High Speed Unified Field Crypto processor for Security Applications using Verilog

N. Kumaresan
Department of ECE, Anna University,
Regional Campus Coimbatore, Tamil Nadu, India
kumaresanauc@gmail.com

S. Kodeeswaran
Department of ECE, Anna University,
Regional Campus Coimbatore, Tamil Nadu, India
skodeeswaranme@gmail.com

***Abstract*—**Traditional cryptographic algorithms are developed on a software platform and provides information security schemes. Also, some processors have performed one of the crypto algorithms (either prime field or binary extension field) on chip level with optimal performance. The objective is to design and implement both symmetric key and public key algorithms of a cryptographic on chip level and make better architecture with pleasing performance. Crypto-processor design, have been designed with unified field instructions to make different processor architecture and improve system performance. The proposed high speed Montgomery modular multiplication and high radix Montgomery multiplication algorithms for pairing computation supports the public key algorithm. This design has been developed using Verilog HDL's and verified using ModelSim-Altera 6.4a, and it has synthesized with Xilinx 9.1 Integrated Synthesis Environment (ISE) tool.

Keywords*-Cryptographic algorithm, Crypto processor, Modular arithmetic*

*****

I. Introduction

Cryptography is a one of the technique to provide high authentication to the users. This secured communication is achieved by various algorithms. These algorithms possess various steps and provide security to the specific system. Communication domain has widely developed and enhanced throughout the world.

Everyone who does shopping, transferring money through their smart cards and smart phone or drawing money from the ATM (Automated Teller Machine) using their ATM card need some security code word to access it. That security code must be confidential or else hackers can use it illegally to withdraw money from other's account. Traditionally, these processors are unique, like the microprocessor and microcontrollers. These controllers only permit the authorized user to communicate with the system. These traditional processors cannot provide secure communication. In this traditional technology, secure communication is done by software platform and not by the hardware. It has led to hacking of personal data like, PIN number, mail account password etc. To make the system faster and reduce the hacking problem, we need a system to do the security process on the hardware platform, so we often call this scenario as Crypto-processor.

A Crypto-processor is a System on Chip (SoC) based microprocessor which carries out the encryption and decryption operations. A secured Crypto-processor has dedicated instructions in the environment. Secure key exchange with public key or secure key algorithm needs various hardware resources. The main objective of this work is to propose a security algorithm which is handled by the dedicated chip itself. So, it becomes handy. Pairing computation does parallel operation and makes the systems performance speedy.

II. Proposed System

The Crypto-processor designing here has both fields of instruction. Also, it contains general purpose instruction which can lead to getting information from the user and making secure communication that can be handled by crypto instruction or otherwise secure instruction. If it is necessary to design a symmetric key algorithm, it has to use prime field instruction whereas for public key algorithm, it has to use binary extension field instruction. So, the proposed processor can meet this requirement and provide an instruction for both fields of operation. This kind of design is called as unified field operation.

The digital system proposed in this paper uses High radix Montgomery multiplication and it will act as Prime Field GF(P) instruction, it could well act as a symmetric key instruction. Another method has been imported in this digital architecture is Binary Extension Field GF (2^M) architecture and it has designed with Galois Field Multiplication Instruction.

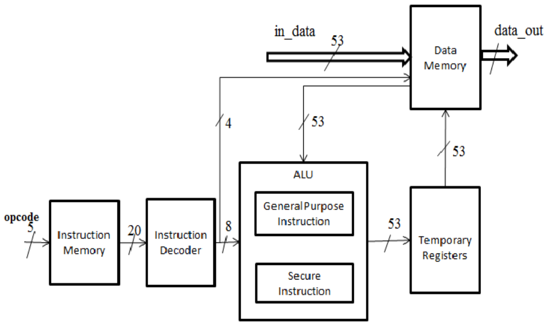


Figure 1. Block diagram of the proposed Crypto processor.

The Crypto-processor is properly connected with a clock and reset. The clock defines the operating speed of the processor and reset signal can make the system start from initial incident at any time. For this processor, there is a set of Opcode, it shows which function processor shall have to do. When initializing Opcode to any of the given instruction, it fetches the source address and destination address for this particular operation that can be handled by the instruction decoder. This information is passed to data memory and ALU. Data memory can provide operand information to ALU unit as per source address and ALU can perform the corresponding operation and it will be stored in temporary memory as well as data memory. Once the sequence of operations is over then output can be presented in the memory to the user.

ALU consists of general purpose instruction and secure instruction in which general purpose instructions that are usual operations followed in typical processor. Also in secure instruction operation consists of dual field or unified field operation which can do both prime field and binary extension field of operation. Instruction sets designed and it is tabulated as in Table. 1. It consists of prime instruction for modular addition, subtraction and high radix modular multiplication, and binary extension field instruction for Galois field multiplication; also it has a general purpose instruction like arithmetic and logic instruction, shift and rotate.

TABLE I. INSTRUCTION SET

Sl. No.	Instruction	Operation
1.	Addition	A + B mod P – Prime field
2.	Subtraction	A – B mod P – Prime field
3.	Multiplication	A*B mod p – Prime field
4.	High Radix Montgomery Multiplication	(AB+CD) R^{-1} – Prime field
5.	Galois field Multiplication	A * B – Binary extension field GF(2^m)
6.	Arithmetic and logical, shift & rotate, LOAD, STORE – General Instruction Set	This includes +, -, *, /, &, |, ^, >>, <<

This design constraint is designing Crypto-processor with unified field instruction. It follows the design flow; finally it could end with two things that are Register Transfer Level (RTL) schematic and physical layout. This work scopes RTL schematic design. For this design, a Verilog code has used for mathematical approach of the algorithm and straight forward design like memory and decoder designs are developed using Hardware Description Language (HDL). Once the design is written in HDL, it is possible to verify using simulation tools. The processor has been developed using Verilog HDL. Also, it produces device utilization summary and timing summary. Finally the design could implement it on any FPGA’s.

Proposed design has explained with a following flow diagram. Each algorithm has had a sequence of steps and Figure 2 shows a flow diagram of High radix Montgomery Multiplication algorithm.

Step 1: Need to choose A, B, C, D and P1, P2 – these values must be large prime numbers. N represents the number of input bit width.

Step 2: Number of iterations to compute the intermediate result depends on width of input.

Step 3: Intermediate results need a modulo operation and simple arithmetic operation like addition and multiplication.

Step 4: Final result is the addition of intermediate remainder and simple expression carried out by using addition and multiplication.

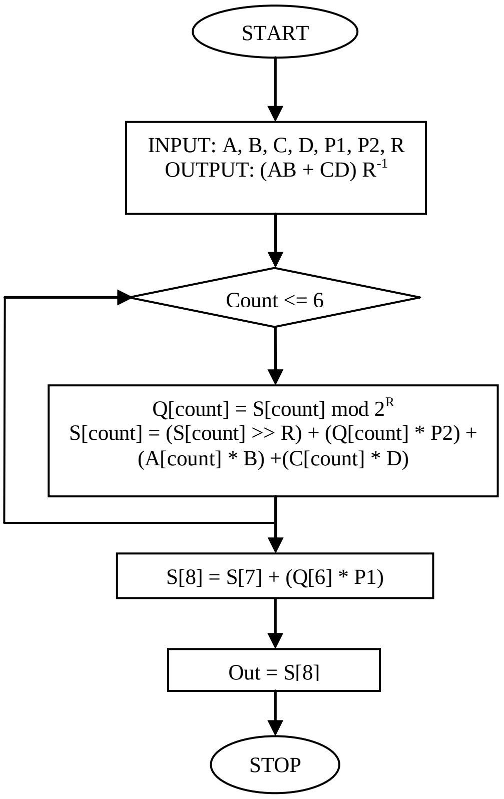

START
INPUT: A, B, C, D, P1, P2, R
OUTPUT: (AB + CD) R-1
Count <= 6
Q[count] = S[count] mod 2R
S[count] = (S[count] >> R) + (Q[count] * P2) + (A[count] * B) +(C[count] * D)
S[8] = S[7] + (Q[6] * P1)
Out = S[8]
STOP

Figure 2. Flow diagram of high radis Montgomery multiplication.

Following flow diagram describes the operation of modular addition.

Step1: Choose highest prime number of A, B and P.

Step2: Then separately calculate remainder value corresponding input A and B.

Step3: Finally, add those values to obtain modular addition.

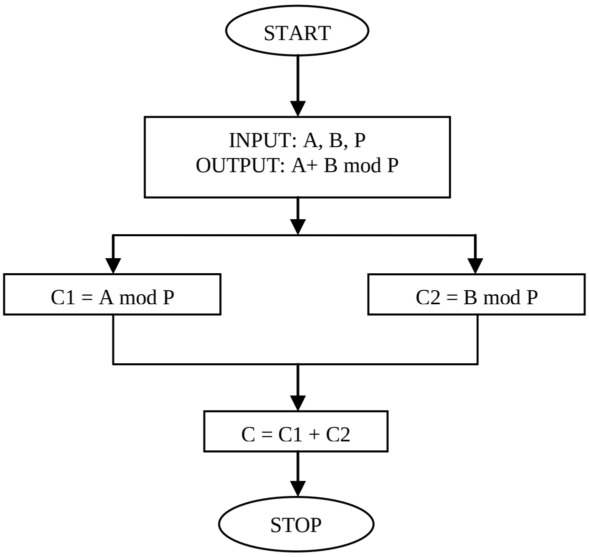


Figure 3. Modular addition.

Following flow diagram describes the operation of modular subtraction.

Step1: Choose highest prime number of A, B and P.

Step2: Then separately calculate remainder value corresponding input A and B.

Step3: Finally, subtract those values to obtain modular Subtraction.

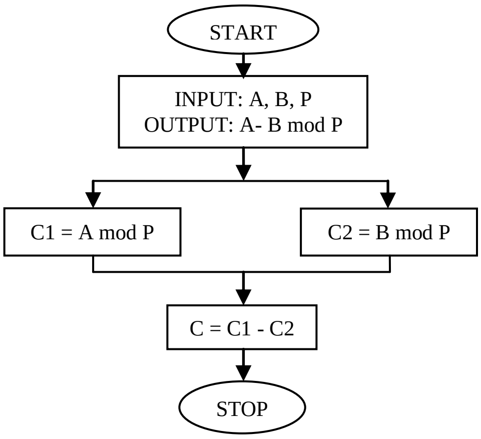


Figure 4. Modular subtraction.

The binary extension field instruction has been explained with flow diagram shown in figure 5. And its steps are as follows:

Step 1: Choose A, B and P prime values.

Step 2: Using Look Up table based design; its corresponding output will be obtained.

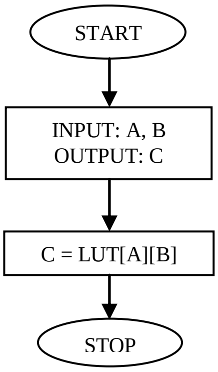


Figure 5. Galois multiplication GF (2^M).

III. Results ans Discussion

The simulation result can describe how the designs can response for a different input stream at various time incidents. With the application of an input to the system, it is possible by writing test bench for a designed system. In a simulated window there are details which clearly show that at how next modules responds as per the input given for each module.

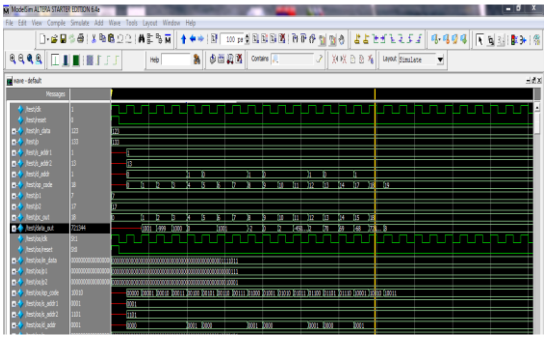


Figure 6. Simulation result of the proposed Crypto processor.

Above screenshot shows the result of crypto-processor, in which clock and reset are operating signal of the processor. The clocks can able to provide 20ns with 50% duty cycle. A reset signal is used to start a processor from the initial value. An Opcode has been used to define which operation currently the processor is executing. The data_out signal represents the output of the processor. The operands for processors are fetched from data memory.

Examine the simulation result with an example, for Opcode input is 0, it could perform addition of two numbers, namely data1 is in decimal ‘1’ and data2 is decimal ‘1000’, it will be added and the result is 1001. For Opcode input is 18, it could perform prime field high radix Montgomery multiplication. This multiplication needs two prime numbers, namely P1 and P2 also it needs four data inputs and its result is 721344.

For an Opcode input is 19, it could perform binary extension field multiplication. It needs two operands and its output taken from Lookup table. Its output is 8, for an input 1 and 8. Once designs are verified using simulation tools, it is time to synthesis a design. Synthesis means that HDL code can be converted into the RTL schematic, so that design can easily implement on any FPGA. The following screen shot has taken for this particular device Virtex5-XC5VLX30-FF324-3.

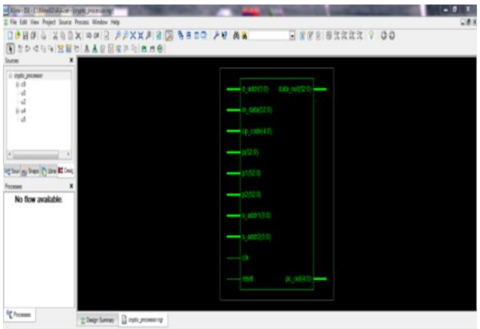


Figure 7. RTL view of Crypto processor.

This result shows the top module of our Crypto-processor, this shows all input and output of the Crypto-processor. Once our HDL code can be converted to RTL as shown in above Figure 7, this can be implemented on any FPGA. This shows the internal design of our Crypto-processor. This internal schematic shows how data are connected between modules.

The Crypto-processor consists of data memory, instruction memory, instruction decoder, ALU, program counter and temporary memory. These details are shown in the following Figure 8.

After completing synthesis, we get design summary, timing summary and details of combinational and sequential logic utilized by design. An area result obtained by the crypto-processor as taken Virtex5 as a target device. As per calculation 56% of the area has occupied by a Crypto-processor.

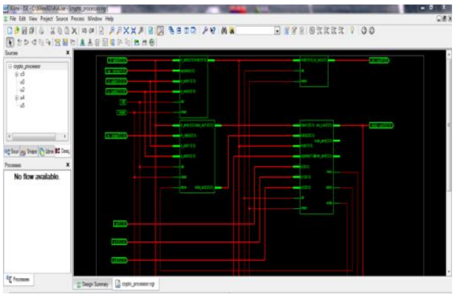
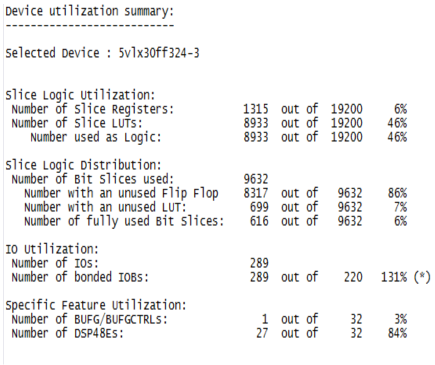
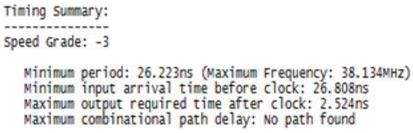


Figure 8. Internal modules of Crypto processor.



```
Device utilization summary:
---------------------------

Selected Device : 5vlx30ff324-3


Slice Logic Utilization:
 Number of Slice Registers:          1315  out of  19200     6%
 Number of Slice LUTs:               8933  out of  19200    46%
    Number used as Logic:            8933  out of  19200    46%

Slice Logic Distribution:
 Number of Bit Slices used:          9632
   Number with an unused Flip Flop   8317  out of   9632    86%
   Number with an unused LUT:         699  out of   9632     7%
   Number of fully used Bit Slices:   616  out of   9632     6%

IO Utilization:
 Number of IOs:                       289
 Number of bonded IOBs:               289  out of    220   131% (*)

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:              1  out of     32     3%
 Number of DSP48Es:                    27  out of     32    84%
```

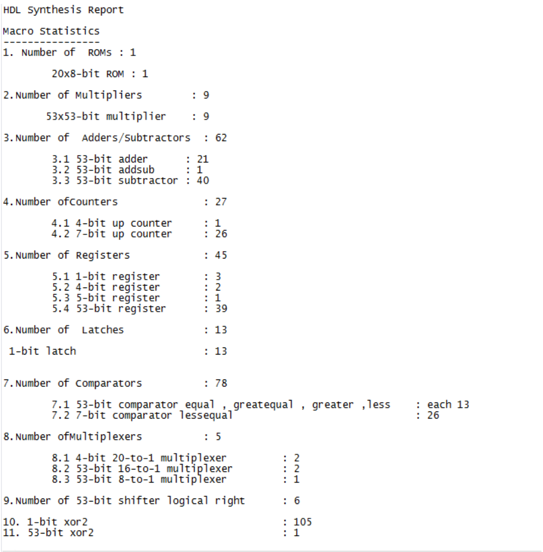
Figure 9. Device utilization summary of a Crypto processor.



```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 26.223ns (Maximum Frequency: 38.134MHz)
   Minimum input arrival time before clock: 26.808ns
   Maximum output required time after clock: 2.524ns
   Maximum combinational path delay: No path found
```

Figure 10. Timing summary of a Crypto processor.

This HDL synthesis report details gives us knowledge about how much combinational and sequential logic that has been used by the Crypto-processor. Combinational logic includes adder/subtractors, multiplier, multiplexer and logic gate. Similarly sequential logic includes counters, registers and latches.



```
HDL Synthesis Report

Macro Statistics
----------------
1. Number of  ROMs : 1

        20x8-bit ROM : 1

2.Number of Multipliers        : 9

       53x53-bit multiplier    : 9

3.Number of  Adders/Subtractors  : 62

        3.1 53-bit adder      : 21
        3.2 53-bit addsub     : 1
        3.3 53-bit subtractor : 40

4.Number ofCounters              : 27

        4.1 4-bit up counter     : 1
        4.2 7-bit up counter     : 26

5.Number of Registers            : 45

        5.1 1-bit register       : 3
        5.2 4-bit register       : 2
        5.3 5-bit register       : 1
        5.4 53-bit register      : 39

6.Number of  Latches             : 13

 1-bit latch                     : 13


7.Number of Comparators          : 78

        7.1 53-bit comparator equal , greatequal , greater ,less    : each 13
        7.2 7-bit comparator lessequal                               : 26

8.Number ofMultiplexers          : 5

        8.1 4-bit 20-to-1 multiplexer         : 2
        8.2 53-bit 16-to-1 multiplexer        : 2
        8.3 53-bit 8-to-1 multiplexer         : 1

9.Number of 53-bit shifter logical right      : 6

10. 1-bit xor2                                : 105
11. 53-bit xor2                               : 1
```

Figure 11. Macro static details of a Crypto processor.

IV. CONCLUSION

A Crypto-processor prototype design has been made and synthesized. It can be able to execute unified field instruction. To achieve creditable performance and having both instruction set in single processor architecture, had invoked both prime and binary field algorithms into single processor. Also, this design used high radix Montgomery multiplication to make pairing computation to attain system as fast as possible. Consequentially, this design is very useful for simulation and synthesis tools for analyzing this architecture and it has been occupied 56% area and timing it is also achieved credibly equally to known architecture.

In future, it could add few more instructions like Modular exponential and Modular inversion. Also to increase the number of pipeline stages to make better performance.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

V. References

[1] V. Bunimov and M. Schimmler, "Area and time efficient modular multiplication of large integers", in Proc. IEEE Int. Conf. Application-specific systems, architectures, and processors, ASAP 2003, The Hague, Netherlands, 2003, pp. 400-409.

[2] R. C. C. Cheung, Sylvain Duquesne, Junfeng Fan, Nicolas Guillermin, Ingrid Verbauwhede, and Gavin Xiaoxu Yao, "FPGA implementation of pairings using residue number system and lazy reduction", in Proc. 13th Int. Conf. on Cryptographic hardware and embedded systems, Nara, Japan, 2011, pp. 421–441.

[3] J. Fan, F. Vercauteren, and I. Verbauwhede, "Efficient hardware implementation of Fp-Arithmetic for pairing-friendly curves", IEEE Transactions on Computers., vol. 61, no. 5, pp. 676-685, 2012.

[4] S. Ghosh, D. Mukhopadhyay, and D. Roy Chowdhury, "High speed flexible pairing cryptoprocessor on FPGA platform", in Proc. 4th Int. Conf. on Pairing-based cryptography, Pairing 2010, Japan, 2010, pp. 450–466.

[5] H. Wu, "Bit-parallel finite field multiplier and squarer using polynomial basis", IEEE Transactions on Computers, vol. 51, no. 7, pp. 750-758, 2002.

[6] J. Han, Y. Li, Z. Yu, and X. Zeng, "A 65nm cryptographic processor for high speed pairing computation", IEEE Transactions on VLSI systems, vol. 23, no. 4, pp. 692-701, 2014.

[7] D. Kammler, D. Zhang, P. Schwabe, H. Scharwaechter, M. Langenberg, D. Auras, G. Ascheid, and R. Mathar, "Designing an ASIP for cryptographic pairing over Barreto-Naehrig curves", in Proc. 11th Int. workshop on Cryptographic hardware and embedded systems, Switzerland, 2009, pp. 254-271.

[8] Y. Li, J. Han, S. Wang, and D. Fang, "An 800Mhz cryptographic pairing processor in 65nm CMOS", in Proc. IEEE Asian Conf. on Solid State Circuits, Kobe, 2012, pp. 217-220.

[9] O. Nibouche, A. Bouridane, and M. Nibouche, "Architectures for Montgomery's multiplication", in Proc. IEE Computers and digital techniques, vol. 150, no. 6, 2003, pp. 361-368.

[10] J. Fan, F. Vercauteren, and I. Verbauwhede, "Faster F_p – arithmetic for cryptographic pairing on Barreto-Naehrig curves", in Proc. 11th Int. workshop on Cryptographic hardware and embedded systems, Switzerland, 2009, pp. 240-253.