is used. The multiplier and adder units works with pipeline of 3 bit inputs and one bit output. At last to enhance the performance of multiplication, karatsuba method is applied recursively. This is about multiply and adder unit and let us discuss about FFT unit.

Forward and reverse networks are formed in FFT unit. Basically, this unit is targeted on high clock frequency and small resource cost. Here constant geometry FFT is applied to FFT computation. The main comparison of in-place and constant geometry FFT is it has same connection network between every adjacent stages. The FFT is designed with six inputs. In this the four inputs forward the digits into BFSs for FFT computation and the other two inputs forward the pre-computed upper bound constraints into FSO.

Next one is RAM unit. RAM unit consists of several RAM sets which stores the pre-computed data, the intermediate results, and the final modular product. In RAM the data storage requirement during FFT-RAM computation is not trivial. Now to well manage the input and output of data and to reduce the wiring workload, RAM unit is built. The remaining are the Ripple Carry Adder (RCA), the Subtracter and the Shift Module units are responsible for the time domain operations, such as modulo R and Q0 reductions, conditional selections. Now to generate all control signals of entire system, control unit is designed. At last it can observe that the proposed system gives better results compared to existed system.

IV. RESULTS

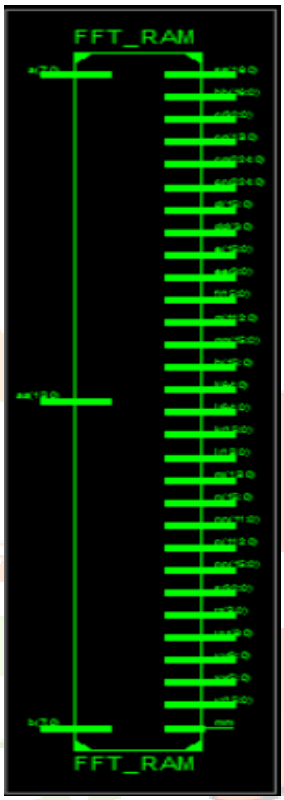

Fig. 3. RTL Schematic

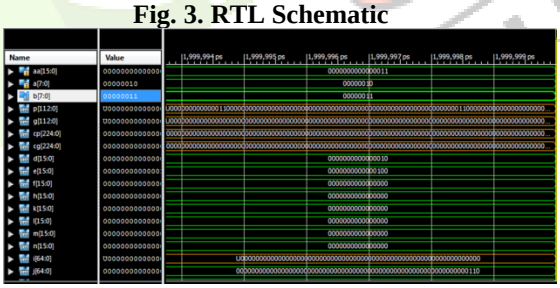

Fig. 4. Input Waveform

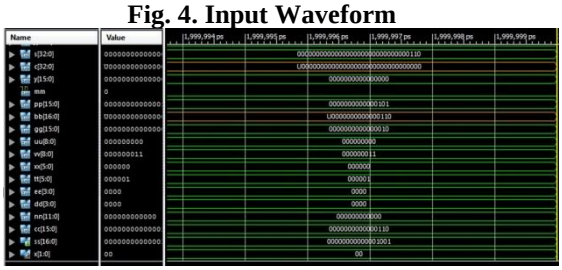

Fig. 5. Output waveform

V. CONCLUSION

In this paper we proposed a modified version of FFT depending upon the Montgomery modular multiplication algorithm under framework (FFT-RAM). To compute the modular multiplication and avoid the zero padding operation, cyclic and nega cyclic convolutions are applied. Because of this the transform length is reduced by half. To obtain high latency efficiency, pipelined architectures are designed with one and two butterfly structures. So the proposed system provides better latency efficiency compared to exist one.

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.
[2] R. L. Rivest, "A description of a single-chip implementation of the RSA cipher," Lambda, vol. 1, no. Oct.–Dec., pp. 14–18, 1980.
[3] "Recommendation for key management," NIST, Tech. Rep. Special Publication 800-57, Part-1, Rev.-3, 2012.
[4] P. L. Montgomery, "Modular multiplication without trial division,"Mathematics Comput., vol. 44, no. 170, pp. 519–521, 1985.
[5] A. Karatsuba and Y. Ofman, "Multiplication of multidigit numbers on automata," Soviet Physics Doklady, vol. 7, 1963, Art. No. 595.
[6] S. A. Cook and S. O. Aanderaa, "On the minimum computation time of functions," Trans.Amer.Math. Soc., vol. 142, pp. 291–314, 1969.
[7] A. Sch€onhageand V. Strassen, "Schnelle multiplikation Großer Zahlen," Computing, vol. 7, no. 3/4, pp. 281–292, 1971.
[8] M. F€urer, "Faster integer multiplication," SIAM J. Comput., vol. 39, no. 3, pp. 979–1005, 2009.
[9] D. Harvey, J. van der Hoeven, and G. Lecerf, "Even faster integer multiplication," CoRR, vol. abs/1407.3360, 2014. [Online]. Available: http://arxiv.org/abs/1407.3360

AJAY KUMAR MALLAVARAPU Completed his B.tech in Malla Reddy Engineering College and Pursuing M.tech in Sai Tirumala Engineering College. His M.tech Specialization is electronics & comm engineering.

M. VENKATESWARA RAO completed his B.Tech at ST.Mary's Engineering College, Guntur and M.Tech at VRS&YRN Engineering College, Chirala. At Present he is working as Assistant Professor at Sai Tirumala Engineering College with Five Years of Teaching Experience.